Inspur

CN12700 Series

INOS Unicast Routing Configuration Guide

(Release 8.x)

Inspur-Cisco Networking Technology Co.,Ltd. provides customers with comprehensive technical support and services. For any assistance, please contact our local office or company headquarters.
Website: http://www.inspur.com/
Technical Support Tel: 400-691-1766
Technical Support Email:inspur_network@inspur.com
Technical Document Support Email:inspur_network@inspur.com
Address: 1036 Langchao Road, Lixia District, Jinan City, Shandong Province
Postal code: 250101

Notice

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute the warranty of any kind, express or implied

Preface

Objectives

This guide describes main functions of the CN12700 Series. To have a quick grasp of the CN12700 Series, please read this manual carefully.

Versions

The following table lists the product versions related to this document.

Product name	Version
CN12700 Series	

Conventions

Symbol conventions

The symbols that may be found in this document are defined as follows.

Symbol	Description
⚠ Warning	Indicates a hazard with a medium or low level of risk which, if not avoided, could result in minor or moderate injury.
⚠ Caution	Indicates a potentially hazardous situation that, if not avoided, could cause equipment damage, data loss, and performance degradation, or unexpected results.
✎ Note	Provides additional information to emphasize or supplement important points of the main text.
🔍 Tip	Indicates a tip that may help you solve a problem or save time.

General conventions

Convention	Description
Boldface	Names of files, directories, folders, and users are in **boldface**. For example, log in as user **root**.
Italic	Book titles are in *italics*.
`Lucida Console`	Terminal display is in `Lucida Console`.

Command conventions

Convention	Description
Boldface	The keywords of a command line are in **boldface**.
Italic	Command arguments are in *italics*.
[]	Items (keywords or arguments) in square brackets [ ] are optional.
{ x | y | ... }	Alternative items are grouped in braces and separated by vertical bars. One is selected.
[x | y | ...]	Optional alternative items are grouped in square brackets and separated by vertical bars. One or none is selected.
{ x | y | ... } *	Alternative items are grouped in braces and separated by vertical bars. A minimum of one or a maximum of all can be selected.
[x | y | ...] *	The parameter before the & sign can be repeated 1 to n times.

GUI conventions

Convention	Description
Boldface	Buttons, menus, parameters, tabs, windows, and dialog titles are in **boldface**. For example, click **OK**.
>	Multi-level menus are in boldface and separated by the ">" signs. For example, choose **File** > **Create** > **Folder**.

Keyboard operation

Format	Description
Key	Press the key. For example, press **Enter** and press **Tab**.

Format	Description
Key 1+Key 2	Press the keys concurrently. For example, pressing **Ctrl+C** means the two keys should be pressed concurrently.
Key 1, Key 2	Press the keys in turn. For example, pressing **Alt**, **A** means the two keys should be pressed in turn.

Mouse operation

Action	Description
Click	Select and release the primary mouse button without moving the pointer.
Double-click	Press the primary mouse button twice continuously and quickly without moving the pointer.
Drag	Press and hold the primary mouse button and move the pointer to a certain position.

Change history

Updates between document versions are cumulative. Therefore, the latest document version contains all updates made to previous versions.

Issue 01 (2020-02-24)

Initial commercial release

Contents

Figures

Tables

CHAPTER 1 Overview

This chapter contains the following sections:

- Overview.

1.1 Overview

This chapter introduces the underlying concepts for the Layer 3 unicast routing protocols in Inspur INOS.

1.1.1 Information About Layer 3 Unicast Routing

Layer 3 unicast routing involves two basic activities: determining optimal routing paths and packet switching. You can use routing algorithms to calculate the optimal path from the router to a destination. This calculation depends on the algorithm selected, route metrics, and other considerations such as load balancing and alternate path discovery.

Routing Fundamentals

Routing protocols use a metric to evaluate the best path to the destination. A metric is a standard of measurement, such as a path bandwidth, that routing algorithms use to determine the optimal path to a destination. To aid path determination, routing algorithms initialize and maintain routing tables that contain route information such as the IP destination address, the address of the next router, or the next hop. Destination and next-hop associations tell a router that an IP destination can be reached optimally by sending the packet to a particular router that represents the next hop on the way to the final destination. When a router receives an incoming packet, it checks the destination address and attempts to associate this address with the next hop.

Routing tables can contain other information, such as the data about the desirability of a path. Routers compare metrics to determine optimal routes, and these metrics differ depending on the design of the routing algorithm used.

Routers communicate with one another and maintain their routing tables by transmitting a variety of messages. The routing update message is one such message that consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture of the network topology. A link-state advertisement, which is another example of a message sent between routers, informs other routers of the link state of the sending router. You can also use link information to enable routers to determine optimal routes to network destinations.

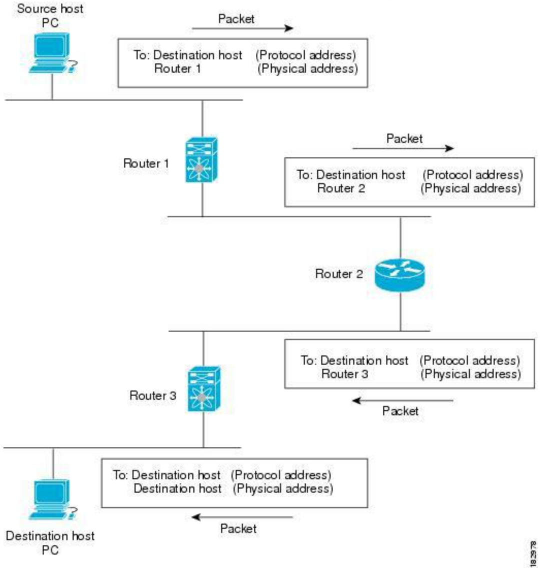
Packet Switching

In packet switching, a host determines that it must send a packet to another host. Having acquired a router address by some means, the source host sends a packet that is addressed specifically to the router physical (Media Access Control [MAC]-layer) address but with the IP (network layer) address of the destination host.

The router examines the destination IP address and tries to find the IP address in the routing table. If the router does not know how to forward the packet, it typically drops the packet. If the router knows how to forward the packet, it changes the destination MAC address to the MAC address of the next-hop router and transmits the packet.

The next hop might be the ultimate destination host or another router that executes the same switching decision process. As the packet moves through the internetwork, its physical address changes, but its protocol address remains constant (see the following figure).

Figure 1 : Packet Header Updates Through a Network


Source host
PC
Packet
To: Destination host (Protocol address)
Router 1 (Physical address)
Router 1
Packet
To: Destination host (Protocol address)
Router 2 (Physical address)
Router 2
To: Destination host (Protocol address)
Router 3 (Physical address)
Router 3
Packet
To: Destination host (Protocol address)
Destination host (Physical address)
Packet
Destination host
PC
182978

Routing Metrics

Routing algorithms use many different metrics to determine the best route. Sophisticated routing algorithms can base route selection on multiple metrics.

Path Length

The path length is the most common routing metric. Some routing protocols allow you to assign arbitrary costs to each network link. In this case, the path length is the sum of the costs associated with each link traversed. Other routing protocols define the hop count, which is a metric that specifies the number of passes through internetworking products, such as routers, that a packet must take from a source to a destination.

Reliability

The reliability, in the context of routing algorithms, is the dependability (in terms of the bit-error rate) of each network link. Some network links might go down more often than others. After a network fails, certain network links might be repaired more easily or more quickly than other links. The reliability factors that you can take into account when assigning the reliability rating are arbitrary numeric values that you usually assign to network links.

Routing Delay

The routing delay is the length of time required to move a packet from a source to a destination through the internetwork. The delay depends on many factors, including the bandwidth of intermediate network links, the port queues at each router along the way, the network congestion on all intermediate network links, and the physical distance that the packet must travel. Because the routing delay is a combination of several important variables, it is a common and useful metric.

Bandwidth

The bandwidth is the available traffic capacity of a link. For example, a 10-Gigabit Ethernet link is preferable to a 1-Gigabit Ethernet link. Although the bandwidth is the maximum attainable throughput on a link, routes through links with greater bandwidth do not necessarily provide better routes than routes through slower links. For example, if a faster link is busier, the actual time required to send apacket to the destination could be greater.

Load

The load is the degree to which a network resource, such as a router, is busy. You can calculate the load in a variety of ways, including CPU usage and packets processed per second. Monitoring these parameters on a continual basis can be resource intensive.

Communication Cost

The communication cost is a measure of the operating cost to route over a link. The communication cost is another important metric, especially if you do not care about performance as much as operating expenditures. For example, the line delay for a private line might be longer than a public line, but you can send packets over your private line rather than through the public lines that cost money for usage time.

Router IDs

Each routing process has an associated router ID. You can configure the router ID to any interface in the system. If you do not configure the router ID, Inspur INOS selects the router ID based on the following criteria:

- Inspur INOS prefers loopback0 over any other interface. If loopback0 does not exist, then Inspur INOS prefers the first loopback interface over any other interface type.
- If you have not configured a loopback interface, Inspur INOS uses the first interface in the configuration file as the router ID. If you configure any loopback interface after Inspur INOS selects the router ID, the loopback interface becomes the router ID. If the loopback interface is not loopback0 and you configure loopback0 with an IP address, the router ID changes to the IP address of loopback0.
- If the interface that the router ID is based on changes, that new IP address becomes the router ID. If any other interface changes its IP address, there is no router ID change.

Autonomous Systems

An autonomous system (AS) is a network controlled by a single technical administration entity. Autonomous systems divide global external networks into individual routing domains, where local routing policies are applied. This organization simplifies routing domain administration and simplifies consistent policy configuration.

Each autonomous system can support multiple interior routing protocols that dynamically exchange routing information through route redistribution. The Regional Internet Registries assign a unique number to each public autonomous system that directly connects to the Internet. This autonomous system number (AS number) identifies both the routing process and the autonomous system.

The Border Gateway Protocol (BGP) supports 4-byte AS numbers that can be represented in asplain and asdot notations:

- asplain—A decimal value notation where both 2-byte and 4-byte AS numbers are represented by their decimal value. For example, 65526 is a 2-byte AS number, and 234567 is a 4-byte AS number.
- asdot—An AS dot notation where 2-byte AS numbers are represented by their decimal value and 4-byte AS numbers are represented by a dot notation. For example, 2-byte AS number 65526 is represented as 65526, and 4-byte AS number 65546 is represented as 1.10.

The BGP 4-byte AS number capability is used to propagate 4-byte-based AS path information across BGP speakers that do not support 4-byte AS numbers. Beginning with Inspur INOS Release 8.4(1), you can configure 4-byte AS numbers in asdot notation. The default value is asplain.

The following table lists the AS number ranges.

Table 1 : AS Numbers

2-Byte Numbers	4-Byte Numbers in AS.dot Notation	4-Byte Numbers in plaintext Notation	Purpose
1 to 64511	N/A	1 to 64511	Public AS (assigned by RIR) [1]
64512 to 65534	N/A	64512 to 65534	Private AS (assigned by local administrator)
65535	N/A	65535	Reserved
N/A	1.0 to 65535.65535	65536 to 4294967295	Public AS (assigned by RIR)

[1] RIR=Regional Internet Registries

Private autonomous system numbers are used for internal routing domains but must be translated by the router for traffic that is routed out to the Internet. You should not configure routing protocols to advertise private autonomous system numbers to external networks. By default, Inspur INOS does not remove private autonomous system numbers from routing updates.

Convergence

A key aspect to measure for any routing algorithm is how much time a router takes to react to network topology changes. When a part of the network changes for any reason, such as a link failure, the routing information in different routers might not match. Some routers will have updated information about the changed topology, while other routers will still have the old information. The convergence is the amount of time before all routers in the network have updated, matching routing information. The convergence time varies depending on the routing algorithm. Fast convergence minimizes the chance of lost packets caused by inaccurate routing information.

Load Balancing and Equal Cost Multipath

Routing protocols can use load balancing or equal cost multipath (ECMP) to share traffic across multiple paths.When a router learns multiple routes to a specific network, it installs the route with the lowest administrative distance in the routing table. If the router receives and installs multiple paths with the same administrative distance and cost to a destination, load balancing can occur. Load balancing distributes the traffic across all the paths, sharing the load. The number of paths used is limited by the number of entries that the routing protocol puts in the routing table. Inspur INOS supports up to 16 paths to a destination.Starting from Inspur INOS Release 8.4(1), the BGP feature supports up to 64 paths to a destination on F3-Series I/O module.

The Enhanced Interior Gateway Routing Protocol (EIGRP) also supports unequal cost load balancing.

Route Redistribution

If you have multiple routing protocols configured in your network, you can configure these protocols to share routing information by configuring route redistribution in each protocol. For example, you can configure the Open Shortest Path First (OSPF) protocol to advertise routes learned from the Border Gateway Protocol (BGP). You can also redistribute static routes into any dynamic routing protocol. The router that is redistributing routes from another protocol sets a fixed route metric for those redistributed routes, which prevents incompatible route metrics between the different routing protocols. For example, routes redistributed from EIGRP into OSPF are assigned a fixed link cost metric that OSPF understands.

Route redistribution also uses an administrative distance to distinguish between routes learned from two different routing protocols. The preferred routing protocol is given a lower administrative distance so that its routes are picked over routes from another protocol with a higher administrative distance assigned.

Administrative Distance

An administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. Typically, a route can be learned through more than one protocol. Administrative distance is used to discriminate between routes learned from more than one protocol. The route with the lowest administrative distance is installed in the IP routing table.

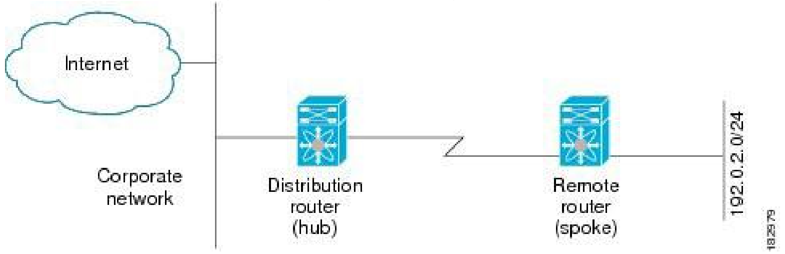
Stub Routing

You can use stub routing in a hub-and-spoke network topology, where one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for IP traffic to follow into the remote router is through a distribution router. This type of configuration is commonly used in WAN topologies in which the distribution router is directly connected to a WAN. The distribution router can be connected to many more remote routers. Often, the distribution router is connected to 100 or more remote routers. In a hub-and-spoke topology, the remote router must forward all nonlocal traffic to a distribution router, so it becomes unnecessary for the remote router to hold a complete routing table. Generally, the distribution router sends only a default route to the remote router.

Only specified routes are propagated from the remote (stub) router. The stub router responds to all queries for summaries, connected routes, redistributed static routes, external routes, and internal routes with the message "inaccessible." A router that is configured as a stub sends a special peer information packet to all neighboring routers to report its status as a stub router.

Any neighbor that receives a packet that informs it of the stub status does not query the stub router for any routes, and a router that has a stub peer does not query that peer. The stub router depends on the distribution router to send the proper updates to all peers. The following figure shows a simple hub-and-spoke network.

Figure 2 : Simple Hub-and-Spoke Network



Stub routing does not prevent routes from being advertised to the remote router. This figure shows that the remote router can access the corporate network and the Internet through the distribution router only. A full route table on the remote router, in this example, serves no functional purpose because the path to the corporate network and the Internet is always through the distribution router. A larger route table only increases the amount of memory consumed by the remote router. The bandwidth and memory used can be lessened by summarizing and filtering routes in the distribution router. In this network topology, the remote router does not need to receive routes that have been learned from other networks because the remote router must send all non-local traffic, regardless of its destination, to the distribution router. To configure a true stub network, you should configure the distribution router to send only a default route to the remote router.

OSPF supports stub areas and EIGRP supports stub routers.

1.1.2 Routing Algorithms

Routing algorithms determine how a router gathers and reports reachability information, how it deals with topology changes, and how it determines the optimal route to a destination. Various types of routing algorithms exist, and each algorithm has a different impact on network and router resources. Routing algorithms use a variety of metrics that affect calculation of optimal routes. You can classify routing algorithms by type, such as static or dynamic, and interior or exterior.

Static Routes and Dynamic Routing Protocols

Static routes are route table entries that you manually configure. These static routes do not change unless you reconfigure them. Static routes are simple to design and work well in environments where network traffic is relatively predictable and where network design is relatively simple.

Because static routing systems cannot react to network changes, you should not use them for large, constantly changing networks. Most routing protocols today use dynamic routing algorithms that adjust to changing network circumstances by analyzing incoming routing update messages. If the message indicates that a network change has occurred, the routing software recalculates routes and sends out new routing update messages. These messages permeate the network, triggering routers to rerun their algorithms and change their routing tables accordingly.

You can supplement dynamic routing algorithms with static routes where appropriate. For example, you should configure each subnetwork with a static route to the IP default gateway or router of last resort (a router to which all unrouteable packets are sent).

Interior and Exterior Gateway Protocols

You can separate networks into unique routing domains or autonomous systems. An autonomous system is a portion of an internetwork under common administrative authority that is regulated by a particular set of administrative guidelines. Routing protocols that route between autonomous systems are called exterior gateway protocols or interdomain protocols. The Border Gateway Protocol (BGP) is an example of an exterior gateway protocol. Routing protocols used within an autonomous system are called interior gateway protocols or intradomain protocols. EIGRP and OSPF are examples of interior gateway protocols.

Distance Vector Protocols

Distance vector protocols use distance vector algorithms (also known as Bellman-Ford algorithms) that call for each router to send all or some portion of its routing table to its neighbors. Distance vector algorithms define routes by distance (for example, the number of hops to the destination) and direction (for example, the next-hop router). These routes are then broadcast to the directly connected neighbor routers. Each router uses these updates to verify and update the routing tables.

To prevent routing loops, most distance vector algorithms use split horizon with poison reverse which means that the routes learned from an interface are set as unreachable and advertised back along the interface that they were learned on during the next periodic update. This process prevents the router from seeing its own route updates coming back.

Distance vector algorithms send updates at fixed intervals but can also send updates in response to changes in route metric values. These triggered updates can speed up the route convergence time. The Routing Information Protocol (RIP) is a distance vector protocol.

Link-State Protocols

The link-state protocols, also known as shortest path first (SPF), share information with neighboring routers. Each router builds a link-state advertisement (LSA) that contains information about each link and directly connected neighbor router.

Each LSA has a sequence number. When a router receives an LSA and updates its link-state database, the LSA is flooded to all adjacent neighbors. If a router receives two LSAs with the same sequence number (from the same router), the router does not flood the last LSA that it received to its neighbors because it wants to prevent an LSA update loop. Because the router floods the LSAs immediately after it receives them, the convergence time for link-state protocols is minimized.

Discovering neighbors and establishing adjacency is an important part of a link state protocol. Neighbors are discovered using special Hello packets that also serve as keepalive notifications to each neighbor router.

Adjacency is the establishment of a common set of operating parameters for the link-state protocol between neighbor routers.

The LSAs received by a router are added to the router's link-state database. Each entry consists of the following parameters:

- Router ID (for the router that originated the LSA)
- Neighbor ID
- Link cost
- Sequence number of the LSA
- Age of the LSA entry

The router runs the SPF algorithm on the link-state database, building the shortest path tree for that router. This SPF tree is used to populate the routing table.

In link-state algorithms, each router builds a picture of the entire network in its routing tables. The link-state algorithms send small updates everywhere, while distance vector algorithms send larger updates only to neighboring routers.

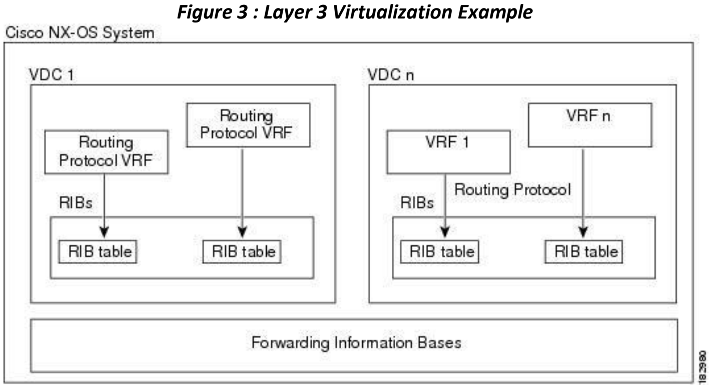
Because they converge more quickly, link-state algorithms are less likely to cause routing loops than distance vector algorithms. However, link-state algorithms require more CPU power and memory than distance vector algorithms and they can be more expensive to implement and support. Link-state protocols are generally more scalable than distance vector protocols.

OSPF is an example of a link-state protocol.

1.1.3 Layer 3 Virtualization

Inspur INOS uses a virtual device context (VDC) to provide separate management domains per VDC and software fault isolation. Each VDC supports multiple virtual routing and forwarding instances and multiple routing information bases (RIBs) to support multiple address domains. Each VRF is associated with a RIB and this information is collected by the Forwarding Information Base (FIB). The following figure shows the relationship between a VDC, a VRF, and a Inspur INOS device.

Figure 3 : Layer 3 Virtualization Example



A VRF represents a Layer 3 addressing domain. Each Layer 3 interface (logical or physical) belongs to one VRF. A VRF belongs to one VDC. Each VDC can support multiple VRFs.

See the Inspur CN12700 Series INOS Virtual Device Context Configuration Guide for information about VDCs.

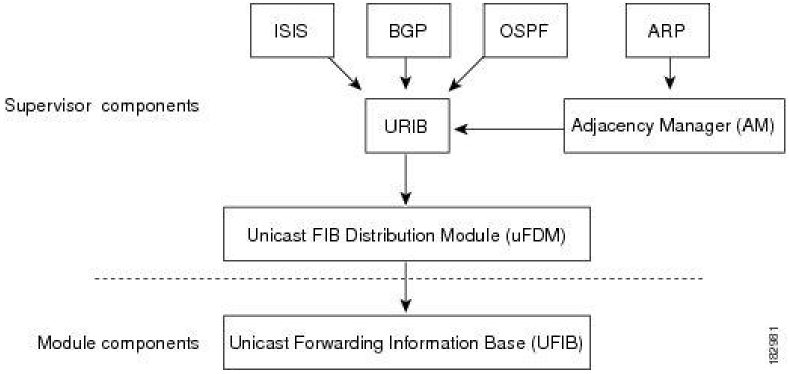
1.1.4 Inspur INOS Forwarding Architecture

The Inspur INOS forwarding architecture is responsible for processing all routing updates and populating the forwarding information to all modules in the chassis.

Unicast RIB

The Inspur INOS forwarding architecture consists of multiple components, as shown in the following figure.

Figure 4 : Inspur INOS Forwarding Architecture



The unicast RIB exists on the active supervisor. It maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols. The unicast RIB also collects adjacency information from sources such as the Address Resolution Protocol (ARP). The unicast RIB determines the best next hop for a given route and populates the unicast forwarding information base (FIB) by using the services of the unicast FIB distribution module (FDM).

Each dynamic routing protocol must update the unicast RIB for any route that has timed out. The unicast RIB then deletes that route and recalculates the best next hop for that route (if an alternate path is available).

Adjacency Manager

The adjacency manager exists on the active supervisor and maintains adjacency information for different protocols including ARP, Neighbor Discovery Protocol (NDP), and static configuration. The most basic adjacency information is the Layer 3 to Layer 2 address mapping discovered by these protocols. Outgoing Layer 2 packets use the adjacency information to complete the Layer 2 header.

The adjacency manager can trigger ARP requests to find a particular Layer 3 to Layer 2 mapping. The new mapping becomes available when the corresponding ARP reply is received and processed. For IPv6, the adjacency manager finds the Layer 3 to Layer 2 mapping information from NDP.

Unicast Forwarding Distribution Module

The unicast Forwarding Distribution Module (FDM) exists on the active supervisor and distributes the forwarding path information from the unicast RIB and other sources. The unicast RIB generates forwarding information that the unicast FIB programs into the hardware forwarding tables on the standby supervisor and the modules. The unicast FDM also downloads the FIB information to newly inserted modules.

The unicast FDM gathers adjacency information, rewrite information, and other platform-dependent information when updating routes in the unicast FIB. The adjacency and rewrite information consists of interface, next hop, and Layer 3 to Layer 2 mapping information. The interface and next-hop information is received in route updates from the unicast RIB. The Layer 3 to Layer 2 mapping is received from the adjacency manager.

FIB

The unicast FIB exists on supervisors and switching modules and builds the information used for the hardware forwarding engine. The unicast FIB receives route updates from the unicast FDM and sends the information to be programmed in the hardware forwarding engine. The unicast FIB controls the addition, deletion, and modification of routes, paths, and adjacencies.

The unicast FIBs are maintained on a per-VRF and per-address-family basis, that is, one for IPv4 and one for IPv6 for each configured VRF. Based on route update messages, the unicast FIB maintains a per-VRF prefix and next-hop adjacency information database. The next-hop adjacency data structure contains the next-hop IP address and the Layer 2 rewrite information. Multiple prefixes could share a next-hop adjacency information structure.

Hardware Forwarding

Inspur INOS supports distributed packet forwarding. The ingress port takes relevant information from the packet header and passes the information to the local switching engine. The local switching engine does the Layer 3 lookup and uses this information to rewrite the packet header. The ingress module forwards the packet to the egress port. If the egress port is on a different module, the packet is forwarded using the switch fabric to the egress module. The egress module does not participate in the Layer 3 forwarding decision.

The forwarding tables are identical on the supervisor and all the modules.

You also use the **show platform fib** or **show platform forwarding** commands to display details on hardware forwarding.

Software Forwarding

The software forwarding path in Inspur INOS is used mainly to handle features that are not supported in the hardware or to handle errors encountered during the hardware processing. Typically, packets with IP options or packets that need fragmentation are passed to the CPU on the active supervisor. All packets that should be switched in the software or terminated go to the supervisor. The supervisor uses the information provided by the unicast RIB and the adjacency manager to make the forwarding decisions. The module is not involved in the software forwarding path.

Software forwarding is controlled by control plane policies and rate limiters. For more information, see the *Inspur CN12700 Series INOS Security Configuration Guide*.

1.1.5 Layer 3 Interoperation with the CN12700-F3 Module

Layer 3 routing functionality comes up automatically when you have one of the CN12700-F Series modules installed in the chassis with the CN12700-F3 module. You would usually position a chassis with both the CN12700-F3 module, or a mixed chassis, at the boundary between the Layer 2 and Layer 3 networks.

You must configure a VLAN interface for each VLAN on the CN12700-F3 module that you want to use the proxy-routing functionality in a mixed chassis. (*See the Inspur CN12700 Series INOS Interfaces Configuration Guide* for information about configuring VLAN interfaces.)

By default, all of the physical interfaces on the CN12700-F series modules in the VDC become proxy routing ports for the VLANs that are configured with VLAN interfaces on the Layer 2-only CN12700-F3 module in the same VDC. The physical interfaces on the M Series module can be administratively down and still pass traffic as proxy forwarding.

Packets that enter an interface on the CN12700-F3 module are automatically forwarded to one of the interfaces on the M Series modules in the same VDC to be routed. The interface on the F Series module also performs egress replication for Layer 3 multicast packets that enter an interface on the CN12700-F3 module in the same VDC.

Because the Layer 3 (proxy routing) traffic from the CN12700-F3 module adds to the traffic that the F Series modules are already processing, the device automatically provides load balancing for the total traffic load among the front panel ports of the available F Series modules in the VDC. If you add or remove interfaces to the F Series module in the VDC, the device automatically rebalances the traffic. Note that proxy routing is sharing the forwarding capacity of the F Series modules. Removing interfaces reduces the amount of capacity available.

Instead of using the automatically configured proxy-routing interfaces on the F Series module, you can optionally configure which interfaces on the F Series module in the VDC performs proxy routing.

1.1.6 Summary of Layer 3 Routing Features

This section provides a brief introduction to the Layer 3 unicast features and protocols supported in Inspur INOS.

IPv4 and IPv6

Layer 3 uses either the IPv4 or IPv6 protocol. IPv6 is a new IP protocol designed to replace IPv4, the Internet protocol that is predominantly deployed and used throughout the world. IPv6 increases the number of network address bits from 32 bits (in IPv4) to 128 bits.

IP Services

IP Services includes Dynamic Host Configuration Protocol (DHCP) and Domain Name System (DNS Client) clients.

OSPF

The Open Shortest Path First (OSPF) protocol is a link-state routing protocol used to exchange network reachability information within an autonomous system. Each OSPF router advertises information about its active links to its neighbor routers. Link information consists of the link type, the link metric, and the neighbor router that is connected to the link. The advertisements that contain this link information are called link-state advertisements.

EIGRP

The Enhanced Interior Gateway Routing Protocol (EIGRP) is a unicast routing protocol that has the characteristics of both distance vector and link-state routing protocols. It is an improved version of IGRP, which is a Inspur proprietary routing protocol. EIGRP relies on its neighbors to provide the routes. It constructs the network topology from the routes advertised by its neighbors, similar to a lin-state protocol, and uses this information to select loop-free paths to destinations.

IS-IS

The Intermediate System-to-Intermediate System (IS-IS) protocol is an intradomain Open System Interconnection (OSI) dynamic routing protocol specified in the International Organization for Standardization (ISO) 10589. The IS-IS routing protocol is a link-state protocol. IS-IS features are as follows:

- Hierarchical routing
- Classless behavior
- Rapid flooding of new information
- Fast Convergence
- Very scalable

BGP

The Border Gateway Protocol (BGP) is an inter-autonomous system routing protocol. A BGP router advertises network reachability information to other BGP routers using Transmission Control Protocol (TCP) as its reliable transport mechanism. The network reachability information includes the destination network prefix, a list of autonomous systems that needs to be traversed to reach the destination, and the next-hop router.

Reachability information contains additional path attributes such as preference to a route, origin of the route, community and others..

RIP

The Routing Information Protocol (RIP) is a distance-vector protocol that uses a hop count as its metric. RIP is widely used for routing traffic in the global Internet and is an Interior Gateway Protocol (IGP), which means that it performs routing within a single autonomous system.

Static Routing

Static routing allows you to enter a fixed route to a destination. This feature is useful for small networks where the topology is simple. Static routing is also used with other routing protocols to control default routes and route distribution.

Layer 3 Virtualization

Virtualization allows you to share physical resources across separate management domains. Inspur INOS supports Virtual Device Contexts (VDCs) that allow you to create separate virtual systems within a Inspur INOS system. Each VDC is isolated from the others, which means that a problem in one VDC does not affect any other VDCs. VDCs are also secure from each other. You can assign separate network operators to each VDC and these network operators cannot control or view the configuration of a different VDC.

Inspur INOS also supports Layer 3 virtualization with virtual routing and forwarding (VRF). VRF provides a separate address domain for configuring Layer 3 routing protocols.

Route Policy Manager

The Route Policy Manager provides a route filtering capability in Inspur INOS. It uses route maps to filter routes distributed across various routing protocols and between different entities within a given routing protocol. Filtering is based on specific match criteria, which is similar to packet filtering by access control lists.

Policy-Based Routing

Policy-based routing uses the Route Policy Manager to create policy route filters. These policy route filters can forward a packet to a specified next hop based on the source of the packet or other fields in the packet header. Policy routes can be linked to extended IP access lists so that routing might be based on protocol types and port numbers.

First Hop Redundancy Protocols

First hop redundancy protocols (FHRP), such as Gateway Load Balancing Protocol (GLBP), Hot Standby Router Protocol (HSRP), and Virtual Router Redundancy Protocol (VRRP), allow you to provide redundant connections to your hosts. If an active first-hop router fails, the FHRP automatically selects a standby router to take over. You do not need to update the hosts with new IP addresses since the address is virtual and shared between each router in the FHRP group.

Object Tracking

Object tracking allows you to track specific objects on the network, such as the interface line protocol state, IP routing, and route reachability, and take action when the tracked object's state changes. This feature allows you to increase the availability of the network and shorten the recovery time if an object state goes down.

1.1.7 Related Documents for Layer 3 Unicast Routing

Feature Name	**Feature Information**
Layer 3 features	*Inspur CN12700 Series INOS Multicast Routing Configuration Guide* *Inspur CN12700 Series INOS High Availability and Redundancy Guide* *Inspur CN12700 Series INOS Virtual Device Context*

CHAPTER 2 Configuring IPv4

This chapter contains the following sections:

- Finding Feature Information.
- Information About IPv4.
- Virtualization Support for IPv4.
- IP Directed Broadcasts.
- Licensing Requirements for IPv4.
- Prerequisites for IPv4.
- Guidelines and Limitations for IPv4.
- Default Settings for IPv4 Parameters.
- Configuring IPv4.
- Verifying the IPv4 Configuration.
- Configuration Examples for IPv4.
- Related Documents for IPv4.
- Standards for IPv4.
- Feature History for IPv4.

2.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

2.2 Information About IPv4

You can configure IP on the device to assign IP addresses to network interfaces. When you assign IP addresses, you enable the interfaces and allow communication with the hosts on those interfaces.

You can configure an IP address as primary or secondary on a device. An interface can have one primary IP address and multiple secondary addresses. All networking device on an interface should share the same primary IP address because the packets that are generated by the device always use the primary IPv4 address. Each IPv4 packet is based on the information from a source or destination IP address.

You can use a subnet to mask the IP addresses. A mask is used to determine what subnet an IP address belongs to. An IP address contains the network address and the host address. A mask identifies the bits that denote the network number in an IP address. When you use the mask to subnet a network, the mask is then referred to as a subnet mask. Subnet masks are 32-bit values that allow the recipient of IP packets to distinguish the network ID portion of the IP address from the host ID portion of the IP address.

The IP feature is responsible for handling IPv4 packets that terminate in the supervisor module, as well as forwarding of IPv4 packets, which includes IPv4 unicast/multicast route lookup, reverse path forwarding (RPF) checks, and software access control list/policy-based routing (ACL/PBR) forwarding. The IP feature also manages the network interface IP address configuration, duplicate address checks, static routes, and packet send/receive interface for IP clients.

2.2.1 Multiple IPv4 Addresses

Inspur INOS supports multiple IP addresses per interface. You can specify an unlimited number of secondary addresses for a variety of situations.

The most common situations are as follows:

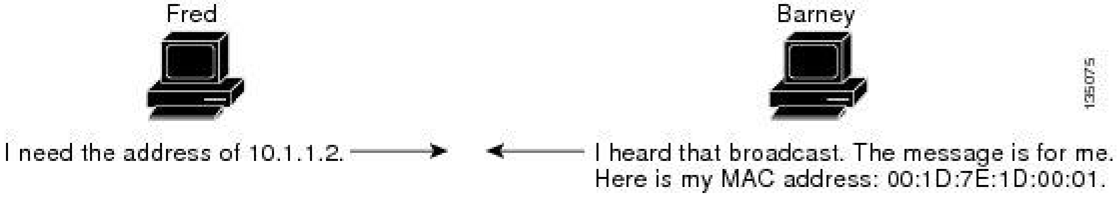
- When there are not enough host IP addresses for a particular network interface. For example, if your subnetting allows up to 254 hosts per logical subnet, but on one physical subnet you must have 300 host addresses, then you can use secondary IP addresses on the routers or access servers to allow you to have two logical subnets that use one physical subnet.
- Two subnets of a single network might otherwise be separated by another network. You can create a single network from subnets that are physically separated by another network by using a secondary address. In these instances, the first network is extended, or layered on top of the second network. A subnet cannot appear on more than one active interface of the router at a time.

2.2.2 Address Resolution Protocol

Networking devices and Layer 3 switches use Address Resolution Protocol (ARP) to map IP (network layer) addresses to (Media Access Control [MAC]-layer) addresses to enable IP packets to be sent across networks. Before a device sends a packet to another device, it looks in its own ARP cache to see if there is a MAC address and corresponding IP address for the destination device. If there is no entry, the source device sends a broadcast message to every device on the network.

Each device compares the IP address to its own. Only the device with the matching IP address replies to the device that sends the data with a packet that contains the MAC address for the device. The source device adds the destination device MAC address to its ARP table for future reference, creates a data-link header and trailer that encapsulates the packet, and proceeds to transfer the data.

Figure 5 : ARP Process


Fred
Barney
135075
I need the address of 10.1.1.2.
I heard that broadcast. The message is for me.
Here is my MAC address: 00:1D:7E:1D:00:01.

When the destination device lies on a remote network that is beyond another device, the process is the same except that the device that sends the data sends an ARP request for the MAC address of the default gateway. After the address is resolved and the default gateway receives the packet, the default gateway broadcasts the destination IP address over the networks connected to it. The device on the destination device network uses ARP to obtain the MAC address of the destination device and delivers the packet. ARP is enabled by default.

The default system-defined CoPP policy rate limits ARP broadcast packets bound for the supervisor module. The default system-defined CoPP policy prevents an ARP broadcast storm from affecting the control plane traffic but does not affect bridged packets.

2.2.3 ARP Caching

ARP caching minimizes broadcasts and limits wasteful use of network resources. The mapping of IP addresses to MAC addresses occurs at each hop (device) on the network for every packet sent over an internetwork, which may affect network performance.

ARP caching stores network addresses and the associated data-link addresses in the memory for a period of time, which minimizes the use of valuable network resources to broadcast for the same address each time that a packet is sent. You must maintain the cache entries that are set to expire periodically because the information might become outdated. Every device on a network updates its tables as addresses are broadcast.

To maintain the ARP entry, active MAC address-table entries and host routing adjacencies, Inspur INOS sends up to 3 unicast ARP request messages to devices that are present in the ARP cache. The first message is sent at 75% of the configured ARP timeout value, followed by two retries 30 and 60 seconds later if the cached entry has not already been refreshed.

2.2.4 Static and Dynamic Entries in the ARP Cache

Static routing requires that you manually configure the IP addresses, subnet masks, gateways, and corresponding MAC addresses for each interface of each device. Static routing requires more work to maintain the route table. You must update the table each time you add or change routes.

Dynamic routing uses protocols that enable the devices in a network to exchange routing table information with each other. Dynamic routing is more efficient than static routing because the route table is automatically updated unless you add a time limit to the cache. The default time limit is 25 minutes but you can modify the time limit if the network has many routes that are added and deleted from the cache.

2.2.5 Devices That Do Not Use ARP

When a network is divided into two segments, a bridge joins the segments and filters traffic to each segment based on MAC addresses. The bridge builds its own address table, which uses MAC addresses only. A device has an ARP cache that contains both IP addresses and the corresponding MAC addresses.

Passive hubs are central-connection devices that physically connect other devices in a network. They send messages out on all their ports to the devices and operate at Layer 1 but do not maintain an address table.

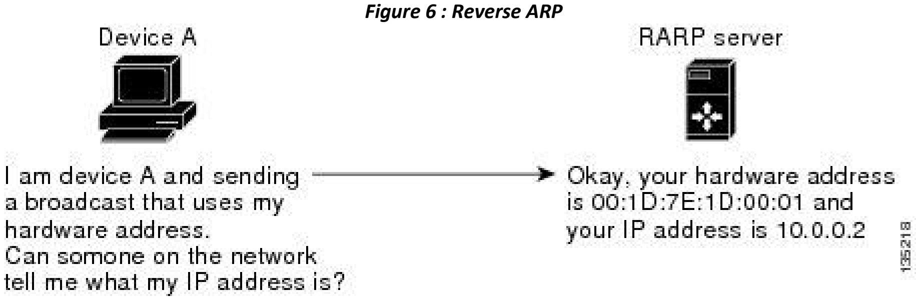
Layer 2 switches determine which port is connected to a device to which the message is addressed and sent only to that port. However, Layer 3 switches are devices that build an ARP cache (table).

2.2.6 Reverse ARP

Reverse ARP (RARP) as defined by RFC 903 works the same way as ARP, except that the RARP request packet requests an IP address instead of a MAC address. RARP often is used by diskless workstations because this type of device has no way to store IP addresses to use when they boot. The only address that is known is the MAC address because it is burned into the hardware.

Use of RARP requires an RARP server on the same network segment as the router interface.

Figure 6 : Reverse ARP



I am device A and sending a broadcast that uses my hardware address. Can somone on the network tell me what my IP address is?

Okay, your hardware address is 00:1D:7E:1D:00:01 and your IP address is 10.0.0.2

RARP has several limitations. Because of these limitations, most businesses use DHCP to assign IP addresses dynamically. DHCP is cost effective and requires less maintenance than RARP. The following are the most important limitations:

- Since RARP uses hardware addresses, if the internetwork is large with many physical networks, a RARP server must be on every segment with an additional server for redundancy. maintaining two servers for every segment is costly.
- Each server must be configured with a table of static mappings between the hardware addresses and IP addresses. Maintenance of the IP addresses is difficult.
- RARP only provides IP addresses of the hosts and not subnet masks or default gateways.

2.2.7 Proxy ARP

Proxy ARP enables a device that is physically located on one network appear to be logically part of a different physical network connected to the same device or firewall. Proxy ARP allows you to hide a device with a public IP address on a private network behind a router and still have the device appear to be on the public network in front of the router. By hiding its identity, the router accepts responsibility for routing packets to the real destination. Proxy ARP can help devices on a subnet reach remote subnets without configuring routing or a default gateway.

When devices are not in the same data link layer network but in the same IP network, they try to transmit data to each other as if they are on the local network. However, the router that separates the devices does not send a broadcast message because routers do not pass hardware-layer broadcasts and the addresses cannot be resolved.

When you enable Proxy ARP on the device and it receives an ARP request, it identifies the request as a request for a system that is not on the local LAN. The device responds as if it is the remote destination for which the broadcast is addressed, with an ARP response that associates the device's MAC address with the remote destination's IP address. The local device believes that it is directly connected to the destination, while in reality its packets are being forwarded from the local subnetwork toward the destination subnetwork by their local device. By default, Proxy ARP is disabled.

2.2.8 Local Proxy ARP

You can use local Proxy ARP to enable a device to respond to ARP requests for IP addresses within a subnet where normally no routing is required. When you enable local Proxy ARP, ARP responds to all ARP requests for IP addresses within the subnet and forwards all traffic between hosts in the subnet. Use this feature only on subnets where hosts are intentionally prevented from communicating directly by the configuration on the device to which they are connected.

2.2.9 Gratuitous ARP

Gratuitous ARP sends a request with an identical source IP address and a destination IP address to detect duplicate IP addresses. Inspur INOS supports enabling or disabling gratuitous ARP requests or ARP cache updates.

2.2.10 Glean Throttling

When forwarding an incoming IP packet in a line card, if the Address Resolution Protocol (ARP) request for the next hop is not resolved, the line card forwards the packets to the supervisor (glean throttling). The supervisor resolves the MAC address for the next hop and programs the hardware.

The Inspur CN12700 Series device hardware has glean rate limiters to protect the supervisor from the glean traffic. If the maximum number of entries is exceeded, the packets for which the ARP request is not resolved continues to be processed in the software instead of getting dropped in the hardware.

When an ARP request is sent, the software adds a /32 drop adjacency in the hardware to prevent the packets to the same next-hop IP address to be forwarded to the supervisor. When the ARP is resolved, the hardware entry is updated with the correct MAC address. If the ARP entry is not resolved before a timeout period, the entry is removed from the hardware.

2.2.11 Path MTU Discovery

Path maximum transmission unit (MTU) discovery is a method for maximizing the use of available bandwidth in the network between the endpoints of a TCP connection. It is described in RFC 1191. Existing connections are not affected when this feature is turned on or off.

2.2.12 ICMP

You can use the Internet Control Message Protocol (ICMP) to provide message packets that report errors and other information that is relevant to IP processing. ICMP generates error messages, such as ICMP destination unreachable messages, ICMP Echo Requests (which send a packet on a round trip between two hosts) and Echo Reply messages. ICMP also provides many diagnostic functions and can send and redirect error packets to the host. By default, ICMP is enabled.

Some of the ICMP message types are as follows:

- Network error messages
- Network congestion messages
- Troubleshooting information
- Timeout announcements

2.3 Virtualization Support for IPv4

IPv4 supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Inspur INOS Virtual Device Context Configuration Guide*.

2.4 IP Directed Broadcasts

An IP directed broadcast is an IP packet whose destination address is a valid broadcast address for an IP subnet, but which originates from a node that is not itself a part of that destination subnet.

A device that is not directly connected to its destination subnet forwards an IP directed broadcast in the same way that it forwards unicast IP packets destined for a host on that subnet. When a directed broadcast packet reaches a device that is directly connected to its destination subnet, that packet is broadcast on the destination subnet.

The destination address in the IP header of the packet is rewritten to the configured IP broadcast address for the subnet, and the packet is sent as a link-layer broadcast.

If directed broadcast is enabled for an interface, incoming IP packets whose addresses identify the packets as directed broadcasts that are intended for the subnet to which that interface is attached, are broadcasted on that subnet.

Use the **ip directed-broadcast** command on an interface to enable software forwarding of all IP directed broadcasts on that interface. Optionally, you can also use the **ip directed-broadcast** *acl-name* command to filter these broadcasts through an IP access list such that only those packets that pass through the access list are broadcast on the subnet. By default, IP directed broadcasts that are intended for the subnet to which a specific interface is attached are not forwarded at that interface if the IP Directed Broadcasts feature has not been enabled on that interface.

Hardware Forwarding of IP Directed Broadcasts

From Inspur INOS Release 8.4(1), all Inspur CN12700 Series I/O modules support hardware forwarding of IP directed broadcasts. This feature is limited to the VDC on which it is applied. Use the **ip directed-broadcast hw-assist** command on an interface to enable hardware forwarding of all IP directed broadcasts on that interface. This command prevents the IP directed broadcasts from being sent to the supervisor. Use the **ip directed-broadcast hw-assist drop** command on an interface to drop all IP directed broadcasts on that interface in the hardware.

You can use the **ip directed-broadcast hw-assist** command on an interface on which you have already used the **ip directed-broadcast** command. This will enable IP directed broadcasts with hardware-assist on that interface, and prevent the IP directed broadcasts from being sent to the supervisor.

If you have to configure hardware forwarding of IP directed broadcasts on an interface along with an ACL to filter the IP directed broadcast packets through an IP access list such that only those packets that pass through the access list are broadcast on the subnet, you have to manually configure an ACL on the egress of the interface on which the **ip directed-broadcast hw-assist** command has been used, and modify the ACL configuration to match the directed broadcast packets.

When you configure **ip directed-broadcast** *acl-name* command with the acl-name as **hw-assist**, you cannot delete this configuration after the ISSU. This is applicable to releases prior to Inspur INOS Release 8.4(1).

The following example shows an ACL sample configuration when you have configured hardware forwarding of IP directed broadcasts:

```
  ip access-list DirectedBroadcasts

   10 remark IOC Softchannels

   20 permit udp any any eq 5064

   30 permit udp any any eq 5065

   40 permit udp any any eq 5066

   50 permit udp any any eq 5067

   70  permit  udp  198.51.100.10/24
any   eq   7777   90   permit   udp
198.51.100.11/24  any  eq  7777  100
permit udp 198.51.100.248/24 any eq
7777
```

The following example shows how the above ACL sample configuration should be modified when hardware forwarding of IP directed broadcasts is enabled:

```
ip access-list DirectedBroadcasts

 10 remark IOC Softchannels

 20 permit udp any 172.26.40.255/24
 eq   5064   30   permit   udp   any
 172.26.40.255/24 eq 5065 40 permit
 udp any 172.26.40.255/24 eq 5066

 50 permit udp any 172.26.40.255/24
 eq 5067

 70   permit   udp   198.51.100.10/24
 172.26.40.255/24 eq 7777

 90   permit   udp   198.51.100.11/24
 172.26.40.255/24 eq 7777

 100  permit  udp  198.51.100.248/24
 172.26.40.255/24 eq 7777

 110 deny any 172.26.40.255/24
```

2.5 Licensing Requirements for IPv4

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you.

2.6 Prerequisites for IPv4

IPv4 has the following prerequisites:

- IPv4 can only be configured on Layer 3 interfaces.

2.7 Guidelines and Limitations for IPv4

IPv4 has the following configuration guidelines and limitations:

- You can configure a secondary IP address only after you configure the primary IP address.

- F3 Series modules do not support IPv4 tunnels.
- If any device on a network segment uses a secondary IPv4 address, other devices on that same network segment that require a secondary address must use a secondary address from the same network or subnet. The inconsistent use of secondary addresses on a network segment can quickly cause routing loops.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

2.8 Default Settings for IPv4 Parameters

Table 2 : Default IPv4 Parameters

Parameters	Default
ARP timeout	1500 seconds
proxy ARP	Disabled
Maximum number of IPv4 ARP entries in the neighbor adjacency table	131,072

2.9 Configuring IPv4

2.9.1 Configuring IPv4 Addressing

You can assign a primary IP address for a network interface.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 routed interface.
Step 4	switch(config-if)# **ip address** *ip-address/length*[*secondary*]	Specifies a primary or secondary IPv4 address for an interface. • The network mask can be a four-part dotted decimal address. For example, 255.0.0.0 indicates that each bit equal to 1 means the corresponding address bit belongs to the network address. • The network mask can be indicated as a slash (/) and a number—a prefix length. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). A slash must precede the decimal value and there is no space between the IP address and the slash.

Step 5	(Optional) switch(config-if)# **show ip interface**	Displays interfaces configured for IPv4.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example
This example shows how to assign an IPv4 address:

```
switch#   configure   terminal
switch(config)#       interface  ethernet   2/3
switch(config-if)# no switchport
switch(config-if)# ip address 192.2.1.1.255.0.0.0
switch(config-if)# copy running-config startup-config
switch(config-if)#
```

2.9.2 Configuring Multiple IPv4 Addresses

You can only add secondary IP addresses after you configure primary IP addresses.

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 routed interface.
Step 4	switch(config-if)# **ip address** *ip-address/length*[*secondary*]	Specifies a the configured address as a secondary IPv4 address.
Step 5	(Optional) switch(config-if)# **show ip interface**	Displays interfaces configured for IPv4.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

2.9.3 Configuring a Static ARP Entry

Configure a static ARP entry on the device to map IP addresses to MAC hardware addresses, including static multicast MAC addresses.

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config)# **no switchport**	Configures the interface as a Layer 3 routed interface.
Step 4	switch(config-if)# **ip arp address** *ip-address mac-address*	Associates an IP address with a MAC address as a static entry.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to assign a static ARP entry:

```
switch#   configure   terminal
switch(config)#      interface ethernet   2/3
switch(config-if)# no switchport
switch(config-if)#          ip          arp 192.2.1.1.0019.076c.1a78
switch(config-if)# copy      running-config      startup-config s
witch(config-if)#
```

2.9.4 Configuring Proxy ARP

Configure proxy ARP on the device to determine the media addresses of hosts on other networks or subnets.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config-if)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 routed interface.
Step 4	switch(config-if)# **ip proxy arp**	Enables proxy ARP on the interface.
Step 5	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure proxy ARP:

```
switch#          configure          terminal
switch(config)#      interface  ethernet   2/3
switch(config-if)# no switchport
switch(config-if)#  ip  proxy-arp

switch(config-if)#  copy  running-config  startup-config

switch(config-if)#
```

2.9.5 Configuring Local Proxy ARP

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config)# **no switchport**	Configures the interface as a Layer 3 routed interface.
Step 4	switch(config-if)# **ip local-proxy-arp**	Enables local proxy ARP on the interface.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure local proxy ARP:

```
switch#  configure  terminal
switch(config)#   interface ethernet 2/3
switch(config-if)# no switchport

switch(config-if)# ip local-proxy-arp

switch(config-if)# copy running-config startup-config

switch(config-if)#
```

2.9.6 Configuring Gratuitous ARP

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 routed interface.

Step 4	switch(config-if)# **ip arp gratuitous** {**request** | **update**]	Enables gratuitous ARP on the interface. Gratuitous ARP is enabled by default.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example
This example shows how to configure gratuitous ARP:

```
switch#           configure          terminal
switch(config)#       interface  ethernet   2/3
switch(config-if)# no switchport

switch(config-if)#  ip arp gratuitous  request
switch(config-if)#     copy     running-config  startup-config
switch(config-if)#
```

2.9.7 Configuring the IP ARP Cache Limit

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip arp cache limit** *max-arp-entries* [**syslog** *syslogs-per-second*]	Configures the maximum number of ARP entries in the neighbor adjacency table. The range is from 1 to 409600. The syslog keyword configures the number of syslogs per second. The range is from 1 to 1000. If you do not configure a limit, system logs appear on the console if you try to add an adjacency after reaching the default limit. If you configure a limit for IPv4 ARP entries, system logs appear if you try to add an adjacency after reaching the configured limit.
Step 3	switch(config)# **show ip adjacency summary**	Displays the global limit of the neighbor adjacency table and a summary of throttle adjacencies.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves this configuration change.

2.9.8 Configuring Glean Optimization

You can configure glean optimization to improve the performance of glean packets by reducing the processing of the packets in the supervisor. Glean optimization applies to glean packets where the destination IP address is part of the same subnet and does not apply to packets where the destination IP address is in a different subnet. The default is enabled.

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# [**no**] **ip arp fast-path**	Enables glean optimization. Use the **no** form of the command to disable this feature.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves this configuration change.

2.9.9 Configuring Path MTU Discovery

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip tcp path-mtu-discovery**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip tcp path-mtu-discovery**	Enables path MTU discovery.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

2.9.10 Configuring IP Packet Verification

Inspur INOS supports an Intrusion Detection System (IDS) that checks for IP packet verification. You can enable or disable these IDS checks.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ip verify address** {**destination zero** | **identical** | **reserved** | **source** {**broadcast** | **multicast**}}	Performs the following IDS checks on the IP address: • destination zero—Drops IP packets if the destination IP address is 0.0.0.0. • identical—Drops IP packets if the source IP address is identical to the destination IP address. • reserved—Drops IP packets if the IP address is in the 127.x.x.x range. • source—Drops IP packets if the IP source address is either 255.255.255.255 (broadcast) or in the 224.x.x.x range (multicast).
Step 3	switch(config)# **hardware ip verify checksum**	Drops IP packets if the packet checksum is invalid.
Step 4	switch(config)# **hardware ip verify fragment**	Drops IP packets if the packet fragment has a nonzero offset and the DF bit is active.
Step 5	switch(config)# **hardware ip verify length** {**consistent** | **maximum** {**max-frag** | **max-tcp** | **udp**} | **minimum**}	Performs the following IDS checks on the IP address: • consistent— Drops IP packets where the Ethernet frame size is greater than or equal to the IP packet length plus the Ethernet header. • maximum max-frag—Drops IP packets if the maximum fragment offset is greater than 65536. • maximum max-tcp—Drops IP packets if the TCP length is greater than the IP payload length. • maximum udp—Drops IP packets if the IP payload length is less than the UDP packet length. • minimum—Drops IP packets if the Ethernet frame length is less than the IP packet length plus four octets (the CRC length).
Step 6	switch(config)# **hardware ip verify tcp tiny-frag**	Drops TCP packets if the IP fragment offset is 1, or if the IP fragment offset is 0 and the IP payload length is less than 16.
Step 7	switch(config)# **hardware ip verify version**	Drops IP packets if the ethertype is not set to 4 (IPv4).

What to do next

Use the **show hardware forwarding ip verify** command to display the IP packet verification configuration.

2.9.11 Enabling Forwarding of IP Directed Broadcasts

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Specify the interface on which forwarding of IP directed broadcasts should be configured and enter interface configuration mode:
switch(config)# **interface** *type slot / port*

Step 3 Enable forwarding of IP directed broadcasts:
switch(config-if)# **ip directed-broadcast** [*acl-name* | **hw-assist [drop]**]

Note

- Use the **ip directed-broadcast** command to enable software forwarding of IP directed broadcasts.
- Use the **ip directed-broadcast** *acl-name* command to filter the IP directed broadcast packets through the specified IP access list.
- Use the **ip directed-broadcast hw-assist** command to enable hardware forwarding of IP directed broadcasts.
- Use the **ip directed-broadcast hw-assist drop** command to enable dropping of all directed broadcast packets on that interface in the hardware.
- You can either use the **ip directed-broadcast** *acl-name* command or the **ip directed-broadcast hw-assist**
  command on an interface. However, you cannot use both the commands on the same interface.

Use the **ip directed-broadcast** command to enable software forwarding of IP directed broadcasts. Use the **ip directed-broadcast** *acl-name* command to filter the IP directed broadcast packets through the specified IP access list. Use the **ip directed-broadcast hw-assist** command to enable hardware forwarding of IP directed broadcasts. Use the **ip directed-broadcast hw-assist drop** command to enable dropping of all directed broadcast packets on that interface in hardware. You can either use the **ip directed-broadcast** *acl-name* command or the **ip directed-broadcast hw-assist** command on an interface. You cannot use both the commands on the same interface.

Step 4 (Optional) Display the running configuration on the specified interface:
switch# **show running-config** *interface*

Step 5 (Optional) Display forwarding information:
switch# **show forwarding interfaces**

Example: Running Configuration

This example shows a running configuration to enable software forwarding of IP directed broadcasts on a specific interface, followed by a verification command that displays the running configuration on that interface:

```
configure terminal
  interface vlan 11
  ip directed-broadcast
  .
  .
  .
switch# show running-config interface vlan 11
!Command: show running-config interface Vlan11
!Time: Fri Jul 21 14:42:00 2017
```

```
version 8.2(1)
interface Vlan11
  ip directed-broadcast
```

This example shows a running configuration to enable software forwarding of IP directed broadcasts on a specific interface along with an ACL to filter the IP directed broadcast packets through a specified IP access list, followed by a verification command that displays running configuration on that interface:

```
configure terminal
interface vlan 11
  ip directed-broadcast acl
  .
  .
  .
switch# show running-config interface vlan 11
!Command: show running-config interface Vlan11
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Vlan11
 ip directed-broadcast acl
```

This example shows a running configuration to enable hardware forwarding of IP directed broadcasts on a specific interface, followed by verification commands that display the running configuration and forwarding information:

```
configure terminal
interface vlan11
  ip directed-broadcast hw-assist
  .
  .
  .
switch# show running-config interface Ethernet2/5
!Command:     show     running-config     interface
Ethernet2/5
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Vlan11
 ip directed-broadcast hw-assist

switch# show forwarding interfaces
slot 2
=======
Vlan11, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-mode = none,
bcast-mode
 = pu
nt, mac address = 0022.557a.5341
sup-eth2, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-mode
= none, bcast-mode =
punt, mac address = 0000.0000.0000
Ethernet2/5, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-mode
= none, bcast-mode
= fwd, mac address = 0022.557a.5341
Ethernet12/17, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-mode
 = none, bcast-mo
de = drop, mac address = 0022.557a.5341
Slot 4
======
.
.
.
switch# show forwarding interfaces | i Ethernet2/5
Ethernet2/5, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-
mode = none, bcast-mode
= fwd, mac address = 0022.557a.5341
```

This example shows a running configuration to enable dropping of all the IP directed broadcasts in the hardware on a specific interface, followed by a verification command that displays the running configuration on that interface:

```
configure terminal
interface vlan 11
  ip directed-broadcast hw-assist drop
  .
  .
  .
switch# show running-config interface vlan 11
!Command: show running-config interface Vlan11
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Vlan11
 ip directed-broadcast hw-assist drop
```

2.9.12 Disabling Forwarding of IP Directed Broadcasts

Step 1 Enter global configuration mode:
switch# **configure terminal**

Step 2 Specify the interface on which forwarding of IP directed broadcasts should be configured and enter interface configuration mode:
switch(config)# **interface** *type slot / port*

Step 3 Enable forwarding of IP directed broadcasts:
switch(config-if)# **ip directed-broadcast** [*acl-name* | **hw-assist [drop]**]

Note

- Use the **no ip directed-broadcast** command to disable forwarding of IP directed broadcasts.
- Use the **no ip directed-broadcast** *acl-name* command to disable forwarding of IP directed broadcasts on a specific interface along with the configured ACL.
- Use the **no ip directed-broadcast hw-assist** command to disable hardware forwarding of IP directed broadcasts and to disable dropping of all directed broadcasts on a specific interface in the hardware if the **ip directed-broadcast hw-assist drop** command has been used.
- Use the **no ip directed-broadcast hw-assist drop** command to disable dropping of all directed broadcasts on a specific interface in the hardware.

Step 4 (Optional) Display the running configuration on the specified interface:
switch# **show running-config** *interface*

Step 5 (Optional) Display forwarding information:
switch# **show forwarding** *interfaces*

Example: Running Configuration

This example shows a running configuration to disable forwarding of IP directed broadcasts on a specific interface, followed by a verification command that displays the running configuration on that interface:

```
configure terminal
interface vlan 11
  no ip directed-broadcast
  .
  .
  .
switch# show running-config interface vlan 11
!Command: show running-config interface Vlan11
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Vlan11
```

This example shows a running configuration to disable forwarding of IP directed broadcasts on a specific interface along with the configured ACL, followed by a verification command that displays the running configuration on that interface:

```
configure terminal
interface vlan 11
  no ip directed-broadcast acl
  .
  .
  .
switch# show running-config interface vlan 11
!Command: show running-config interface Vlan11
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Vlan11
```

This example shows a running configuration to disable hardware forwarding of IP directed broadcasts on an interface, followed by verification commands that display the running configuration and forwarding information:

```
configure terminal
interface Ethernet2/5
  no ip directed-broadcast hw-assist
  .
  .
  .
switch# show running-config interface Ethernet2/5
!Command: show running-config interface Ethernet2/5
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Ethernet2/5

switch# show forwarding interfaces
slot 2
=======

Vlan11, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-mode =
none, bcast-mode
= pu
nt, mac address = 0022.557a.5341
sup-eth2, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-
mode = none, bcast-mode =
punt, mac address = 0000.0000.0000
Ethernet2/5, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-
mode = none, bcast-mode
= punt, mac address = 0022.557a.5341
Ethernet12/17, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-
 mode = none, bcast-mo
de = drop, mac address = 0022.557a.5341
Slot 4
======
.
.
.

switch# show forwarding interfaces | i Ethernet2/5
Ethernet2/5, v4 adj-count = 0, v6 adj-count = 0, v4_rpf-mode = none, v6_rpf-
mode = none, bcast-mode
```

```
= punt, mac address = 0022.557a.5341
```

This example shows a running configuration to disable dropping of all IP directed broadcasts in the hardware on a specific interface, followed by a verification command that displays the running configuration on that interface:

```
configure terminal
interface vlan 11
  no ip directed-broadcast hw-assist drop
.
.
.
switch# show running-config interface vlan 11
!Command: show running-config interface Vlan11
!Time: Fri Jul 21 14:42:00 2017
version 8.2(1)
interface Vlan11
```

2.9.13 Configuring IP Glean Throttling

Inspur INOS software supports glean throttling rate limiters to protect the supervisor from the glean traffic.

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ip glean throttle**	Enables ARP throttling.
Step 3	switch(config)# **no hardware ip glean throttle**	Disables ARP throttling.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example
This example shows how to enable IP glean throttling:

```
switch# configure terminal
switch(config)# hardware ip glean throttle
switch(config-if)#  copy  running-config  startup-config
```

2.9.14 Configuring the Hardware IP Glean Throttle Maximum

You can limit the maximum number of drop adjacencies that are installed in the Forwarding Information Base (FIB).

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose

Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ip glean throttle maximum** *count*	Configures the number of drop adjacencies that are installed in the FIB.
Step 3	switch(config)# **no hardware ip glean throttle maximum** *count*	Applies the default limits. The default value is 1000. The range is from 0 to 32767 entries.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to limit the maximum number of drop adjacencies that are installed in the FIB:

```
switch# configure terminal

switch(config)# hardware ip glean throttle maximum 2134

switch(config-if)# copy running-config startup-config
```

2.9.15 Configuring the Hardware IP Glean Throttle Timeout

You can configure a timeout for the installed drop adjacencies to remain in the Forwarding Information Base (FIB).

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ip glean throttle maximum timeout** *timeout-in-seconds*	Configures the timeout for the installed drop adjacencies to remain in the FIB.
Step 3	switch(config)# **no hardware ip glean throttle maximum timeout** *timeout-in-seconds*	Applies the default limits. The timeout value is in seconds. The range is from 300 seconds (5 minutes) to 1800 seconds (30 minutes). **Note** After the timeout period is exceeded, the drop adjacencies are removed from the FIB.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

2.9.16 Configuring the Hardware IP Glean Throttle Syslog

You can a syslog if the number of packets that get dropped for a specific flow exceeds the configured packet count.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ip glean throttle syslog** *packet-count*	Generates a syslog if the number of packets that get dropped for a specific flow exceed the configured packet count.
Step 3	switch(config)# **no hardware ip glean throttle syslog** *packet-count*	Applies the default limits. The default is 10000 packets. The range is from 0 to 65535 packets. **Note** After the timeout period is exceeded, the drop adjacencies are removed from the FIB.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to generate a syslog if the number of packets that get dropped for a specific flow exceeds the configured packet count:

```
switch# configure terminal

switch(config)# hardware ip glean throttle maximum timeout 300

switch(config-if)#  copy running-config startup-config
```

2.10 Verifying the IPv4 Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show forwarding interfaces	Displays forwarding information.
show hardware forwarding ip verify	Shows the IP packet verification configuration.
show ip adjacency	Displays the adjacency table.
show ip adjacency summary	Displays the summary of number of throttle adjacencies.
show ip arp	Displays the ARP table.
show ip arp summary	Displays the summary of the number of throttle adjacencies.

show ip adjacency throttle statistics	Displays only the throttle adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics [**vrf** *vrf-name*]	Displays the ARP statistics.
show running-config *interface*	Displays the running configuration on the specified interface.

2.11 Configuration Examples for IPv4

2.11.1 Example: Reserving All Ports on a Module for Proxy Routing

This example shows how to reserve all ports on a module for proxy routing:

Step 1: Determine which modules are present in the device:

```
switch# show module

Mod Ports Module-Type Model Status

--- ----- -------------------------------- ------------------ ------------

1 32 10 Gbps Ethernet Module CN12700-M132XP-12 ok
2 48 10/100/1000 Mbps Ethernet Module CN12700-M148GT-11 ok
3 48 1000 Mbps Optical Ethernet Modul CN12700-M148GS-11 ok
5 0 Supervisor module-1X CN12700-SUP1 active *
6 0 Supervisor module-1X CN12700-SUP1 ha-standby
8 32 1/10 Gbps Ethernet Module CN12700-F132XP-15 ok
```

The F3 module is in Slot 8, and the F3 modules are in Slots 1 to 3.
Step 2: Determine which ports are available in the VDC:

```
switch#  show  vdc  membership  |  end
"Ethernet3/48"  vdc_id:  0  vdc_name:
Unallocated  interfaces:  vdc_id:  1
vdc_name:     switch     interfaces:
Ethernet1/9 Ethernet1/10 Ethernet1/11
Ethernet1/12             Ethernet1/13
Ethernet1/14             Ethernet1/15
Ethernet1/16             Ethernet1/17
Ethernet1/18             Ethernet1/19
Ethernet1/20             Ethernet1/21
Ethernet1/22             Ethernet1/23
Ethernet1/24             Ethernet1/25
Ethernet1/26             Ethernet1/27
Ethernet1/28             Ethernet1/29
Ethernet1/30             Ethernet1/31
Ethernet1/32  Ethernet2/1  Ethernet2/2
Ethernet2/3

Ethernet2/4     Ethernet2/5
Ethernet2/6     Ethernet2/7
Ethernet2/8     Ethernet2/9
Ethernet2/10   Ethernet2/11
Ethernet2/12   Ethernet2/25
Ethernet2/26   Ethernet2/27
Ethernet2/28   Ethernet2/29
Ethernet2/30   Ethernet2/31
Ethernet2/32   Ethernet2/33
```

```
Ethernet2/34   Ethernet2/35
Ethernet2/36   Ethernet2/37
Ethernet2/38   Ethernet2/39
Ethernet2/40   Ethernet2/41
Ethernet2/42   Ethernet2/43
Ethernet2/44   Ethernet2/45
Ethernet2/46   Ethernet2/47
Ethernet2/48    Ethernet3/1
Ethernet3/2     Ethernet3/3
Ethernet3/4     Ethernet3/5
Ethernet3/6     Ethernet3/7
Ethernet3/8     Ethernet3/9
Ethernet3/10   Ethernet3/11
Ethernet3/12   Ethernet3/13
Ethernet3/14   Ethernet3/15
Ethernet3/16   Ethernet3/17
Ethernet3/18   Ethernet3/19
Ethernet3/20 Ethernet3/21

Ethernet3/22      Ethernet3/23
Ethernet3/24      Ethernet3/25
Ethernet3/26      Ethernet3/27
Ethernet3/28      Ethernet3/29
Ethernet3/30      Ethernet3/31
Ethernet3/32      Ethernet3/33
Ethernet3/34      Ethernet3/35
Ethernet3/36      Ethernet3/37
Ethernet3/38      Ethernet3/39
Ethernet3/40      Ethernet3/41
Ethernet3/42      Ethernet3/43
Ethernet3/44      Ethernet3/45
Ethernet3/46      Ethernet3/47
Ethernet3/48
```

Step 3: Determine which ports are available for proxy routing:

```
switch# show hardware proxy layer-3 detail

Global Information:

F3 Modules: Count: 1 Slot: 8

F3 Modules: Count: 3 Slot: 1-3
Replication   Rebalance   Mode:
Manual Number of proxy layer-3
forwarders: 13 Number of proxy
layer-3     replicators:     8
Forwarder   Interfaces   Status
Reason

------------------------------------------------------------------------------

Eth1/9, Eth1/11, Eth1/13, Eth1/15 up
SUCCESS  Eth1/10,  Eth1/12,  Eth1/14,
Eth1/16 up SUCCESS Eth1/17, Eth1/19,
Eth1/21, Eth1/23 up SUCCESS Eth1/18,
Eth1/20, Eth1/22, Eth1/24 up SUCCESS
Eth1/25, Eth1/27, Eth1/29, Eth1/31 up
SUCCESS  Eth1/26,  Eth1/28,  Eth1/30,
Eth1/32   up   SUCCESS   Eth2/1-12   up
SUCCESS
```

```
Eth2/25-36  up  SUCCESS  Eth2/37-48  up
SUCCESS Eth3/1-12 up SUCCESS Eth3/13-
24  up  SUCCESS  Eth3/25-36  up  SUCCESS
Eth3/37-48 up SUCCESS

Replicator Interfaces #Interface-Vlan
Interface-Vlan

-------------------------------------------------------------------------------

Eth1/1,  Eth1/3,  Eth1/5,  Eth1/7,
Eth1/9,   0   Eth1/11,   Eth1/13,
Eth1/15

Eth1/2,  Eth1/4,  Eth1/6,  Eth1/8,
Eth1/10,   0   Eth1/12,   Eth1/14,
Eth1/16

Eth1/17,   Eth1/19,   Eth1/21,
Eth1/23,  0 Eth1/25, Eth1/27,
Eth1/29,   Eth1/31   Eth1/18,
Eth1/20,  Eth1/22, Eth1/24,  0
Eth1/26,   Eth1/28,   Eth1/30,
Eth1/32 Eth2/1-24 0

Eth2/25-48 0

Eth3/1-24 0

Eth3/25-48 0

switch#
```

Step 4: Reserve a module for unicast and multicast proxy routing:

```
switch# configure terminal

switch(config)# hardware proxy layer-3 forwarding use module 2

switch(config)# hardware proxy layer-3 replication use module 2
```

Step 5: Verify this configuration:

```
switch(config)# show hardware proxy layer-3 detail

Global Information:

F3 Modules: Count: 1 Slot: 8

F3  Modules:  Count:  3  Slot:
1-3  Replication  Rebalance
Mode:   Manual   Number   of
proxy layer-3 forwarders: 3
Number   of   proxy   layer-3
replicators:   2  Forwarder
Interfaces Status Reason

-------------------------------------------------------------------------------

Eth2/1-12 up SUCCESS

Eth2/25-
36      up
```

```
                    SUCCESS
                    Eth2/37-
                    48      up
                    SUCCESS

                    Replicator Interfaces #Interface-Vlan Interface-Vlan

------------------------------------------------------------------------------

                    Eth2/1-24 0

                    Eth2/25-48 0

                    switch(config)#
```

2.11.2 Example: Reserving Ports for Proxy Routing

This example shows how to reserve some ports on a module for proxy routing: Step 1: Reserve a subset of ports on a module:

```
switch(config)# hardware proxy layer-3 forwarding use interface ethernet
2/1-6

switch(config)# hardware proxy layer-3 replication use interface ethernet 2/1-6
 <----subset of port group
```

This example reserves a subset of ports from a port group. Step 2: Verify this configuration:

```
                    switch(config)# show hardware proxy layer-3 detail

                    Global Information:

                    F3 Modules: Count: 1 Slot: 8

                    F3 Modules: Count: 3 Slot:
                    1-3  Replication  Rebalance
                    Mode:   Manual   Number   of
                    proxy layer-3 forwarders: 1
                    Number   of   proxy   layer-3
                    replicators:   1   Forwarder
                    Interfaces Status Reason

------------------------------------------------------------------------------

                    Eth2/1-12 up SUCCESS

                    Replicator Interfaces #Interface-Vlan Interface-Vlan

------------------------------------------------------------------------------

                    Eth2/1-24 0 switch(config)#
```

2.11.3 Example: Excluding Ports From Proxy Routing

The following example excludes some ports on a module for proxy routing:

```
switch(config)#  hardware  proxy  layer-3  forwarding  exclude  interface ethernet  2/1-12

switch(config)#  hardware  proxy  layer-3  replication exclude  interface  ethernet  2/1-12

switch(config)#  show  hardware  proxy layer-3 detail

Global Information:

F3 Modules: Count: 1 Slot: 8
```

```
F3  Modules:  Count:  3
Slot:  1-3  Replication
Rebalance Mode: Manual



Number   of   proxy   layer-3
forwarders: 12 Number of proxy
layer-3     replicators:     7
Forwarder  Interfaces  Status
Reason

------------------------------------------------------------------------------

Eth1/9, Eth1/11, Eth1/13, Eth1/15 up
SUCCESS  Eth1/10,  Eth1/12,  Eth1/14,
Eth1/16 up SUCCESS Eth1/17, Eth1/19,
Eth1/21, Eth1/23 up SUCCESS Eth1/18,
Eth1/20, Eth1/22, Eth1/24 up SUCCESS
Eth1/25, Eth1/27, Eth1/29, Eth1/31 up
SUCCESS  Eth1/26,  Eth1/28,  Eth1/30,
Eth1/32  up  SUCCESS  Eth2/25-36  up
SUCCESS

Eth2/37-48 up
SUCCESS
Eth3/1-12  up
SUCCESS
Eth3/13-24 up
SUCCESS
Eth3/25-36 up
SUCCESS
Eth3/37-48 up
SUCCESS

Replicator  Interfaces  #Interface-Vlan  Interface-
Vlan

------------------------------------------------------------------------------

Eth1/1,  Eth1/3,  Eth1/5,  Eth1/7,
Eth1/9,   0   Eth1/11,   Eth1/13,
Eth1/15

Eth1/2,  Eth1/4,  Eth1/6,  Eth1/8,
Eth1/10,   0   Eth1/12,   Eth1/14,
Eth1/16

Eth1/17,   Eth1/19,   Eth1/21,
Eth1/23,  0  Eth1/25,  Eth1/27,
Eth1/29,   Eth1/31   Eth1/18,
Eth1/20,  Eth1/22,  Eth1/24,  0
Eth1/26,   Eth1/28,   Eth1/30,
Eth1/32 Eth2/25-48 0

Eth3/1-24 0

Eth3/25-48 0

switch(config)#
```

2.12 Standards for IPv4

Standards	Title
No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.	—

2.13 Feature History for IPv4

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 3: Feature History for IPv4

Feature Name	Release	Feature Information
Hardware Forwarding of IP Directed Broadcasts	8.4(1)	This feature enables hardware forwarding of IP directed broadcasts. This feature is limited to the VDC on which it is applied.
Glean optimization	8.4(1)	This feature was introduced.
ARP	8.4(1)	Added the ability to configure the maximum number of ARP entries in the neighbor adjacency table.
IP	8.4(1)	Updated for F3 Series modules.
ACL filter for IP directed broadcasts	8.4(1)	Added support to filter IP directed broadcasts through an IP access list.
Glean throttling	8.4(1)	Added support for IPv4 glean throttling.
ARP	8.4(1)	Added support to protect against an ARP broadcast storm.
IP	8.4(1)	Changed the **platform ip verify** command to the **hardware ip verify** command.
ARP	8.4(1)	Added support for gratuitous ARP. The **ip arp gratuitous** {**request** | **update**} command was added.
IP	8.4(1)	This feature was introduced.

CHAPTER 3 Configuring IPv6

This chapter contains the following sections:

- Finding Feature Information.
- Information About IPv6.
- Virtualization Support for IPv6.
- Licensing Requirements for IPv6.
- Prerequisites for IPv6.
- Guidelines and Limitations for Configuring IPv6.
- Default Settings for IPv6.
- Configuring IPv6.
- Verifying the IPv6 Configuration.
- Configuration Example for IPv6.
- Related Documents for IPv6.
- Standards for IPv6.
- Feature History for IPv6.

3.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

3.2 Information About IPv6

IPv6, which is designed to replace IPv4, increases the number of network address bits from 32 bits (in IPv4) to 128 bits. IPv6 is based on IPv4 but it includes a much larger address space and other improvements such as a simplified main header and extension headers.

The larger IPv6 address space allows networks to scale and provide global reachability. The simplified IPv6 packet header format handles packets more efficiently. The flexibility of the IPv6 address space reduces the need for private addresses and the use of Network Address Translation (NAT), which translates private (not globally unique) addresses into a limited number of public addresses. IPv6 enables new application protocols that do not require special processing by border routers at the edge of networks.

IPv6 functionality, such as prefix aggregation, simplified network renumbering, and IPv6 site multihoming capabilities, enable more efficient routing. IPv6 supports Routing Information Protocol (RIP), Integrated Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF) for IPv6, and multiprotocol Border Gateway Protocol (BGP).

3.2.1 IPv6 Address Formats

An IPv6 address has 128 bits or 16 bytes. The address is divided into eight, 16-bit hexadecimal blocks separated by colons (:) in the format x:x:x:x:x:x:x:x.

Two examples of IPv6 addresses are as follows:

```
2001:0DB8:7654:3210:FEDC:BA98:7654:3210

2001:0DB8:0:0:8:800:200C:417A
```

IPv6 addresses contain consecutive zeros within the address. You can use two colons (::) at the beginning, middle, or end of an IPv6 address to replace the consecutive zeros.

You can use a double colon as part of the IPv6 address when consecutive 16-bit values are denoted as zero. You can configure multiple IPv6 addresses per interface but only one link-local address.

The hexadecimal letters in IPv6 addresses are not case sensitive.

Table 4 : Compressed IPv6 Address Formats

IPv6 AddressType	Preferred Format	Compressed Format
Unicast	2001:0:0:0:0:DB8:800:200C:417A	2001::0DB8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

A node may use the loopback address listed in the table to send an IPv6 packet to itself. The loopback address in IPv6 is the same as the loopback address in IPv4.

The IPv6-prefix is in the form documented in RFC 2373 where the IPv6 address is specified in hexadecimal using 16-bit values between colons. The prefix length is a decimal value that indicates how many of the high-order contiguous bits of the address comprise the prefix (the network portion of the address). For example, 2001:0DB8:8086:6502::/32 is a valid IPv6 prefix.

3.2.2 IPv6 Unicast Addresses

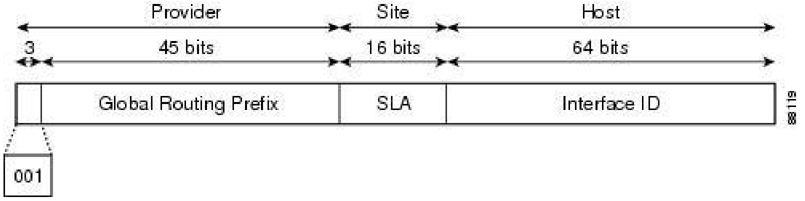
An IPv6 unicast address is an identifier for a single interface on a single node. A packet that is sent to a unicast address is delivered to the interface identified by that address.

Aggregatable Global Addresses

An aggregatable global address is an IPv6 address from the aggregatable global unicast prefix. The structure of aggregatable global unicast addresses enables strict aggregation of routing prefixes that limits the number of routing table entries in the global routing table. Aggregatable global addresses are used on links that are aggregated upward through organizations and eventually to the Internet service providers (ISPs).

Aggregatable global IPv6 addresses are defined by a global routing prefix, a subnet ID, and an interface ID. Except for addresses that start with binary 000, all global unicast addresses have a 64-bit interface ID. The IPv6 global unicast address allocation uses the range of addresses that start with binary value 001 (2000::/3). The figure shows the structure of an aggregatable global address.

Figure 7 : Aggregatable Global Addresses



Addresses with a prefix of 2000::/3 (001) through E000::/3 (111) are required to have 64-bit interface identifiers in the extended universal identifier (EUI)-64 format. The Internet Assigned Numbers Authority (IANA) allocates the IPv6 address space in the range of 2000::/16 to regional registries.

The aggregatable global address consists of a 48-bit global routing prefix and a 16-bit subnet ID or Site-Level Aggregator (SLA). In the IPv6 aggregatable global unicast address format document (RFC 2374), the global routing prefix included two other hierarchically structured fields called Top-Level Aggregator (TLA) and Next-Level Aggregator (NLA). The IETF decided to remove the TLS and NLA fields from the RFCs because these fields are policy based. Some existing IPv6 networks deployed before the change might still use networks that are on the older architecture.

A subnet ID, which is a 16-bit subnet field, can be used by individual organizations to create a local addressing hierarchy and to identify subnets. A subnet ID is similar to a subnet in IPv4, except that an organization with an IPv6 subnet ID can support up to 65,535 individual subnets.

An interface ID identifies interfaces on a link. The interface ID is unique to the link. In many cases, an interface ID is the same as or based on the link-layer address of an interface. Interface IDs used in aggregatable global unicast and other IPv6 address types have 64 bits and are in the modified EUI-64 format.

Interface IDs are in the modified EUI-64 format in one of the following ways:

- For all IEEE 802 interface types (for example, Ethernet, and Fiber Distributed Data interfaces), the first three octets (24 bits) are the Organizationally Unique Identifier (OUI) of the 48-bit link-layer address (MAC address) of the interface, the fourth and fifth octets (16 bits) are a fixed hexadecimal value of FFFE, and the last three octets (24 bits) are the last three octets of the MAC address. The Universal/Local (U/L) bit, which is the seventh bit of the first octet, has a value of 0 or 1. Zero indicates a locally administered identifier; 1 indicates a globally unique IPv6 interface identifier.
- For all other interface types (for example, serial, loopback, ATM, Frame Relay, and tunnel interface types-except tunnel interfaces used with IPv6 overlay tunnels), the interface ID is similar to the interface ID for IEEE 802 interface types; however, the first MAC address from the pool of MAC addresses in the router is used as the identifier (because the interface does not have a MAC address).
- For tunnel interface types that are used with IPv6 overlay tunnels, the interface ID is the IPv4 address assigned to the tunnel interface with all zeros in the high-order 32 bits of the identifier

If no IEEE 802 interface types are in the router, link-local IPv6 addresses are generated on the interfaces in the router in the following sequence:

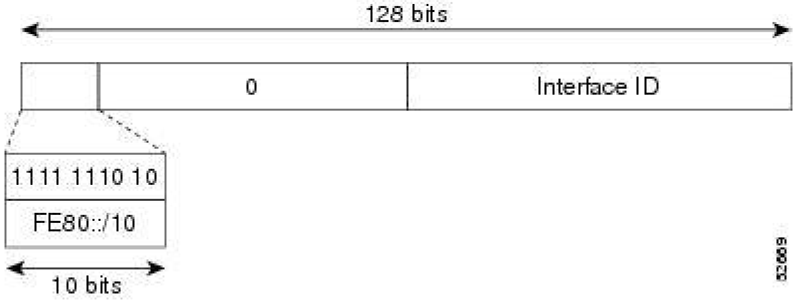
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, the router uses a Message Digest 5 (MD5) hash to determine the MAC address of the router from the hostname of the router.

Link-Local Addresses

A link-local address is an IPv6 unicast address that can be automatically configured on any interface using the link-local prefix FE80::/10 (1111 1110 10) and the interface identifier in the modified EUI-64 format. Link-local addresses are used in the Neighbor Discovery Protocol (NDP) and the stateless autoconfiguration process. Nodes on a local link can use link-local addresses to communicate; the nodes do not need globally unique addresses to communicate.

IPv6 routers cannot forward packets that have link-local source or destination addresses to other links.

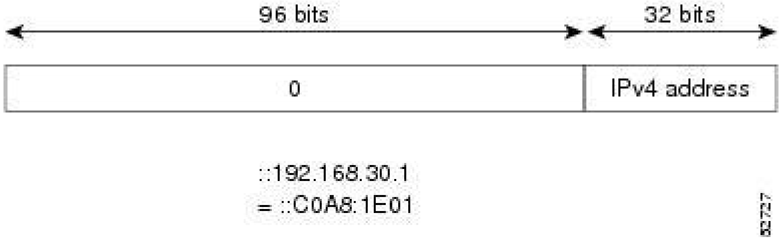
Figure 8 : Link-Local Address Format



IPv4-Compatible IPv6 Addresses

An IPv4-compatible IPv6 address is an IPv6 unicast address that has zeros in the high-order 96 bits of the address and an IPv4 address in the low-order 32 bits of the address. The format of an IPv4-compatible IPv6 address is 0:0:0:0:0:0:A.B.C.D or ::A.B.C.D. The entire 128-bit IPv4-compatible IPv6 address is used as the IPv6 address of a node and the IPv4 address embedded in the low-order 32 bits is used as the IPv4 address of the node. IPv4-compatible IPv6 addresses are assigned to nodes that support both the IPv4 and IPv6 protocol stacks and are used in automatic tunnels.

Figure 9 : IPv4-Compatible IPv6 Address Format



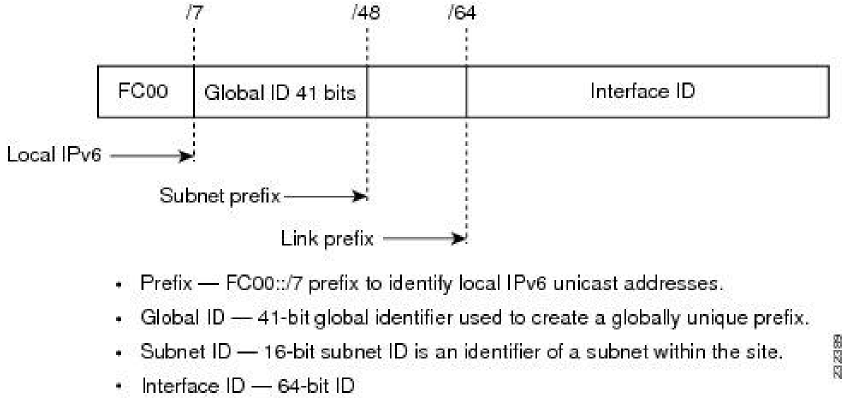
Unique Local Addresses

A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications may treat unique local addresses like global scoped addresses.

A unique local address has the following characteristics:

- It has a globally unique prefix (it has a high probability of uniqueness).
- It has a well-known prefix to allow for easy filtering at site boundaries
- It allows sites to be combined or privately interconnected without creating any address conflicts or requiring renumbering of interfaces that use these prefixes.
- It is ISP-independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity.
- If it is accidentally leaked outside of a site through routing or the Domain Name Server (DNS), there is no conflict with any other addresses.

Figure 10 : Unique Local Address Structure



- Prefix — FC00::/7 prefix to identify local IPv6 unicast addresses.
- Global ID — 41-bit global identifier used to create a globally unique prefix.
- Subnet ID — 16-bit subnet ID is an identifier of a subnet within the site.
- Interface ID — 64-bit ID

Site Local Addresses

Because RFC 3879 deprecates the use of site-local addresses, you should follow the recommendations of unique local addressing (ULA) in RFC 4193 when you configure private IPv6 addresses.

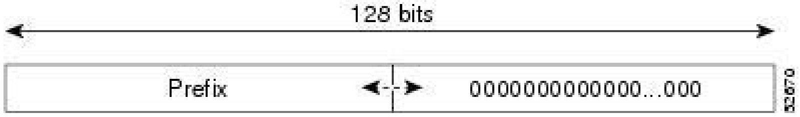
3.2.3 IPv6 Anycast Addresses

An anycast address is an address that is assigned to a set of interfaces that belong to different nodes. A packet sent to an anycast address is delivered to the closest interface-as defined by the routing protocols in

use-identified by the anycast address. Anycast addresses are syntactically indistinguishable from unicast addresses because anycast addresses are allocated from the unicast address space. Assigning a unicast address to more than one interface turns a unicast address into an anycast address. You must configure the nodes to which the anycast address to recognize that the address is an anycast address.

The following figure shows the format of the subnet router anycast address; the address has a prefix concatenated by a series of zeros (the interface ID). The subnet router anycast address can be used to reach a router on the link that is identified by the prefix in the subnet router anycast address.

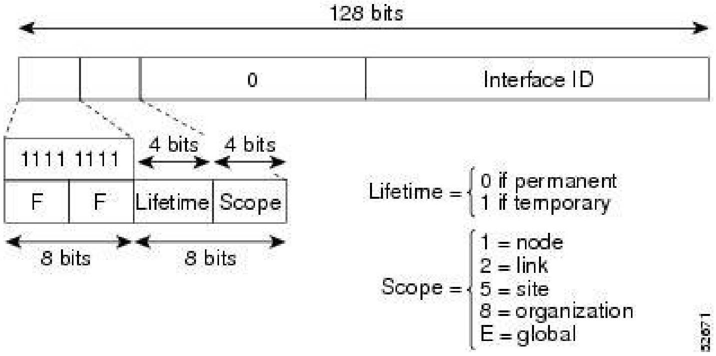
Figure 11 : Subnet Router Anycast Address Format



3.2.4 IPv6 Multicast Addresses

An IPv6 multicast address is an IPv6 address that has a prefix of FF00::/8 (1111 1111). An IPv6 multicast address is an identifier for a set of interfaces that belong to different nodes. A packet sent to a multicast address is delivered to all interfaces identified by the multicast address. The second octet following the prefix defines the lifetime and scope of the multicast address. A permanent multicast address has a lifetime parameter equal to 0; a temporary multicast address has a lifetime parameter equal to 1. A multicast address that has the scope of a node, link, site, or organization, or a global scope, has a scope parameter of 1, 2, 5, 8, or E, respectively. For example, a multicast address with the prefix FF02::/16 is a permanent multicast address with a link scope.

Figure 12: IPv6 Multicast Address Format



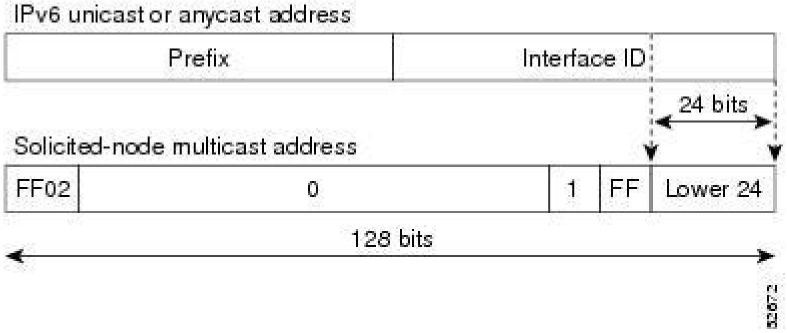
IPv6 nodes (hosts and routers) are required to join (where received packets are destined for) the following multicast groups:

- All-nodes multicast group FF02:0:0:0:0:0:0:1 (the scope is link-local)
- Solicited-node multicast group FF02:0:0:0:0:1:FF00:0000/104 for each of its assigned unicast and anycast addresses

IPv6 routers must also join the all-routers multicast group FF02:0:0:0:0:0:0:2 (the scope is link-local).

The solicited-node multicast address is a multicast group that corresponds to an IPv6 unicast or anycast address. IPv6 nodes must join the associated solicited-node multicast group for every unicast and anycast address to which it is assigned. The IPv6 solicited-node multicast address has the prefix FF02:0:0:0:0:1:FF00:0000/104 concatenated with the 24 low-order bits of a corresponding IPv6 unicast or anycast address. For example, the solicited-node multicast address that corresponds to the IPv6 address 2037::01:800:200E:8C6C is FF02::1:FF0E:8C6C. Solicited-node addresses are used in neighbor solicitation messages.

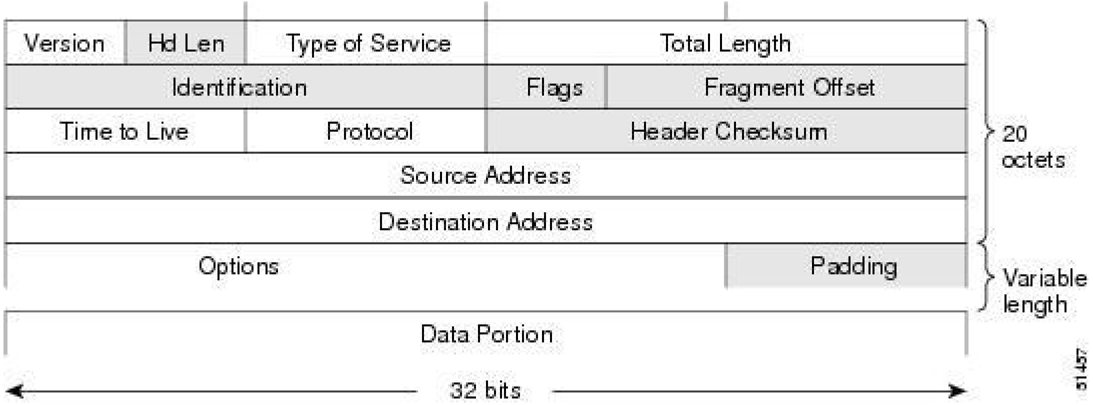
Figure 13 : Pv6 Solicited-Node Multicast Address Format


IPv6 unicast or anycast address
Prefix
Interface ID
24 bits
Solicited-node multicast address
FF02
0
1
FF
Lower 24
128 bits
52672

3.2.5 IPv4 Packet Header

The base IPv4 packet header has 12 fields with a total size of 20 octets (160 bits). The 12 fields may be followed by an Options field, which is followed by a data portion that is usually the transport-layer packet. The variable length of the Options field adds to the total size of the IPv4 packet header. The shaded fields of the IPv4 packet header are not included in the IPv6 packet header.

Figure 14 : IPv4 Packet Header Format



3.2.6 Simplified IPv6 Packet Header

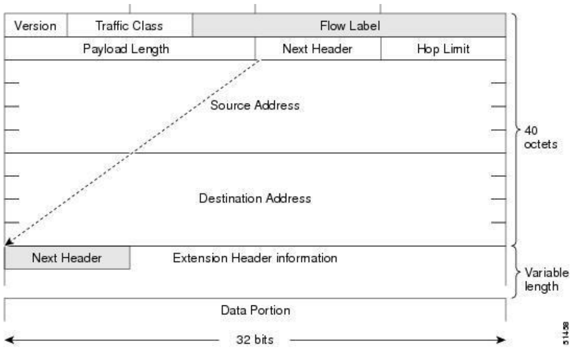
The base IPv6 packet header has 8 fields with a total size of 40 octets (320 bits). Fragmentation is handled by the source of a packet and checksums at the data link layer and transport layer are used. The User Datagram Protocol (UDP) checksum checks the integrity of the inner packet and the base IPv6 packet header and Options field are aligned to 64 bits, which can facilitate the processing of IPv6 packets.

Table 5 : Base IPv6 Packet Header Fields

Field	Description

Version	Similar to the Version field in the IPv4 packet header, except that the field lists number 6 for IPv6 instead of number 4 for IPv4.
Traffic Class	Similar to the Type of Service field in the IPv4 packet header. The Traffic Class field tags packets with a traffic class that is used in differentiated services.
Flow Label	New field in the IPv6 packet header. The Flow Label field tags packets with a specific flow that differentiates the packets at the network layer.
Payload Length	Similar to the Total Length field in the IPv4 packet header. The Payload Length field indicates the total length of the data portion of the packet.
Next Header	Similar to the Protocol field in the IPv4 packet header. The value of the Next Header field determines the type of information that follows the base IPv6 header. The type of information that follows the base IPv6 header can be a transport-layer packet, for example, a TCP or UDP packet, or an Extension Header.
Hop Limit	Similar to the Time to Live field in the IPv4 packet header. The value of the Hop Limit field specifies the maximum number of routers that an IPv6 packet can pass through before the packet is considered invalid. Each router decrements the value by one. Because no checksum is in the IPv6 header, the router can decrement the value without needing to recalculate the checksum, which saves processing resources.
Source Address	Similar to the Source Address field in the IPv4 packet header, except that the field contains a 128-bit source address for IPv6 instead of a 32-bit source address for IPv4.
Destination Address	Similar to the Destination Address field in the IPv4 packet header, except that the field contains a 128-bit destination address for IPv6 instead of a 32-bit destination address for IPv4.

Figure 15: IPv6 Packet Header Format



Optional extension headers and the data portion of the packet are after the eight fields of the base IPv6 packet header. If present, each extension header is aligned to 64 bits. There is no fixed number of extension headers in an IPv6 packet. Each extension header is identified by the Next Header field of the previous header.

Typically, the final extension header has a Next Header field of a transport-layer protocol, such as TCP or UDP.

Figure 16 : IPv6 Extension Header Format

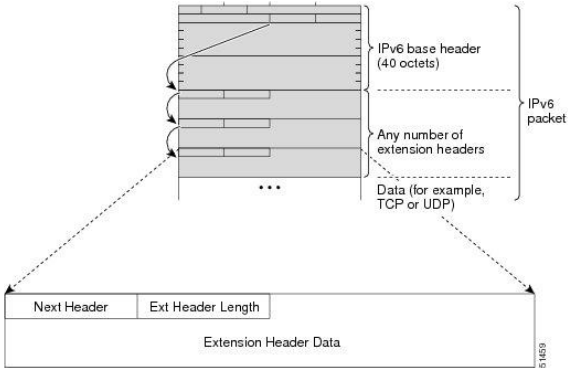

IPv6 base header (40 octets)
IPv6 packet
Any number of extension headers
Data (for example, TCP or UDP)
Next Header
Ext Header Length
Extension Header Data
51459

Table 6 : IPv6 Extension Header Types

HeaderType	Next Header Value	Description
Hop-by-Hop options header	0	Header that is processed by all hops in the path of a packet. When present, the hop-by-hop options header always follows immediately after the base IPv6 packet header.

Destination Header Options	60	Header that can follow any hop-by-hop options header. The header is processed at the final destination and at each visited address specified by a routing header. Alternatively, the destination options header can follow any Encapsulating Security Payload (ESP) header. The destination options header is processed only at the final destination.
Routing Header	43	Header that is used forsource routing.
Fragment Header	44	Header that is used when a source fragments a packet that is larger than the maximum transmission unit (MTU) for the path between itself and a destination. The Fragment header is used in each fragmented packet.
Upper-Layer Headers	6 (TCP) 17 (UDP)	Headers that are used insidea packet to transport the data. The two main transport protocols are TCP and UDP.

3.2.7 DNS for IPv6

IPv6 supports DNS record types that are supported in the DNS name-to-address and address-to-name lookup processes. The DNS record types support IPv6 addresses.

Table 7 : IPv6 DNS Record Types

Record Type	**Description**	**Format**
AAAA	Maps a hostname to an IPv6 address. (Equivalent to an A record in IPv4.)	www.abc.test AAAA 3FFE:YYYY:C18:1::2
PTR	Maps an IPv6 address to a hostname. (Equivalent to a PTR record in IPv4.)	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0. 0.8.1.c.0.y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

3.2.8 Path MTU Discovery for IPv6

As in IPv4, you can use path MTU discovery in IPv6 to allow a host to dynamically discover and adjust to differences in the MTU size of every link along a data path. In IPv6, however, fragmentation is handled by the source of a packet when the path MTU of one link along a given data path is not large enough to accommodate the size of the packets. Having IPv6 hosts handle packet fragmentation saves IPv6 router processing resources and helps IPv6 networks run more efficiently. Once the path MTU is reduced by the arrival of an ICMP Too Big message, Inspur INOS retains the lower value. The connection does not increase the segment size to gauge the throughput.

3.2.9 CDP IPv6 Address Support

You can use the Inspur Discovery Protocol (CDP) IPv6 address support for the neighbor information feature to transfer IPv6 addressing information between two Inspur devices. Inspur Discovery Protocol support for IPv6 addresses provides IPv6 information to network management products and troubleshooting tools.

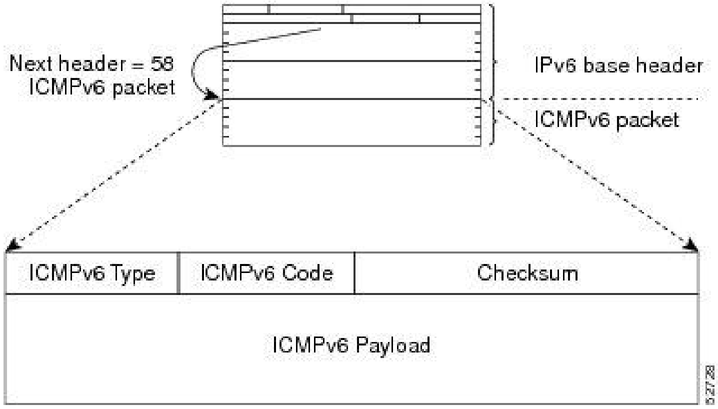
3.2.10 ICMP for IPv6

You can use ICMP in IPv6 to provide information about the health of the network. ICMPv6, the version that works with IPv6, reports errors if packets cannot be processed correctly and sends informational messages about the status of the network. For example, if a router cannot forward a packet because it is too large to be sent out on another network, the router sends out an ICMPv6 message to the originating host. Additionally, ICMP packets in IPv6 are used in IPv6 neighbor discovery and path MTU discovery. The path MTU discovery process ensures that a packet is sent using the largest possible size that is supported on a specific route.

A value of 58 in the Next Header field of the base IPv6 packet header identifies an IPv6 ICMP packet. The ICMP packet follows all the extension headers and is the last piece of information in the IPv6 packet.Within the IPv6 ICMP packets, the ICMPv6 Type and ICMPv6 Code fields identify IPv6 ICMP packet specifics, such as the ICMP message type. The value in the Checksum field is computed by the sender and checked by the receiver from the fields in the IPv6 ICMP packet and the IPv6 pseudo header.

The ICMPv6 Payload field contains error or diagnostic information that relates to IP packet processing.

Figure 17 : IPv6 ICMP Packet Header Format



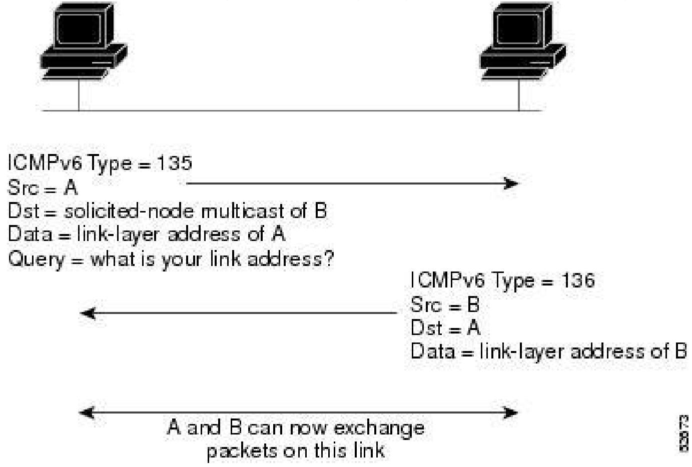
3.2.11 IPv6 Neighbor Discovery

You can use the IPv6 Neighbor Discovery Protocol (NDP) to determine whether a neighboring router is reachable. IPv6 nodes use neighbor discovery to determine the addresses of nodes on the same network (local link), to find neighboring routers that can forward their packets, to verify whether neighboring routers are reachable or not, and to detect changes to link-layer addresses. NDP uses ICMP messages to detect whether packets are sent to neighboring routers that are unreachable.

3.2.12 IPv6 Neighbor Solicitation Message

A node sends a neighbor solicitation message, which has a value of 135 in the Type field of the ICMP packet header, on the local link when it wants to determine the link-layer address of another node on the same local link. The source address is the IPv6 address of the node that sends the neighbor solicitation message. The destination address is the solicited-node multicast address that corresponds to the IPv6 address of the destination node. The neighbor solicitation message also includes the link-layer address of the source node.

Figure 18 : IPv6 Neighbor Discovery-Neighbor Solicitation Message


ICMPv6 Type = 135
Src = A
Dst = solicited-node multicast of B
Data = link-layer address of A
Query = what is your link address?
ICMPv6 Type = 136
Src = B
Dst = A
Data = link-layer address of B
A and B can now exchange packets on this link

After receiving the neighbor solicitation message, the destination node replies by sending a neighbor advertisement message, which has a value of 136 in the Type field of the ICMP packet header, on the local link. The source address is the IPv6 address of the node (the IPv6 address of the node interface that sends the neighbor advertisement message). The destination address is the IPv6 address of the node that sends the neighbor solicitation message. The data portion includes the link-layer address of the node that sends the neighbor advertisement message.

After the source node receives the neighbor advertisement, the source node and destination node can communicate.

Neighbor solicitation messages can verify the reachability of a neighbor after a node identifies the link-layer address of a neighbor. When a node wants to verify the reachability of a neighbor, it uses the destination address in a neighbor solicitation message as the unicast address of the neighbor.

Neighbor advertisement messages are also sent when there is a change in the link-layer address of a node on a local link. When there is a change, the destination address for the neighbor advertisement is the all-nodes multicast address.

Neighbor unreachability detection identifies the failure of a neighbor or the failure of the forward path to the neighbor and is used for all paths between hosts and neighboring nodes (hosts or routers). Neighbor unreachability detection is performed for neighbors to which only unicast packets are being sent and is not performed for neighbors to which multicast packets are being sent.

A neighbor is considered reachable when a positive acknowledgment is returned from the neighbor (indicating that packets previously sent to the neighbor have been received and processed). A positive

acknowledgment-from an upper-layer protocol (such as TCP)-indicates that a connection is making forward progress (reaching its destination). If packets are reaching the peer, they are also reaching the next-hop neighbor of the source. Forward progress is also a confirmation that the next-hop neighbor is reachable.

For destinations that are not on the local link, forward progress implies that the first-hop router is reachable. When acknowledgments from an upper-layer protocol are not available, a node probes the neighbor using unicast neighbor solicitation messages to verify that the forward path is still working. The return of a solicited neighbor advertisement message from the neighbor is a positive acknowledgment that the forward path is still working (neighbor advertisement messages that have the solicited flag set to a value of 1 are sent only in response to a neighbor solicitation message). Unsolicited messages confirm only the one-way path from the source to the destination node; solicited neighbor advertisement messages indicate that a path is working in both directions.

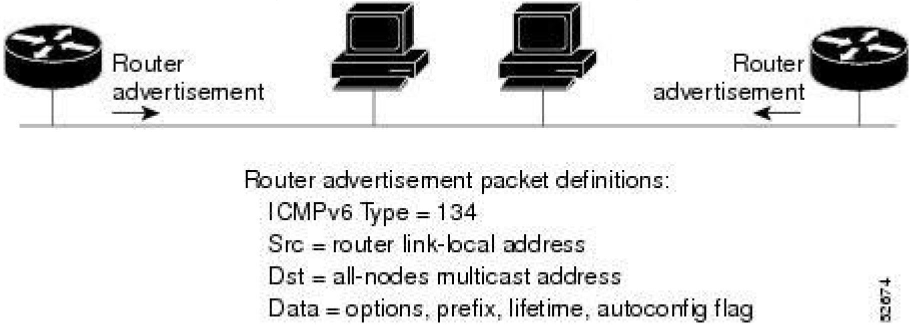
Neighbor solicitation messages are also used in the stateless autoconfiguration process to verify the uniqueness of unicast IPv6 addresses before the addresses are assigned to an interface. Duplicate address detection is performed first on a new, link-local IPv6 address before the address is assigned to an interface (the new address remains in a tentative state while duplicate address detection is performed). A node sends a neighbor solicitation message with an unspecified source address and a tentative link-local address in the body of the message. If another node is already using that address, the node returns a neighbor advertisement message that contains the tentative link-local address. If another node is simultaneously verifying the uniqueness of the same address, that node also returns a neighbor solicitation message. If no neighbor advertisement messages are received in response to the neighbor solicitation message and no neighbor solicitation messages are received from other nodes that are attempting to verify the same tentative address, the node that sent the original neighbor solicitation message considers the tentative link-local address to be unique and assigns the address to the interface.

3.2.13 IPv6 Router Advertisement Message

Router advertisement (RA) messages, which have a value of 134 in the Type field of the ICMP packet header, are periodically sent out to each configured interface of an IPv6 router. For stateless autoconfiguration to work properly, the advertised prefix length in RA messages must always be 64 bits.

The RA messages are sent to the all-nodes multicast address.

Figure 19 : Neighbor Discovery—RA Message



Router advertisement packet definitions:
ICMPv6 Type = 134
Src = router link-local address
Dst = all-nodes multicast address
Data = options, prefix, lifetime, autoconfig flag

RA messages typically include the following information:

- One or more onlink IPv6 prefixes that nodes on the local link can use to automatically configure their IPv6 addresses
- Life-time information for each prefix included in the advertisement
- Sets of flags that indicate the type of autoconfiguration (stateless or stateful) that can be completed

- Default router information (whether the router sending the advertisement should be used as a default router and, if so, the amount of time in seconds that the router should be used as a default router)

Additional information for hosts, such as the hop limit and MTU that a host should use in packets that it originates

RAs are also sent in response to router solicitation messages. Router solicitation messages, which have a value of 133 in the Type field of the ICMP packet header, are sent by hosts at system startup so that the host can immediately autoconfigure without needing to wait for the next scheduled RA message. The source address is usually the unspecified IPv6 address (0:0:0:0:0:0:0:0). If the host has a configured unicast address, the unicast address of the interface that sends the router solicitation message is used as the source address in the message. The destination address is the all-routers multicast address with a scope of the link. When an RA is sent in response to a router solicitation, the destination address in the RA message is the unicast address of the source of the router solicitation message.

You can configure the following RA message parameters:

- The time interval between periodic RA messages
- The router life-time value, which indicates the usefulness of a router as the default router (for use by all nodes on a given link)
- The network prefixes in use on a given link
- The time interval between neighbor solicitation message retransmissions (on a given link)
- The amount of time that a node considers a neighbor reachable (for use by all nodes on a given link)

The configured parameters are specific to an interface. The sending of RA messages (with default values) is automatically enabled on Ethernet interfaces. For other interface types, you must enter the **no ipv6 nd suppress-ra** command to send RA messages. You can disable the RA message feature on individual interfaces by entering the **ipv6 nd suppress-ra** command.

3.2.14 IPv6 Router Advertisement Options for DNS Configuration

Most of the internet services are identified by a Domain Name Server (DNS) name. Inspur INOS IPv6 Router Advertisement (RA) provides the following two options to allow IPv6 hosts to perform automatic DNS configuration:

- Recursive DNS Server (RDNSS)
- DNS Search List (DNSSL)

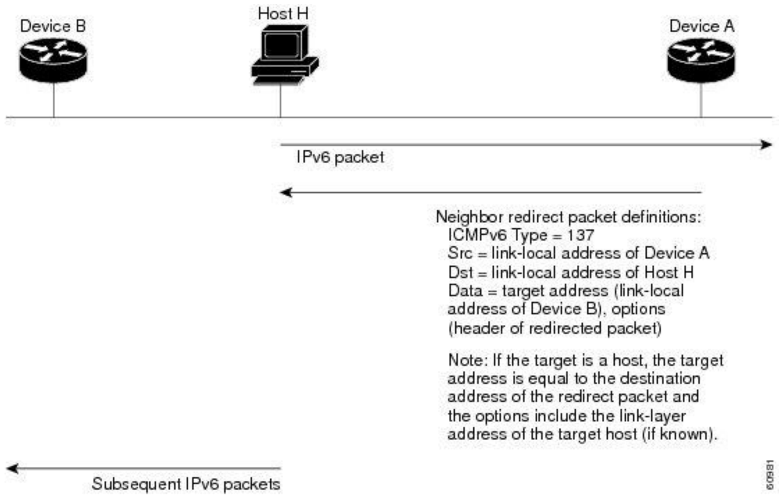
RDNSS contains the address of recursive DNS servers that help in DNS name resolution in IPv6 hosts. DNS Search List is a list of DNS suffix domain names used by IPv6 hosts when they perform DNS query searches.

For more information on RA options for DNS configuration, refer IETF RFC 6106.

3.2.15 IPv6 Neighbor Redirect Message

Routers send neighbor redirect messages to inform hosts of better first-hop nodes on the path to a destination. A value of 137 in the Type field of the ICMP packet header identifies an IPv6 neighbor redirect message.

Figure 20 : IPv6 Neighbor Discovery—Neighbor Redirect Message


Device B
Host H
Device A
IPv6 packet
Neighbor redirect packet definitions:
ICMPv6 Type = 137
Src = link-local address of Device A
Dst = link-local address of Host H
Data = target address (link-local address of Device B), options (header of redirected packet)
Note: If the target is a host, the target address is equal to the destination address of the redirect packet and the options include the link-layer address of the target host (if known).
Subsequent IPv6 packets
60981

After forwarding a packet, a router sends a redirect message to the source of the packet under the following circumstances:

- The destination address of the packet is not a multicast address.
- The packet was not addressed to the router.
- The packet is about to be sent out the interface on which it was received.
- The router determines that a better first-hop node for the packet resides on the same link as the source of the packet.
- The source address of the packet is a global IPv6 address of a neighbor on the same link or a link-local address.

3.3 Virtualization Support for IPv6

IPv6 supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

3.4 Licensing Requirements for IPv6

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide.*

3.5 Prerequisites for IPv6

IPv6 has the following prerequisites:

- You must be familiar with IPv6 basics such as IPv6 addressing, IPv6 header information, ICMPv6, and the IPv6 Neighbor Discovery (ND) Protocol.
- Ensure that you follow the memory/processing guidelines when you make a device a dual-stack device (IPv4/IPv6).

3.6 Guidelines and Limitations for Configuring IPv6

IPv6 has the following configuration guidelines and limitations:

• IPv6 packets are transparent to Layer 2 LAN switches because the switches do not examine Layer 3 packet information before forwarding IPv6 frames. IPv6 hosts can be directly attached to Layer 2 LAN switches.

• You can configure multiple IPv6 global addresses within the same prefix on an interface. However, multiple IPv6 link-local addresses on an interface are not supported.

• It supports contiguous masks only for both IPv4 and IPv6 addresses and does not support discontiguous masks IPv6 and IPv4 filters.

• Each interface can be configured with a maximum of 255 global IPv6 addresses and a maximum of 255 anycast IPv6 addresses.

• Because RFC 3879 deprecates the use of site-local addresses, you should configure private IPv6 addresses according to the recommendations of unique local addressing (ULA) in RFC 4193.

On F3 Series modules, you must disable IGMP optimized multicast flooding (OMF) on any VLANs that require any IPv6 packet forwarding (unicast or multicast). IPv6 neighbor discovery functions correctly only in a VLAN with the OMF feature disabled. To disable OMF, use the **no ip igmp snooping optimised-multicast-flood** command in VLAN configuration mode. With OMF disabled, unknown IPv4 multicast traffic (as well as all IPv6 multicast traffic) is flooded to all ports in the VLAN. Note that unknown multicast traffic refers to multicast packets with an active source but no receivers (and therefore no group forwarding entry in the hardware) in the ingress VLAN.

• IPv6 static route next hop link-local address cannot be configured at any local interface.

3.7 Default Settings for IPv6

Parameters	Default
ND reachable time	0 milliseconds
neighbor solicitation retransmit interval	1000 milliseconds

3.8 Configuring IPv6

3.8.1 Configuring IPv6 Addressing

You must configure an IPv6 address on an interface so that the interface can forward IPv6 traffic. When you configure a global IPv6 address on an interface, it automatically configures a link-local address and activates IPv6 for that interface.

Before you begin

Ensure that you are in the correct VDC (or use the switchto vdc command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.

Step 3	switch(config-if)# **ipv6 address** {*address* [**eui64**] [**route-preference** *preference*] [**secondary**] **tag** *tag-id*] or switch(config-if)# **ipv6 address** *ipv6-address* **use-link-local-only**	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. Entering the **ipv6 address** command configures global IPv6 addresses with an interface identifier (ID) in the low-order 64 bits of the IPv6 address. Only the 64-bit network prefix for the address needs to be specified; the last 64 bits are automatically computed from the interface ID. Entering the **ipv6 address use-link-local-only**command configures a link-local address on the interface that is used instead of the link-local address that is automatically configured when IPv6 is enabled on the interface. This command enables IPv6 processing on an interface without configuring an IPv6 address.
Step 4	(Optional) switch(config-if)# **show ip interface**	Displays interfaces configured for IPv4.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example
This example shows how to assign an IPv6 address:

```
switch# configure terminal

switch(config)# interface ethernet 2/3

switch(config-if)#        ipv6       address  2001:db8::/64    eui64
switch(config-if)#    copy       running-config       startup-config
switch(config-if)#
```

This example shows how to display an IPv6 interface:

```
switch# configure terminal

switch(config)# show ipv6 interface ethernet 3/1

Ethernet3/1,  Interface  status:  protocol-down/link-down/admin-
down,          iod:          36          IPv6          address:
0dc3:0dc3:0000:0000:0218:baff:fed8:239d

IPv6 subnet:  0dc3:0dc3:0000:0000:0000:0000:0000:0000/64

IPv6              link-local              address:
fe80::0218:baff:fed8:239d     (default)     IPv6
multicast routing: disabled

IPv6 multicast groups locally joined:

ff02::0001:ffd8:239d          ff02::0002          ff02::0001
ff02::0001:ffd8:239d

IPv6  multicast  (S,G)  entries
joined:  none  IPv6  MTU:  1500
(using link MTU)
```

```
IPv6  RP  inbound  packet-filtering
policy:  none  IPv6  RP  outbound
packet-filtering policy: none IPv6
inbound  packet-filtering  policy:
none   IPv6   outbound   packet-
filtering   policy:   none   IPv6
interface  statistics  last  reset:
never

IPv6       interface       RP-traffic       statistics:
(forwarded/originated/consumed) Unicast packets: 0/0/0

Unicast bytes:
0/0/0
Multicast
packets:
0/0/0
Multicast
bytes: 0/0/0
```

3.8.2 Configuring IPv6 Neighbor Discovery

You can configure IPv6 neighbor discovery on the router. Neighbor Discovery (ND) enables IPv6 nodes and routers to determine the link-layer address of a neighbor on the same link, find neighboring routers, and keep track of neighbors.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.

Step 3	switch(config-if)# **ipv6 nd** [**hop-limit** *hop-limit* | **managed-config-flag** | **mtu** *mtu* | **ns-interval** *interval* | **other-config-flag** | **prefix** | **ra-interval** *interval* | **ra-lifetime** *lifetime* | **reachable-time** *time* | **redirects** | **retrans-timer** *time* | **suppress-ra**]	Specifies an IPv6 address assigned to the interface and enables IPv6 processing on the interface. • **hop-limit** *hop-limit*— Advertises the hop limit in IPv6 neighbor discovery packets. The range is from 0 to 255. • **managed-config-flag**— Advertises in ICMPv6 router-advertisement messages to use stateful address auto-configuration to obtain address information. • **mtu** *mtu*—Advertises the maximum transmission unit (MTU) in ICMPv6 router-advertisement messages on this link. The range is from 1280 to 65535 bytes. • **ns-interval** *interval*—Configures the retransmission interval between IPv6 neighbor solicitation messages. The range is from 1000 to 3600000 milliseconds. • **other-config-flag**—Indicates in ICMPv6 router-advertisement messages that hosts use stateful auto-configuration to obtain nonaddress related information. • **prefix**—Advertises the IPv6 prefix in the router-advertisement messages. • **ra-interval** *interval*—Configures the interval between sending ICMPv6 router-advertisement messages. The range is from 4 to 1800 seconds. • **ra-lifetime** *lifetime*—Advertises the lifetime of a default router in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • **reachable-time** *time*—Advertises the time when a node considers a neighbor up after receiving a reachability confirmation in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • **redirects**—Enables sending ICMPv6 redirect messages. • **retrans-timer** *time*—Advertises the time between neighbor-solicitation messages in ICMPv6 router-advertisement messages. The range is from 0 to 9000 seconds. • **suppress-ra**— Disables sending ICMPv6 router-advertisement messages.

	Command or Action	Purpose
Step 4	Required: switch(config-if)# **ipv6 nd prefix** {*ipv6-address/prefix-length* | **default**} {**valid-lifetime** | **infinite** | **no-advertise**} {**preferred-lifetime** | **infinite**} [**no-autoconfig**] [**no-onlink**] [**off-link**]	Advertises the IPv6 prefix in the router advertisement messages. • *valid-lifetime*—The amount of time (in seconds) that the specified IPv6 prefix is advertised as being valid. • **infinite**—Specifies that the valid lifetime is infinite. • **no-advertise**—Specifies that the prefix is not advertised. • *preferred-lifetime*—The amount of time (in seconds) that the specified IPv6 prefix is advertised as being preferred. • **no-autoconfig**—Indicates to hosts on the local link that the specified prefix cannot be used for IPv6 autoconfiguration. The prefix will be advertised with the A-bit clear. • **no-onlink**—Configures the specified prefix as not on-link. The prefix will be advertised with the L-bit clear. • **off-link**—Configures the specified prefix as off-link. The prefix will be advertised with the L-bit clear. The prefix will not be inserted into the routing table as a connected prefix. If the prefix is already present in the routing table as a connected prefix (for example, because the prefix was also configured using the **ipv6 address** command), it will be removed.
Step 5	(Optional) switch(config-if)# **show ip nd interface**	Displays interfaces configured for IPv6 neighbor discovery.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure IPv6 neighbor discovery reachable time:

```
switch# configure terminal

switch(config)# interface ethernet 3/1

switch(config-if)# ipv6 nd reachable-time 10
switch(config-if)#      copy      running-config  startup-config
switch(config-if)#
```

This example shows how to display an IPv6 interface:

```
switch# configure terminal
```

```
switch(config)#  show  ipv6  nd  interface  ethernet 3/1



ICMPv6 ND Interfaces for VRF "default"

Ethernet3/1,    Interface    status:    protocol-down/link-down/
admin-down                    IPv6                      address:
0dc3:0dc3:0000:0000:0218:baff:fed8:239d

ICMPv6 active timers:

Last Neighbor-Solicitation sent:

never      Last      Neighbor-
Advertisement  sent:  never  Last
Router-Advertisement sent:never

Next  Router-Advertisement  sent  in:
0.000000       Router-Advertisement
parameters:

Periodic interval: 200 to 600 seconds

Send   "Managed  Address  Configuration"
flag:   false   Send   "Other   Stateful
Configuration" flag: false Send "Current
Hop Limit" field: 64

Send "MTU" option value: 1500

Send  "Router  Lifetime"  field:
1800 secs Send "Reachable Time"
field:  10  ms   Send  "Retrans
Timer"  field:  0  ms  Neighbor-
Solicitation parameters:

NS  retransmit  interval:
1000   ms   ICMPv6   error
message parameters: Send
redirects: false

Send unreachables: false
```

This example shows how to include the IPv6 prefix 2001:0DB8::/35 in router advertisements that are sent out Ethernet interface 0/0 with a valid lifetime of 1000 seconds and a preferred lifetime of 900 seconds:

```
switch(config)# interface ethernet 0/0

switch(config-if)#  ipv6  nd  prefix  2001:0DB8::/35 1000 900
```

3.8.3 Configuring Optional IPv6 Neighbor Discovery

You can use the following optional IPv6 neighbor discovery commands:

Command	Purpose
ipv6 nd cache limit *max-nd-adj* [**syslog** *syslogs-per-second*]	Configures the maximum number of entries in the neighbor adjacency table. The range is from 1 to 409600. The **syslog** keyword configures the number of system logs per second. The range is from 1 to 1000. If you configure a limit for IPv6 neighbor discovery entries, system logs appear if you try to add an adjacency after reaching the configured limit. **Note** You cannot unconfigure the cache limit until the total number of current adjacencies is less than 131,072.
ipv6 nd dad attempts *number*	Sets the number of consecutive neighbor solicitation messages that the device sends from the IPv6 interface for duplicate address detection (DAD) validation. The default value is 1 attempt.
ipv6 nd fast-path	Improves the performance of glean packets by reducing the processing of the packets in the supervisor. It applies to glean packets where the destination IP address is part of the same subnet and does not apply to packets where the destination IP address is in a different subnet. The default is enabled.
ipv6 nd hop-limit	Configures the maximum number of hops used in router advertisements and all IPv6 packets that are originated by the router.
ipv6 nd managed-config-flag	Sets the managed address configuration flag in IPv6 router advertisements.
ipv6 nd mtu	Sets the maximum transmission unit (MTU) size of IPv6 packets sent on an interface.
ipv6 nd ns-interval	Configures the interval between IPv6 neighbor solicitation retransmissions on an interface.
ipv6 nd other-config-flag	Configures the other stateful configuration flag in IPv6 router advertisements.
ipv6 nd ra-interval	Configures the interval between IPv6 router advertisement (RA) transmissions on an interface.
ipv6 nd ra-lifetime	Configures the router lifetime value in IPv6 router advertisements on an interface.

ipv6 nd reachable-time	Configures the amount of time that a remote IPv6 node is considered reachable after some reachability confirmation event has occurred.
ipv6 nd redirects	Enables ICMPv6 redirect messages to be sent.
ipv6 nd retrans-timer	Configures the advertised time between neighbor solicitation messages in router advertisements.
ipv6 nd suppress-ra	Suppresses IPv6 router advertisement transmissions on a LAN interface.

3.8.4 Configuring Recursive DNS Server (RDNSS)

You can configure up to eight DNS servers to advertise with Router Advertisement. You can also remove one or more DNS servers from the advertising list by using the no form of the command.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)#**interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# **ipv6 nd ra dns server** *ipv6-addr* [*rdnss-life* | **infinite**] **sequence** *sequence-num*	Configures the recursive DNS server. You can specify the life time and the sequence of the server.
Step 4	switch(config-if)# **show ipv6 nd ra dns server** [**interface** *interface*]	(Optional) Displays the configured RDNSS list.
Step 5	switch(config-if)# **ipv6 nd ra dns server suppress**	(Optional) Disables the configured server list.
Step 6	switch(config-if)# **no ipv6 nd ra dns server** *ipv6-addr* [*rdnss-life* | **infinite**] **sequence** *sequence-num*	Removes a server from the RDNSS list.

Example

The following example shows how to configure Recursive DNS Server list on Ethernet 3/3 and verify the same.

```
switch# configure terminal

switch(config)# interface ethernet 3/3

switch(config-if)# ipv6 nd ra dns server 1::1 1000 sequence 0

switch(config-if)# ipv6 nd ra dns server 2::1 infinite sequence 1

switch(config)# show ipv6 nd ra dns server

Recursive  DNS  Server  List
on:   mgmt0   Suppress   DNS
Server List: No
```

```
Recursive  DNS  Server  List  on:
  Ethernet3/3 Suppress DNS Server
  List: No

  DNS Server 1: 1::1 Lifetime:1000 seconds
  Sequence:0  DNS  Server  2:  2::1  Infinite
  Sequence:1
```

3.8.5 Configuring DNS Search List (DNSSL)

You can configure up to eight DNS search lists to advertise with Router Advertisement. You can also remove one or more DNS search lists from the advertising list by using the no form of the command.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters the global configuration mode.
Step 2	switch(config)#**interface ethernet** *number*	Enters interface configuration mode.
Step 3	switch(config-if)# **ipv6 nd ra dns search-list** *list* [*dnssl-life* | **infinite**] **sequence** *sequence-num*	Configures the DNS search list. You can specify the life time and the sequence of the search list.
Step 4	switch(config-if)# **show ipv6 nd ra dns search-list** [**interface** *interface*]	(Optional) Displays the configured DNS search list.
Step 5	switch(config-if)# **ipv6 nd ra dns search-list suppress**	(Optional) Disables the configured search list.
Step 6	switch(config-if)# **no ipv6 nd ra dns search-list** *list* [*dnssl-life* | **infinite**] **sequence** *sequence-num*	(Optional) Removes a search list from the RA.

Example

The following example shows how to configure DNS Search list on Ethernet 3/3 and verify the same.

```
switch# configure terminal

switch(config)# interface ethernet 3/3

switch(config-if)#  ipv6  nd  ra  dns  search-list  Inspur.com  100
sequence 1

switch(config-if)#  ipv6  nd  ra  dns  search-list  ind.Inspur.com  100  sequence
2

switch(config)# show ipv6 nd ra dns search-list

DNS  Search  List
on:         mgmt0
Suppress      DNS
Search List: No

 DNS  Search  List  on:
  Ethernet3/3
```

```
Suppress DNS Search
List: No

DNS  Server  1:Inspur.com  100
Sequence:1       DNS    Server
2:ind.Inspur.com          100
Sequence:2
```

3.8.6 Configuring IPv6 Packet Verification

Inspur INOS supports an Intrusion Detection System (IDS) that checks for IPv6 packet verification. You can enable or disable these IDS checks.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware ip verify address** {**destination zero** | **identical** | **reserved** | **source multicast**}	Performs the following IDS checks on the IPv6 address: • destination zero—Drops IPv6 packets if the destination IP address is ::. • identical—Drops IPv6 packets if the source IPv6 address is identical to the destination IPv6 address. • reserved—Drops IPv6 packets if the IPv6 address is ::1. • source multicast—Drops IPv6 packets if the IPv6 source address is in the FF00::/8 range (multicast).
Step 3	switch(config)# **hardware ip verify length** {**consistent** | **maximum** {**max-frag** | **max-tcp** | **udp**}}	Performs the following IDS checks on the IPv6 address: • consistent—Drops IPv6 packets where the Ethernet frame size is greater than or equal to the IPv6 packet length plus the Ethernet header. • maximum max-frag—Drops IPv6 packets if the formula (IPv6 Payload Length - IPv6 Extension Header Bytes) + (Fragment Offset * 8) is greater than 65536.. • maximum max-tcp—Drops IPv6 packets if the TCP length is greater than the IP payload length. • maximum max-udp—Drops IPv6 packets if the TCP length is less than the UDP packet length.
Step 4	switch(config)# **hardware ipv6 verify tcp tiny-frag**	Drops TCP packets if the IPv6 fragment offset is 1, or if the IPv6 fragment offset is 0 and the IP payload length is less than 16.
Step 5	switch(config)# **hardware ipv6 verify version**	Drops TCP packets if the EtherType is not set to 6 (IPv6).
Step 6	switch(config)# **show hardware forwarding ip verify**	Displays the IPv6 packet verification configuration.

Step 7	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

3.9 Verifying the IPv6 Configuration

Use one of the following commands to verify the configuration:

Command	**Purpose**
show hardware forwarding ip verify	Shows the IPv4 and IPv6 packet verification configuration.
show ipv6 interface	Displays IPv6-related interface information.
show ipv6 adjacency	Displays the adjacency table.
show ipv6 icmp	Displays ICMP IPv6 information.
show ipv6 nd	Displays IPv6 neighbor discovery information.
show ipv6 neighbor	Displays IPv6 neighbor entry.

3.10 Configuration Example for IPv6

```
switch# configure terminal

switch(config)#  interface  ethernet  3/1

switch(config-if)#ipv6 address 2001:db8::/64 eui64

switch(config-if)#ipv6 nd reachable-time 10

switch(config-if)#
```

3.11 Related Documents for IPv6

For more information related to IP CLI commands, see the *Inspur CN12700 Series INOS Unicast Routing Command Reference.*

3.12 Standards for IPv6

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

3.13 Feature History for IPv6

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 8 : Feature History for IPv6

Feature Name	**Release**	**Feature Information**

Duplicate address detection	8.4(1)	Added the ability to set the number of consecutive neighbor solicitation messages that the device sends from the IPv6 interface.
Glean optimization	8.4(1)	Added the **fast-path** keyword to the **ipv6 nd** command to improve the performance of glean packets by reducing the processing of the packets in the supervisor.
IPv6	8.4(1)	Added the ability to configure the maximum number of neighbor discovery entries in the neighbor adjacency table.
IPv6	8.4(1)	Updated for F3 Series modules.
IPv6	8.4(1)	Added support for IPv6 path MTU discovery.
IPv6	8.4(1)	Changed **platform** {**ip** | **ipv6**} **verify** command to the **hardware** {**ip** | **ipv6**} **verify** command.
IPv6	8.4(1)	Added the **tag** keyword to the**ipv6 address** command.
IPv6	8.4(1)	This feature was introduced.

CHAPTER 4 Configuring DNS

This chapter contains the following sections:

- Finding Feature Information.
- Information About DNS Clients.
- Licensing Requirements for DNS Clients.
- Prerequisites for DNS Clients.
- Guidelines and Limitations for DNS Clients.
- Default Settings for DNS Client Parameters.
- Configuring DNS Clients.
- Verifying the DNS Client Configuration.
- Configuration Examples for DNS Clients.
- Related Documents for DNS Clients.
- Standards for DNS Clients.
- Feature History for DNS.

4.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

4.2 Information About DNS Clients

4.2.1 DNS Client Overview

If your network devices require connectivity with devices in networks for which you do not control the name assignment, you can assign device names that uniquely identify your devices within the entire internetwork using the domain name server (DNS). DNS uses a hierarchical scheme for establishing host names for network nodes, which allows local control of the segments of the network through a client-server scheme. The DNS system can locate a network device by translating the hostname of the device into its associated IP address.

On the Internet, a domain is a portion of the naming hierarchy tree that refers to general groupings of networks based on the organization type or geography. Domain names are pieced together with periods (.) as the delimiting characters. For example, Inspur is a commercial organization that the Internet identifies by a com domain, so its domain name is Inspur.com. A specific hostname in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.Inspur.com.*

DNS Name Servers

Name servers keep track of domain names and know the parts of the domain tree for which they have complete information. A name server may also store information about other parts of the domain tree. To map domain names to IP addresses in Inspur INOS, you must identify the hostnames, specify a name server, and enable the DNS service.

Inspur INOS allows you to statically map IP addresses to domain names. You can also configure Inspur INOS to use one or more domain name servers to find an IP address for a host name.

DNS Operation

A name server handles client-issued queries to the DNS server for locally defined hosts within a particular zone as follows:

- An authoritative name server responds to DNS user queries for a domain name that is under its zone of authority by using the permanent and cached entries in its own host table. If the query is for a domain name that is under its zone of authority but for which it does not have any configuration information, the authoritative name server replies that no such information exists.
- A name server that is not configured as the authoritative name server responds to DNS user queries by using information that it has cached from previously received query responses. If no router is configured as the authoritative name server for a zone, queries to the DNS server for locally defined hosts receive nonauthoritative responses.

Name servers answer DNS queries (forward incoming DNS queries or resolve internally generated DNS queries) according to the forwarding and lookup parameters configured for the specific domain.

4.2.2 High Availability for DNS Clients

Inspur INOS supports stateless restarts for the DNS client. After a reboot or supervisor switchover, Inspur INOS applies the running configuration.

4.2.3 Virtualization Support for DNS Clients

Inspur INOS supports multiple instances of the DNS clients that run on the same system. You can configure a DNS client in each virtual device connect (VDC).You can optionally have a different DNS client configuration in each virtual routing and forwarding (VRF) instance within a VDC. By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. See the *Inspur INOS Virtual Device Context Configuration Guide* .

4.3 Licensing Requirements for DNS Clients

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide.*

4.4 Prerequisites for DNS Clients

You must have a DNS name server on your network.

- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur INOS Virtual Device Context Configuration Guide).*

4.5 Guidelines and Limitations for DNS Clients

- You configure the DNS client in a specific VRF. If you do not specify a VRF, Inspur INOS uses the default VRF.
- Inspur INOS does not support underscore in a DNS name. Hence do not use underscore in a DNS name.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

4.6 Default Settings for DNS Client Parameters

The table below lists the default settings for DNS client parameters.

Table 9 : Default DNS Client Parameters

Parameters	**Default**
DNS client	Enabled

4.7 Configuring DNS Clients

4.7.1 Configuring the DNS Client

Before you begin

- Ensure that you have a domain name server on your network.
- Ensure that you are in the correct VDC (or use the switchto vdc command).

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip host** *name address1* [*address2... address6*]	Defines up to six static hostname-to-address mappings in the hostname cache. The address can be either an IPv4 address or an IPv6 address.
Step 3	(Optional) switch(config)# **ip domain-name** *name* [**use-vrf** *vrf-name*]	Defines the default domain name that Inspur INOS uses to complete unqualified host names. You can optionally define a VRF that Inspur INOS uses to resolve this domain name if it cannot be resolved in the VRF that you configured this domain name under. Inspur INOS appends the default domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup. use-vrf is used as a DNS query supposed to be sending on a different VRF and listening for the reply on a different VRF. Example: DNS query is sent over VRF RED while the response should come on VRF Default.
Step 4	switch(config)# **ip dns source-interface** [*loopback X different interface*]	Defines what will be the source IP for the DNS Query which will be sent out. When DNS server tries to answer back, it will use the Loopback0 as the destination and there should be a valid return route.
Step 5	(Optional) switch(config)# **ip domain-list** *name* [**use-vrf** *vrf-name*]	Defines additional domain names that Inspur INOS can use to complete unqualified hostnames. You can optionally define a VRF that Inspur INOS uses to resolve these domain names if they cannot be resolved in the VRF that you configured this domain name under. Inspur INOS uses each entry in the domain list to append that domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup. Inspur INOS continues this process for each entry in the domain list until it finds a match.

Step 6	(Optional) switch(config)# **ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*]	Defines up to six name servers. The address can be either an IPv4 address or an IPv6 address. You can optionally define a VRF that Inspur INOS uses to reach this name server if it cannot be reached in the VRF that you configured this name server under.
Step 7	(Optional) switch(config)# **ip domain-lookup**	Enables DNS-based address translation. This feature is enabled by default.
Step 8	(Optional) switch(config)# **show hosts**	Displays information about DNS.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

4.7.2 Configuring Virtualization on a DNS Client

You can configure a DNS client within a VRF. If you do not enter VRF configuration mode, your DNS client configuration applies to the default VRF.

You can optionally configure a DNS client to use a specified VRF other than the VRF under which you configured the DNS client as a backup VRF. For example, you can configure a DNS client in the Red VRF but use the Blue VRF to communicate with the DNS server if the server cannot be reached through the Red VRF.

Before you begin

- Ensure that you have a domain name server on your network.
- Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*
3. (Optional) switch(config-vrf)# **ip domain-name** *name* [**use-vrf** *vrf-name*]
4. (Optional) switch(config-vrf)# **ip domain-list** *name* [**use-vrf** *vrf-name*]
5. (Optional) switch(config-vrf)# **ip name-server** *address1* [*address2... address6*] [**use-vrf***vrf-name*]
6. (Optional) switch(config-vrf)# **show hosts**
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a VRF and enters VRF configuration mode.

Step 3	(Optional) switch(config-vrf)# **ip domain-name** *name* [**use-vrf** *vrf-name*]	Defines the default domain name server that Inspur INOS uses to complete unqualified hostnames. You can optionally define a VRF that Inspur INOS uses to resolve this domain name server if it cannot be resolved in the VRF under which you configured this domain name. Inspur INOS appends the default domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup.
Step 4	(Optional) switch(config-vrf)# **ip domain-list** *name* [**use-vrf** *vrf-name*]	Defines additional domain name servers that Inspur INOS can use to complete unqualified hostnames. You can optionally define a VRF that Inspur INOS uses to resolve this domain name server if it cannot be resolved in the VRF under which you configured this domain name. Inspur INOS uses each entry in the domain list to append that domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup. Inspur INOS continues this process for each entry in the domain list until it finds a match.
Step 5	(Optional) switch(config-vrf)# **ip name-server** *address1* [*address2... address6*] [**use-vrf** *vrf-name*]	Defines up to six name servers. The address can be either an IPv4 address or an IPv6 address. You can optionally define a VRF that Inspur INOS uses to reach this name server if it cannot be reached in the VRF that you configured this name server under.
Step 6	(Optional) switch(config-vrf)# **show hosts**	Displays information about DNS.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

4.8 Verifying the DNS Client Configuration

To display the DNS client configuration, perform the following task:

Command	**Purpose**
show hosts	Displays information about DNS.

4.9 Configuration Examples for DNS Clients

This example shows how to establish a domain list with several alternate domain names:

```
ip domain list csi.com

ip   domain
list telecomprog.edu
ip domain
list merit.edu
```

This example shows how to configure the hostname-to-address mapping process and specify IP DNS-based translation. The example also configures the addresses of the name servers and the default domain name.

```
ip domain lookup

ip  name-server  192.168.1.111
192.168.1.2  ip  domain  name
Inspur.com
```

4.10 Related Documents for DNS Clients

Related Topic	
DNS Client CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide, Release 5.x*

4.11 Standards for DNS Clients

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

4.12 Feature History for DNS

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	**Release**	**Feature Information**
DNS	8.4(1)	This feature was introduced.

CHAPTER 5 Configuring WCCPv2

This chapter contains the following sections:

- Finding Feature Information.
- Information About WCCPv2.
- Licensing Requirements for WCCPv2.
- Prerequisites for WCCPv2.
- Guidelines and Limitations for WCCPv2.
- WCCPv2 Default Settings.
- Configuring WCCPv2.
- Verifying the WCCPv2 Configuration.
- Configuration Examples for WCCPv2.
- Related Documents for WCCPv2.
- Standards for the WCCPv2.
- Feature History for WCCPv2.

5.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

5.2 Information About WCCPv2

5.2.1 WCCPv2 Overview

WCCPv2 enables the Inspur INOS router to transparently redirect packets to cache engines. WCCPv2 does not interfere with normal router operations. Using WCCPv2, the router can redirect requests on configured interfaces to cache engines rather than to intended host sites. With WCCPv2, the router can balance traffic loads across a cluster of cache engines (cache cluster) and ensure fault-tolerant and fail-safe operation in the cluster. As you add or delete cache engines from a cache cluster, WCCPv2 dynamically redirects the packets to the currently available cache engines.

WCCPv2 accepts the traffic at the cache engine and establishes the connection with the traffic originator (the client). The cache engine acts as if it were the original destination server. If the requested object is not available on the cache engine, the cache engine establishes its own connection out to the original destination server to retrieve the object.

Until Release 8.4(1), WCCPv2 is supported only on the Layer3 or SVI interfaces, for Inspur CN12700 Series Switches.

Beginning from Release 8.4(1), WCCPv2 feature is supported on L3VNI BDI interfaces as an ingress feature. This feature is supported on Inspur CN12700 Series on and F3 modules only.

WCCPv2 communicates between routers and cache engines on UDP port 2048.

By allowing a cache cluster to connect to multiple routers, WCCPv2 provides redundancy and a distributed architecture for instances when a cache engine must connect to many interfaces. In addition, WCCPv2 allows you to keep all the cache engines in a single cluster, which avoids the unnecessary duplication of web pages across several clusters.

WCCPv2 Service Types

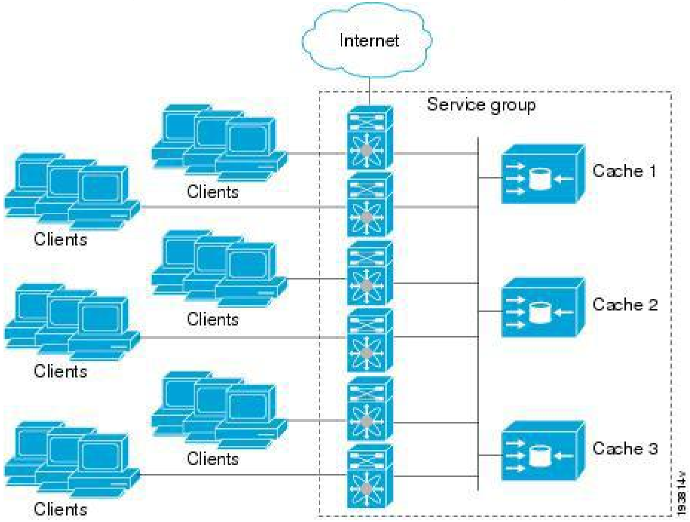
A service is a defined traffic type that the router redirects to a cache engine with the WCCPv2 protocol. You can configure the router to run one of the following cache-related services:

- Well-known —The router and the cache engine know the traffic type, for example the web cache service on TCP port 80 for HTTP.
- Dynamic service—A service in which the cache engine describes the type of redirected traffic to the router.

WCCPv2 Service Groups

A service group is a subset of cache engines within a cluster and the routers connected to the cluster that are running the same service. The figure shows a service group within a cache cluster. The cache engines and the routers can be a part of multiple service groups.

Figure 21 : WCCPv2 Cache Cluster and Service Group



You can configure a service group as open or closed. An open service group forwards traffic without redirection if there is no cache engine to redirect the traffic to. A closed service group drops traffic if there is no cache engine to redirect the traffic to.

The service group defines the traffic that is redirected to individual cache engines in that service group. The service group definition consists of the following:

- Service ID (0–255)
- Service Type
- Priority of the service group
- Protocol (TCP or UDP) of redirected traffic
- Service flags
- Up to eight TCP or UDP port numbers (either all source or all destination port numbers)

WCCPv2 Service Group Lists

WCCPv2 requires that each cache engine be aware of all the routers in the service group. You can configure a list of router addresses for each of the routers in the group on each cache engine.

The following sequence of events details how WCCPv2 configuration works:

1. You configure each cache engine with a list of routers.
2. Each cache engine announces its presence and generates a list of all routers with which it has established communications.
3. The routers reply with their view (list) of cache engines in the group.

The cache engines and routers exchange control messages every 10 seconds by default.

WCCPv2 designates one cache engine as the lead. If there is a group of cache engines, the one seen by all routers and the one that has the lowest IP address becomes the designated cache engine. The designated cache engine determines how traffic should be allocated across cache engines. The traffic assignment method is passed to the entire service group from the designated cache engine so that the routers of the group can redirect the packets and the cache engines of the group can manage their traffic load better.

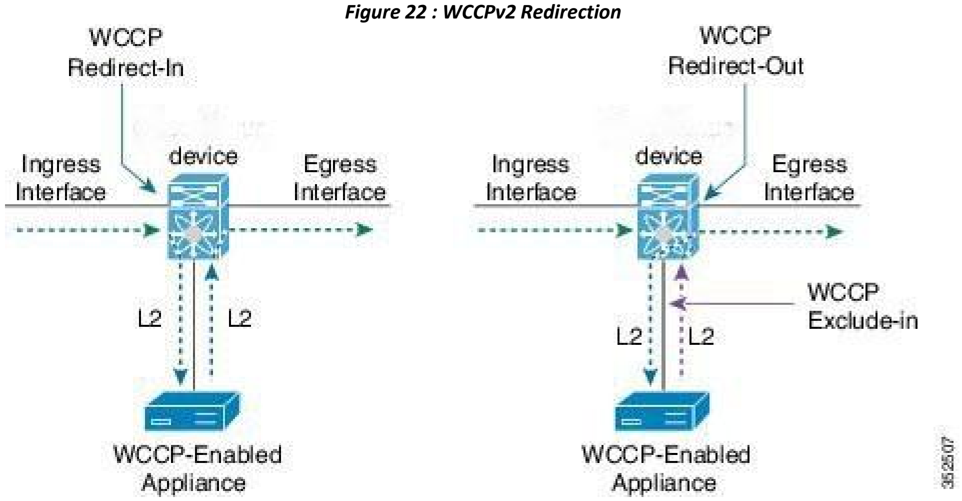
Inspur INOS uses the mask method to assign traffic. The designated cache engine assigns the mask and value sets to the router in the WCCP Redirect Assignment message. The router matches these mask and value sets to the source IP address, destination IP address, source port, and destination port of each packet. The router redirects the packet to the cache engine if the packet matches an assigned mask and value set. If the packet does not match an assigned mask and value set, the router forwards the packet without any redirection.

WCCPv2 Redirection

You can use an IP access list as a redirect list to specify a subset of traffic to redirect with WCCPv2. You can apply this access list for ingress or egress traffic on an interface. The figure shows how redirection applies to ingress or egress traffic.

You can also exclude ingress traffic on an interface but allow egress redirection on that interface.

Figure 22 : WCCPv2 Redirection



5.2.2 WCCPv2 Authentication

WCCPv2 can authenticate a device before it adds that device to the service group. Message Digest (MD5) authentication allows each WCCPv2 service group member to use a secret key to generate a keyed MD5 digest string that is part of the outgoing packet. At the receiving end, a keyed digest of an incoming packet is generated. If the MD5 digest within the incoming packet does not match the generated digest, WCCP ignores the packet.

WCCPv2 rejects packets in any of the following cases:

- The authentication schemes differ on the router and in the incoming packet.
- The MD5 digests differ on the router and in the incoming packet.

You must configure the same authentication on all members of a WCCPv2 service group.

5.2.3 WCCPv2 Redirection Method

WCCPv2 negotiates the packet redirection method between the router and the cache engine. Inspur INOS uses this traffic redirection method for all cache engines in a service group.

WCCPv2 redirects packets using Layer 2 Destination MAC rewrite method, where WCCPv2 replaces the destination MAC address of the packet with the MAC address of the cache engine that needs to handle the packet. The cache engine and the router must be adjacent to Layer 2.

You can also configure an access control list (ACL), called a redirect list, for a WCCPv2 service group. This ACL can either permit a packet to go through the WCCPv2 redirection process or deny the WCCP redirection and send the packet through the normal packet forwarding procedure.

5.2.4 WCCPv2 Packet Return Method

WCCPv2 filters packets to determine which redirected packets have been returned from the cache engine and which packets have not. WCCPv2 does not redirect the returned packets, because the cache engine has determined that these packets should not be cached. WCCPv2 returns packets that the cache engine does not service to the router that transmitted them.

A cache engine may return a packet for one of the following reasons:

- The cache engine is overloaded and cannot service the packets.
- The cache engine is filtering certain conditions that make caching packets counterproductive, for example, when IP authentication has been turned on.

WCCPv2 negotiates the packet return method between the router and the cache engine. Inspur INOS uses this traffic return method for all cache engines in a service group.

WCCPv2 returns packets using the Destination MAC rewrite method, where WCCPv2 replaces the destination MAC address of the packet with the MAC address of the router that originally redirected the packet. The cache engine and the router must be adjacent to Layer 2.

5.2.5 High Availability for WCCPv2

WCCPv2 supports stateful restarts and stateful switchovers. A stateful restart occurs when the WCCPv2 process fails and is restarted. A stateful switchover occurs when the active supervisor switches to the standby supervisor. Inspur INOS applies the running configuration after a switchover.

5.2.6 Virtualization Support for WCCPv2

WCCPv2 supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

WCCP redirection occurs within a VRF. You must configure the WCCP cache engine so that the forward and return traffic to and from the cache engine occurs from interfaces that are a part of the same VRF.

The VRF used for the WCCP on an interface should match the VRF configured on that interface.

If you change the VRF membership of an interface, Inspur INOS removes all layer 3 configuration, including WCCPv2.

For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

5.2.7 WCCPv2 Error Handling for SPM Operations

The Service Policy Manager (SPM) supervisor component acts as a data path manager for the WCCP Manager. The WCCP manager is shielded from the underlying platform specifics by the SPM and is portable to platform variations. The WCCP manager has a set of SPM APIs to pass the configurations that are mapped and programmed in the hardware. These APIs can process and parse the application data that is implemented and maintained in one single handler.

The interface redirects that failed to be programmed by the SPM are stored until there is a service group configuration change through the CLI or an RA message. The WCCP manager retries programming policies that failed previously.

The WCCP manager sends policy updates to the SPM in intervals to program TCAM entries in the hardware. These policy updates can be triggered by the CLI or through RA (Redirect-Assign) messages. When the WCCP is notified of an SPM error, a syslog message appears.

5.3 Licensing Requirements for WCCPv2

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide.*

5.4 Prerequisites for WCCPv2

WCCPv2 has the following prerequisites:

- You must globally enable the WCCPv2 feature.
- You can only configure WCCPv2 on Layer 3 or VLAN interfaces (see the Inspur CN12700 Series INOS Interfaces Configuration Guide).
- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*).

5.5 Guidelines and Limitations for WCCPv2

WCCPv2 has the following configuration guidelines and limitations:

- A WCCPv2 service group supports up to 32 routers and 32 cache engines.
- All cache engines in a cluster must include all routers that service the cluster in its configuration. If a cache engine within a cluster does not include one or more of the routers in its configuration, the service group detects the inconsistency and the cache engine is not allowed to operate within the service group.
- The cache engine cannot be on the same SVI with a redirect out statement.
- WCCPv2 works with IPv4 networks only.
- Any traffic that is coming from anF3 module interface and going towards a Traffic Engineering (TE) Class-based Tunnel Selection (CBTS) tunnel will be dropped if you have configured the **ip wccp redirect exclude in** command on the inbound F3-Series module interface or Switch Virtual Interface (SVI).
- WCCPv2 supports multiple service groups in the same direction (either inbound or outbound) on any Layer 3 interface, under the following conditions:
    - The access-list used must not have **deny ip any any** entry.
    - The access-list used for multiple service groups must not contain overlapping entries. The following is an example of an overlapping entry:

```
ip access-list wccp_acl1

 permit   tcp   10.0.0.0/8
10.0.0.0/8 ip access-list
wccp_acl2

 permit tcp 10.10.10.1/32 10.10.10.10/32
```

- Inspur INOS removes all Layer 3 configuration on an interface when you change the VDC, interface VRF membership, port-channel membership, or the port mode to Layer 2.
- Inspur INOS does not support WCCPv2 on tunnel interfaces.
- WCCPv2 is supported on all types of FEX devices.

- WCCP requires the client, server, and WCCP client to be on separate interfaces. If you migrate a topology from a Inspur Catalyst 6500 Series switch deployment, it might not be supported.
- WCCPv2 redirect-in and redirect-out is fully supported in Inspur INOS Release 8.4(1) in non-mixed module VDCs. WCCPv2 is also support in mixed module VDC scenarios for most module combinations.
- For egress WCCPv2, traffic is not redirected when the ingress includes F3 series modules, and the next-hop is pointing to an SVI interface or subinterface of any module. If the egress WCCP policy is applied on a SVI or subinterface and if the packet ingresses on a F3 module, the same limitation applies.
- Beginning with Inspur INOS Release 8.4(1),policy-based routing and WCCPv2 are supported on the same interface. However, policy-based routing with statistics and WCCPv2 is supported on the same interface only if bank chaining is disabled.
- GRE redirection/return and hash assignment are not supported on a Inspur CN12700 Series switch.
- Traffic might encounter a vPC loop and drop if you have Web Cache Control Protocol (WCCP) and vPC on your Inspur CN12700 Series switch and the traffic migrates from a Inspur CN 65xx switch to your switch. Traffic that comes from a vPC member port and crosses a vPC peer-link is not permitted to egress any vPC member port. However, it can egress any other type of port, such a Layer 3 port or an orphan port. This behavior is expected.
- If traffic drops after you configure WCCP and vPC on your CN12700 Series switch and based on your design, you can perform one of the following tasks to avoid the vPC loop:
  - Configure a Layer 2 trunk to carry the traffic in question.
  - Enable a peer gateway.
  - Shut down one of the member ports in the vPC.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.
- The following restrictions apply to the redirect-list, ACL:
  - Permit statements in the redirect ACL will consume more security TCAM entries compared to deny statements. Ensure the TCAM does not become oversubscribed.
  - The ACL must be an IPV4 simple ACL.
  - The protocol must be IP or TCP.
  - Only individual source or destination port numbers may be specified; port ranges cannot be specified.
  - The use of fragments or options is not permitted.
- From Inspur INOS Release 8.4(1), the following guidelines and limitations are applicable for WCCPv2:
  - WCCPv2 is supported for the L3 Virtual Network Identifier (VNI) Bridge Domain Interface (BDI), if it is applied on the ingress traffic only by using the **ip wccp** *service* **redirect in** command.
  - WCCPv2 is not supported for the L2VNI BDI.
  - The commands **ip wccp** *service* **redirect out** and **ip wccp redirect exclude in** are not supported on L3VNI BDI.
  - **ip wccp web-cache redirect out** command is not supported in WCCP on BDI interface.

5.6 WCCPv2 Default Settings

Parameters	**Default**
Authentication	No authentication
WCCPv2	Disable

5.7 Configuring WCCPv2

To configure WCCPv2, perform these tasks in this chapter:

Step 1 Enable the WCCPv2 feature.

Step 2 Configure a WCCPv2 service group.

Step 3 Apply WCCPv2 redirection to an interface.

5.7.1 Enabling and Disabling WCCPv2

Before you begin

- Enable the WCCPv2 feature.
- Ensure you are in the correct VDC (or use the **switchto vdc** command

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(config)# [**no**] **feature wccp**	Enables or disables the WCCPv2 feature in a VDC. Use the **no** form of the command to disable the feature.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

5.7.2 Configuring a WCCPv2 Service Group

Before you begin

- Enable the WCCPv2 feature.
- Ensure you are in the correct VDC (or use the **switchto vdc** command

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip wccp** {*service-number* | **web-cache**} [**mode** {**open** [**redirect-list** *acl-name*] | **closed service-list** *acl-name*}][**password** [**0-7**] *pwstring*]	Creates an open or closed mode service group. The service list identifies a named extended IP access list that defines the packets that will match the service. This list is required only when the service is defined as closed mode Optional parameters are as follows:

	Command or Action	Purpose
		• **mode**—Configures the service group in open or closed mode. On a service list, the mode controls the traffic type that the service group handles. The default is open. For closed mode, use this keyword to configure an IP access list to define the traffic type that matches this service. Closed mode for dynamic service groups requires a service list ACL that specifies the protocol and port information that is used for the service group. If there are no members in the service group, packets matching the service-list ACL are dropped. • **password**—Configures MD5 authentication for a service group. Use **password** *0 pwstring* to store the password in clear text. Use **password 7** *pwstring* to store the password in encrypted form. You can use the **password 7** keywords for an already encrypted password. • **redirect-list**—Configures a global WCCPv2 redirection list for the service group to control the traffic that is redirected to the cache engine. • **service-list**—Configures an IP access list that defines the traffic type redirected by the service group. • The *service-number* range is from 1 to 255. The *acl-name* can be any case-sensitive, alphanumeric string up to 64 characters. The *pwstring* can be any case-sensitive, alphanumeric string up to eight characters.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

5.7.3 Applying WCCPv2 Redirection to an Interface

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface ethernet** *number*	Enters interface configuration mode.

Step 3	switch(config-if)# **ip wccp** {*service-number* **redirect**{**in** | **out**} | **web-cache** | **redirect** {**in** | **out**}}	Applies the specified type of WCCPv2 redirection to the interface. The command examples show the following: • WCCPv2 redirection applied on the ingress or egress traffic for this interface. • WCCPv2 redirection applied on the ingress or egress web cache traffic for this interface. • Ingress traffic excluded from WCCP redirection on this interface. **Note** **ip wccp web-cache redirect out** command is not supported in WCCP on BDI interface.
Step 4	switch(config)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a router to redirect web-related packets without a destination of 19.20.2.1 to the web cache:

```
switch(config)# access-list 100

switch(config-acl)# deny ip any host 192.0.2.1

switch(config-acl)#  permit ip any any

switch(config-acl)# exit

switch(config)# ip wccp web-cache redirect-list 100
switch(config)# interface ethernet 2/1
switch(config-if)#  ip  wccp  web-cache redirect out
```

This example shows sample configuration for un-supported features:

```
switch# configure terminal

switch(config)# interface Bdi555

switch(config-if)# ip wccp redirect exclude in

This will remove all redirect-in on the interface. Proceed (y/n)?  [no] y

ERROR: Exclude in not supported on BDI



switch(config-if)# ip wccp 62 redirect out

ERROR: Redirect out not supported on BDI
```

This example shows a running-configuration, followed by a verification command that displays the L3VNI-BDI configuration details. Replace the placeholders with relevant values for your setup. The example considers that interface 555 is configured for BDI.

```
switch (config)# show running-configuration interface bdi 555
```

```
!Command: show running-config wccp

!Time: Thu Sep 25 02:46:02 2017

version
8.2(1)
interfac
e Bdi555

   description
   L3VNI-BDI  no
   shutdown

   vrf member
   vrf5000 no
   ip
   redirects
   ip forward

   ip    pin
 sparse-
 mode     ip
 wccp     61
 redirect
 in
```

This example show running-configuration for WCCP configuration on BDI interface. Replace the placeholders with relevant values for your setup.

```
switch (config)# show running-configuration wccp

!Command: show running-config wccp

!Time: Thu Sep 25 02:46:02 2017



version 8.2(1) feature wccp



vrf context vrf5000 ip wccp web-cache ip wccp 61

   ip wccp 62

interface Bdi555

   vrf member vrf5000

   ip wccp 61 redirect in
```

5.7.3 Configuring WCCPv2 in a VRF

Before you begin

- Enable the WCCPv2 feature.
- Ensure you are in the correct VDC (or use the **switchto vdc** command

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **vrf context** *vrf-name*	Enters VRF configuration mode. The vrf-name can be any case-sensitive, alphanumeric string up to 63 characters.
Step 3	switch(config)# **ip wccp** {*service-number* | **web-cache**} [**mode** {**open** [**redirect-list** *acl-name*] | **closed service-list** *acl-name*}][**password** [**0-7**] *pwstring*]	Creates an open or closed mode service group. The service list identifies a named extended IP access list that defines the packets that will match the service. This list is required only when the service is defined as closed mode Optional parameters are as follows: **mode**—Configures the service group in open or closed mode. On a service list, the mode controls the traffic type that the service group handles. The default is open. For closed mode, use this keyword to configure an IP access list to define the traffic type that matches this service. Closed mode for dynamic service groups requires a service list ACL that specifies the protocol and port information that is used for the service group. If there are no members in the service group, packets matching the service-list ACL are dropped. • **password**—Configures MD5 authentication for a service group. Use **password** *0 pwstring* to store the password in clear text. Use **password 7** *pwstring* to store the password in encrypted form. You can use the **password 7** keywords for an already encrypted password. • **redirect-list**—Configures a global WCCPv2 redirection list for the service group to control the traffic that is redirected to the cache engine. • **service-list**—Configures an IP access list that defines the traffic type redirected by the service group. • The *service-number* range is from 1 to 255. The *acl-name* can be any case-sensitive, alphanumeric string up to 64 characters. The *pwstring* can be any case-sensitive, alphanumeric string up to eight characters.
Step 4	(Optional) switch(config-vrf)# **show ip wccp** [**vrf** *vrf-name*]	Displays information about WCCPv2. The vrf-name can be any case-sensitive, alphanumeric string up to 64 characters.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure WCCPv2 in VRF Red on interface Ethernet 2/1:

```
switch# configure terminal
```

```
switch(config)# vrf context Red

switch(config-vrf)# ip wccp web-cache password Test1 redirect-list httpTest

switch(config-vrf)# interface ethernet 2/1

switch(config-if)# vrf member Red

switch(config-if)# ip wccp web-cache redirect out
```

5.8 Verifying the WCCPv2 Configuration

Use one of the following commands to verify the configuration:

Command	**Purpose**
show ip wccp [**vrf** *vrf-name*] [*service-number* | **web-cache**]	Displays the WCCPv2 status for all groups or one group in a VRF.
show ip interface [*ethernet-number*]	Displays the WCCPv2 interface information.
show ip wccp [*service-number* | **web-cache**]	Displays the WCCPv2 service group status.
show ip wccp [*service-number* | **web-cache**] **detail**	Displays the clients in a WCCPv2 service group.
show ip wccp [*service-number* | **web-cache**] **mask**	Displays the WCCPv2 mask assignment.
show ip wccp [*service-number* | **web-cache**]**service**	Displays the WCCPv2 service group definition.
show ip wccp [*service-number* | **web-cache**] **view**	Displays the WCCPv2 group membership.

5.9 Configuration Examples for WCCPv2

This example shows how to configure WCCPv2 authentication on router redirect web-related packets without a destination of 192.0.2.1 to the web cache:

```
access-list 100

 deny   ip   any
 host  192.0.2.1
 permit  ip  any
 any

feature wccp

ip   wccp   web-cache   password   0   Test1
redirect-list 100 interface ethernet 1/2

 ip  wccp  web-cache
 redirect   out   no
 shutdown
```

This example shows the sample output when WCCP is configuration in a VRF.

```
switch(config)# show ip wccp vrf vrf5000

VRF  vrf5000  WCCP
    information:
```

```
Router
information:

    Router Identifier:                   50.50.50.1

    Protocol Version:                    2.0

Service Identifier: web-cache

    Number of Service Group Clients:     1

    Number of Service Group Routers:     1

    Service mode:                        Open

    Service Access-list:                 -none-

    Redirect Access-list:                -none- Service Identifier: 61

    Number of Service Group Clients:     1

    Number of Service Group Routers:     1

    Service mode:                        Open

    Service Access-list:                 -none-

    Redirect Access-list:                -none- Service Identifier: 62

    Number of Service Group Clients:     1

    Number of Service Group Routers:     1

    Service mode:                        Open

    Service Access-list:                 -
    none-

    Redirect Access-list:                -
    none-
```

The following example shows a verification command to display the kind of service for WCCP.

```
switch(config)#  show  ip  wccp  vrf  vrf5000  61
service

WCCP   service   information
    definition:        Type:
                   Dynamic

    Id:            61

    Priority:      34

    Protocol:      6

    Options:       0x00000501

    --------

        Mask/Value sets:  1

        Value elements :  16
```

```
                        Ports:            -none-
```

The following example shows a verification command to display cache engine information, after the connection with the cache engine is established

```
switch(config)# show ip wccp vrf vrf5000 61 view

WCCP      Router
Informed     of:
50.50.50.1

WCCP  Cache  Engines
Visible: 10.10.10.3

WCCP Cache Engines Not Visible:

-none-
```

5.10 RelatedDocumentsforWCCPv2

Related Topic	Document Title
WCCPv2 CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
IP ACLs	*Inspur CN12700 Series INOS Security Configuration Guide, Release*8.4(1)

5.11 Standards for the WCCPv2

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

5.12 FeatureHistoryforWCCPv2

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Release	Feature Information
WCCPv2 on BDI	8.4(1)	Added support on BDI interface.
WCCPv2 Redirection	8.4(1)	Added support for F3 module.
WCCPv2	8.4(1)	Added support for policy-based routing and WCCPv2 on the same interface if bank chaining is disabled.
WCCPv2 Error Handling for SPM Operations	8.4(1)	This feature was added.
WCCPv2	8.4(1)	This feature was introduced.

CHAPTER 6 Configuring OSPFv2

This chapter contains the following sections:

- Finding Feature Information.
- Information About OSPFv2.
- Licensing Requirements for OSPFv2.
- Prerequisites for OSPFv2.
- Guidelines and Limitations for OSPFv2.
- Default Settings for OSPFv2.
- Configuring Basic OSPFv2.
- Configuring Advanced OSPFv2.
- Verifying the OSPFv2 Configuration.
- Monitoring OSPFv2 .
- Configuration Examples for OSPFv2.
- Related Documents for OSPFv2.
- Feature History for OSPFv2.

6.1 Finding Feature Information

Your software release might not support all the features documented in this module.To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

6.2 Information About OSPFv2

OSPFv2 is an IETF link-state protocol for IPv4 networks. An OSPFv2 router sends a special message, called a hello packet, out each OSPF-enabled interface to discover other OSPFv2 neighbor routers. Once a neighbor is discovered, the two routers compare information in the to determine if the routers have compatible configurations. The neighbor routers try to establish , which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv2 routing information. Adjacent routers share (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv2 routers eventually have identical link-state databases. When all OSPFv2 routers have identical link-state databases, the network is converged. Each router then uses Dijkstra′s Shortest Path First (SPF) algorithm to build its route table.

You can divide OSPFv2 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv2 supports IPv4, while OSPFv3 supports IPv6. For more information, see the "Configuring OSPFv3" chapter.

6.2.1 Hello Packet

OSPFv2 routers periodically send Hello packets on every OSPF-enabled interface. The hello interval determines how frequently the router sends these Hello packets and is configured per interface. OSPFv2 uses Hello packets for the following tasks:

- Neighbor discovery
- Keepalives
- Bidirectional communications

- Designated router election

The Hello packet contains information about the originating OSPFv2 interface and router, including the assigned OSPFv2 cost of the link, the , and optional capabilities of the originating router. An OSPFv2 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table.

Hello packets also include a list of router IDs for the routers that the originating interface has communicated with. If the receiving interface sees its own router ID in this list, then bidirectional communication has been established between the two interfaces.

OSPFv2 uses Hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a Hello packet by the configured (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

6.2.2 Neighbors

An OSPFv2 interface must have a compatible configuration with a remote interface before the two can be considered neighbors. The two OSPFv2 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID
- Authentication
- Optional capabilities

If there is a match, the following information is entered into the neighbor table:

- Neighbor ID—The router ID of the neighbor.
- Priority—Priority of the neighbor. The priority is used for designated router election.
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of the time since the last Hello packet was received from this neighbor.
- IP Address—The IP address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared as the designated router or as the backup designated router.
- Local interface—The local interface that received the Hello packet for this neighbor.

6.2.3 Adjacency

Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not.

Adjacency is established using Database Description packets, Link State Request packets, and Link State Update packets in OSPF. The Database Description packet includes just the LSA headers from the link-state database of the neighbor. The local router compares these headers with its own link-state database and determines which LSAs are new or updated. The local router sends a Link State Request packet for each LSA that it needs new or updated information on. The neighbor responds with a Link State Update packet. This exchange continues until both routers have the same link-state information.

6.2.4 Designated Routers

Networks with multiple routers present a unique situation for OSPF. If every router floods the network with LSAs, the same link-state information is sent from multiple sources. Depending on the type of network, OSPFv2 might use a single router, the (DR), to control the LSA floods and represent the network to the rest of the OSPFv2 area. If the DR fails, OSPFv2 selects a (BDR). If the DR fails, OSPFv2 uses the BDR.

Network types are as follows:

• Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.

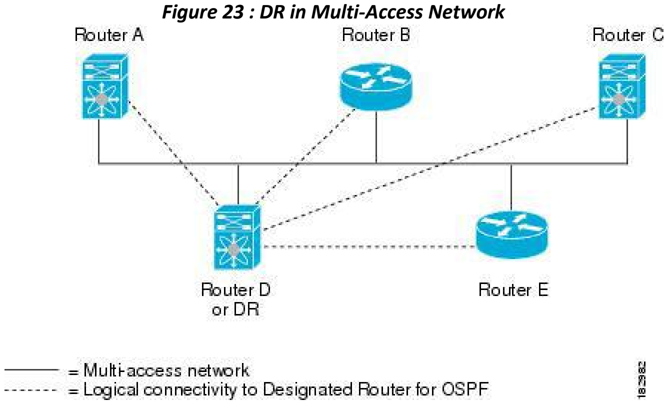
• Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv2 routers establish a DR and BDR that controls LSA flooding on the network. OSPFv2 uses the well-known IPv4 multicast addresses 224.0.0.5 and a MAC address of 0100.5300.0005 to communicate with neighbors.

The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final tie breaker, OSPFv2 chooses the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR and use the IPv4 multicast address 224.0.0.6 to send LSA updates to the DR and BDR. Figure 3-1 shows this adjacency relationship between all routers and the DR.

DRs are based on a router interface. A router might be the DR for one network and not for another network on a different interface.

Figure 23 : DR in Multi-Access Network



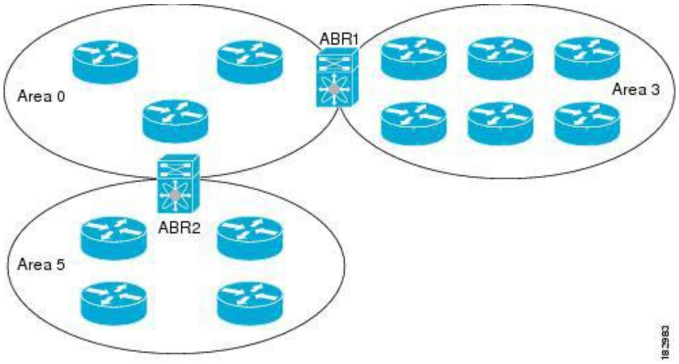
6.2.5 Areas

You can limit the CPU and memory requirements that OSPFv2 puts on the routers by dividing an OSPFv2 network into . An area is a logical division of routers and links within an OSPFv2 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database is limited to links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that you can enter as a number or in dotted decimal notation, such as 10.2.3.1.

Inspur INOS always displays the area in dotted decimal notation.

If you define more than one area in an OSPFv2 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become (ABRs). An ABR connects to both the backbone area and at least one other defined area.

Figure 24 : OSPFv2 Areas



The ABR has a separate link-state database for each area to which it connects. The ABR sends Network Summary (type 3) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In the OSPFv2 Areas Figure, Area 0 sends summarized information about Area 5 to Area 3.

OSPFv2 defines one other router type: the autonomous system boundary router (ASBR). This router connects an OSPFv2 area to another autonomous system. An autonomous system is a network controlled by a single technical administration entity. OSPFv2 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system.

6.2.6 Link-State Advertisements

Link-State Advertisements Types

OSPFv2 uses link-state advertisements (LSAs) to build its routing table.

Names	Description
Router LSA	LSA sent by every router. This LSA includes the state and the cost of all links and a list of all OSPFv2 neighbors on the link. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to local OSPFv2 area.
Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network. Network LSAs trigger an SPF recalculation.
Network Summary LSA	LSA sent by the area border router to an external area for each destination in the local area. This LSA includes the link cost from the area border router to the local destination.
ASBR Summary LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only.
AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system.

NSSA External LSA	LSA generated by the ASBR within a not-so-stubby area (NSSA). This LSA includes the link cost to an external autonomous system destination. NSSA External LSAs are flooded only within the local NSSA.
Opaque LSAs	LSA used to extend OSPF.

Link Cost

Each OSPFv2 interface is assigned a . The cost is an arbitrary number. By default, Inspur INOS assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. The link cost is carried in the LSA updates for each link.

Flooding and LSA Group Pacing

When an OSPFv2 router receives an LSA, it forwards that LSA out every OSPF-enabled interface, flooding the OSPFv2 area with this information. This LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv2 area configuration. The LSAs are flooded based on the (every 30 minutes by default). Each LSA has its own link-state refresh time.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer utilization. This feature groups LSAs with similar link-state refresh times to allow OSPFv2 to pack multiple LSAs into an OSPFv2 Update message.

By default, LSAs with link-state refresh times within 10 seconds of each other are grouped together. You should lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv2 load on your network.

Link-State Database

Each router maintains a link-state database for the OSPFv2 network. This database contains all the collected LSAs, and includes information on all the routes through the network. OSPFv2 uses this information to calculate the best path to each destination and populates the routing table with these best paths.

LSAs are removed from the link-state database if no LSA update has been received within a set interval, called the MaxAge. Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from being aged out. Inspur INOS supports the LSA grouping feature to prevent all LSAs from refreshing at the same time.

Opaque LSAs

Opaque LSAs allow you to extend OSPF functionality. Opaque LSAs consist of a standard LSA header followed by application-specific information. This information might be used by OSPFv2 or by other applications. OSPFv2 uses Opaque LSAs to support OSPFv2 Graceful Restart capability. Three Opaque LSA types are defined as follows:

- LSA type 9—Flooded to the local network.
- LSA type 10—Flooded to the local area.
- LSA type 11—Flooded to the local autonomous system.

6.2.7 OSPFv2 and the Unicast RIB

OSPFv2 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The resultant shortest path for each destination is then put in the OSPFv2 route table. When the OSPFv2 network is converged, this route table feeds into the unicast RIB. OSPFv2 communicates with the unicast RIB to do the following:

- Add or remove routes
- Handle route redistribution from other protocols
- Provide convergence updates to remove stale OSPFv2 routes and for stub router advertisements

OSPFv2 also runs a modified Dijkstra algorithm for fast recalculation for summary and external (type 3, 4, 5, and 7) LSA changes.

6.2.8 Authentication

You can configure authentication on OSPFv2 messages to prevent unauthorized or invalid routing updates in your network. Inspur INOS supports two authentication methods:

- Simple password authentication
- MD5 authentication digest

You can configure the OSPFv2 authentication for an OSPFv2 area or per interface.

Simple Password Authentication

Simple password authentication uses a simple clear-text password that is sent as part of the OSPFv2 message. The receiving OSPFv2 router must be configured with the same clear-text password to accept the OSPFv2 message as a valid route update. Because the password is in clear text, anyone who can watch traffic on the network can learn the password.

MD5 Authentication

You should use MD5 authentication to authenticate OSPFv2 messages. You configure a password that is shared at the local router and all remote OSPFv2 neighbors. For each OSPFv2 message, Inspur INOS creates an MD5 one-way message digest based on the message itself and the encrypted password. The interface sends this digest with the OSPFv2 message. The receiving OSPFv2 neighbor validates the digest using the same encrypted password. If the message has not changed, the digest calculation is identical and the OSPFv2 message is considered valid.

MD5 authentication includes a sequence number with each OSPFv2 message to ensure that no message is replayed in the network.

6.2.9 Advanced Features for OSPFv2

Inspur INOS supports advanced OSPFv2 features that enhance the usability and scalability of OSPFv2 in the network.

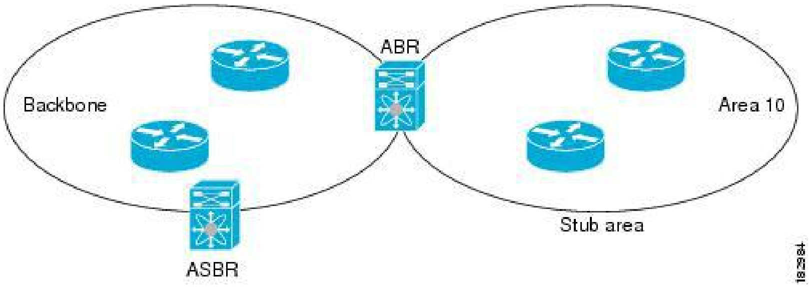
Stub Area

You can limit the amount of external routing information that floods an area by making it a stub area. A stub area is an area that does not allow AS External (type 5) LSAs. These LSAs are usually flooded throughout the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area are stub routers.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

The following figure shows an example of an OSPFv2 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 25 : Stub Area



Stub areas use a default route for all traffic that needs to go through the backbone area to the external autonomous system. The default route is 0.0.0.0 for IPv4.

Not-So-Stubby Area

A Not-so-Stubby Area (NSSA) is similar to a stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates NSSA External (type 7) LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this NSSA External LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv2 autonomous system.

Summarization and filtering are supported during the translation.

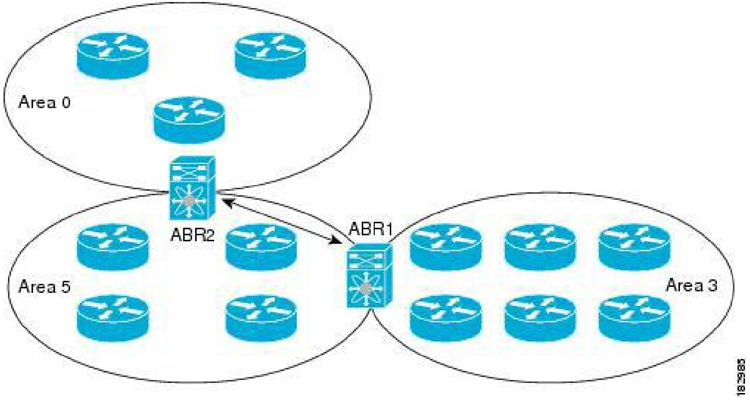
You can, for example, use NSSA to simplify administration if you are connecting a central site using OSPFv2 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv2 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv2 to cover the remote connection by defining the area between the corporate router and remote router as an NSSA.

The backbone Area 0 cannot be an NSSA.

Virtual Links

Virtual links allow you to connect an OSPFv2 area ABR to a backbone area ABR when a direct physical connection is not available. The figure shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 26 : Virtual Links



You can also use virtual links to temporarily recover from a partitioned area, which occurs when a link within the area fails, isolating part of the area from reaching the designated ABR to the backbone area.

Route Redistribution

OSPFv2 can learn routes from other routing protocols by using route redistribution. You configure OSPFv2 to assign a link cost for these redistributed routes or a default link cost for all redistributed routes.

Route redistribution uses route maps to control which external routes are redistributed. You must configure a route map with the redistribution to control which routes are passed into OSPFv2. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. You can use route maps to modify parameters in the AS External (type 5) and NSSA External (type 7) LSAs before these external routes are advertised in the local OSPFv2 autonomous system.

OSPFv2 sets the type-5 LSA's forwarding address as described below:

- If the next-hop for the route is an attached-route then the forwarding address is the next-hop address for that route.
- If the next-hop for the route is a recursive route and next-hop's next-hop is an attached route then the forwarding address is the next-hop's next-hop address.

Route Summarization

Because OSPFv2 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents

all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas. The two types of summarization are as follows

- Inter-area route summarization
- External route summarization

You configure inter-area route summarization on ABRs, summarizing routes between areas in the autonomous system. To take advantage of summarization, you should assign network numbers in areas in a contiguous way to be able to lump these addresses into one range.

External route summarization is specific to external routes that are injected into OSPFv2 using route redistribution. You should make sure that external ranges that are being summarized are contiguous. Summarizing overlapping ranges from two different routers could cause packets to be sent to the wrong destination. Configure external route summarization on ASBRs that are redistributing routes into OSPF.

When you configure a summary address, Inspur INOS automatically configures a discard route for the summary address to prevent routing black holes and route loops.

High Availability and Graceful Restart

Inspur INOS provides a multilevel high-availability architecture. OSPFv2 supports stateful restart, which is also referred to as non-stop routing (NSR). If OSPFv2 experiences problems, it attempts to restart from its previous run-time state. The neighbors do not register any neighbor event in this case. If the first restart is not successful and another problem occurs, OSPFv2 attempts a graceful restart.

A graceful restart, or nonstop forwarding (NSF), allows OSPFv2 to remain in the data forwarding path through a process restart. When OSPFv2 needs to perform a graceful restart, it sends a link-local opaque (type 9) LSA, called a grace LSA. This restarting OSPFv2 platform is called NSF capable.

The grace LSA includes a grace period, which is a specified time that the neighbor OSPFv2 interfaces hold onto the LSAs from the restarting OSPFv2 interface. (Typically, OSPFv2 tears down the adjacency and discards all LSAs from a down or restarting OSPFv2 interface.) The participating neighbors, which are called NSF helpers, keep all LSAs that originate from the restarting OSPFv2 interface as if the interface was still adjacent.

When the restarting OSPFv2 interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its LSA updates again. At this point, the NSF helpers recognize that the graceful restart has finished.

Stateful restart is used in the following scenarios:

- First recovery attempt after the process experiences problems
- ISSU
- User-initiated switchover using the **system switchover** command
- Active supervisor reload using the **reload module** *active-sup* command Graceful restart is used in the following scenarios:
- Second recovery attempt after the process experiences problems within a 4-minute interval
- Manual restart of the process using the restart ospf command
- Active supervisor removal

OSPFv2 Stub Router Advertisements

You can configure an OSPFv2 interface to act as a stub router using the OSPFv2 Stub Router Advertisements feature. Use this feature when you want to limit the OSPFv2 traffic through this router, such as when you want to introduce a new router to the network in a controlled manner or limit the load on a router that is already overloaded. You might also want to use this feature for various administrative or traffic engineering reasons.

OSPFv2 stub router advertisements do not remove the OSPFv2 router from the network topology, but they do prevent other OSPFv2 routers from using this router to route traffic to other parts of the network. Only the traffic that is destined for this router or directly connected to this router is sent.

OSPFv2 stub router advertisements mark all stub links (directly connected to the local router) to the cost of the local OSPFv2 interface. All remote links are marked with the maximum cost (0xFFFF).

Multiple OSPFv2 Instances

Inspur INOS supports multiple instances of the OSPFv2 protocol that run on the same node. You cannot configure multiple instances over the same interface. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv2 autonomous system.

SPF Optimization

Inspur INOS optimizes the SPF algorithm in the following ways:

- Partial SPF for Network (type 2) LSAs, Network Summary (type 3) LSAs, and AS External (type 5) LSAs—When there is a change on any of these LSAs, Inspur INOS performs a faster partial calculation rather than running the whole SPF calculation.
- SPF timers—You can configure different timers for controlling SPF calculations. These timers include exponential backoff for subsequent SPF calculations. The exponential backoff limits the CPU load of multiple SPF calculations.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol that provides fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules. See the Inspur CN12700 Series INOS Interfaces Configuration Guide for more information.

Virtualization Support for OSPFv2

OSPFv2 supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

Inspur INOS Release 8.4(1) or later supports more than four process instances for OSPFv2 per VDC. However, only the first four configured OSPFv2 instances are supported with MPLS LDP and MPLS TE. Each OSPFv2 instance can support multiple VRFs, up to the system limit. For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide and the Inspur CN12700 Series INOS Verified Scalability Guide.*

6.3 Licensing Requirements for OSPFv2

The following table shows the licensing requirements for this feature:

Product	**LicenseRequirement**
Inspur INOS	OSPFv2 requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *Inspur INOS Licensing Guide*.

6.4 Prerequisites for OSPFv2

OSPFv2 has the following prerequisites:

- You must be familiar with routing fundamentals to configure OSPF.
- You are logged on to the switch.
- You have configured at least one interface for IPv4 that can communicate with a remote OSPFv2 neighbor.
- You have installed the Enterprise Services license.
- You have completed the OSPFv2 network strategy and planning for your network. For example, you must decide whether multiple areas are required.
- You have enabled the OSPF feature.

You have installed the appropriate license and entered the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide* for configuration information and the Inspur INOS Licensing Guide for licensing information) if you are configuring VDCs.

6.5 Guidelines and Limitations for OSPFv2

OSPFv2 has the following configuration guidelines and limitations:

- CE devices installs type 3 LSAs with DN-bit or Type 5 LSAs with DN-bit and VPN Route TAG in the RIB (non-default VRF). This behaviour is applicable prior to Inspur INOS Release 8.4(1).
- The default-information originate command must be configured so that the MPLS default route is advertised to the CE-VRF. When using default-information originate command, the DN-bit in type 3 5 LSAs options and Route TAGs in Type 5 LSAs are not set for the default route only.
- The Inspur CN12700 supports the Internet Engineering Task Force (IETF) version only. As a result, NSF IETF must be explicitly configured under the routing protocols in the Virtual Switching System (VSS). No additional configuration is required on the Inspur CN12700 pairs because they run NSF IETF graceful-restart by default. However, each neighbor device that will become Layer 3 adjacent must have NSF configured and the same mode of NSF must be enabled to successfully operate a graceful failover.

- Inspur INOS displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.
- All OSPFv2 routers must operate in the same RFC compatibility mode. OSPFv2 for Inspur INOS complies with RFC 2328. Use the **rfc1583compatibility** command in router configuration mode if your network includes routers that support only RFC 1583.
- In scaled scenarios, when the number of interfaces and link-state advertisements in an OSPF process is large, the snmp-walk on OSPF MIB objects is expected to time out with a small-values timeout at the SNMP agent. If your observe a timeout on the querying SNMP agent while polling OSPF MIB objects, increase the timeout value on the polling SNMP agent.
- MTU configured at interface level works in either the data plane or in the control plane but not at both planes at the same time.

When you configure MTU with a size lower than the supported size in data and control planes a few features that have minimum MTU requirements may not work in both the planes.

For example, MPLS VPN is supported in the data plane since this plane supports the MTU of 1500 bytes that the MPLS VPN requires. But control plane does not support MPLS VPN because this plane cannot handle the 1500-byte packets.

To make the configured MTU work in control plane for MPLS VPN, you need to manually configure the OSPF packet size (by using the **packet-size***size* command) so that OSPF works on the control plane. This is applicable from Inspur INOS Release 8.4(1) onwards.

The **packet-size***size* command is supported on the Ethernet, SVI, and GRE tunnel interfaces.

- Inspur INOS Release 8.4(1) or later supports more than four process instances for OSPFv2 per VDC. However, only the first four configured OSPFv2 instances are supported with MPLS LDP and MPLS TE.
- The following guidelines and limitations apply to the administrative distance feature, which is supported beginning with Inspur INOS Release 8.4(1):
    - When an OSPF route has two or more equal cost paths, configuring the administrative distance is non-deterministic for the **match ip route-source** command.
    - Configuring the administrative distance is supported only for the **match route-type**, **match ip address prefix-list**, and **match ip route-source prefix-list** commands. The other match statements are ignored.
    - There is no preference among the **match route-type**, **match ip address**, and **match ip route-source** commands for setting the administrative distance of OSPF routes. In this way, the behavior of the table map for setting the administrative distance in Inspur INOS OSPF is different from that in Inspur IOS OSPF
    - The discard route is always assigned an administrative distance of 220. No configuration in the table map applies to OSPF discard routes.
    - In Inspur INOS Release 8.4(1) and later releases, you can filter next-hop paths for an OSPF route to prevent the path from being added to the RIB. Before Inspur INOS Release 8.4(1), filtering on a specific path was ignored and the entire route was not added to the RIB.

6.6 Default Settings for OSPFv2

Table 10: Default OSPFv2 Parameters

Parameters	Default
Administrative distance	110
Hello interval	10 seconds
Dead interval	40 seconds
Discard routes	Enabled

Graceful restart grace period	60 seconds
OSPFv2 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	10 seconds
SPF calculation initial delay time	200 milliseconds
SPF minimum hold time	5000 milliseconds
SPF calculation initial delay time	1000 milliseconds

6.7 Configuring Basic OSPFv2

6.7.1 Enabling OSPFv2

You must enable the OSPFv2 feature before you can configure OSPFv2.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature ospf**	Enables the OSPFv2 feature. **Note** Use the **no** form of this command to disable the OSPFv2 feature and remove all associated configuration.
Step 3	(Optional) switch(config)# **show feature**	Displays enabled and disabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

6.7.2 Creating an OSPFv2 Instance

The first step in configuring OSPFv2 is to create an OSPFv2 instance. You assign a unique instance tag for this OSPFv2 instance. The instance tag can be any string.

Before you begin

Ensure that you have enabled the OSPF feature.

Use the **show ip ospf** instance-tag command to verify that the instance tag is not in use.

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

Ensure that you are in the correct VDC (or use the switch to vdc command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag. **Note** Use the **no** form of this command in global configuration mode to remove the OSPFv2 instance and all associated configurations. Using the **no** form of this command in the interface configuration mode does not remove the OSPF configuration. You must manually remove any OSPFv2 commands configured in interface mode.
Step 3	(Optional) switch(config-router)# **router-id** *ip-address*	Configures the OSPFv2 router ID. This IP address identifies this OSPFv2 instance and must exist on a configured interface in the system. This command restarts the OSPF process automatically and changes the router id after it is configured.
Step 4	(Optional) switch(config-router)# **show ip ospf** *instance-tag*	Displays OSPF information.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

6.7.3 Configuring OSPF Packet Size

MTU configured at interface level works in either the data plane or in the control plane but not at both planes at the same time.

When you configure MTU with a size lower than the supported size in data and control planes a few features that have minimum MTU requirements may not work in both the planes.

For example, MPLS VPN is supported in the data plane since this plane supports the MTU of 1500 bytes that the MPLS VPN requires. But control plane does not support MPLS VPN because this plane cannot handle the 1500-byte packets.

To make the configured MTU work in control plane for MPLS VPN, you need to manually configure the OSPF packet size so that OSPF works on the control plane. This is applicable from Inspur INOS Release 8.4(1)onwards.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **router ospf** *instance-tag*
3. switch(config-router)# **router-id** *ip-address*
4. switch(config-router)# **packet-size** *size*

5. (Optional) switch(config-router)# **show ip ospf interface** *interface-number*

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **router ospf** *instance-tag*	Creates a new OSPF instance with the configured instance tag. **Note** Use the **no** form of this command in global configuration mode to remove the OSPFv2 instance and all associated configurations. Using the **no** form of this command in the interface configuration mode does not remove the OSPF configuration. You must manually remove any OSPFv2 commands configured in interface mode.
Step 3	switch(config-router)# **router-id** *ip-address*	Configures the OSPFv2 router ID. This IP address identifies this OSPFv2 instance and must exist on a configured interface in the system. This command restarts the OSPF process automatically and changes the router id after it is configured.
Step 4	switch(config-router)# **packet-size** *size*	• Configures the OSPFv2 packet size. The size range is from 572 to 9212 bytes. • You can configure the packet-size in the interface configuration mode also. • You can configure the **packet-size** *size* command even if the**ip ospf mtu-ignore** command is already configured in the network.
Step 5	(Optional) switch(config-router)# **show ip ospf interface** *interface-number*	Displays OSPF information.

Example

This example shows how to configure the OSPF packet-size:

```
router ospf 1

  router-id 3.3.3.3

  [no] packet-size 2000
```

This example shows the display of the OSPF packet-size:

```
Switch (config-router)# show ip ospf interface ethernet 1/25
```

```
       Ethernet1/25  is  up,  line
           protocol   is   up   IP
           address 1.0.0.1/24

---------snip ---------------

           Number  of  opaque  link  LSAs:  0,
           checksum  sum  0  Max  Packet  Size:
           2000
```

6.7.4 Configuring Optional Parameters on an OSPFv2 Instance

You can configure optional parameters for OSPF. The following commands are available in the router configuration mode.

For more information about OSPFv2 instance parameters, see the 〝Configuring Advanced OSPFv2〞 section

Before you begin

Ensure that you have enabled the OSPF feature.

OSPFv2 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch(config-router)# **distance** *number*
2. switch(config-router)# **log-adjacency-changes [detail]**
3. switch(config-router)# **maximum-paths** *path-number*
4. switch(config-router)# [**no**]**name-lookup** *path-number*
5. switch(config-router)# **passive-interface default**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **distance** *number*	Configures the administrative distance for this OSPFv2 instance. The range is from 1 to 255. The default is 110.
Step 2	switch(config-router)# **log-adjacency-changes [detail]**	Generates a system message whenever a neighbor changes state.
Step 3	switch(config-router)# **maximum-paths** *path-number*	Configures the maximum number of equal OSPFv2 paths to a destination in the route table. This command is used for load balancing. The range is from 1 to 16. The default is 8.

Step 4	switch(config-router)# [**no**]**name-lookup** *path-number*	Enables the translation of OSPF router IDs to host names, either by looking up the local hosts database or querying DNS names in IPv6. This command makes it easier to identify a device because it displays the device by name rather than by its router ID or neighbor ID. **Note** To stop displaying OSPF router IDs as DNS names, use the no form of this command.
Step 5	switch(config-router)# **passive-interface default**	Suppresses routing updates on all interfaces. This command is overridden by the VRF or interface command mode configuration.

Example

This example shows how to create an OSPFv2 instance:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# copy running-config startup-config
```

6.7.5 Configuring Networks in OSPFv2

You can configure a network to OSPFv2 by associating it through the interface that the router uses to connect to that network. You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

Before you begin

Ensure that you have enabled the OSPF feature

Ensure that you are in the correct VDC (or use the switchto vdc command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip address** *ip-prefix/length*	Assigns an IP address and subnet mask to this interface.
Step 4	switch(config-if)# **ip router ospf** *instance-tag* **area** *area-id* [**secondaries none**]	Adds the interface to the OSPFv2 instance and area.
Step 5	(Optional) switch(config-if)# **show ip ospf** *instance-tag* **interface** *interface-type slot/port*	Displays OSPF information.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Step 7	(Optional) switch(config)# **ip ospf cost** *number*	Configures the OSPFv2 cost metric for this interface. The default is to calculate cost metric, based on reference bandwidth and interface bandwidth. The range is from 1 to 65535.
Step 8	(Optional) switch(config)# **ip ospf dead-interval** *seconds*	Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
Step 9	(Optional) switch(config)# **ip ospf hello-interval** *seconds*	Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
Step 10	(Optional) switch(config)# **ip ospf mtu-ignore**	Configures OSPFv2 to ignore any IP MTU mismatch with a neighbor. The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.
Step 11	(Optional) switch(config)# **[default | no] ip ospf passive-interface**	Suppresses routing updates on the interface. This command overrides the router or VRF command mode configuration. The default option removes this interface mode command and reverts to the router or VRF configuration, if present.
Step 12	(Optional) switch(config)# **ip ospf priority** *number*	Configures the OSPFv2 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1.
Step 13	(Optional) switch(config)# **ip ospf shutdown**	Shuts down the OSPFv2 instance on this interface.

Example

This example shows how to add a network area 0.0.0.10 in OSPFv2 instance 201:

```
switch# configure terminal

switch(config)# interface ethernet 1/2



switch(config-if)#        ip        address   192.0.2.1/16
switch(config-if)#  ip   router   ospf   201   area   0.0.0.10
switch(config-if)# copy running-config startup-config
```

Use the **show ip ospf interface** command to verify the interface configuration. Use the **show ip ospf neighbor** command to see the neighbors for this interface.

6.7.6 Configuring Authentication for an Area

You can configure authentication for all networks in an area or for individual interfaces in the area. Interface authentication configuration overrides area authentication.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

Create the key chain for this authentication configuration. See the *Inspur CN12700 Series INOS Security Configuration Guide*

Ensure that you are in the correct VDC (or use the switchto vdc command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **area** *area-id* **authentication** [**message-digest**]	Configures the authentication mode for an area.
Step 4	switch(config-router)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 5	(Optional) Configure one of the following commands: • **ip ospf authentication-key** [**0** | **3**] *key* • **ip ospf message-digest-key** *key-id* **md5** [**0** | **3**] *key*	The first command configures simple password authentication for this interface. Use this command if the authentication is not set to key-chain or message-digest. The **0** keyword configures the password in clear text. The **3** keyword configures the password as 3DES encrypted. The second command configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The key-id range is from 1 to 255. The MD5 option 0 configures the password in clear text and 3 configures the pass key as 3DES encrypted.
Step 6	(Optional) switch(config)# **show ip ospf** *instance-tag* **interface** *interface-type slot/port*	Displays OSPF information.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

6.7.7 Configuring Authentication for an Interface

You can configure authentication for all networks in an area or for individual interfaces in the area. Interface authentication configuration overrides area authentication.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that all neighbors on an interface share the same authentication configuration, including the shared authentication key.

Create the key chain for this authentication configuration. See the *Inspur CN12700 Series INOS Security Configuration Guide*.

Ensure that you are in the correct VDC (or use the switchto vdc command).

Procedure

	Command or Action	Purpose

Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip ospf authentication** [**message-digest**]	Enables interface authentication mode for OSPFv2 for either cleartext or message-digest type. Overrides area-based authentication for this interface. All neighbors must share this authentication type.
Step 4	(Optional) switch(config-if)# **ip ospf authentication key-chain** *key-name*	Configures interface authentication to use key chains for OSPFv2. For details on key chains, see the *Inspur CN12700 Series INOS Security Configuration Guide*.
Step 5	(Optional) switch(config-if)# **ip ospf authentication-key** [**0** | **3** | **7**] *key*	Configures simple password authentication for this interface. Use this command if the authentication is not set to key-chain or message-digest. The options are as follows: • **0**—configures the password in clear text. • **3**—configures the pass key as 3DES encrypted. • **7**—configures the key as Inspur type 7 encrypted.
Step 6	(Optional) switch(config-if)# **ip ospf message-digest-key** *key-id* **md5** [**0** | **3** | **7**] *key*	Configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The key-id range is from 1 to 255. The MD5 options are as follows: • **0**—configures the password in clear text. • **3**—configures the pass key as 3DES encrypted. • **7**—configures the key as Inspur type 7 encrypted.
Step 7	(Optional) switch(config-if)# **show ip ospf** *instance-tag* **interface** *interface-type slot/port*	Displays OSPF information.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to set an interface for simple, unencrypted passwords and set the password for Ethernet interface 1/2:

```
switch#        configure        terminal
switch(config)#    router     ospf   201
switch(config-router)#              exit
switch(config)# interface ethernet 1/2

switch(config-if)#  ip  router  ospf  201  area  0.0.0.10
switch(config-if)#    ip    ospf authentication
switch(config-if)#  ip  ospf authentication-key 0 mypass
switch(config-if)# copy running-config startup-config
```

6.8 Configuring Advanced OSPFv2

6.8.1 Configuring Filter Lists for Border Routers

You can separate your OSPFv2 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv2 domains can connect to external domains as well, through an autonomous system border router (ASBR).

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas.
- Filter list—Filters the Network Summary (type 3) LSAs that are allowed in from an external area.

ASBRs also support filter lists.

Before you begin

Ensure that you have enabled the OSPF feature.

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Network Summary (type 3) LSAs.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **area** *area-id* **filter-list route-map** *map-name* {**in** | **out**}
4. (Optional) switch(config-if)# **show ip ospf policy statistics area** *id* **filter-list** {**in** | **out**}
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **area** *area-id* **filter-list route-map** *map-name* {**in** | **out**}	Filters incoming or outgoing Network Summary (type 3) LSAs on an ABR.
Step 4	(Optional) switch(config-if)# **show ip ospf policy statistics area** *id* **filter-list** {**in** | **out**}	Displays OSPF policy information.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a filter list in area 0.0.0.10:

```
switch# configure terminal
```

```
switch(config)# router ospf 201

switch(config-router)#  area  0.0.0.10  filter-list  route-map FilterLSAs in

switch(config-router)# copy running-config startup-config
```

6.8.2 Configuring Stub Areas

You can configure a stub area for part of an OSPFv2 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs and limit unnecessary routing to and from selected networks. You can optionally block all summary routes from going into the stub area.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that there are no virtual links or ASBRs in the proposed stub area. Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **area** *area-id* **stub**
4. (Optional) switch(config-router)# **area** *area-id* **default-cost** *cost*
5. (Optional) switch(config-if)# **show ip ospf** *instance-tag*
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **area** *area-id* **stub**	Creates this area as a stub area.
Step 4	(Optional) switch(config-router)# **area** *area-id* **default-cost** *cost*	Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215. The default is 1.
Step 5	(Optional) switch(config-if)# **show ip ospf** *instance-tag*	Displays OSPF information.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a stub area:

```
switch#    configure    terminal
switch(config)#  router  ospf  201
switch(config-router)#      area 0.0.0.10 stub
switch(config-router)#copy running-config startup-config
```

6.8.3 Configuring a Totally Stubby Area

You can create a totally stubby area and prevent all summary route updates from going into the stub area. To create a totally stubby area, use the following command in router configuration mode:

Command	**Purpose**
router ospf *instance-tag*	Creates this area as a totally stubby area.

6.8.4 Configuring NSSA

You can configure an NSSA for part of an OSPFv2 domain where limited external traffic is required. You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv2 domain with this routing information. An NSSA can be configured with the following optional parameters:

- No redistribution—Redistributed routes bypass the NSSA and are redistributed to other areas in the OSPFv2 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- Default information originate—Generates an NSSA External (type 7) LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- Route map—Filters the external routes so that only those routes that you want are flooded throughout the NSSA and other areas.
- Translate—Translates NSSA External LSAs to AS External LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv2 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs. If you choose this option, the forwarding address is set to 0.0.0.0.
- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that there are no virtual links in the proposed NSSA and that it is not the backbone area. Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)#**area** *area-id* **nssa [no-redistribution] [default-information-originate]originate** [**route-map** *map-name*]] [**no-summary**] [**translate type7** {**always | never**} [**suppress-fa**]]
4. (Optional) switch(config-router)# **area** *area-id* **default-cost** *cost*
5. (Optional) switch(config-if)# **show ip ospf** *instance-tag*
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.

Step 3	switch(config-router)# **area** *area-id* **nssa [no-redistribution] [default-information-originate]originate** [**route-map** *map-name*]] [**no-summary**] [**translate type7** {**always | never**} [**suppress-fa**]]	Creates this area as an NSSA.
Step 4	(Optional) switch(config-router)# **area** *area-id* **default-cost** *cost*	Sets the cost metric for the default summary route sent into this NSSA.
Step 5	(Optional) switch(config-if)# **show ip ospf** *instance-tag*	Displays OSPF information.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)#  area  0.0.0.10  nssa  no-summary
switch(config-router)#copy running-config startup-config
```

This example shows how to create an NSSA that generates a default route:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# area 0.0.0.10 nssa default-info-originate

switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that filters external routes and blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that always translates NSSA External (type 5) LSAs to AS External (type 7) LSAs:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)#copy running-config startup-config
```

6.8.5 Configuring Virtual Links

A virtual link connects an isolated area to the backbone area through an intermediate area. You can configure the following optional parameters for a virtual link:

- Authentication—Sets a simple password or MD5 message digest authentication and associated keys.
- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

You cannot add a virtual link to a stub area.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **area** *area-id* **virtual link** *router-id*
4. (Optional) switch(config-router-vlink)# **show ip ospf virtual-link** [**brief**]
5. (Optional) switch(config-router-vlink)# **authentication** [**key-chain** *key-id* **message-digest** | **null**]
6. (Optional) switch(config-router-vlink)# **authentication-key** [**0** | **3**] *key*
7. (Optional) switch(config-router-vlink)# **dead-interval** *seconds*
8. (Optional) switch(config-router-vlink)# **hello-interval** *seconds*
9. (Optional) switch(config-router-vlink)# **message-digest-key** *key-id* **md5** [**0** | **3**] *key*
10. (Optional) switch(config-router-vlink)# **retransmit-interval** *seconds*
11. (Optional) switch(config-router-vlink)# **transmit-delay** *seconds*
12. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **area** *area-id* **virtual link** *router-id*	Creates one end of a virtual link to a remote router. You must create the virtual link on that remote router to complete the link.
Step 4	(Optional) switch(config-router-vlink)# **show ip ospf virtual-link** [**brief**]	Displays OSPF virtual link information.
Step 5	(Optional) switch(config-router-vlink)# **authentication** [**key-chain** *key-id* **message-digest** | **null**]	Overrides area-based authentication for this virtual link.

Step 6	(Optional) switch(config-router-vlink)# **authentication-key** [**0** | **3**] *key*	Configures a simple password for this virtual link. Use this command if the authentication is not set to key-chain or message-digest. 0 configures the password in clear text. 3 configures the password as 3DES encrypted.
Step 7	(Optional) switch(config-router-vlink)# **dead-interval** *seconds*	Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
Step 8	(Optional) switch(config-router-vlink)# **hello-interval** *seconds*	Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
Step 9	(Optional) switch(config-router-vlink)# **message-digest-key** *key-id* **md5** [**0** | **3**] *key*	Configures message digest authentication for this virtual link. Use this command if the authentication is set to message-digest. 0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted.
Step 10	(Optional) switch(config-router-vlink)# **retransmit-interval** *seconds*	Configures the OSPFv2 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
Step 11	(Optional) switch(config-router-vlink)# **transmit-delay** *seconds*	Configures the OSPFv2 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.
Step 12	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a simple virtual link between two ABRs. The configuration for ABR 1 (router ID 27.0.0.55) is as follows:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)#  area  0.0.0.10  virtual-link 10.1.2.3
switch(config-router)# copy running-config startup-config
```

The configuration for ABR 2 (Router ID 10.1.2.3) is as follows:

```
switch# configure terminal

switch(config)# router ospf 101

switch(config-router)# area 0.0.0.10 virtual-link 27.0.0.55

switch(config-router)# copy running-config startup-config
```

6.8.6 Configuring Redistribution

You can redistribute routes learned from other routing protocols into an OSPFv2 autonomous system through the ASBR.

You can configure the following optional parameters for route redistribution in OSPF:

- Default information originate—Generates an AS External (type 5) LSA for a default route to the external autonomous system.
- Default metric—Sets all redistributed routes to the same cost metric.

Before you begin

Ensure that you have enabled the OSPF feature.

Create the necessary route maps used for redistribution.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* **ospf** *id* **rip** *id* | **static**} **route-map** *map-name*
4. switch(config-router)# **default-information originate** [**always**] [**route-map** *map-name*]
5. switch(config-router)# **default-metric** [*cost*]
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* **ospf** *id* **rip** *id* | **static**} **route-map** *map-name*	Redistributes the selected protocol into OSPF through the configured route map. **Note** If you redistribute static routes, Inspur INOS also redistributes the default static route.
Step 4	switch(config-router)# **default-information originate** [**always**] [**route-map** *map-name*]	Creates a default route into this OSPF domain if the default route exists in the RIB. Use the following optional keywords: • **always** —Always generate the default route of 0.0.0. even if the route does not exist in the RIB • **route-map**—Generate the default route if the route map returns true. **Note** This command ignores **match** statements in the route map.

Step 5	switch(config-router)# **default-metric** [*cost*]	Sets the cost metric for the redistributed routes. This command does not apply to directly connectedroutes. Use a route map to set the default metric for directly connected routes.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# redistribute bgp route-map FilterExternalBGP

switch(config-router)# copy running-config startup-config
```

6.8.7 Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the OSPFv2 route table. You can configure a maximum limit to the number of routes accepted from external protocols. OSPFv2 provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when OSPFv2 reaches the configured maximum. OSPFv2 does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where OSPFv2 logs a warning when that threshold is passed.
- Warning only—Logs a warning only when OSPFv2 reaches the maximum. OSPFv2 continues to accept redistributed routes.
- Withdraw—Starts the timeout period when OSPFv2 reaches the maximum. After the timeout period, OSPFv2 requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, OSPFv2 withdraws all redistributed routes. You must clear this condition before OSPFv2 accepts more redistributed routes.
- You can optionally configure the timeout period.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **redistribute** {**bgp** *id* **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map**
4. *map-name*
5. switch(config-router)# **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw**
6. [*num-retries timeout*]]
7. (Optional) switch(config-router)# **show running-config ospf**
8. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **redistribute** {**bgp** *id* **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name*	Redistributes the selected protocol into OSPF through the configured route map.
Step 4	switch(config-router)# **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]]	Specifies a maximum number of prefixes that OSPFv2 distributes. The range is from 0 to 65536. Optionally specifies the following: • *threshold*—Percent of maximum prefixes that trigger a warning message. • **warning-only**—Logs an warning message when the maximum number of prefixes is exceeded. • **withdraw**—Withdraws all redistributed routes. Optionally tries to retrieve the redistributed routes. The *num-retries* range is from 1 to 12. The *timeout* is 60 to 600 seconds. The default is 300 seconds. Use the **clear ip ospf redistribution** command if all routes are withdrawn.
Step 5	(Optional) switch(config-router)# **show running-config ospf**	Displays the OSPFv2 configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to limit the number of redistributed routes into OSPF:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# redistribute bgp route-map Filter External BGP

switch(config-router)# redistribute maximum-prefix 1000 75
```

6.8.8 Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. You can also configure route summarization for external, redistributed routes by configuring a summary address for those routes on an ASBR.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. Configure one of the following commands:
   - **area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*]
   - **summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag*]
4. (Optional) switch(config-router)# [**no**] **discard route** {**internal** | **external**}
5. (Optional) switch(config-router)# **show ip ospf summary-address**
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	Configure one of the following commands: • **area** *area-id* **range** *ip-prefix/length* [**no-advertise**] [**cost** *cost*] • **summary-address** *ip-prefix/length* [**no-advertise** | **tag** *tag*]	The first command creates a summary address on an ABR for a range of addresses and optionally does not advertise this summary address in a Network Summary (type 3) LSA. The cost range is from 0 to 16777215. The second command creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.
Step 4	(Optional) switch(config-router)# [**no**] **discard route** {**internal** | **external**}	When you configure a summary address, Inspur INOS software automatically configures a discard route for the summary address to prevent routing black holes and route loops. You can use the **no** form of this command to prevent the discard routes from being created.
Step 5	(Optional) switch(config-router)# **show ip ospf summary-address**	Displays information about OSPF summary addresses.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create summary addresses between areas on an ABR:

```
switch# configure terminal

switch(config)# router ospf 201
switch(config-router)#  area 0.0.0.10 range 10.3.0.0/16

switch(config-router)# copy running-config startup-config
```

This example shows how to create summary addresses on an ASBR:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# summary-address 10.5.0.0/16

switch(config-router)# copy running-config startup-config
```

6.8.9 Configuring Stub Route Advertisements

Use stub route advertisements when you want to limit the OSPFv2 traffic through this router for a short time. Stub route advertisements can be configured with the following optional parameters:

- **on startup—**Sends stub route advertisements for the specified announce time.
- **wait-for bgp—**Sends stub router advertisements until BGP converges.

Before you begin

Ensure that you have enabled the OSPF feature.
Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** [*seconds*]] [**wait-for bgp** *tag*] [**summary-lsa** [*max-metric-value*]]
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**include-stub**] [**on-startup** [*seconds*]] [**wait-for bgp** *tag*] [**summary-lsa** [*max-metric-value*]]	Configures OSPFv2 stub route advertisements.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the stub router advertisements on startup for the default 600 seconds:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# max-metric router-lsa on-startup

switch(config-router)# copy running-config startup-config
```

6.8.10 Configuring the Administrative Distance of Routes

Beginning with Inspur INOS Release 8.4(1), you can set the administrative distance of routes added by OSPFv2 into the RIB.

The administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. Typically, a route can be learned through more than one routing protocol. The administrative distance is used to discriminate between routes learned from more than one routing protocol. The route with the lowest administrative distance is installed in the IP routing table.

Before you begin

Ensure that you have enabled OSPF.

Ensure that you are in the correct VDC (or use the **switchto vdc** command). See the guidelines and limitations for this feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# [**no**] **table-map** *map-name* [**filter**]
4. switch(config-router)# **exit**
5. switch(config)# **route-map** *map-name* [**permit** | **deny**] [*seq*]
6. switch(config-route-map)# **match route-type** *route-type*
7. switch(config-route-map)# **match ip route-source prefix-list** *name*
8. switch(config-route-map)# **match ip address prefix-list** *name*
9. switch(config-route-map)# **set distance** *value*
10. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# [**no**] **table-map** *map-name* [**filter**]	Configures the policy for filtering or modifying OSPFv2 routes before sending them to the RIB. You can enter up to 63 alphanumeric characters for the map name. The **filter** keyword specifies that only routes that are permitted by the route map(*map-name*) configuration are downloaded to the routing information base (RIB).
Step 4	switch(config-router)# **exit**	Exits router configuration mode.

Step 5	switch(config)# **route-map** *map-name* [**permit** | **deny**] [*seq*]	Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map. **Note** The **permit** option enables you to set the distance. If you use the **deny** option, the default distance is applied.
Step 6	switch(config-route-map)# **match route-type** *route-type*	Matches against one of the following route types: • external: The external route (BGP, EIGRP, and OSPF type 1 or 2) • inter-area: OSPF inter-area route • internal: The internal route (including the OSPF intra- or inter-area) • intra-area: OSPF intra-area route • nssa-external: The NSSA external route (OSPF type 1 or 2) • type-1: The OSPF external type 1 route • type-2: The OSPF external type 2 route
Step 7	switch(config-route-map)# **match ip route-source prefix-list** *name*	Matches the IPv4 route source address or router ID of a route to one or more IP prefix lists. Use the **ip prefix-list** command to create the prefix list.
Step 8	switch(config-route-map)# **match ip address prefix-list** *name*	Matches against one or more IPv4 prefix lists. Use the **ip prefix-list** command to create the prefix list.
Step 9	switch(config-route-map)# **set distance** *value*	Sets the administrative distance of routes for OSPFv2. The range is from 1 to 255.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the OSPFv2 administrative distance for inter-area routes to 150, for external routes to 200, and for all prefixes in prefix list p1 to 190:

```
switch#   configure   terminal
switch(config)# router ospf 201
switch(config-router)#        table-map        foo
switch(config-router)# exit
switch(config)#  route-map foo permit 10

switch(config-route-map)#   match   route-type inter-area
switch(config-route-map)#       set     distance      150
switch(config)#      route-map     foo     permit      20
switch(config-route-map)#    match  route-type  external
switch(config-route-map)#     set      distance      200
switch(config)#  route-map foo permit 30
switch(config-route-map)#   match   ip   route-source prefix-list  p1
switch(config-route-map)#    match    ip    address    prefix-list    p1
switch(config-route-map)# set distance 190
```

The following example shows how to configure a route map for blocking the next hops that are learned through VLAN 10:

```
switch(config)# route-map Filter-OSPF 10 deny

switch(config-route-map)#   match interface  VLAN  10
switch(config-route-map)# exit

switch(config)# route-map Filter-OSPF 20 permit
```

The following example shows how to configure the table-map command with the filter keyword to use a route map (Filter-OSPF) to remove the next-hop path that is learned through VLAN 10 but not the next-hop path that is learned through VLAN 20:

```
switch(config)# route ospf p1

switch(config-router)# table-map Filter-OSPF filter
```

6.8.11 Modifying the Default Timers

OSPFv2 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv2 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs that arrive from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message.
- Throttle LSAs—Sets the rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

At the interface level, you can also control the following timers:

- Retransmit interval—Sets the estimated time between successive LSAs
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

Before you begin

Ensure that you have enabled the OSPF feature.

Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **timers lsa-arrival** *msec*
4. switch(config-router)# **timers lsa-group-pacing** *seconds*
5. switch(config-router)# **timers throttle lsa** *start-time hold-interval max-time*
6. switch(config-router)# **timers throttle spf** *delay-time hold-time max-wait*
7. switch(config)# **interface** *type slot/port*

8. switch(config-if)# **ip ospf hello-interval** *seconds*
9. switch(config-if)# **ip ospf dead-interval** *seconds*
10. switch(config-if)# **ip ospf retransmit-interval** *seconds*
11. switch(config-if)# **ip ospf transmit-delay** *seconds*
12. (Optional) switch(config-if)# **show ip ospf**
13. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **timers lsa-arrival** *msec*	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	switch(config-router)# **timers lsa-group-pacing** *seconds*	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 240 seconds.
Step 5	switch(config-router)# **timers throttle lsa** *start-time hold-interval max-time*	Sets the rate limit in milliseconds for generating LSAs with the following timers: • *start-time*—The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. • *hold-interval*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. • *max-time*—The range is from 50 to 30,000 milliseconds. The default value is 5000milliseconds
Step 6	switch(config-router)# **timers throttle spf** *delay-time hold-time max-wait*	Sets the SPF best path schedule initial delay time and the minimum hold time in seconds between SPF best path calculations. The range is from 1 to 600000. The default is no delay time and 5000 millisecond hold time.
Step 7	switch(config)# **interface** *type slot/port*	Enters interface configuration mode.
Step 8	switch(config-if)# **ip ospf hello-interval** *seconds*	Sets the hello interval this interface. The range is from 1 to 65535. The default is 10.
Step 9	switch(config-if)# **ip ospf dead-interval** *seconds*	Sets the dead interval for this interface. The range is from 1 to 65535.
Step 10	switch(config-if)# **ip ospf retransmit-interval** *seconds*	Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.
Step 11	switch(config-if)# **ip ospf transmit-delay** *seconds*	Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.

Step 12	(Optional) switch(config-if)# **show ip ospf**	Displays information about OSPF.
Step 13	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to control LSA flooding with the lsa-group-pacing option:

```
switch# configure terminal

switch(config)# router ospf 201

switch(config-router)# timers lsa-group-pacing 300

 switch(config-router)# copy running-config startup-config
```

6.8.12 Configuring Graceful Restart

Graceful restart is enabled by default. You can configure the following optional parameters for graceful restart in an OSPFv2 instance:

- Grace period—Configures how long neighbors should wait after a graceful restart has started before tearing down adjacencies.
- Helper mode disabled—Disables helper mode on the local OSPFv2 instance. OSPFv2 does not participate in the graceful restart of a neighbor.
- Planned graceful restart only—Configures OSPFv2 to support graceful restart only in the event of a planned restart.

Before you begin

Ensure that you have enabled OSPF.

Ensure that all neighbors are configured for graceful restart with matching optional parameters set. Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **graceful-restart**
4. (Optional) switch(config-router)# **graceful-restart grace-period** *seconds*
5. (Optional) switch(config-router)# **graceful-restart helper-disable**
6. (Optional) switch(config-router)# **graceful-restart planned-only**
7. (Optional) switch(config-if)# **show ip ospf** *instance-tag*
8. (Optional) switch(config)# **copy running-config startup-config**
9.

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	switch(config-router)# **graceful-restart**	Enables a graceful restart. A graceful restart is enabled by default.
Step 4	(Optional) switch(config-router)# **graceful-restart grace-period** *seconds*	Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	(Optional) switch(config-router)# **graceful-restart helper-disable**	Disables helper mode. This feature is enabled by default.
Step 6	(Optional) switch(config-router)# **graceful-restart planned-only**	Configures a graceful restart for planned restarts only.
Step 7	(Optional) switch(config-if)# **show ip ospf** *instance-tag*	Displays OSPF information.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable a graceful restart if it has been disabled and set the grace period to 120 seconds:

```
switch#   configure   terminal
switch(config)# router ospf 201
switch(config-router)# graceful-restart

switch(config-router)#   graceful-restart   grace-period 120
switch(config-router)# copy running-config startup-config
```

6.8.13 Restarting an OSPFv2 Instance

You can restart an OSPv2 instance. This action clears all neighbors for the instance.

To restart an OSPFv2 instance and remove all associated neighbors, use the following command:

Command	Purpose
restart ospf *instance-tag*	Restarts the OSPFv2 instance and removes all neighbors.

6.8.14 Configuring OSPFv2 with Virtualization

You can configure multiple OSPFv2 instances in each VDC. You can also create multiple VRFs within each VDC and use the same or multiple OSPFv2 instances in each VRF. You assign an OSPFv2 interface to a VRF.

Before you begin

Create the VDCs.
Ensure that you have enabled the OSPF feature.
Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*
3. switch(config)# **router ospf** *instance-tag*
4. switch(config-router)# **vrf** *vrf-name*
5. (Optional) switch(config-router-vrf)# **maximum-paths** *path*
6. switch(config-router-vrf)# **interface** *interface-type slot/port*
7. switch(config-if)# **vrf member** *vrf-name*
8. switch(config-if)# **ip address** *ip-prefix/length*
9. switch(config-if)# **ip router ospf** *instance-tag* **area** *area-id*
10. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode.
Step 3	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv2 instance with the configured instance tag.
Step 4	switch(config-router)# **vrf** *vrf-name*	Enters VRF configuration mode.
Step 5	(Optional) switch(config-router-vrf)# **maximum-paths** *path*	Configures the maximum number of equal OSPFv2 paths to a destination in the route table for this VRF. Used for load balancing.
Step 6	switch(config-router-vrf)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 7	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 8	switch(config-if)# **ip address** *ip-prefix/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 9	switch(config-if)# **ip router ospf** *instance-tag* **area** *area-id*	Assigns this interface to the OSPFv2 instance and area configured.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create summary addresses between areas on an ABR:

```
switch#      configure      terminal
switch(config)# vrf context NewVRF
switch(config)#    router         ospf         201
switch(config)#   interface ethernet 1/2

switch(config-if)#  vrf  member  NewVRF
switch(config-if)#        ip        address    192.0.2.1/16
switch(config-if)#      ip   router   ospf   201   area   0
switch(config)# copy running-config startup-config
```

6.9 Verifying the OSPFv2 Configuration

To display the OSPFv2 configuration, perform one of the following tasks:

Command	Purpose
show ip ospf [*instance-tag*] [**vrf** *vrf-name*]	Displays the information about one or more OSPFv2 routing instances. The output includes the following area-level counts: • Interfaces in this area—A count of all interfaces added to this area (configured interfaces). • Active interfaces—A count of all interfaces considered to be in router link states and SPF (UP interfaces). • Passive interfaces—A count of all interfaces considered to be OSPF passive (no adjacencies will be formed). • Loopback interfaces—A count of all local loopback interfaces.
show ip ospf border-routers [**vrf** *{vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv2 link-state database summary.
show ip ospf interface *number* [**vrf** *{vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv2 interface configuration.
show ip ospf lsa-content-changed-list *neighbor-id interface-type number* [**vrf** {**vrf-name** | **all** | **default** | **management**}]	Displays the OSPFv2 LSAs that have changed.
show ip ospf neighbors [*neighbor-id*] [**detail**] [*interface-type number*] [**vrf** {*vrf-name* | **all** | **default** | **management**}] [**summary**]	Displays the list of OSPFv2 neighbors.
show ip ospf request-list *neighbor-id interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the list of OSPFv2 link-state requests.
show ip ospf retransmission-list *neighbor-id interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the list of OSPFv2 link-state retransmissions.
show ip ospf route [*ospf-route*] [**summary**] [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the internal OSPFv2 routes.

show ip ospf summary-address [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays information about the OSPFv2 summary addresses.
show ip ospf virtual-links [**brief**] [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays information about OSPFv2 virtual links.
show ip ospf vrf {*vrf-name* | **all** | **default** | **management**}	Displays information about VRF-based OSPFv2 configuration.
show running-configuration ospf	Displays the current running OSPFv2 configuration.

6.10 Monitoring OSPFv2

To display OSPFv2 statistics, use the following commands:

Command	**Purpose**
show ip ospf policy statistics area *area-id* **filter-list** {**in** | **out**} [**vrf** *vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv2 route policy statistics for an area.
show ip ospf policy statistics redistribute {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv2 route policy statistics.
show ip ospf statistics [**vrf***number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv2 event counters.
show ip ospf traffic *interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv2 packet counters.

6.11 Configuration Examples for OSPFv2

```
feature  ospf  router ospf 201

router-id   290.0.2.1 interface    ethernet 1/2

ip  router  ospf  201 area 0.0.0.10 ip ospf authentication
ip ospf authentication-key 0 mypass
```

6.12 Feature History for OSPFv2

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 11: Feature History for OSPFv2

Feature Name	**Release**	**Feature Information**

OSPF Packet-size	8.4(1)	Added support for configuring OSPF packet-size.
OSPF—Distribute List to Filter Paths	8.4(1)	Added support for filtering next-hop paths for an OSPF route to prevent the path from being added to the RIB.
Administrative distance of routes	8.4(1)	Added the **filter** keyword to the **table-map** command to specify that only routes permitted by the route map are downloaded to the RIB.
Route summarization	8.4(1)	Added the ability to prevent discard routes from being created
OSPFv2	8.4(1)	Added support for the optional name lookup parameter for OSFPv2 instances.
OSPFv2	8.4(1)	Added support for more than four process instances for OSPFv2 per VDC.
OSPFv2	8.4(1)	Added support for configuring the administrative distance of routes for OSPFv2.
Passive interface	8.4(1)	Added support for setting the passive interface mode on all interfaces in the router or VRF.
OSPFv2	8.4(1)	Added options for the **max-metric router-lsa** command.
BFD	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide* for more information.
OSPFv2	8.4(1)	This feature was introduced.

CHAPTER 7 Configuring OSPFv3

This chapter contains the following sections:

7.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

7.2 Information About OSPFv3

OSPFv3 is an IETF link-state protocol. An OSPFv3 router sends a special message, called a hello packet, out each OSPF-enabled interface to discover other OSPFv3 neighbor routers. Once a neighbor is discovered, the two routers compare information in the Hello packet to determine if the routers have compatible configurations. The neighbor routers attempt to establish adjacency, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv3 routing information. Adjacent routers share link-state advertisements (LSAs) that include information about the operational state of each link, the cost of the link, and any other neighbor information. The routers then flood these received LSAs out every OSPF-enabled interface so that all OSPFv3 routers eventually have identical link-state databases. When all OSPFv3 routers have identical link-state databases, the network is converged. Each router then uses Dijkstra′s Shortest Path First (SPF) algorithm to build its route table.

You can divide OSPFv3 networks into areas. Routers send most LSAs only within one area, which reduces the CPU and memory requirements for an OSPF-enabled router.

OSPFv3 supports IPv6.

7.2.1 Comparison of OSPFv3 and OSPFv2

Much of the OSPFv3 protocol is the same as in OSPFv2. OSPFv3 is described in RFC 2740. The key differences between the OSPFv3 and OSPFv2 protocols are as follows:

OSPFv3 expands on OSPFv2 to provide support for IPv6 routing prefixes and the larger size of IPv6 addresses.

・LSAs in OSPFv3 are expressed as prefix and prefix length instead of address and mask.
・The router ID and area ID are 32-bit numbers with no relationship to IPv6 addresses.
・OSPFv3 uses link-local IPv6 addresses for neighbor discovery and other features.

- OSPFv3 can use the IPv6 authentication trailer (RFC 6506) or IPSec (RFC 4552) for authentication. However, neither of these options is supported on Inspur INOS.
- OSPFv3 redefines LSA types.

7.2.2 Hello Packet

OSPFv3 routers periodically send Hello packets on every OSPF-enabled interface. The hello interval determines how frequently the router sends these Hello packets and is configured per interface. OSPFv3 uses Hello packets for the following tasks:

- Neighbor discovery
- Keepalives
- Bidirectional communications
- Designated router election

The Hello packet contains information about the originating OSPFv3 interface and router, including the assigned OSPFv3 cost of the link, the hello interval, and optional capabilities of the originating router. An OSPFv3 interface that receives these Hello packets determines if the settings are compatible with the receiving interface settings. Compatible interfaces are considered neighbors and are added to the neighbor table.

Hello packets also include a list of router IDs for the routers that the originating interface has communicated with. If the receiving interface sees its own router ID in this list, then bidirectional communication has been established between the two interfaces.

OSPFv3 uses Hello packets as a keepalive message to determine if a neighbor is still communicating. If a router does not receive a Hello packet by the configured dead interval (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

7.2.3 Neighbors

An OSPFv3 interface must have a compatible configuration with a remote interface before the two can be considered neighbors. The two OSPFv3 interfaces must match the following criteria:

- Hello interval
- Dead interval
- Area ID
- Optional capabilities

If there is a match, the information is entered into the neighbor table:

- If there is a match, the information is entered into the neighbor table:
- Priority—Priority of the neighbor router. The priority is used for designated router election.
- State—Indication of whether the neighbor has just been heard from, is in the process of setting up bidirectional communications, is sharing the link-state information, or has achieved full adjacency.
- Dead time—Indication of how long since the last Hello packet was received from this neighbor.
- Link-local IPv6 Address—The link-local IPv6 address of the neighbor.
- Designated Router—Indication of whether the neighbor has been declared the designated router or backup designated router.
- Local interface—The local interface that received the Hello packet for this neighbor.

When the first Hello packet is received from a new neighbor, the neighbor is entered into the neighbor table in the initialization state. Once bidirectional communication is established, the neighbor state becomes two-way. ExStart and exchange states come next, as the two interfaces exchange their link-state database. Once this is all complete, the neighbor moves into the full state, which signifies full adjacency. If the neighbor fails to send any Hello packets in the dead interval, then the neighbor is moved to the down state and is no longer considered adjacent.

7.2.4 Adjacency

Not all neighbors establish adjacency. Depending on the network type and designated router establishment, some neighbors become fully adjacent and share LSAs with all their neighbors, while other neighbors do not.

Adjacency is established using Database Description packets, Link State Request packets, and Link State Update packets in OSPFv3. The Database Description packet includes the LSA headers from the link-state database of the neighbor. The local router compares these headers with its own link-state database and determines which LSAs are new or updated. The local router sends a Link State Request packet for each LSA that it needs new or updated information on. The neighbor responds with a Link State Update packet. This exchange continues until both routers have the same link-state information.

7.2.5 Designated Routers

Networks with multiple routers present a unique situation for OSPFv3. If every router floods the network with LSAs, the same link-state information is sent from multiple sources. Depending on the type of network, OSPFv3 might use a single router, the designated router (DR), to control the LSA floods and represent the network to the rest of the OSPFv3 area. If the DR fails, OSPFv3 uses the BDR.

Network types are as follows:

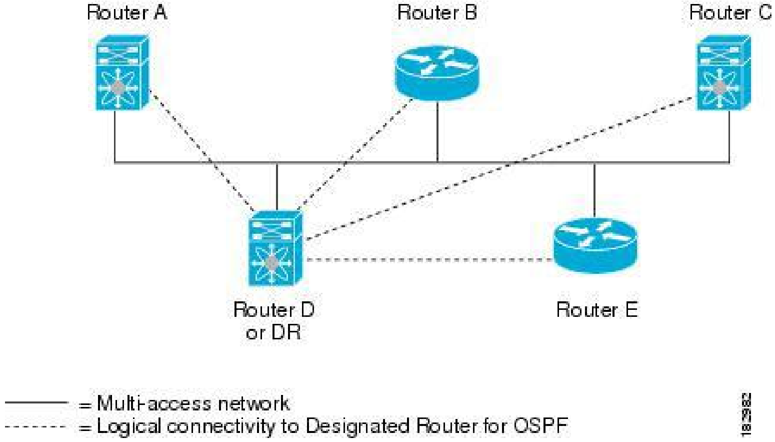
- Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv3 routers establish a DR and BDR that controls LSA flooding on the network. OSPFv3 uses the well-known IPv6 multicast addresses, FF02::5, and a MAC address of 0100.5300.0005 to communicate with neighbors.

The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR fields and the priority field in the Hello packet. As a final determinant, OSPFv3 chooses the highest router IDs as the DR and BDR.

All other routers establish adjacency with the DR and the BDR and use the IPv6 multicast address FF02::6 to send LSA updates to the DR and BDR. The Figure shows this adjacency relationship between all routers and the DR.

DRs are based on a router interface. A router might be the DR for one network and not for another network on a different interface.

Figure 27 : DR in Multi-Access Network


Router A
Router B
Router C
Router D
or DR
Router E
——— = Multi-access network
------- = Logical connectivity to Designated Router for OSPF
182982

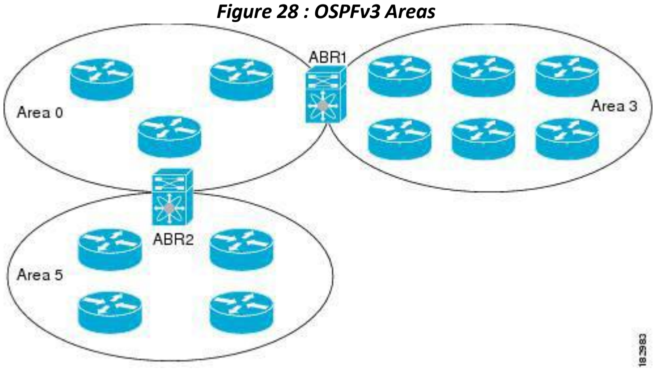
7.2.6 Areas

You can limit the CPU and memory requirements that OSPFv3 puts on the routers by dividing an OSPFv3 network into areas. An area is a logical division of routers and links within an OSPFv3 domain that creates separate subdomains. LSA flooding is contained within an area, and the link-state database is limited to links within the area. You can assign an area ID to the interfaces within the defined area. The Area ID is a 32-bit value that can be expressed as a number or in dotted decimal notation, such as 10.2.3.1.

Inspur INOS always displays the area in dotted decimal notation.

If you define more than one area in an OSPFv3 network, you must also define the backbone area, which has the reserved area ID of 0. If you have more than one area, then one or more routers become area border routers (ABRs). An ABR connects to both the backbone area and at least one other defined area.

Figure 28 : OSPFv3 Areas



The ABR has a separate link-state database for each area which it connects to. The ABR sends Inter-Area Prefix (type 3) LSAs from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In the figure, Area 0 sends summarized information about Area 5 to Area 3.

OSPFv3 defines one other router type: the autonomous system boundary router (ASBR). This router connects an OSPFv3 area to another autonomous system. An autonomous system is a network controlled by a single technical administration entity. OSPFv3 can redistribute its routing information into another autonomous system or receive redistributed routes from another autonomous system.

7.2.7 Link-State Advertisement Types

OSPFv3 uses link-state advertisements (LSAs) to build its routing table.

	Names	Description
1	Router LSA	LSA sent by everyrouter. This LSA includes the state and cost of all links but does not include prefix information. Router LSAs trigger an SPF recalculation. Router LSAs are flooded to the local OSPFv3 area.

2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network but does not include prefix information. Network LSAs trigger an SPF recalculation.
3	Inter-Area Prefix LSA	LSA sent by the area border router to an external area for each destination in local area. This LSA includes the link cost from the border router to the local destination.
4	Inter-Area Router LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only.
5	AS External LSA	LSA generated by the ASBR. This LSA includes the link cost to an external autonomous system destination. AS External LSAs are flooded throughout the autonomous system.
7	Type-7 LSA	LSA generated by the ASBR within an NSSA. This LSA includes the link cost to an external autonomous system destination. Type-7 LSAs are flooded only within the local NSSA.
8	Link LSA	LSA sent by every router, using a link-local flooding scope. This LSA includes the link-local address and IPv6 prefixes for this link.

9	Intra-Area Prefix LSA	LSA sent by every router. This LSA includes any prefix or link state changes. Intra-Area Prefix LSAs are flooded to the local OSPFv3 area. This LSA does not trigger an SPF recalculation.
10	Grace LSAs	LSA sent by a restarting router, using a link-local flooding scope. This LSA is used for a graceful restart of OSPFv3.

Link Cost

Each OSPFv3 interface is assigned a link cost. The cost is an arbitrary number. By default, Inspur INOS assigns a cost that is the configured reference bandwidth divided by the interface bandwidth. By default, the reference bandwidth is 40 Gb/s. The link cost is carried in the LSA updates for each link.

Flooding and LSA Group Pacing

OSPFv3 floods LSA updates to different sections of the network, depending on the LSA type. OSPFv3 uses the following flooding scopes:

- Link-local—LSA is flooded only on the local link. Used for Link LSAs and Grace LSAs.
- Area-local—LSA is flooded throughout a single OSPF area only. Used for Router LSAs, Network LSAs, Inter-Area-Prefix LSAs, Inter-Area-Router LSAs, and Intra-Area-Prefix LSAs.
- AS scope—LSA is flooded throughout the routing domain. An AS scope is used for AS External LSAs.

LSA flooding guarantees that all routers in the network have identical routing information. LSA flooding depends on the OSPFv3 area configuration. The LSAs are flooded based on the link-state refresh time (every 30 minutes by default). Each LSA has its own link-state refresh time.

You can control the flooding rate of LSA updates in your network by using the LSA group pacing feature. LSA group pacing can reduce high CPU or buffer utilization. This feature groups LSAs with similar link-state refresh times to allow OSPFv3 to pack multiple LSAs into an OSPFv3 Update message.

By default, LSAs with link-state refresh times within 10 seconds of each other are grouped together. You should lower this value for large link-state databases or raise it for smaller databases to optimize the OSPFv3 load on your network.

Link-State Database

Each router maintains a link-state database for the OSPFv3 network. This database contains all the collected LSAs and includes information on all the routes through the network. OSPFv3 uses this information to calculate the bast path to each destination and populates the routing table with these best paths.

LSAs are removed from the link-state database if no LSA update has been received within a set interval, called the MaxAge. Routers flood a repeat of the LSA every 30 minutes to prevent accurate link-state information from being aged out. Inspur INOS supports the LSA grouping feature to prevent all LSAs from refreshing at the same time.

7.2.8 Multi-Area Adjacency

OSPFv3 multi-area adjacency allows you to configure a link on the primary interface that is in more than one area. This link becomes the preferred intra-area link in those areas. Multi-area adjacency establishes a

point-to-point unnumbered link in an OSPFv3 area that provides a topological path for that area. The primary adjacency uses the link to advertise an unnumbered point-to-point link in the Router LSA for the corresponding area when the neighbor state is full.

The multi-area interface exists as a logical construct over an existing primary interface for OSPF; however, the neighbor state on the primary interface is independent of the multi-area interface. The multi-area interface establishes a neighbor relationship with the corresponding multi-area interface on the neighboring router.

7.2.9 OSPFv3 and the IPv6 Unicast RIB

OSPFv3 runs the Dijkstra shortest path first algorithm on the link-state database. This algorithm selects the best path to each destination based on the sum of all the link costs for each link in the path. The shortest path for each destination is then put in the OSPFv3 route table. When the OSPFv3 network is converged, this route table feeds into the IPv6 unicast RIB. OSPFv3 communicates with the IPv6 unicast RIB to do the following:

- Add or remove routes
- Handle route redistribution from other protocols
- Provide convergence updates to remove stale OSPFv3 routes and for stub router advertisements.

OSPFv3 also runs a modified Dijkstra algorithm for fast recalculation for Inter-Area Prefix, Inter-Area Router, AS-External, type-7, and Intra-Area Prefix (type 3, 4, 5, 7, 8) LSA changes.

7.2.10 Address Family Support

Inspur INOS supports multiple address families, such as unicast IPv6 and multicast IPv6. OSPFv3 features that are specific to an address family are as follows:

- Default routes
- Route summarization
- Route redistribution
- Filter lists for border routers
- SPF optimization

Use the **address-family ipv6 unicast** command to enter the IPv6 unicast address family configuration mode when configuring these features.

7.2.11 Authentication

You can configure authentication on OSPFv3 messages to prevent unauthorized or invalid routing updates in the network. OSPFv3 uses the Inspur INOS IPSecV6 secure sockets API to add authentication and encryption to its packets. It uses IPSec in transport mode with manually configured security association (SA) shared by all OSPFv3 routers in a link.

Inspur INOS OSPFv3 uses IPSec AH header with MD5 or SHA1 authentication. You can configure IPSec with a security policy, which is a combination of the security policy index (SPI) and a key.

OSPFv3 authentication can be configured at the following levels:

- Router / Process
- Area
- Interface

If you configure IPSec for an OSPFv3 area, the authentication is applied to all the interfaces in that area, except for the interfaces that have IPSec configured directly. If you configure IPSec for an OSPFv3 process, the authentication is applied on each interface in every area of that process. A security policy applied on an interface overrides the policy applied at the process or the area level.

7.3 Advanced Features

Inspur INOS supports advanced OSPFv3 features that enhance the usability and scalability of OSPFv3 in the network.

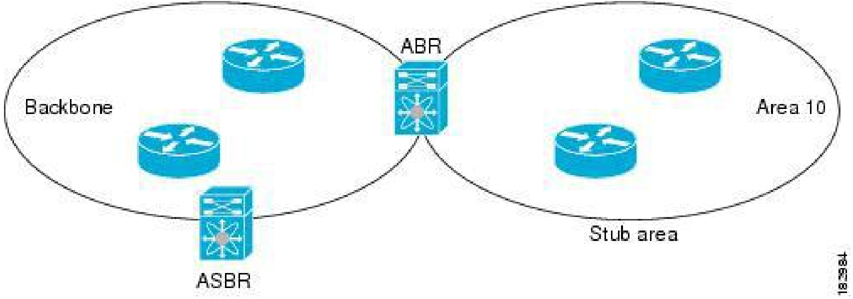
7.3.1 Stub Area

You can limit the amount of external routing information that floods an area by making it a stub area. A stub area is an area that does not allow AS External (type 5) LSAs. These LSAs are usually flooded throughout the local autonomous system to propagate external route information. Stub areas have the following requirements:

- All routers in the stub area are stub routers.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

The figure shows an example an OSPFv3 autonomous system where all routers in area 0.0.0.10 have to go through the ABR to reach external autonomous systems. Area 0.0.0.10 can be configured as a stub area.

Figure 29 : Stub Area



Stub areas use a default route for all traffic that needs to go through the backbone area to the external autonomous system. The default route is an Inter-Area-Prefix LSA with the prefix length set to 0 for IPv6.

7.3.2 Not-So-Stubby Area

A Not-So-Stubby Area (NSSA) is similar to the stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates type-7 LSAs that it floods throughout the NSSA. You can optionally configure the ABR that connects the NSSA to other areas to translate this type-7 LSA to AS External (type 5) LSAs. The ABR then floods these AS External LSAs throughout the OSPFv3 autonomous system. Summarization and filtering are supported during the translation.

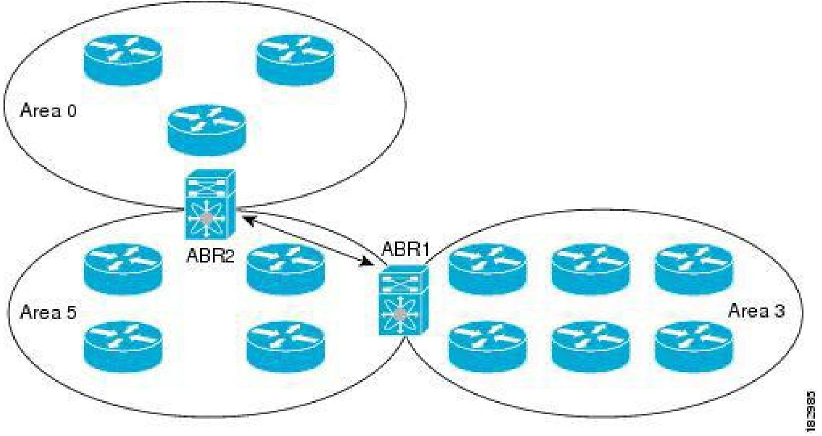
You can, for example, use NSSA to simplify administration if you are connecting a central site using OSPFv3 to a remote site that is using a different routing protocol. Before NSSA, the connection between the corporate site border router and a remote router could not be run as an OSPFv3 stub area because routes for the remote site could not be redistributed into a stub area. With NSSA, you can extend OSPFv3 to cover the remote connection by defining the area between the corporate router and remote router as an NSSA.

The backbone Area 0 cannot be an NSSA

7.3.3 Virtual Links

Virtual links allow you to connect an OSPFv3 area ABR to a backbone area ABR when a direct physical connection is not available. The figure shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 30 : Virtual Links



You can also use virtual links to temporarily recover from a partitioned area, which occurs when a link within the area fails, isolating part of the area from reaching the designated ABR to the backbone area.

7.3.4 Route Redistribution

OSPFv3 can learn routes from other routing protocols by using route redistribution. You configure OSPFv3 to assign a link cost for these redistributed routes or a default link cost for all redistributed routes.

Route redistribution uses route maps to control which external routes are redistributed. You must configure a route map with the redistribution to control which routes are passed into OSPFv3. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on. You can use route maps to modify parameters in the AS External (type 5) and NSSA External (type 7) LSAs before these external routes are advertised in the local OSPFv3 autonomous system.

OSPFv3 sets the type-5 LSA's forwarding address as described below:

- If the next-hop for the route is an attached-route then the forwarding address is the next-hop address for that route.
- If the next-hop for the route is a recursive route and next-hop's next-hop is an attached route then the forwarding address is the next-hop's next-hop address.

7.3.5 Route Summarization

Because OSPFv3 shares all learned routes with every OSPF-enabled router, you might want to use route summarization to reduce the number of unique routes that are flooded to every OSPF-enabled router. Route summarization simplifies route tables by replacing more-specific addresses with an address that represents all the specific addresses. For example, you can replace 2010:11:22:0:1000::1 and 2010:11:22:0:2000:679:1 with one summary address, 2010:11:22::/32.

Typically, you would summarize at the boundaries of area border routers (ABRs). Although you could configure summarization between any two areas, it is better to summarize in the direction of the backbone so that the backbone receives all the aggregate addresses and injects them, already summarized, into other areas. The two types of summarization are as follows:

- Inter-area route summarization
- External route summarization

You configure inter-area route summarization on ABRs, summarizing routes between areas in the autonomous system. To take advantage of summarization, assign network numbers in areas in a contiguous way to be able to lump these addresses into one range.

External route summarization is specific to external routes that are injected into OSPFv3 using route redistribution. You should make sure that external ranges that are being summarized are contiguous. Summarizing overlapping ranges from two different routers could cause packets to be sent to the wrong destination. Configure external route summarization on ASBRs that are redistributing routes into OSPF.

When you configure a summary address, Inspur INOS automatically configures a discard route for the summary address to prevent routing black holes and route loops.

7.3.6 High Availability and Graceful Restart

Inspur INOS provides a multilevel high-availability architecture. OSPFv3 supports stateful restart, which is also referred to as non-stop routing (NSR). If OSPFv3 experiences problems, it attempts to restart from its previous run-time state. The neighbors do not register any neighbor event in this case. If the first restart is not successful and another problem occurs, OSPFv3 attempts a graceful restart.

A graceful restart, or non-stop forwarding (NSF), allows OSPFv3 to remain in the data forwarding path through a process restart. When OSPFv3 needs to perform a graceful restart, it sends a link-local Grace (type 11) LSA. This restarting OSPFv3 platform is called NSF capable.

The Grace LSA includes a grace period, which is a specified time that the neighbor OSPFv3 interfaces hold onto the LSAs from the restarting OSPFv3 interface. (Typically, OSPFv3 tears down the adjacency and discards all LSAs from a down or restarting OSPFv3 interface.) The participating neighbors, which are called NSF helpers, keep all LSAs that originate from the restarting OSPFv3 interface as if the interface was still adjacent.

When the restarting OSPFv3 interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its LSA updates again. At this point, the NSF helpers recognize that the graceful restart has finished.

Stateful restart is used in the following scenarios:

- First recovery attempt after the process experiences problems
- ISSU
- User-initiated switchover using the **system switchover** command

Graceful restart is used in the following scenarios:

- Second recovery attempt after the process experiences problems within a 4-minute interval
- Manual restart of the process using the **restart ospfv3** command
- Active supervisor removal
- Active supervisor reload using the **reload module** active-supcommand

7.3.7 Multiple OSPFv3 Instances

Inspur INOS supports multiple instances of the OSPFv3 protocol. By default, every instance uses the same system router ID. You must manually configure the router ID for each instance if the instances are in the same OSPFv3 autonomous system.

The OSPFv3 header includes an instance ID field to identify that OSPFv3 packet for a particular OSPFv3 instance. You can assign the OSPFv3 instance. The interface drops all OSPFv3 packets that do not have a matching OSPFv3 instance ID in the packet header.

Inspur INOS allows only one OSPFv3 instance on an interface.

7.3.8 SPF Optimization

Inspur INOS optimizes the SPF algorithm in the following ways:

- Partial SPF for Network (type 2) LSAs, Inter-Area Prefix (type 3) LSAs, and AS External (type 5) LSAs—When there is a change on any of these LSAs, Inspur INOS performs a faster partial calculation rather than running the whole SPF calculation.
- SPF timers—You can configure different timers for controlling SPF calculations. These timers include exponential backoff for subsequent SPF calculations. The exponential backoff limits the CPU load of multiple SPF calculations.

7.3.9 Virtualization Support

OSPFv3 supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. Each OSPFv3 instance can support multiple VRFs, up to the system limit. For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

7.4 Licensing Requirements for OSPFv3

OSPFv3 requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *License and Copyright Information for Inspur INOS Software.*

7.5 Prerequisites for OSPFv3

OSPFv3 has the following prerequisites:

- You must be familiar with routing fundamentals to configure OSPFv3.
- You must be logged on to the switch.
- You have configured at least one interface for IPv6 that is capable of communicating with a remote OSPFv3 neighbor.
- You have installed the Enterprise Services license.
- You have completed the OSPFv3 network strategy and planning for your network. For example, you must decide whether multiple areas are required.
- You have enabled OSPF.
- You have installed the Advanced Services license and entered the desired VDC (*see the Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*) if you are configuring VDCs.
- You are familiar with IPv6 addressing and basic configuration.

7.6 Guidelines and Limitations for OSPFv3

OSPFv3 has the following configuration guidelines and limitations:

- You can have up to four instances of OSPFv3 in a VDC.
- Before Inspur INOS Release 8.4(1), Bidirectional Forwarding Detection (BFD) was not supported for OSPFv3. In Inspur INOS Release 8.4(1)and later releases, BFD includes a client for OSPFv3.
- Inspur INOS displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.

- MTU configured at interface level works in either the data plane or in the control plane but not at both planes at the same time.

When you configure MTU with a size lower than the supported size in data and control planes a few features that have minimum MTU requirements may not work in both the planes.

For example, MPLS VPN is supported in the data plane since this plane supports the MTU of 1500 bytes that the MPLS VPN requires. But control plane does not support MPLS VPN because this plane cannot handle the 1500-byte packets.

To make the configured MTU work in control plane for MPLS VPN, you need to manually configure the OSPF packet size (by using the **packet-size***size* command) so that OSPF works on the control plane. This is applicable from Inspur INOS Release 8.4(1) nwards.

The **packet-size***size* command is supported on the Ethernet, SVI, and GRE tunnel interfaces.

- If you configure OSPFv3 in a virtual port channel (vPC) environment, use the following timer commands in router configuration mode on the core switch to ensure fast OSPF convergence when a vPC peer link is shut down:

```
switch (config-router)# timers throttle spf 1 50 50

switch (config-router)# timers lsa-arrival 10
```

- The value of object OSPFv3 router ID differs from RFC 5643 for traps ospfv3 Nbr Restart Helper Status Change and ospfv3VirtNbrRestartHelperStatusChange. As per the RFC 5643, the value of object OSPFv3 router ID should be the router ID of the originator of the trap. But the current implementation will provide the router ID of the neighbor for both ospfv3NbrRestartHelperStatusChange and ospfv3VirtNbrRestartHelperStatusChange.
- Only the first four OSPFv3 instances are supported with MPLS LDP and MPLS TE.
- In scaled scenarios, when the number of interfaces and link-state advertisements in an OSPFv3 process is large, the snmp-walk on OSPF MIB objects is expected to time out with a small-value timeout at the SNMP agent. If you observe a timeout on the querying SNMP agent while polling OSPF MIB objects, increase the timeout value on the polling SNMP agent.
- If there is a particular OSPFv3 prefix that is learnt through type-5 as well as type-7, and both have different forwarding addresses, then these two route types are not comparable as per RFC3101, Section 2.5, step 6(e). (This applies only if the same destination/cost/non-zero forwarding addresses are there). OSPF will therefore do ECMP with all available next-hops.
- INOS OSPF and U6RIB store only one route-type per route. If there is a mix of route-type across next-hops, only one of them, (the new path type) will be shown for all next hops.

Currently, route-type is a route property, and not a next-hop property.

- The following guidelines and limitations apply to the administrative distance feature, which is supported beginning with Inspur INOS Release 8.4(1):
  - When an OSPF route has two or more equal cost paths, configuring the administrative distance is non-deterministic for the **match ip route-source** command.
  - For matching route sources in OSPFv3 routes, you must configure **match ip route-source** instead of **match ipv6 route-source** because the route sources and router IDs for OSPFv3 are IPv4 addresses.
  - Configuring the administrative distance is supported only for the **match route-type**, **match ipv6 address prefix-list**, and **match ip route-source prefix-list** commands. The other match statements are ignored.
  - The discard route is always assigned an administrative distance of 220. No configuration in the table map applies to OSPF discard routes.
  - There is no preference among the **match route-type**, **match ipv6 address**, and **match ip route-source** commands for setting the administrative distance of OSPF routes. In this way, the behavior of the table map for setting the administrative distance in Inspur INOS OSPF is different from that in Inspur IOS OSPF.

In Inspur INOS Release 8.4(1) and later releases, you can filter next-hop paths for an OSPF route to prevent the path from being added to the RIB. Before Inspur INOS Release 8.4(1), filtering on a specific path was ignored and the entire route was not added to the RIB.

• If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

7.7 Default Settings for OSPFv3

Table 12 : Default OSPFv3 Parameters

Parameters	Default
Administrative distance	110
Hello interval	10 seconds
Dead interval	40 seconds
Discard routes	Enabled
Graceful restart grace period	60 seconds
Graceful restart notify period	15 seconds
OSPFv3 feature	Disabled
Stub router advertisement announce time	600 seconds
Reference bandwidth for link cost calculation	40 Gb/s
LSA minimal arrival time	1000 milliseconds
LSA group pacing	10 seconds
SPF calculation initial delay time	0 milliseconds
SPF calculation hold time	5000 milliseconds
SPF calculation initial delay time	0 milliseconds

7.8 Configuring Basic OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

7.8.1 Enabling OSPFv3

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **feature ospfv3**

3. (Optional) switch(config)# **show feature**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature ospfv3**	Enables OSPFv3. To disable the OSPFv3 feature and remove all associated configurations, use the **no** form of the command.
Step 3	(Optional) switch(config)# **show feature**	Displays enabled and disabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

7.8.2 Creating an OSPFv3 Instance

The first step in configuring OSPFv3 is to create an instance or OSPFv3 instance. You assign a unique instance tag for this OSPFv3 instance. The instance tag can be any string. For each OSPFv3 instance, you can also configure the following optional parameters:

- Router ID—Configures the router ID for this OSPFv3 instance. If you do not use this parameter, the router ID selection algorithm is used.
- Administrative distance—Rates the trustworthiness of a routing information source.
- Log adjacency changes—Creates a system message whenever an OSPFv3 neighbor changes its state.
- Name lookup—Translates OSPF router IDs to host names, either by looking up the local hosts database or querying DNS names in IPv6.
- Maximum paths—Sets the maximum number of equal paths that OSPFv3 installs in the route table for a particular destination. Use this parameter for load balancing between multiple paths.
- Reference bandwidth—Controls the calculated OSPFv3 cost metric for a network. The calculated cost is the reference bandwidth divided by the interface bandwidth. You can override the calculated cost by assigning a link cost when a network is added to the OSPFv3 instance.

Before you begin

You must enable OSPFv3.

Ensure that the OSPFv3 instance tag that you plan on using is not already in use on this router. Use the **show ospfv3** *instance-tag* command to verify that the instance tag is not in use.

OSPFv3 must be able to obtain a router identifier (for example, a configured loopback address) or you must configure the router ID option.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **router ospfv3** *instance-tag*
3. (Optional) switch(config-router)# **router-id** *ip-address*
4. (Optional) switch(config-router)# **show ipv6 ospfv3** *instance-tag*

5. (Optional) switch(config-router)# **log-adjacency-changes** [**detail**]
6. (Optional) switch(config-router)# **passive-interface default**
7. (Optional) switch(config-router-af)# **distance** *numbers*
8. switch(config-router-af)# **maximum-paths** *paths*
9. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag. **Note** The **no router ospfv3** *instance tag* command does not remove OSPF configuration in interface mode. You must manually remove any OSPFv3 commands configured in interface mode.
Step 3	(Optional) switch(config-router)# **router-id** *ip-address*	Configures the OSPFv3 router ID. This ID uses the dotted decimal notation and identifies this OSPFv3 instance and must exist on a configured interface in the system. This command restarts the OSPF process automatically and changes the router id after it is configured.
Step 4	(Optional) switch(config-router)# **show ipv6 ospfv3** *instance-tag*	Displays OSPFv3 information.
Step 5	(Optional) switch(config-router)# **log-adjacency-changes** [**detail**]	Generates a system message whenever a neighbor changes state.
Step 6	(Optional) switch(config-router)# **passive-interfacedefault**	Suppresses routing updates on all interfaces. This command is overridden by the VRF or interface command mode configuration.
Step 7	(Optional) switch(config-router-af)# **distance** *numbers*	Configures the administrative distance for this OSPFv3 instance. The range is from 1 to 255. The default is 110.
Step 8	switch(config-router-af)# **maximum-paths** *paths*	Configures the maximum number of equal OSPFv3 paths to a destination in the route table. The range is from 1 to16. The default is 8. This command is used for load balancing.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an OSPFv3 instance:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)#      copy      running-config
startup-config
```

7.8.3 Configuring OSPFv3 Packet Size

MTU configured at interface level works in either the data plane or in the control plane but not at both planes at the same time.

When you configure MTU with a size lower than the supported size in data and control planes a few features that have minimum MTU requirements may not work in both the planes.

For example, MPLS VPN is supported in the data plane since this plane supports the MTU of 1500 bytes that the MPLS VPN requires. But control plane does not support MPLS VPN because this plane cannot handle the 1500-byte packets.

To make the configured MTU work in control plane for MPLS VPN, you need to manually configure the OSPF packet size so that OSPF works on the control plane. This is applicable from Inspur INOS Release 8.4(1)onwards.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **router ospfv3** *instance-tag*
3. switch(config-router)# **router-id** *ip-address*
4. switch(config-router)# **ospfv3 packet-size** *size*
5. (Optional) switch(config-router)# **show ospfv3 interface**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag. **Note** The **no router ospfv3** *instance-tag* command does not remove OSPF configuration in interface mode. You must manually remove any OSPFv3 commands configured in interface mode.
Step 3	switch(config-router)# **router-id** *ip-address*	Configures the OSPFv3 router ID. This ID uses the dotted decimal notation and identifies this OSPFv3 instance and must exist on a configured interface in the system. This command restarts the OSPF process automatically and changes the router id after it is configured.

Step 4	switch(config-router)# **ospfv3 packet-size** *size*	• Configures the OSPFv3 packet size. The size range is from 1280 to 9212 bytes. • You can configure the packet-size in the interface configuration mode also. • You can configure the **packet-size** *size* command even if the**ip ospf mtu-ignore** command is already configured in the network.
Step 5	(Optional) switch(config-router)# **show ospfv3 interface**	Displays OSPF information.

Example
This example shows how to configure the OSPFv3 packet-size:

```
router ospf 1

  router-id 3.3.3.3

  [no] packet-size 2000
```

This example shows the display of the configured OSPFv3 packet-size:

```
Switch (config-router)# show ospfv3 interface ethernet 1/25

Ethernet1/25  is  up,  line
    protocol   is   up   IP
    address 1.0.0.1/24

    ---------    snip ---------------

    Number  of  opaque  link  LSAs:  0,
    checksum  sum  0  Max  Packet  Size:
    2000
```

7.8.4 Configuring Networks in OSPFv3

You can configure a network to OSPFv3 by associating it through the interface that the router uses to connect to that network. You can add all networks to the default backbone area (Area 0), or you can create new areas using any decimal number or an IP address.

Before you begin
You must enable OSPFv3.
Ensure that you are in the correct VDC (or use the switchto vdc command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **ipv6 address** *ipv6-prefix/length*
4. switch(config-if)# **ipv6 router ospfv3** *instance-tag* **area** *area-id* [**secondaries none**]
5. (Optional) switch(config-if)# **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*
6. (Optional) switch(config-if)# **ospfv3 cost** *number*
7. (Optional) switch(config-if)# **ospfv3 dead-interval** *seconds*

8. (Optional) switch(config-if)# **ospfv3 hello-interval** *seconds*
9. (Optional) switch(config-if)# **ospfv3 instance** *instance*
10. (Optional) switch(config-if)# **ospfv3 mtu-ignore**
11. (Optional) switch(config-if)# **ospfv3 network** {**broadcast** | **point-point**}
12. (Optional) switch(config-if)# **ospfv3 priority** *number*
13. (Optional) switch(config-if)# **ospfv3 shutdown**
14. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ipv6 address** *ipv6-prefix/length*	Assigns an IPv6 address to this interface.
Step 4	switch(config-if)# **ipv6 router ospfv3** *instance-tag* **area** *area-id* [**secondaries none**]	Adds the interface to the OSPFv3 instance and area.
Step 5	(Optional) switch(config-if)# **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*	Displays OSPFv3 information.
Step 6	(Optional) switch(config-if)# **ospfv3 cost** *number*	Configures the OSPFv3 cost metric for this interface. The default is to calculate a cost metric, based on the reference bandwidth and interface bandwidth. The range is from 1 to 65535.
Step 7	(Optional) switch(config-if)# **ospfv3 dead-interval** *seconds*	Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
Step 8	(Optional) switch(config-if)# **ospfv3 hello-interval** *seconds*	Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
Step 9	(Optional) switch(config-if)# **ospfv3 instance** *instance*	Configures the OSPFv3 instance ID. The range is from 0 to 255. The default is 0. The instance ID is link-local in scope.
Step 10	(Optional) switch(config-if)# **ospfv3 mtu-ignore**	Configures OSPFv3 to ignore any IP maximum transmission unit (MTU) mismatch with a neighbor. The default is to not establish adjacency if the neighbor MTU does not match the local interface MTU.
Step 11	(Optional) switch(config-if)# **ospfv3 network** {**broadcast** | **point-point**}	Sets the OSPFv3 network type.
Step 12	(Optional) switch(config-if)# **ospfv3 priority** *number*	Configures the OSPFv3 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1.
Step 13	(Optional) switch(config-if)# **ospfv3 shutdown**	Shuts down the OSPFv3 instance on this interface.

Step 14	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add a network area 0.0.0.10 in OSPFv3 instance 201:

```
switch# configure terminal

switch(config)#       interface       ethernet      1/2
switch(config-if)#    ipv6    address   2001:0DB8::1/48
switch(config-if)# ipv6 ospfv3 201 area 0.0.0.10

switch(config-if)# copy running-config startup-config
```

7.9 Configuring Advanced OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

7.9.1 Configuring Filter Lists for Border Routers

You can separate your OSPFv3 domain into a series of areas that contain related networks. All areas must connect to the backbone area through an area border router (ABR). OSPFv3 domains can connect to external domains as well through an autonomous system border router (ASBR).

ABRs have the following optional configuration parameters:

- Area range—Configures route summarization between areas.
- Filter list—Filters the Inter-Area Prefix (type 3) LSAs on an ABR that are allowed in from an external area.

ASBRs also support filter lists.

Before you begin

Create the route map that the filter list uses to filter IP prefixes in incoming or outgoing Inter-Area Prefix (type 3) LSAs.

You must enable OSPFv3.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **address-family ipv6 unicast**
4. switch(config-router-af)# **area** *area-id* **filter-list route-map** *map-name* {**in** |**out**}
5. (Optional) switch(config-if)# **show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** |**out**}
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag

Step 3	switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 4	switch(config-router-af)# **area** *area-id* **filter-listroute-map** *map-name* {**in** | **out**}	Filters incoming or outgoing Inter-Area Prefix (type 3) LSAs on an ABR.
Step 5	(Optional) switch(config-if)# **show ipv6 ospfv3 policy statistics area** *id* **filter-list** {**in** | **out**}	Displays OSPFv3 policy information.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable graceful restart if it has been disabled:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# address-family ipv6 unicast

switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in

switch(config-router-af)# copy running-config startup-config
```

7.9.2 Configuring Stub Areas

You can configure a stub area for part of an OSPFv3 domain where external traffic is not necessary. Stub areas block AS External (type 5) LSAs, limiting unnecessary routing to and from selected networks. You can optionally block all summary routes from going into the stub area.

Before you begin

You must enable OSPF.

Ensure that there are no virtual links or ASBRs in the proposed stub area. Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **area** *area-id* **stub**
4. (Optional) switch(config-router)# **address-family ipv6 unicast**
5. (Optional) switch(config-router-af)# **area** *area-id* **default cost** *cost*
6. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.

Step 3	switch(config-router)# **area** *area-id* **stub**	Creates this area as a stub area.
Step 4	(Optional) switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 5	(Optional) switch(config-router-af)# **area** *area-id* **default cost** *cost*	Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This shows how to create a stub area that blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# area 0.0.0.10 stub no-summary

switch(config-router)# copy running-config startup-config
```

7.9.3 Configuring a Totally Stubby Area

You can create a totally stubby area and prevent all summary route updates from going into the stub area. To create a totally stubby area, use the following command in router configuration mode:

SUMMARY STEPS

1. switch(config-router)# **area** *area-id* **stub no-summary**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **area** *area-id* **stub no-summary**	Creates this area as a totally stubby area.

7.9.4 Configuring NSSA

You can configure an NSSA for part of an OSPFv3 domain where limited external traffic is required. You can optionally translate this external traffic to an AS External (type 5) LSA and flood the OSPFv3 domain with this routing information. An NSSA can be configured with the following optional parameters:

- No redistribution—Redistributes routes that bypass the NSSA to other areas in the OSPFv3 autonomous system. Use this option when the NSSA ASBR is also an ABR.
- Default information originate—Generates a Type-7 LSA for a default route to the external autonomous system. Use this option on an NSSA ASBR if the ASBR contains the default route in the routing table. This option can be used on an NSSA ABR whether or not the ABR contains the default route in the routing table.
- Route map—Filters the external routes so that only those routes you want are flooded throughout the NSSA and other areas.
- Translate—Translates Type-7 LSAs to AS External (type 5) LSAs for areas outside the NSSA. Use this command on an NSSA ABR to flood the redistributed routes throughout the OSPFv3 autonomous system. You can optionally suppress the forwarding address in these AS External LSAs.

- No summary—Blocks all summary routes from flooding the NSSA. Use this option on the NSSA ABR.

Before you begin

You must enable OSPF.

Ensure that there are no virtual links in the proposed NSSA and that it is not the backbone area. Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**route-map**
4. *map-name*] [**no-summary**] [**translate type7** {**always** | **never**} [**suppress-fa**]]
5. (Optional) switch(config-router)# **address-family ipv6 unicast**(Optional)
6. switch(config-router-af)# **area** *area-id* **default cost** *cost*
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **area** *area-id* **nssa** [**no-redistribution**] [**default-information-originate**] [**route-map** *map-name*] [**no-summary**] [**translate type7** {**always** | **never**} [**suppress-fa**]]	Creates this area as an NSSA.
Step 4	(Optional) switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 5	(Optional) switch(config-router-af)# **area** *area-id* **default cost** *cost*	Sets the cost metric for the default summary route sent into this NSSA. The range is from 0 to 16777215.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# area 0.0.0.10 nssa no-summary

switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that generates a default route:

```
switch# configure terminal
```

```
switch(config)# router ospfv3 201
switch(config-router)#  area  0.0.0.10  nssa  default-info-originate
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that filters external routes and blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa route-map ExternalFilter no-summary
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that always translates Type-7 LSAs to AS External (type 5) LSAs:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)# area 0.0.0.10 nssa translate type 7 always
switch(config-router)# copy running-config startup-config
```

This example shows how to create an NSSA that blocks all summary route updates:

```
switch# configure terminal
switch(config)# router ospfv3 201
switch(config-router)#  area  0.0.0.10  nssa  no-summary
switch(config-router)# copy running-config startup-config
```

7.9.5 Configuring Multi-Area Adjacency

You can add more than one area to an existing OSPFv3 interface. The additional logical interfaces support multi-area adjacency.

Before you begin

You must enable OSPF.

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Ensure that you have configured a primary area for the interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **ipv6 router ospfv3** *instance-tag* **multi-area** *area-id*
4. (Optional) switch(config-if)# **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ipv6 router ospfv3** *instance-tag* **multi-area** *area-id*	Adds the interface to another area.
Step 4	(Optional) switch(config-if)# **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*	Displays OSPFv3 information.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add a second area to an OSPFv3 interface:

```
switch# configure terminal

switch(config)#  interface  ethernet  1/2
switch(config-if)#        ipv6        address    2001:0DB8::1/48
switch(config-if)#  ipv6 ospfv3 201 area 0.0.0.10

switch(config-if)# ipv6 ospfv3 201 multi-area 20

switch(config-if)# copy running-config startup-config
```

7.9.6 Configuring Virtual Links

A virtual link connects an isolated area to the backbone area through an intermediate area. You can configure the following optional parameters for a virtual link:

- Dead interval—Sets the time that a neighbor waits for a Hello packet before declaring the local router as dead and tearing down adjacencies.
- Hello interval—Sets the time between successive Hello packets.
- Retransmit interval—Sets the estimated time between successive LSAs.
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

Before you begin

You must enable OSPF.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **area** *area-id* **virtual-link** *router-id*
4. (Optional) switch(config-if)# **show ipv6 ospfv3 virtual-link** [**brief**]
5. (Optional) switch(config-router-vlink)# **dead-interval** *seconds*
6. (Optional) switch(config-router-vlink)# **hello-interval** *seconds*
7. (Optional) switch(config-router-vlink)# **retransmit-interval** *seconds*
8. (Optional) switch(config-router-vlink)# **transmit-delay** *seconds*
9. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **area** *area-id* **virtual-link***router-id*	Creates one end of a virtual link to a remote router. You must create the virtual link on that remote router to complete the link.
Step 4	(Optional) switch(config-if)# **show ipv6 ospfv3 virtual-link** [**brief**]	Displays OSPFv3 virtual link information.
Step 5	(Optional) switch(config-router-vlink)# **dead-interval** *seconds*	Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
Step 6	(Optional) switch(config-router-vlink)# **hello-interval** *seconds*	Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
Step 7	(Optional) switch(config-router-vlink)# **retransmit-interval** *seconds*	Configures the OSPFv3 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
Step 8	(Optional) switch(config-router-vlink)# **transmit-delay** *seconds*	Configures the OSPFv3 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

These examples show how to create a simple virtual link between two ABRs: Configuration for ABR 1 (router ID 2001:0DB8::1) is as follows:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::10

switch(config-router)# copy running-config startup-config
```

Configuration for ABR 2 (router ID 2001:0DB8::10) is as follows:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1

switch(config-router)# copy running-config startup-config
```

7.9.7 Configuring Redistribution

You can redistribute routes learned from other routing protocols into an OSPFv3 autonomous system through the ASBR.

You can configure the following optional parameters for route redistribution in OSPF:

- Default information originate—Generates an AS External (type 5) LSA for a default route to the external autonomous system.
- Default metric—Sets all redistributed routes to the same cost metric.

Before you begin

Create the necessary route maps used for redistribution. You must enable OSPF.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **address-family ipv6 unicast**
4. switch(config-router-af)# **redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map***map-name*
5. switch(config-router-af)# **default-information originate** [**always**] [**route-map** *map-name*]
6. switch(config-router-af)# **default-metric** *cost*
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 4	switch(config-router-af)# **redistribute** {**bgp***id*| **direct**|**isis** *id* | **rip** *id* | **static**} **route-map** *map-name*	Redistributes the selected protocol into OSPFv3 through the configured route map. **Note** If you redistribute static routes, Inspur INOS also redistributes the default static route.
Step 5	switch(config-router-af)# **default-information originate** [**always**] [**route-map** *map-name*]	Creates a default route into this OSPFv3 domain if the default route exists in the RIB. Use the following optional keywords: • **always** —Always generates the default route of 0.0.0. even if the route does not exist in the RIB. • **route-map**—Generates the default route if the route map returns true. **Note** This command ignores **match** statements in the route map.

Step 6	switch(config-router-af)# **default-metric** *cost*	Sets the cost metric for the redistributed routes. The range is from 1 to 16777214. This command does not apply to directly connected routes. Use a route map to set the default metric for directly connected routes.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to redistribute the Border Gateway Protocol (BGP) into OSPFv3:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# address-family ipv6 unicast

switch(config-router-af)# redistribute bgp route-map FilterExternalBGP

switch(config-router-af)# copy running-config startup-config
```

7.9.8 Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the OSPFv3 route table. You can configure a maximum limit to the number of routes accepted from external protocols. OSPFv3 provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when OSPFv3 reaches the configured maximum. OSPFv3 does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where OSPFv3 logs a warning when that threshold is passed.
- Warning only—Logs a warning only when OSPFv3 reaches the maximum. OSPFv3 continues to accept redistributed routes.
- Withdraw—Starts the configured timeout period when OSPFv3 reaches the maximum. After the timeout period, OSPFv3 requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, OSPFv3 withdraws all redistributed routes. You must clear this condition before OSPFv3 accepts more redistributed routes. You can optionally configure the timeout period.

Before you begin

You must enable OSPF.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **address-family ipv6 unicast**
4. switch(config-router)# **redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name*
5. switch(config-router)# **redistribute maximum-prefix***max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timemout*]]
6. (Optional) **show running-config ospfv3**
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 4	switch(config-router)# **redistribute** {**bgp***id* | **direct** | **isis** *id* | **rip** *id* | **static**} **route-map** *map-name*	Redistributes the selected protocol into OSPFv3 through the configured route map.
Step 5	switch(config-router)# **redistribute maximum-prefix***max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timemout*]]	Specifies a maximum number of prefixes that OSPFv2 distributes. The range is from 0 to 65536. Optionally, specifies the following: • *threshold*—Percent of maximum prefixes that triggers a warning message. • **warning-only**—Logs an warning message when the maximum number of prefixes is exceeded. • **withdraw**—Withdraws all redistributed routes and optionally tries to retrieve the redistributed routes. The num-retries range is from 1 to 12. The timeout range is from 60 to 600 seconds. The default is 300 seconds.
Step 6	(Optional) **show running-config ospfv3** **Example:** `switch(config-router)# show running-config ospf`	Displays the OSPFv3 configuration.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to limit the number of redistributed routes into OSPF:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# address-family ipv6 unicast

switch(config-router-af)# redistribute bgp route-map filterExternalBGP
switch(config-router-af)# copy running-config startup-config
```

7.9.9 Configuring Route Summarization

You can configure route summarization for inter-area routes by configuring an address range that is summarized. You can also configure route summarization for external, redistributed routes by configuring a summary address for those routes on an ASBR.

Before you begin

You must enable OSPF.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **address-family ipv6 unicast**
4. switch(config-router-af)# **area** *area-id* **range** *ipv6-prefix/length* [**no-advertise**] [**cost***cost*]
5. switch(config-router-af)# **summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*]
6. (Optional) switch(config-router)# **show ipv6 ospfv3 summary-address**
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.
Step 4	switch(config-router-af)# **area** *area-id* **range** *ipv6-prefix/length* [**no-advertise**] [**cost** *cost*]	Creates a summary address on an ABR for a range of addresses and optionally advertises this summary address in a Inter-Area Prefix (type 3) LSA. The cost range is from 0 to 16777215.
Step 5	switch(config-router-af)# **summary-address** *ipv6-prefix/length* [**no-advertise**] [**tag** *tag*]	Creates a summary address on an ASBR for a range of addresses and optionally assigns a tag for this summary address that can be used for redistribution with route maps.
Step 6	(Optional) switch(config-router)# **show ipv6 ospfv3 summary-address**	Displays information about OSPFv3 summary addresses.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This shows how to create a stub area that blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)#     address-family     ipv6    unicast
switch(config-router)#  area  0.0.0.10  range   2001:0DB8::/48
switch(config-router)# copy running-config startup-config
```

This example shows how to create summary addresses on an ASBR:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)#      address-family      ipv6    unicast
switch(config-router)#        summary-address     2001:0DB8::/48
switch(config-router)# copy running-config startup-config
```

7.9.10 Configuring the Administrative Distance of Routes

Beginning with Inspur INOS Release 8.4(1), you can set the administrative distance of routes added by OSPFv3 into the RIB.

The administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. Typically, a route can be learned through more than one routing protocol. The administrative distance is used to discriminate between routes learned from more than one routing protocol. The route with the lowest administrative distance is installed in the IP routing table.

Before you begin

Ensure that you have enabled OSPFv3.

Ensure that you are in the correct VDC (or use the **switchto vdc** command). See the guidelines and limitations for this feature.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospf** *instance-tag*
3. switch(config-router)# **address-family ipv6 unicast**
4. switch(config-router-af)# [**no**] **table-map** *map-name* [**filter**]
5. switch(config-router-af)# **exit**
6. switch(config-router)# **exit**
7. switch(config)# **route-map** *map-name* [**permit** | **deny**] [*seq*]
8. switch(config-route-map)# **match route-type** *route-type*
9. switch(config-route-map)# **match ip route-source prefix-list** *name*
10. switch(config-route-map)# **match ipv6 address prefix-list** *name*
11. switch(config-route-map)# **set distance** *value*
12. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospf** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.

Step 4	switch(config-router-af)# [**no**] **table-map** *map-name* [**filter**]	Configures the policy for filtering or modifying OSPFv2 routes before sending them to the RIB. You can enter up to 63 alphanumeric characters for the map name. The **filter** keyword specifies that only routes that are permitted by the route map(*map-name*) configuration are downloaded to the routing information base (RIB).
Step 5	switch(config-router-af)# **exit**	Exits router address-family configuration mode.
Step 6	switch(config-router)# **exit**	Exits router configuration mode.
Step 7	switch(config)# **route-map** *map-name* [**permit** | **deny**] [*seq*]	Creates a route map or enters route-map configuration mode for an existing route map. Use *seq* to order the entries in a route map. **Note** The **permit** option enables you to set the distance. If you use the **deny** option, the default distance is applied.
Step 8	switch(config-route-map)# **match route-type** *route-type*	Matches against one of the following route types: • external—The external route (BGP, EIGRP, and OSPF type 1 or 2) • inter-area—OSPF inter-area route • internal—The internal route (including the OSPF intra- or inter-area) • intra-area—OSPF intra-area route • nssa-external—The NSSA external route (OSPF type 1 or 2) • type-1—The OSPF external type 1 route • type-2—The OSPF external type 2 route
Step 9	switch(config-route-map)# **match ip route-source prefix-list** *name*	Matches the IPv6 route source address or router ID of a route to one or more IP prefix lists. Use the **ip prefix-list** command to create the prefix list. **Note** For OSPFv3, the router ID is 4 bytes.
Step 10	switch(config-route-map)# **match ipv6 addressprefix-list** *name*	Matches against one or more IPv6 prefix lists. Use the **ip prefix-list** command to create the prefix list.
Step 11	switch(config-route-map)# **set distance** *value*	Sets the administrative distance of routes for OSPFv3. The range is from 1 to 255.
Step 12	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure the OSPFv3 administrative distance for inter-area routes to 150, for external routes to 200, and for all prefixes in prefix list p1 to 190:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)#   address-family ipv6  unicast

switch(config-router-af)# table-map   foo

switch(config-router)# exit

switch(config)# exit

switch(config)#  route-map  foo  permit  10
switch(config-route-map)#   match   route-type  inter-area
switch(config-route-map)# set  distance  150
switch(config)#  route-map  foo  permit  20
switch(config-route-map)#     match     route-type     external
switch(config-route-map)#   set   distance 200
switch(config)#  route-map  foo  permit 30
switch(config-route-map)#   match   ip   route-source prefix-list   p1
switch(config-route-map)#    match   ipv6   address   prefix-list   p1
switch(config-route-map)# set distance 190
```

The following example shows how to configure a route map for blocking the next hops that are learned through VLAN 10:

```
switch(config)#  route-map  Filter-OSPF 10   deny
switch(config-route-map)# match      interface      VLAN      10
switch(config-route-map)# exit

switch(config)# route-map Filter-OSPF 20 permit
```

The following example shows how to configure the **table-map** command with the **filter** keyword to use a route map (Filter-OSPF) to remove the next-hop path that is learned through VLAN 10 but not the next-hop path that is learned through VLAN 20:

```
switch(config)# route ospfv3 p1

switch(config-router)# table-map Filter-OSPF filter
```

7.9.11 Modifying the Default Timers

OSPFv3 includes a number of timers that control the behavior of protocol messages and shortest path first (SPF) calculations. OSPFv3 includes the following optional timer parameters:

- LSA arrival time—Sets the minimum interval allowed between LSAs arriving from a neighbor. LSAs that arrive faster than this time are dropped.
- Pacing LSAs—Sets the interval at which LSAs are collected into a group and refreshed, checksummed, or aged. This timer controls how frequently LSA updates occur and optimizes how many are sent in an LSA update message.

- Throttle LSAs—Sets rate limits for generating LSAs. This timer controls how frequently LSAs are generated after a topology change occurs.
- Throttle SPF calculation—Controls how frequently the SPF calculation is run.

At the interface level, you can also control the following timers:

- Retransmit interval—Sets the estimated time between successive LSAs
- Transmit delay—Sets the estimated time to transmit an LSA to a neighbor.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **timers lsa-arrival** *msec*
4. switch(config-router)# **timers lsa-group-pacing** *seconds*
5. switch(config-router)# **timers throttle lsa** *start-time hold-interval max-time*
6. switch(config-router)# **address-family ipv6 unicast**
7. switch(config-router)# **timers throttle spf** *delay-time hold-time*
8. switch(config)# **interface** *type slot/port*
9. switch(config-if)# **ospfv3 retransmit-interval** *seconds*
10. switch(config-if)# **ospfv3 transmit-delay** *seconds*
11. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	switch(config-router)# **timers lsa-arrival** *msec*	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	switch(config-router)# **timers lsa-group-pacing** *seconds*	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 10 seconds.
Step 5	switch(config-router)# **timers throttle lsa** *start-time hold-interval max-time*	Sets the rate limit in milliseconds for generating LSAs. You can configure the following timers: • *start-time*—The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. • *hold-interval*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. • *max-time*—The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	switch(config-router)# **address-family ipv6 unicast**	Enters IPv6 unicast address family mode.

Step 7	switch(config-router)# **timers throttle spf** *delay-time hold-time*	Sets the SPF best path schedule initial delay time and the minimum hold time in seconds between SPF best path calculations. The range is from 1 to 600000. The default is no delay time and 5000 millisecond hold time.
Step 8	switch(config)# **interface** *type slot/port*	Enters interface configuration mode.
Step 9	switch(config-if)# **ospfv3 retransmit-interval** *seconds*	Sets the estimated time in seconds between LSAs transmitted from this interface. The range is from 1 to 65535. The default is 5.
Step 10	switch(config-if)# **ospfv3 transmit-delay** *seconds*	Sets the estimated time in seconds to transmit an LSA to a neighbor. The range is from 1 to 450. The default is 1.
Step 11	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This shows how to create a stub area that blocks all summary route updates:

```
switch# configure terminal

switch(config)# router ospfv3 201

switch(config-router)# timers lsa-group-pacing 300

switch(config-router)# copy running-config startup-config
```

7.9.12 Configuring the OSPFv3 Max-Metric Router LSA

You can configure OSPFv3 to advertise its locally generated router LSAs with the maximum metric value possible (the infinity metric 0xFFF). This feature allows OSPFv3 processes to converge but not attract transit traffic through the device if there are better alternate paths. After a specified timeout or a notification from BGP, OSPFv3 advertises the LSAs with normal metrics.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**stub-prefix-lsa**] [**on-startup** [*seconds*]] [**wait-for bgp** *tag*] [**inter-area-prefix-lsa** [*max-metric-value*]]
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.

Step 3	switch(config-router)# **max-metric router-lsa** [**external-lsa** [*max-metric-value*]] [**stub-prefix-lsa**] [**on-startup** [*seconds*]] [**wait-for bgp** *tag*] [**inter-area-prefix-lsa** [*max-metric-value*]]	Configures a device that is running the OSPFv3 protocol to advertise a maximum metric so that other devices do not prefer the device as an intermediate hop in their SPF calculations.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a router to advertise a maximum metric for the stub links:

```
switch(config)# router ospfv3 200

switch(config-router)# max-metric router-lsa stub-prefix-lsa
```

7.9.13 Configuring Graceful Restart

Graceful restart is enabled by default. You can configure the following optional parameters for graceful restart in an OSPFv3 instance:

- Grace period—Configures how long neighbors should wait after a graceful restart has started before tearing down adjacencies.
- Helper mode disabled—Disables helper mode on the local OSPFv3 instance. OSPFv3 does not participate in the graceful restart of a neighbor.
- Planned graceful restart only—Configures OSPFv3 to support graceful restart only in the event of a planned restart.

Before you begin

You must enable OSPF.

Ensure that all neighbors are configured for graceful restart with matching optional parameters set. Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router ospfv3** *instance-tag*
3. switch(config-router)# **graceful restart**
4. switch(config-router)# **graceful-restart grace-period** *seconds*
5. switch(config-router)# **graceful-restart helper-disable**
6. switch(config-router)# **graceful-restart planned-only**
7. (Optional) switch(config-if)# **show ipv6 ospfv3** *instance-tag*
8. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.

Step 3	switch(config-router)# **graceful restart**	Enables graceful restart. A graceful restart is enabled by default.
Step 4	switch(config-router)# **graceful-restart grace-period** *seconds*	Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	switch(config-router)# **graceful-restart helper-disable**	Disables helper mode. Enabled by default.
Step 6	switch(config-router)# **graceful-restart planned-only**	Configures graceful restart for planned restarts only.
Step 7	(Optional) switch(config-if)# **show ipv6 ospfv3** *instance-tag*	Displays OSPFv3 information.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example
This shows how to create a stub area that blocks all summary route updates:

```
switch#  configure  terminal
switch(config)#         router    ospfv3     201
switch(config-router)# graceful-restart
switch(config-router)# graceful-restart grace-period 120
switch(config-router)# copy running-config startup-config
```

7.9.14 Restarting an OSPFv3 Instance

You can restart an OSPv3 instance. This action clears all neighbors for the instance.
To restart an OSPFv3 instance and remove all associated neighbors, use the following command:

SUMMARY STEPS

1. switch(config)# **restart ospfv3** *instance-tag*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **restart ospfv3** *instance-tag*	Restarts the OSPFv3 instance and removes all neighbors.

7.9.15 Configuring OSPFv3 with Virtualization

You can configure multiple OSPFv3 instances in each VDC. You can also create multiple VRFs within each VDC and use the same or multiple OSPFv3 instances in each VRF. You assign an OSPFv3 interface to a VRF.

Before you begin
Create the VDCs.
You must enable OSPF.
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*
3. switch(config)# **router ospfv3** *instance-tag*
4. switch(config-router)# **vrf** *vrf-name*
5. (Optional) switch(config-router-vrf)# **maximum-paths** *paths*
6. switch(config)# **interface** *type slot/port*
7. switch(config-if)# **vrf member** *vrf-name*
8. switch(config-if)# **ipv6 address** *ipv6-prefix/length*
9. switch(config-if)# **ipv6 ospfv3** *instance-tag* **area** *area-id*
10. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode.
Step 3	switch(config)# **router ospfv3** *instance-tag*	Creates a new OSPFv3 instance with the configured instance tag.
Step 4	switch(config-router)# **vrf** *vrf-name*	Enters VRF configuration mode.
Step 5	(Optional) switch(config-router-vrf)# **maximum-paths** *paths*	Configures the maximum number of equal OSPFv3 paths to a destination in the route table for this VRF. Use this command for load balancing.
Step 6	switch(config)# **interface** *type slot/port*	Enters interface configuration mode.
Step 7	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 8	switch(config-if)# **ipv6 address** *ipv6-prefix/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 9	switch(config-if)# **ipv6 ospfv3** *instance-tag* **area** *area-id*	Assigns this interface to the OSPFv3 instance and area configured.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal
switch(config)#vrf context NewVRF
switch(config-vrf)# exit
  switch(config)# router ospfv3 201
  switch(config-router)# exit
  switch(config)#interface ethernet 1/2
  switch(config-if)# vrf member NewVRF
  switch(config-if)#      ipv6      address 2001:0DB8::1/48
  switch(config-if)#  ipv6 ospfv3 201 area 0
  switch(config-if)# copy running-config startup-config
```

7.9.16 Configuring OSPFv3 Authentication at Router Level

You can enable authentication of OSPFv3 packets on a per-interface basis at the Router level using the following commands.

Before you begin

Ensure you have enabled OSPF.

Ensure that you are in the correct VDC(or use the **switchto vdc** command) Enable the authentication package.

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Enable the authentication package:
switch(config)# **feature imp**

Step 3 Create a new OSPFv3 instance with the configured instance tag:
switch(config)# **router ospfv3** *instance-tag*

Step 4 Enable IPSec AH Authentication:
switch(config-router)# **authentication ipsec spi** *spi auth* **[0 | 3 | 7]** *key*

You can specify the security policy index through spi and define the authentication algorithm through auth which can be md5 or sha1. Numbers 0, 3 and 7 specify the format of key.

Step 5 (Optional) Display OSPFv3 information:
switch(config)# **show running-config ospfv3**

7.9.17 Configuring OSPFv3 Authentication at Area Level

Authentication of OSPFv3 packets is enabled on a per-interface basis at the Area level using the following commands.

Before you begin

Ensure you have enabled OSPF.

Ensure that you are in the correct VDC(or use the **switchto vdc** command) Enable the authentication package.

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Enable the authentication package:
switch(config)# **feature imp**

Step 3 Create a new OSPFv3 instance with the configured instance tag:
switch(config)#**router ospfv3** *instance-tag*

Step 4 Enable IPSec AH Authentication:
switch(config-router)#**area** *area-num* **authentication ipsec spi** *spi auth* **[0 | 3 | 7]** *key*

You can specify the security policy index through spi and define the authentication algorithm through auth which can be md5 or sha1. Numbers 0, 3 and 7 specify the format of *key*.

Step 5 (Optional) Display OSPFv3 information:
switch(config)# **show running-config ospfv3**

7.9.18 Configuring OSPFv3 Authentication at Interface Level

You can configure the authentication of OSPFv3 packets per interface using the following commands.

Before you begin
Ensure you have enabled OSPF.
Ensure that you are in the correct VDC(or use the **switchto vdc** command) Enable the authentication package.

Step 1 Enter the global configuration mode:
switch# **configure terminal**

Step 2 Enables the authentication mode:
switch(config)# **feature imp**

Step 3 Enters the interface configuration mode:
switch(config)# **interface ethernet** *interface*

Step 4 Change the port mode to Layer 3 interface:
switch(config-if)# **no switchport**

Step 5 Specify the OSPFv3 instance and area for the interface:
switch(config-if)# **ipv6 router** *ospfv3* **instance-tag** *area* **area-id**

Step 6 Enable IPSec AH Authentication:
switch(config-if)# **ospfv3 authentication ipsec spi** *spi auth* **[0 | 3 | 7]** *key*

You can specify the security policy index through spi and define the authentication algorithm through auth which can be md5 or sha1. Numbers 0, 3 and 7 specify the format of key.

Step 7 (Optional) Display the running configuration on the interface:
switch(config-if)#**show run interface** *interface*

Configuration Example
The following example shows how to enable security for Ethernet interface 2/1.

```
switch# configure terminal
switch(config)#  interface ethernet 2/1
switch(config-if)# no switchport
switch(config-if)# ipv6 router ospfv3 1 area 0
switch(config-if)#    ospfv3    authentication    ipsec    spi    256    md5    0
11111111111111111111111111111111 switch(config-if)# show run interface ethernet
2/1

!Command: show running-config interface Ethernet2/1

!Time: Mon Oct 26 09:19:30 2015


version         7.2(0)D1(1)
interface Ethernet2/1

shutdown

no switchport medium p2p

  ospfv3      authentication      ipsec      spi      256      md5      3
b54dc5a961fb42098f6902e512cb6e099d44 d3239f4e48e73668de6f52254f0e

  ipv6 router ospfv3 1 area
0.0.0.0 switch(config-if)#
```

7.10 Verifying the OSPFv3 Configuration

To display the OSPFv3 configuration, perform one of the following tasks:

Command	**Purpose**
show ipv6 ospfv3 [*instance-tag*] [**vrf** *vrf-name*]	Displays the information about one or more OSPFv3 routing instances. The output includes the following area-level counts: • Interfaces in this area—A count of all interfaces added to this area (configured interfaces). • Active interfaces—A count of all interfaces considered to be in router link states and SPF (UP interfaces). • Passive interfaces—A count of all interfaces considered to be OSPF passive (no adjacencies will be formed). • Loopback interfaces—A count of all local loopback interfaces.
show ipv6 ospfv3 border-routers	Displays the internal OSPF routing table entries to an ABR and ASBR.
show ipv6 ospfv3 database	Displays lists of information related to the OSPFv3 database for a specific router.
show ipv6 ospfv3 interface *type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 interface configuration.
show ipv6 ospfv3 neighbors	Displays the neighbor information. Use the **clear ospfv3 neighbors** command to remove adjacency with all neighbors.

show ipv6 ospfv3 request-list	Displays a list of LSAs requested by a router.
show ipv6 ospfv3 retransmission-list	Displays a list of LSAs waiting to be retransmitted.
show ipv6 ospfv3 summary-address	Displays a list of all summary address redistribution information configured under an OSPFv3 instance.
show running-configuration ospfv3	Displays the current running OSPFv3 configuration.

7.11 Monitoring OSPFv3

To display OSPFv3 statistics, use the following commands:

Command	**Purpose**
show ipv6 ospfv3 memory	Displays the OSPFv3 memory usage statistics.
show ipv6 ospfv3 policy statistics area *area-id* **filter-list** {**in** | **out**} [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 route policy statistics for an area.
show ipv6 ospfv3 policy statistics redistribute { **bgp** *id* | **direct** | **isis** *id* | **rip** *id* | **static vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 route policy statistics.
show ipv6 ospfv3 statistics [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 event counters.
show ipv6 ospfv3 traffic *interface-type number* [**vrf** {*vrf-name* | **all** | **default** | **management**}]	Displays the OSPFv3 packet counters.

7.12 Configuration Examples for OSPFv3

This example shows how to configure OSPFv3:

```
This   example   shows   how   to
configure OSPFv3: feature ospfv3

router ospfv3 201

 router-id 290.0.2.1

interface ethernet 1/2

 ipv6       address
 2001:0DB8::1/48
 ipv6   ospfv3   201
 area 0.0.0.10
```

7.13 Related Documents for OSPFv3

Related Topic	**Document Title**

OSPFv3 CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs	*Inspur CN12700 Series INOS Virtual Device Context Command Reference*

7.14 FeatureHistoryforOSPFv3

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 13 : Feature History for OSPFv3

Feature Name	Release	Feature Information
OSPF—Distribute List to Filter Paths	8.4(1)	Added support for filtering next-hop paths for an OSPF route to prevent the path from being added to the RIB.
Administrative distance of routes	8.4(1)	Added the **filter** keyword to the **table-map** command to specify that only routes permitted by the route map are downloaded to the RIB.
Route summarization	8.4(1)	Added the ability to prevent discard routes from being created.
OSPFv3	8.4(1)	• Bidirectional Forwarding Detection (BFD) was enhanced to add a client for OSPFv3 • Added the ability to advertise locally generated router LSAs with the maximum metric value possible. • Added the optional **name-lookup** parameter for OSPFv3 instances.
MIBs	8.4(1)	Added OSPFv3 SNMP/trap support.
OSPFv3	8.4(1)	Added support for configuring the administrative distance of routes for OSPFv3.
Passive interface	8.4(1)	Added support for setting the passive interface mode on all interfaces in the router or VRF.
OSPFv3	8.4(1)	This feature was introduced.

CHAPTER 8 Configuring EIGRP

This chapter contains the following sections:

- Finding Feature Information.
- Information About EIGRP.
- Licensing Requirements for EIGRP.
- Prerequisites for EIGRP.
- Guidelines and Limitations for EIGRP.
- Default Settings for EIGRP Parameters.
- Configuring Basic EIGRP.
- Configuring Advanced EIGRP.
- Configuring Virtualization for EIGRP.
- Verifying the EIGRP Configuration.
- Displaying EIGRP Statistics.
- Configuration Example for EIGRP.
- Related Documents for EIGRP.
- MIBs.
- Feature History for EIGRP.

8.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

8.2 Information About EIGRP

EIGRP combines the benefits of distance vector protocols with the features of link-state protocols. EIGRP sends out periodic Hello messages for neighbor discovery. Once EIGRP learns a new neighbor, it sends a one-time update of all the local EIGRP routes and route metrics. The receiving EIGRP router calculates the route distance based on the received metrics and the locally assigned cost of the link to that neighbor. After this initial full route table update, EIGRP sends incremental updates to only those neighbors affected by the route change. This process speeds convergence and minimizes the bandwidth used by EIGRP.

8.2.1 EIGRP Components

EIGRP has the following basic components:

- Reliable Transport Protocol
- Neighbor Discovery and Recovery
- Diffusing Update Algorithm

Reliable Transport Protocol

The Reliable Transport Protocol guarantees ordered delivery of EIGRP packets to all neighbors. The Reliable Transport Protocol supports an intermixed transmission of multicast and unicast packets. The reliable transport can send multicast packets quickly when unacknowledged packets are pending. This provision helps to ensure that the convergence time remains low for various speed links.

The Reliable Transport Protocol includes the following message types:

- Hello—Used for neighbor discovery and recovery. By default, EIGRP sends a periodic multicast Hello message on the local network at the configured hello interval. By default, the hello interval is 5 seconds.
- Acknowledgement—Verify reliable reception of Updates, Queries, and Replies.

- Updates—Send to affected neighbors when routing information changes. Updates include the route destination, address mask, and route metrics such as delay and bandwidth. The update information is stored in the EIGRP topology table.
- Queries and Replies—Sent as part of the Diffusing Update Algorithm used by EIGRP.

Neighbor Discovery and Recovery

EIGRP uses the Hello messages from the Reliable Transport Protocol to discover neighboring EIGRP routers on directly attached networks. EIGRP adds neighbors to the neighbor table. The information in the neighbor table includes the neighbor address, the interface it was learned on, and the hold time, which indicates how long EIGRP should wait before declaring a neighbor unreachable. By default, the hold time is three times the hello interval or 15 seconds.

EIGRP sends a series of Update messages to new neighbors to share the local EIGRP routing information. This route information is stored in the EIGRP topology table. After this initial transmission of the full EIGRP route information, EIGRP sends Update messages only when a routing change occurs. These Update messages contain only the new or changed information and are sent only to the neighbors affected by the change.

EIGRP also uses the Hello messages as a keepalive to its neighbors. As long as Hello messages are received, Inspur INOS can determine that a neighbor is alive and functioning.

Diffusing Update Algorithm

The Diffusing Update Algorithm (DUAL) calculates the routing information based on the destination networks in the topology table. The topology table includes the following information:

- IPv4 or IPv6 address/mask—The network address and network mask for this destination.
- Successors—The IP address and local interface connection for all feasible successors or neighbors that advertise a shorter distance to the destination than the current feasible distance.
- Feasibility distance (FD)—The lowest calculated distance to the destination. The feasibility distance is the sum of the advertised distance from a neighbor plus the cost of the link to that neighbor.

DUAL uses the distance metric to select efficient, loop-free paths. DUAL selects routes to insert into the unicast Routing Information Base (RIB) based on feasible successors. When a topology change occurs, DUAL looks for feasible successors in the topology table. If there are feasible successors, DUAL selects the feasible successor with the lowest feasible distance and inserts that into the unicast RIB, avoiding unnecessary recomputation.

When there are no feasible successors but there are neighbors advertising the destination, DUAL transitions from the passive state to the active state and triggers a recomputation to determine a new successor or next-hop router to the destination. The amount of time required to recompute the route affects the convergence time. EIGRP sends Query messages to all neighbors, searching for feasible successors. Neighbors that have a feasible successor send a Reply message with that information. Neighbors that do not have feasible successors trigger a DUAL recomputation.

8.2.2 EIGRP Route Updates

When a topology change occurs, EIGRP sends an Update message with only the changed routing information to affected neighbors. This Update message includes the distance information to the new or updated network destination.

The distance information in EIGRP is represented as a composite of available route metrics, including bandwidth, delay, load utilization, and link reliability. Each metric has an associated weight that determines if the metric is included in the distance calculation. You can configure these metric weights. You can fine-tune link characteristics to achieve optimal paths, but we recommend that you use the default settings for most configurable metrics.

Internal Route Metrics

Internal routes are routes that occur between neighbors within the same EIGRP autonomous system. These routes have the following metrics:

- Next hop—The IP address of the next-hop router.
- Delay—The sum of the delays configured on the interfaces that make up the route to the destination network. The delay is configured in tens of microseconds.
- Bandwidth—The calculation from the lowest configured bandwidth on an interface that is part of the route to the destination.
- MTU—The smallest maximum transmission unit value along the route to the destination.
- Hop count—The number of hops or routers that the route passes through to the destination. This metric is not directly used in the DUAL computation.
- Reliability—An indication of the reliability of the links to the destination.
- Load—An indication of how much traffic is on the links to the destination.

By default, EIGRP uses the bandwidth and delay metrics to calculate the distance to the destination. You can modify the metric weights to include the other metrics in the calculation.

Wide Metrics

EIGRP supports wide (64-bit) metrics to improve route selection on higher-speed interfaces or bundled interfaces. Routers supporting wide metrics can interoperate with routers that do not support wide metrics as follows:

- A router that supports wide metrics—Adds local wide metrics values to the received values and sends the information on.
- A router that does not support wide metrics— Sends any received metrics on without changing the values.

EIGRP uses the following equation to calculate path cost with wide metrics:

metric = [k1 x bandwidth + (k2 x bandwidth)/(256 − load) + k3 x delay + k6 xextended attributes] x [k5/(reliability + k4)]

Because the unicast RIB cannot support 64-bit metric values, EIGRP wide metrics use the following equation with a RIB scaling factor to convert the 64-bit metric value to a 32-bit value:

RIB Metric = (Wide Metric / RIB scale value)

where the RIB scale value is a configurable parameter.

EIGRP wide metrics introduce the following two new metric values represented as k6 in the EIGRP metrics configuration:

- Jitter—(Measured in microseconds) accumulated across all links in the route path. Routes lower jitter values are preferred for EIGRP path selection.
- Energy—(Measured in watts per kilobit) accumulated across all links in the route path. Routes lower energy values are preferred for EIGRP path selection.

EIGRP prefers a path with no jitter or energy metric values or lower jitter or metric values over a path with higher values.

External Route Metrics

External routes are routes that occur between neighbors in different EIGRP autonomous systems. These routes have the following metrics:

- Next hop—The IP address of the next-hop router.
- Router ID—The router ID of the router that redistributed this route into EIGRP.
- AS number—The autonomous system number of the destination.
- Protocol ID—A code that represents the routing protocol that learned the destination route.
- Tag—An arbitrary tag that can be used for route maps.
- Metric—The route metric for this route from the external routing protocol.

EIGRP and the Unicast RIB

EIGRP adds all learned routes to the EIGRP topology table and the unicast RIB. When a topology change occurs, EIGRP uses these routes to search for a feasible successor. EIGRP also listens for notifications from the unicast RIB for changes in any routes redistributed to EIGRP from another routing protocol.

8.2.3 Advanced EIGRP

You can use the advanced features of EIGRP to optimize your EIGRP configuration.

Address Families

EIGRP supports both IPv4 and IPv6 address families. For backward compatibility, you can configure EIGRPv4 in route configuration mode or in IPv4 address family mode. You must configure EIGRP for IPv6 in address family mode.

Address family configuration mode includes the following EIGRP features:

- Authentication
- AS number
- Default route
- Metrics
- Distance
- Graceful restart
- Logging
- Load balancing
- Redistribution
- Router ID
- Stub router
- Timers

You cannot configure the same feature in more than one configuration mode. For example, if you configure the default metric in router configuration mode, you cannot configure the default metric in address family mode.

Authentication

You can configure authentication on EIGRP messages to prevent unauthorized or invalid routing updates in your network. EIGRP authentication supports MD5 authentication digest.

You can configure the EIGRP authentication per virtual routing and forwarding (VRF) instance or interface using key-chain management for the authentication keys. Key-chain management allows you to control changes to the authentication keys used by MD5 authentication digest. See the Inspur CN12700 Series INOS Security Configuration Guide, for more details about creating key chains.

For MD5 authentication, you configure a password that is shared at the local router and all remote EIGRP neighbors. When an EIGRP message is created, Inspur INOS creates an MD5 one-way message digest based on the message itself and the encrypted password and sends this digest along with the EIGRP message. The receiving EIGRP neighbor validates the digest using the same encrypted password. If the message has not changed, the calculation is identical and the EIGRP message is considered valid.

MD5 authentication also includes a sequence number with each EIGRP message that is used to ensure that no message is replayed in the network.

Stub Routers

You can use the EIGRP stub routing feature to improve network stability, reduce resource usage, and simplify stub router configuration. Stub routers connect to the EIGRP network through a remote router.

When using EIGRP stub routing, you need to configure the distribution and remote routers to use EIGRP and configure only the remote router as a stub. EIGRP stub routing does not automatically enable summarization on the distribution router. In most cases, you need to configure summarization on the distribution routers.

Without EIGRP stub routing, even after the routes that are sent from the distribution router to the remote router have been filtered or summarized, a problem might occur. For example, if a route is lost somewhere in the corporate network, EIGRP could send a query to the distribution router. The distribution router could then send a query to the remote router even if routes are summarized. If a problem communicating over the WAN link between the distribution router and the remote router occurs, EIGRP could get stuck in an active condition and cause instability elsewhere in the network. EIGRP stub routing allows you to prevent queries to the remote router.

Route Summarization

You can configure a summary aggregate address for a specified interface. Route summarization simplifies route tables by replacing a number of more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

If more specific routes are in the routing table, EIGRP advertises the summary address from the interface with a metric equal to the minimum metric of the more specific routes.

Route Redistribution

You can use EIGRP to redistribute static routes, routes learned by other EIGRP autonomous systems, or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into EIGRP. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on.

You also configure the default metric that is used for all imported routes into EIGRP.

You use distribute lists to filter routes from routing updates. These filtered routes are applied to each interface with the **ip distribute-list eigrp** command.

Load Balancing

Split Horizon

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the usage of network segments, which increases effective network bandwidth.

Inspur INOS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the EIGRP route table and the unicast RIB. You can configure EIGRP to load balance traffic across some or all of those paths.

You can use split horizon to ensure that EIGRP never advertises a route out of the interface where it was learned.

Split horizon is a method that controls the sending of EIGRP update and query packets. When you enable split horizon on an interface, Inspur INOS does not send update and query packets for destinations that were learned from this interface. Controlling update and query packets in this manner reduces the possibility of routing loops.

Split horizon with poison reverse configures EIGRP to advertise a learned route as unreachable back through that the interface that EIGRP learned the route from.

EIGRP uses split horizon or split horizon with poison reverse in the following scenarios:

- Exchanging topology tables for the first time between two routers in startup mode.
- Advertising a topology table change.
- Sending a Query message.

By default, the split horizon feature is enabled on all interfaces.

BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide*, for more information.

Virtualization Support for EIGRP

Inspur INOS supports multiple instances of EIGRP that runs on the same system. EIGRP supports Virtual Routing and Forwarding instances (VRFs). VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. See the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*, for more information.

Graceful Restart and High Availability

Inspur INOS supports nonstop forwarding and graceful restart for EIGRP.

You can use nonstop forwarding for EIGRP to forward data packets along known routes in the FIB while the EIGRP routing protocol information is being restored following a failover. With nonstop forwarding (NSF), peer networking devices do not experience routing flaps. During failover, data traffic is forwarded through intelligent modules while the standby supervisor becomes active.

If a Inspur INOS system experiences a cold reboot, the device does not forward traffic to the system and removes the system from the network topology. In this scenario, EIGRP experiences a stateless restart, and all neighbors are removed. Inspur INOS applies the startup configuration, and EIGRP rediscovers the neighbors and shares the full EIGRP routing information again.

A dual supervisor platform that runs Inspur INOS can experience a stateful supervisor switchover. Before the switchover occurs, EIGRP uses a graceful restart to announce that EIGRP will be unavailable for some time. During a switchover, EIGRP uses nonstop forwarding to continue forwarding traffic based on the information in the FIB, and the system is not taken out of the network topology.

The graceful restart-capable router uses Hello messages to notify its neighbors that a graceful restart operation has started. When a graceful restart-aware router receives a notification from a graceful restart-capable neighbor that a graceful restart operation is in progress, both routers immediately exchange their topology tables. The graceful restart-aware router performs the following actions to assist the restarting router as follows:

- The router expires the EIGRP Hello hold timer to reduce the time interval set for Hello messages. This process allows the graceful restart-aware router to reply to the restarting router more quickly and reduces the amount of time required for the restarting router to rediscover neighbors and rebuild the topology table.
- The router starts the route-hold timer. This timer sets the period of time that the graceful restart-aware router will hold known routes for the restarting neighbor. The default time period is 240 seconds.
- The router notes in the peer list that the neighbor is restarting, maintains adjacency, and holds known routes for the restarting neighbor until the neighbor signals that it is ready for the graceful restart-aware router to send its topology table or the route-hold timer expires. If the route-hold timer expires on the graceful restart-aware router, the graceful restart-aware router discards held routes and treats the restarting router as a new router that joins the network and reestablishes adjacency.

After the switchover, Inspur INOS applies the running configuration, and EIGRP informs the neighbors that it is operational again.

Multiple EIGRP Instances

Inspur INOS supports multiple instances of the EIGRP protocol that run on the same system. Every instance uses the same system router ID. You can optionally configure a unique router ID for each instance. For the number of supported EIGRP instances, see the *Inspur CN12700 Series INOS Verified Scalabilty Guide*.

8.3 Licensing Requirements for EIGRP

EIGRP requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the Inspur INOS Licensing Guide.

8.4 Prerequisites for EIGRP

You must enable EIGRP.

If you configure VDCs, you must install the Advanced Services license and enter the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*).

8.5 Guidelines and Limitations for EIGRP

• A metric configuration (either through the default-metric configuration option or through a route map) is required for redistribution from any other protocol, connected routes, or static routes.

• For graceful restart, an NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in a graceful restart operation.

• For graceful restart, neighboring devices participating in the graceful restart must be NSF-aware or NSF-capable.

• Inspur INOS EIGRP is compatible with EIGRP in the Inspur IOS software.

• Do not change the metric weights without a good reason. If you change the metric weights, you must apply the change to all EIGRP routers in the same autonomous system.

• A mix of standard metrics and wide metrics in an EIGRP network with interface speeds of 1 Gigabit or greater may result in suboptimal routing.

• Consider using stubs for larger networks.

• Avoid redistribution between different EIGRP autonomous systems because the EIGRP vector metric will not be preserved.

• The **no {ip | ipv6} next-hop-self** command does not guarantee reachability of the next hop.

• The **{ip | ipv6} passive-interface eigrp** command suppresses neighbors from forming.

• Inspur INOS does not support IGRP or connecting IGRP and EIGRP clouds.

• Autosummarization is disabled by default and cannot be enabled.

• Inspur INOS supports only IP.

• EIGRPv6 adjacency cannot be formed over an interface that only has IPv6 link local address.

• Global IPv6 address is required on the interface for EIGRPv6 neighbour adjacency to be formed over such interface.

• High availability is not supported with EIGRP aggressive timers.

• If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

8.6 Default Settings for EIGRP Parameters

Table 14 : Default Settings for EIGRP Parameters

Parameters	Default
Administrative distance	Internal routes—90 External routes—170
Bandwidth percent	50 percent

Default metric for redistributed routes	Bandwidth—100000 Kb/s Delay— 100 (10 microsecond units) Reliability—255 Loading—1 MTU—1500
EIGRP feature	Disabled
Hello interval	5 seconds
Hold time	15 seconds
Equal-cost paths	8
Metric weights	1 0 1 0 0 0
Next-hop address advertised	IP address of local interface
NSF convergence time	120
NSF route-hold time	240
NSF signal time	20
Redistribution	Disabled
Split horizon	Enabled

8.7 Configuring Basic EIGRP

8.7.1 Enabling or Disabling the EIGRP Feature

You must enable the EIGRP feature before you can configure EIGRP.

Before you begin
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature eigrp**	Enables the EIGRP feature. The **no** option disables the EIGRP feature and removes all associated configuration.
Step 3	(Optional) switch(config)# **show feature**	Displays information about enabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example enables EIGRP:

```
switch# configure terminal

switch(config)# feature eigrp

switch(config)#  copy  running-config  startup-config
```

8.7.2 Creating an EIGRP Instance

You can create an EIGRP instance and associate an interface with that instance. You assign a unique autonomous system number for this EIGRP process. Routes are not advertised or accepted from other autonomous systems unless you enable route redistribution.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- EIGRP must be able to obtain a router ID (for example, a configured loopback address) or you must configure the router ID option.
- If you configure an instance tag that does not qualify as an AS number, you must configure the AS number explicitly or this EIGRP instance remains in the shutdown state. For IPv6, this number must be configured under address family.
- Confirm that you are in the correct VDC. To change the VDC, use the switchto vdc command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. If you configure an instance tag that does not qualify as an AS number, you must use the **autonomous-system** command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state. The **no** option deletes the EIGRP process and all associated configuration. You should also remove any EIGRP commands configured in interface mode if you remove the EIGRP process.
Step 3	(Optional) switch(config-router)# **autonomous-system** *as-number*	Configures a unique AS number for this EIGRP instance. The range is from 1 to 65535.
Step 4	(Optional) switch(config-router)# **log-adjacency-changes**	Generates a system message whenever an adjacency changes state. This command is enabled by default.

Step 5	(Optional) switch(config-router)# **log-neighbor-warnings** [*seconds*]	Generates a system message whenever a neighbor warning occurs. You can configure the time between warning messages, from 1 to 65535, in seconds. The default is 10 seconds. This command is enabled by default.
Step 6	switch(config-router)# **interface** *interface-type slot*/*port*	Enters interface configuration mode. Use **?** to determine the slot and port ranges.
Step 7	switch(config-if)# {**ip** | **ipv6**} **router eigrp** *instance-tag*	Associates this interface with the configured EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 8	(Optional) switch(config-if)# **show** {**ip** | **ipv6**} **eigrp interfaces**	Displays information about EIGRP interfaces.
Step 9	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an EIGRP process and configure an interface for EIGRP:

```
switch#   configure   terminal
switch(config)#   router   eigrp   Test1
switch(config)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1


switch(config-if)# no shutdown

switch(config-if)# copy running-config startup-config
```

8.7.3 Restarting an EIGRP Instance

You can restart an EIGRP instance. This action clears all neighbors for the instance.

Procedure

	Command or Action	**Purpose**
Step 1	(Optional) switch(config)# **flush-routes**	Flushes all EIGRP routes in the unicast RIB when this EIGRP instance restarts.
Step 2	Required: switch(config)# **restart eigrp** *instance-tag*	Restarts the EIGRP instance and removes all neighbors. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to restart an EIGRP instance:

```
switch(config)# flush-routes
switch(config)# restart eigrp Test1

switch(config)#  copy  running-config  startup-config
```

8.7.4 Shutting Down an EIGRP Instance

You can gracefully shut down an EIGRP instance. This action removes all routes and adjacencies but preserves the EIGRP configuration.

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **shutdown**	Disables this instance of EIGRP. The EIGRP router configuration remains.
Step 2	(Optional) switch(config-router)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to shut down an EIGRP instance:

```
switch(config-router)# shutdown

switch(config-router)# copy running-config startup-config
```

8.7.5 Configuring a Passive Interface for EIGRP

You can configure a passive interface for EIGRP. A passive interface does not participate in EIGRP adjacency, but the network address for the interface remains in the EIGRP topology table.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# {**ip** | **ipv6**} **passive-interface eigrp** *instance-tag*	Suppresses EIGRP hellos, which prevents neighbors from forming and sending routing updates on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. **Note** To configure all EIGRP interfaces as passive by default, use the **passive-interface default** command.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure a passive interface for EIGRP:

```
switch(config-if)# ip passive-interface eigrp tag10

switch(config-if)# copy running-config startup-config
```

8.7.6 Shutting Down EIGRP on an Interface

You can gracefully shut down EIGRP on an interface. This action removes all adjacencies and stops EIGRP traffic on this interface but preserves the EIGRP configuration.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# {**ip**|**ipv6**} **eigrp** *instance-tag* **shutdown**	Disables EIGRP on this interface. The EIGRP interface configuration remains. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to shut down EIGRP on an interface:

```
switch(config-if)# ip eigrp Test1 shutdown

switch(config-if)# copy running-config startup-config
```

8.8 Configuring Advanced EIGRP

8.8.1 Configuring Authentication in EIGRP

You can configure authentication between neighbors for EIGRP. You can configure EIGRP authentication for the EIGRP process or for individual interfaces. The interface EIGRP authentication configuration overrides the EIGRP process-level authentication configuration.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- Ensure that all neighbors for an EIGRP process share the same authentication configuration, including the shared authentication key.
- Create the key chain for this authentication configuration. For more information, see the *Inspur CN12700 Series INOS Security Configuration Guide*.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. If you configure an instance tag that does not qualify as an AS number, you must use the **autonomous-system** command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters the address-family configuration mode. This command is optional for IPv4.

Step 4	switch(config-router-af)# **authentication key-chain** *key-chain*	Associates a key chain with this EIGRP process for this VRF. The key chain can be any case-sensitive, alphanumeric string up to 20 characters.
Step 5	switch(config-router-af)# **authentication mode md5**	Configures MD5 message digest authentication mode for this VRF.
Step 6	switch(config-router-af)# **interface** *interface-typeslot/port*	Enters interface configuration mode. Use **?** to find the supported interfaces.
Step 7	switch(config-if)# {**ip** | **ipv6**} **router eigrp** *instance-tag*	Associates this interface with the configured EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 8	switch(config-if)# {**ip** | **ipv6**} **authentication key-chain eigrp** *instance-tag key-chain*	Associates a key chain with this EIGRP process for this interface. This configuration overrides the authentication configuration set in the router VRF mode. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 9	switch(config-if)# {**ip** | **ipv6**} **authentication mode eigrp** *instance-tag* **md5**	Configures the MD5 message digest authentication mode for this interface. This configuration overrides the authentication configuration set in the router VRF mode. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 10	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure MD5 message digest authentication for EIGRP over Ethernet interface 1/2:

```
switch# configure terminal
switch(config)# router eigrp Test1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1
switch(config-if)# ip authentication key-chain eigrp Test1 routeKeys
switch(config-if)# ip authentication mode eigrp Test1 md5
switch(config-if)# copy running-config startup-config
```

8.8.2 Configuring EIGRP Stub Routing

To configure a router for EIGRP stub routing, use these commands in address-family configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router-af)# **stub** [**direct** | **receive-only** | **redistributed** [**direct**] **leak-map** *map-name*]	Configures a remote router as an EIGRP stub router. The map name can be any case-sensitive, alphanumeric string up to 20 characters.

Step 2	(Optional) switch(config-router-af)# **show ip eigrp neighbor detail**	Verifies that the router has been configured as a stub router.
Step 3	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure a stub router to advertise directly connected and redistributed routes. The last line of the output for the **show ip eigrp neighbor detail** command shows the stub status of the remote or spoke router.

```
switch# configure terminal

switch(config)# router eigrp Test1

switch(config-router)#    address-family    ipv4   unicast
switch(config-router-af)#     stub   direct   redistributed
switch(config-router-af)# show ip eigrp neighbor detail

IP-EIGRP neighbors for process 201

H   Address                 Interface   Hold Uptime   SRTT   RTO
                                        Q  Seq Type  (sec)    (ms)
                                                             Cnt
                                        Num

0   10.1.1.2                Se3/1         11 00:00:59    1  4500  0  7

   Version 12.1/1.2, Retrans: 2, Retries: 0

   Stub  Peer  Advertising  ( CONNECTED  SUMMARY  )

Routes  switch(config-router-af)#  copy  running-config startup-config
```

8.8.3 Configuring a Summary Address for EIGRP

You can configure a summary aggregate address for a specified interface. If any more specific routes are in the routing table, EIGRP advertises the summary address out the interface with a metric equal to the minimum of all more specific routes.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# {**ip** | **ipv6**} **summary-address eigrp** *instance-tag ip-prefix/length* [*distance* | **leak-map** *map-name*]	Configures a summary aggregate address as either an IP address and network mask or an IP prefix/length. The instance tag and map name can be any case-sensitive, alphanumeric string up to 20 characters. You can optionally configure the administrative distance for this aggregate address. The default administrative distance is 5 for aggregate addresses.

Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to cause EIGRP to summarize network 192.0.2.0 out Ethernet 1/2 only:

```
switch(config)# interface ethernet 1/2


switch(config-if)#   ip   summary-address   eigrp   Test1   192.0.2.0 255.255.255.0

switch(config-if)# copy running-config startup-config
```

8.8.4 Redistributing Routes into EIGRP

You can redistribute routes in EIGRP from other routing protocols.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- You must configure the metric (either through the default-metric configuration option or through a route map) for routes redistributed from any other protocol.
- You must create a route map to control the types of routes that are redistributed into EIGRP.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. If you configure an instance tag that does not qualify asan AS number, you must use the **autonomous-system** command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	switch(config-router-af)# **redistribute** {**bgp** *as* | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **direct** | **static**} **route-map** *map-name*	Injects routes from one routing domain into EIGRP. The instance tag and map name can be any case-sensitive, alphanumeric string up to 20 characters.

Step 5	switch(config-router-af)# **default-metric** *bandwidth delay reliability loading mtu*	Sets the metrics assigned to routes learned through route redistribution. The default values are as follows: • bandwidth—100000 Kb/s • delay—100 (10 microsecond units) • reliability—255 • loading—1 • MTU—1492
Step 6	(Optional) switch(config-router-af)# **show** {**ip** | **ipv6**} **eigrp route-map statistics redistribute**	Displays information about EIGRP route map statistics.
Step 7	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to redistribute BGP into EIGRP for IPv4:

```
switch# configure terminal

switch(config)# router eigrp Test1

switch(config-router)#   redistribute   bgp   100   route-map   BGPFilter
switch(config-router)#   default-metric   500000   30   200   1   1500
switch(config-router)#  copy running-config startup-config
```

8.8.5 Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the EIGRP route table. You can configure a maximum limit to the number of routes accepted from external protocols. EIGRP provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when EIGRP reaches the configured maximum. EIGRP does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where EIGRP logs a warning when that threshold is passed.
- Warning only—Logs a warning only when EIGRP reaches the maximum. EIGRP continues to accept redistributed routes.
- Withdraw—Starts the timeout period when EIGRP reaches the maximum. After the timeout period, EIGRP requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, EIGRP withdraws all redistributed routes. You must clear this condition before EIGRP accepts more redistributed routes. You can optionally configure the timeout period.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag.
Step 3	switch(config-router)# **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name*	Redistributes the selected protocol into EIGRP through the configured route map.
Step 4	switch(config-router)# **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]]	Specifies a maximum number of prefixes that EIGRP distributes. The range is from 0 to 65536. Optionally specifies the following: • *threshold*—Percent of maximum prefixes that triggers a warning message. • **warning-only**—Logs a warning message when the maximum number of prefixes is exceeded. • **withdraw**—Withdraws all redistributed routes. Optionally tries to retrieve the redistributed routes. The *num-retries* range is from 1 to 12. The timeout is from 60 to 600 seconds. The default is 300 seconds. Use the **clear ip eigrp redistribution** command if all routes are withdrawn.
Step 5	(Optional) switch(config-router)# **show running-config eigrp**	Displays the EIGRP configuration.
Step 6	(Optional) switch(config-router)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to limit the number of redistributed routes into EIGRP:

```
switch# configure terminal

switch(config)# router eigrp Test1

switch(config-router)#    redistribute    bgp    route-map FilterExternalBGP

switch(config-router)# redistribute maximum-prefix 1000 75
```

8.8.6 Configuring the Administrative Distance of Routes

You can set the administrative distance of routes added by EIGRP into the RIB.

Before you begin

You must enable EIGRP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router eigrp** *instance-tag*
3. switch(config-router)# **table-map** *route-map-name* [**filter**]
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP instance and enters router configuration mode.
Step 3	switch(config-router)# **table-map** *route-map-name* [**filter**]	Configures a table map with route map information. You can enter up to 63 alphanumeric characters for the map name. The **filter** keyword filters routes rejected by the route map and does not download them to the RIB.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

8.8.7 Configuring Route-Map Filtering

You can enable EIGRP to interoperate with other protocols to leverage additional routing functionality by filtering inbound and outbound traffic based on route-map options.

Before you begin

You must enable EIGRP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **route-map** *map-tag* [**permit** | **deny**] [*sequence-number*]	Enters route-map configuration mode.
Step 3	switch(config-route-map)# **match metric** *metric-value* [**+-** *deviation-number*] [... *metric-value* [**+-** *deviation-number*]]	Specifies a match clause that filters inbound updates that match an internal or external protocol metric. The *metric-value* argument is an internal protocol metric that can be an EIGRP five-part metric. The range is from 1 to 4294967295. The **+-** *deviation-number* argument represents a standard deviation, which can be any number. When you specify a metric deviation with the + and **-** keywords, the router matches any metric that falls inclusively in that range.
Step 4	switch(config-route-map)# **match source-protocol** *source-protocol* [*as-number*]	Specifies a match clause that matches external routes from sources that match the source protocol. The *source-protocol* argument is the protocol to match. The valid options are **bgp**, **connected**, **eigrp**, **isis**, **ospf**, **rip**, and **static**. The *as-number* argument does not apply to the **connected**, **rip**, and **static** options. The range is from 1 to 65535.

Step 5	switch(config-route-map)# **set tag** *tag-value*	Sets a tag value on the route in the destination routing protocol when all the match criteria of a route map are met.
Step 6	switch(config-route-map)# **exit**	Exits route-map configuration mode.
Step 7	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP instance and enters router configuration mode.
Step 8	switch(config-router)# **exit**	Exits router configuration mode.
Step 9	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode. Use ? to determine the slot and port ranges.
Step 10	switch(config-if)# **ip address** *ip-address*	Specifies an IP address for the EIGRP routing process.
Step 11	switch(config-if)# **ip router eigrp** *as-number*	Configures the EIGRP routing process and enters the router configuration mode.
Step 12	switch(config-if)# **ip distribute-list eigrp** *as-number* **route-map** *map-tag* **in**	Filters networks received in updates.
Step 13	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

8.8.8 Configuring Load Balancing in EIGRP

You can configure the number of Equal Cost Multiple Path (ECMP) routes using the **maximum-paths** option.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. If you configure an instance tag that does not qualify as an AS number, you must use the **autonomous-system** command to configure the AS number explicitly or this EIGRP instance remains in the shutdown state.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	switch(config-router-af)# **maximum-paths** *num-paths*	Sets the number of equal cost paths that EIGRP accepts in the route table. The range is from 1 to 16. The default is 8.

Step 5	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure equal cost load balancing for EIGRP over IPv4 with a maximum of six equal cost paths:

```
switch# configure terminal

switch(config)# router eigrp Test1

switch(config-router)#   address-family   ipv4 unicast

switch(config-router-af)# maximum-paths 6

switch(config-router-af)# copy running-config startup-config
```

8.8.9 Configuring Graceful Restart for EIGRP

You can configure graceful restart or nonstop forwarding for EIGRP.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- An NSF-aware router must be up and completely converged with the network before it can assist an NSF-capable router in a graceful restart operation.
- Neighboring devices participating in the graceful restart must be NSF aware or NSF capable.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. If you configure an instance tag that does not qualify as an AS number, you must use the **autonomous-system** command to configure the AS number explicitly or this EIGRP instance remains in the shutdown state.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	switch(config-router-af)# **graceful-restart**	Enables graceful restart. This feature is enabled by default.
Step 5	switch(config-router-af)# **timers nsf converge** *seconds*	Sets the time limit for the convergence after a switchover. The range is from 60 to 180 seconds. The default is 120.

Step 6	switch(config-router-af)# **timers nsf route-hold** *seconds*	Sets the hold time for routes learned from the graceful restart-aware peer. The range is from 20 to 300 seconds. The default is 240.
Step 7	switch(config-router-af)# **timers nsf signal** *seconds*	Sets the time limit for signaling a graceful restart. The range is from 10 to 30 seconds. The default is 20.
Step 8	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure graceful restart for EIGRP over IPv6 using the default timer values:

```
switch# configure terminal

switch(config)# router eigrp Test1

switch(config-router)# address-family ipv6 unicast

switch(config-router-af)# graceful-restart

switch(config-router-af)#  copy  running-config  startup-config
```

8.8.10 Adjusting the Interval Between Hello Packets and the Hold Time

You can adjust the interval between Hello messages and the hold time.

By default, Hello messages are sent every 5 seconds. The hold time is advertised in Hello messages and indicates to neighbors the length of time that they should consider the sender valid. The default hold time is three times the hello interval, or 15 seconds.

On very congested and large networks, the default hold time might not be sufficient time for all routers to receive hello packets from their neighbors. In this case, you might want to increase the hold time.

SUMMARY STEPS

1. switch(config-if)# {**ip** | **ipv6**} **hello-interval eigrp** *instance-tag seconds*
2. switch(config-if)# {**ip** | **ipv6**} **hold-time eigrp** *instance-tag seconds*
3. (Optional) switch(config-if)# **show ip eigrp interface detail**
4. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-if)# {**ip** | **ipv6**} **hello-interval eigrp** *instance-tag seconds*	Configures the hello interval for an EIGRP routing process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The range is from 1 to 65535 seconds. The default is 5.
Step 2	switch(config-if)# {**ip** | **ipv6**} **hold-time eigrp** *instance-tag seconds*	Configures the hold time for an EIGRP routing process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The range is from 1 to 65535 seconds.
Step 3	(Optional) switch(config-if)# **show ip eigrp interface detail**	Verifies the timer configuration.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to change the interval between Hello packets and the hold time:

```
switch(config)#  interface  ethernet  1/2
switch(config-if)#   ip   hello-interval  eigrp   Test1   30
switch(config-if)#    ip   hold-time    eigrp    Test1    30
switch(config-if)# show ip eigrp interface detail

switch(config-if)# copy running-config startup-config
```

8.8.11 Disabling Split Horizon

You can use split horizon to block route information from being advertised by a router out of any interface from which that information originated. Split horizon usually optimizes communications among multiple routing devices, particularly when links are broken.

By default, split horizon is enabled on all interfaces.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# **no** {**ip** | **ipv6**} **split-horizon eigrp** *instance-tag*	Disables split horizon.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to disable split horizon on a particular interface:

```
switch(config)# interface ethernet 1/2

switch(config-if)# no ip split-horizon eigrp Test1

switch(config-if)# copy running-config startup-config
```

8.8.12 Enabling Wide Metrics

You can enable wide metrics in router or address family configuration mode.

SUMMARY STEPS

1. switch(config-router)# **metrics version 64bit**
2. switch(config-router)# **metrics rib-scale** *value*
3. (Optional) switch(config-router)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **metrics version 64bit**	Enables 64-bit metric values.
Step 2	switch(config-router)# **metrics rib-scale** *value*	(Optional) Configures the scaling factor used to convert the 64-bit metric values to 32 bit in the RIB. The range is from 1 to 255. The default value is 128.

Step 3	(Optional) switch(config-router)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable wide metrics in router configuration mode:

```
switch# configure terminal

switch(config)#   router   eigrp Test1
switch(config-router)#  metrics        version        64bit
switch(config-router)#   metrics rib-scale 128

switch(config-router)#    copy    running-config startup-config
```

8.8.13 Tuning EIGRP

You can configure optional parameters to tune EIGRP for your network. Some of the parameters can be configured in address-family configuration mode, and others can be configured in interface configuration mode.

Procedure

	Command or Action	Purpose
Step 1	(Optional) switch(config-router-af)# **default-information originate** [**always** | **route-map** *map-name*]	Originates or accepts the default route with prefix 0.0.0.0/0. When a route-map is supplied, the default route is originated only when the route map yields a true condition. The route-map name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 2	(Optional) switch(config-router-af)# **distance** *internal external*	Configures the administrative distance for this EIGRP process. The range is from 1 to 255. The internal value sets the distance for routes learned from within the same autonomous system (the default value is 90). The external value sets the distance for routes learned from an external autonomous system (the default value is 170).
Step 3	(Optional) switch(config-router-af)# **metric max-hops** *hop-count*	Sets the maximum allowed hops for an advertised route. Routes over this maximum are advertised as unreachable. The range is from 1 to 255. The default is 100.

Step 4	(Optional) switch(config-router-af)# **metric weights** *tos k1 k2 k3 k4 k5 k6*	Adjusts the EIGRP metric or K value. EIGRP uses the following formula to determine the total metric to the network: metric = [k1 x bandwidth + (k2 x bandwidth)/(256 – load) + k3 x delay + k6 x extended attributes] x [k5/(reliability + k4)] Default values and ranges are as follows: • TOS—0. The range is from 0 to 8. • k1—1. The range is from 0 to 255. • k2—0. The range is from 0 to 255. • k3—1. The range is from 0 to 255. • k4—0. The range is from 0 to 255. • k5—0. The range is from 0 to 255. • k6—0. The range is from 0 to 255.
Step 5	(Optional) switch(config-router-af)# **timers active-time** {*time-limit* | **disabled**}	Sets the time the router waits in minutes (after sending a query) before declaring the route to be stuck in the active (SIA) state. The range is from 1 to 65535. The default is 3.
Step 6	switch(config-router-af)# **exit**	Exits address-family configuration mode.
Step 7	switch(config-router)# **exit**	Exits router configuration mode.
Step 8	switch(config)# **interface ethernet** *slot//port*	Enters interface configuration mode.
Step 9	(Optional) switch(config-if)# {**ip**|**ipv6**} **bandwidth eigrp** *instance-tag bandwidth*	Configures the bandwidth metric for EIGRP on an interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The bandwidth range is from 1 to 2,560,000,000 Kb/s.
Step 10	(Optional) switch(config-if)# {**ip** | **ipv6**} **bandwidth-percent eigrp** *instance-tagpercent*	Configures the percentage of bandwidth that EIGRP might use on an interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The percent range is from 0 to 100. The default is 50.
Step 11	(Optional) switch(config-if)# **no** {**ip** | **ipv6**} **delay eigrp** *instance-tag delay*	Configures the delay metric for EIGRP on an interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. The delay range is from 1 to 16777215 (in tens of microseconds).
Step 12	(Optional) switch(config-if)# {**ip** | **ipv6**} **distribute-list eigrp** *instance-tag* {**prefix-list** *name* | **route-map** *map-name*} {**in** | **out**}	Configures the route filtering policy for EIGRP on this interface. The instance tag, prefix list name, and route-map name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 13	(Optional) switch(config-if)# **no** {**ip** | **ipv6**} **next-hop-self eigrp** *instance-tag*	Configures EIGRP to use the received next-hop address rather than the address for this interface. The default is to use the IP address of this interface for the next-hop address. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

Step 14	(Optional) switch(config-if)# {**ip** | **ipv6**} **offset-list eigrp** *instance-tag* {**prefix-list** *name* | **route-map** *map-name*} {**in** | **out**} *offset*	Adds an offset to incoming and outgoing metrics to routes learned by EIGRP. The instance tag, prefix list name, and route-map name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 15	(Optional) switch(config-if)# {**ip** | **ipv6**} **passive-interface eigrp** *instance-tag*	Suppresses EIGRP hellos, which prevents neighbors from forming and sending routing updates on an EIGRP interface. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 16	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure optional parameters in address-family configuration mode to tune EIGRP for your network:

```
switch# configure terminal

switch(config)# router eigrp Test1

switch(config-router)# address-family ipv4 unicast
switch(config-router-af)#      default-information originate     always
switch(config-router-af)# distance 25 100

switch(config-router-af)#      metric      max-hops     70
switch(config-router-af)# metric weights  0  1  3  2  1  0
switch(config-router-af)# timers active-time 200

switch(config-router-af)# copy running-config startup-config
```

The following example shows how to configure optional parameters in interface configuration mode to tune EIGRP for your network:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)#     ip     bandwidth     eigrp   Test1      30000
switch(config-if)#    ip bandwidth-percent    eigrp    Test1     30
switch(config-if)#  ip  delay  eigrp  Test1 100

switch(config-if)# ip distribute-list eigrp Test1 route-map EigrpTest in

switch(config-if)# ip next-hop-self eigrp Test1

switch(config-if)# ip offset-list eigrp Test1 prefix-list EigrpList in

switch(config-if)#  ip  passive-interface  eigrp Test1

switch(config-if)# copy running-config startup-config
```

8.9 Configuring Virtualization for EIGRP

You can configure multiple EIGRP processes in each VDC. You can also create multiple VRFs within each VDC and use the same or multiple EIGRP processes in each VRF. You assign an interface to a VRF.

Before you begin

- Ensure that you have enabled the EIGRP feature.
- Create the VDCs and VRFs.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*
3. switch(config-vrf)# **router eigrp** *instance-tag*
4. switch(config-router)# **interface ethernet** *slot//port*
5. switch(config-if)# **vrf member** *vrf-name*
6. switch(config-if)# {**ip** | **ipv6**} **router eigrp** *instance-tag*
7. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode. The VRF name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 3	switch(config-vrf)# **router eigrp** *instance-tag*	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters. If you configure an instance tag that does not qualify as an AS number, you must use the **autonomous-system** command to configure the AS number explicitly or this EIGRP instance remains in the shutdown state.
Step 4	switch(config-router)# **interface ethernet** *slot//port*	Enters interface configuration mode. Use **?** to find the slot and port ranges.
Step 5	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF. The VRF name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 6	switch(config-if)# {**ip** | **ipv6**} **router eigrp** *instance-tag*	Adds this interface to the EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 7	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a VRF and add an interface to the VRF:

```
switch# configure interface
```

```
switch(config)# vrf context NewVRF
switch(config-vrf)# router eigrp Test1
switch(config-router)# interface ethernet 1/2
switch(config-if)# ip router eigrp Test1
switch(config-if)# vrf member NewVRF
switch(config-if)# copy running-config startup-config
```

8.10 Verifying the EIGRP Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show {**ip** | **ipv6**} **eigrp** [*instance-tag*]	Displays a summary of the configured EIGRP processes.
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **interfaces** [*type number*] [**brief**] [**detail**]	Displays information about all configured EIGRP interfaces.
show {**ip** | **ipv6**} **eigrp** *instance-tag* **neighbors** [*type number*] [**detail**]	Displays information about all the EIGRP neighbors. Use this command to verify the EIGRP neighbor configuration.
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **route** [*ip-prefix/length*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**vrf** *vrf-name*]	Displays information about all the EIGRP routes.
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **topology** [*ip-prefix/length*] [**active**] [**all-links**] [**detail-links**] [**pending**] [**summary**] [**zero-successors**] [**vrf** *vrf-name*]	Displays information about the EIGRP topology table.
show running-configuration eigrp	Displays the current running EIGRP configuration.

8.11 Displaying EIGRP Statistics

Use one of the following commands to display EIGRP statistics:

Command	Purpose
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **accounting** [**vrf** *vrf-name*]	Displays accounting statistics for EIGRP.
show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **route-map statistics redistribute**	Displays redistribution statistics for EIGRP.

show {**ip** | **ipv6**} **eigrp** [*instance-tag*] **traffic** [**vrf** *vrf-name*]	Displays traffic statistics for EIGRP.

8.12 Configuration Example for EIGRP

```
switch# configure terminal
switch(config)# feature eigrp
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 192.0.2.55/24
switch(config-if)# ip router eigrp Test1
switch(config)# exit
switch(config)# no shutdown
switch(config)# router eigrp Test1
switch(config-router)# router-id 192.0.2.1
```

The following example shows how to use a route map with the distribute-list command to filter routes that are dynamically received from (or advertised to) EIGRP peers. The example configures a route table with a metric of 50, a source protocol of BGP, and an autonomous system number of 45000. When the match clauses is true, the tag value of the destination routing protocol is set to 5. The route map is used to distribute incoming packets for an EIGRP process

```
switch(config)# route-map metric-range
switch(config-route-map)# match metric 50
switch(config-route-map)# match source-protocol bgp 45000
switch(config-route-map)# set tag 5
switch(config-route-map)# exit
switch(config)#router eigrp 1
switch(config-router)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip address 172.16.0.0
switch(config-if)# ip router eigrp 1
switch(config-if)# ip distribute-list eigrp 1 route-map metric-range in
```

The following example shows how to use a route map with the redistribute command to allow routes that are redistributed from the routing table to be filtered with a route map before being admitted into an EIGRP topology table. The example shows how to configure a route map to match EIGRP routes with a metric of 110, 200, or an inclusive range of 700 to 800. When the match clause is true, the tag value of the destination routing protocol is set to 10. The route map is used to redistribute EIGRP packets.

```
switch(config)# route-map metric-eigrp

switch(config-route-map)# match metric 110 200 750 +- 50

switch(config-route-map)#   set   tag   10
switch(config-route-map)#              exit
switch(config)#router eigrp 1

switch(config-router)#    redistribute    eigrp    route-map metric-eigrp

switch(config-router)#exit

switch(config)#  interface ethernet 1/2


switch(config-if)# ip address 172.16.0.0
switch(config-if)# ip router eigrp 1
```

8.13 Related Documents for EIGRP

Related Topic	Document Title
EIGRP CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*
EIGRP overview	-
EIGRP FAQs	-

8.14 Feature History for EIGRP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 15 : Feature History for EIGRP

Feature Name	Release	Feature Information
EIGRP	8.4(1)	Added support for route-map filtering.
EIGRP	8.4(1)	Added support for configuring the administrative distance of routes.
EIGRP	8.4(1)	Added the ability to configure all EIGRP interfaces as passive by default.
Wide metrics	8.4(1)	Added support for EIGRP wide metrics.

BFD	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide* for more information.
Graceful shutdown	8.4(1)	Added support to gracefully shut down an EIGRP instance or EIGRP on an interface but preserve the EIGRP configuration.
EIGRP instance tag	8.4(1)	Changed the length to 20 characters.
Limits on redistributed routes	8.4(1)	Added support for limiting the number of redistributed routes.
EIGRP IPv6 support	8.4(1)	Added support for IPv6.
Authentication	8.4(1)	Added the ability to configure authentication within a VRF for EIGRP.
EIGRP	8.4(1)	This feature was introduced.

CHAPTER 9 Configuring IS-IS

This chapter contains the following sections:

- Finding Feature Information.
- Information About IS-IS.
- Licensing Requirements for IS-IS.
- Prerequisites for IS-IS.
- Guidelines and Limitations for IS-IS.
- Default Settings for IS-IS.
- Configuring IS-IS.
- Verifying the IS-IS Configuration.
- Monitoring IS-IS.
- Configuration Examples for IS-IS.
- Related Documents for IS-IS.
- Standards for IS-IS.
- Feature History for IS-IS.

9.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

9.2 Information About IS-IS

IS-IS is an Interior Gateway Protocol (IGP) based on Standardization (ISO)/International Engineering Consortium (IEC) 10589. Inspur INOS supports Internet Protocol version 4 (IPv4), and beginning with Inspur INOS Release 8.4(1), Inspur INOS supports IPv6. IS-IS is a dynamic link-state routing protocol that can detect changes in the network topology and calculate loop-free routes to other nodes in the network. Each router maintains a link-state database that describes the state of the network and sends packets on every configured link to discover neighbors. IS-IS floods the link-state information across the network to each neighbor. The router also sends advertisements and updates on the link-state database through all the existing neighbors.

9.2.1 IS-IS Overview

IS-IS sends a hello packet out every configured interface to discover IS-IS neighbor routers. The hello packet contains information, such as the authentication, area, and supported protocols, which the receiving interface uses to determine compatibility with the originating interface. The hello packets are also padded to ensure that IS-IS establishes adjacencies only with interfaces that have matching maximum transmission unit (MTU) settings. Compatible interfaces form adjacencies, which update routing information in the link-state database through link-state update messages (LSPs). By default, the router sends a periodic LSP refresh every 10 minutes and the LSPs remain in the link-state database for 20 minutes (the LSP lifetime). If the router does not receive an LSP refresh before the end of the LSP lifetime, the router deletes the LSP from the database.

The LSP interval must be less than the LSP lifetime or the LSPs time out before they are refreshed.

IS-IS sends periodic hello packets to adjacent routers. If you configure transient mode for hello packets, these hello packets do not include the excess padding used before IS-IS establishes adjacencies. If the MTU value on adjacent routers changes, IS-IS can detect this change and send padded hello packets for a period of time. IS-IS uses this feature to detect mismatched MTU values on adjacent routers.

IS-IS Areas

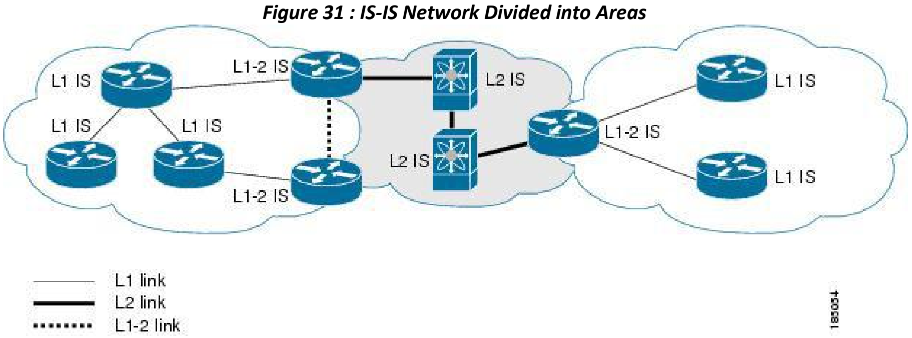
You can design IS-IS networks as a single area that includes all routers in the network or as multiple areas that connect into a backbone or Level 2 area. Routers in a nonbackbone area are Level 1 routers that establish adjacencies within a local area (intra-area routing). Level 2 area routers establish adjacencies to other Level 2 routers and perform routing between Level 1 areas (inter-area routing). A router can have both Level 1 and Level 2 areas configured. These Level 1/Level 2 routers act as area border routers that route information from the local area to the Level 2 backbone area

Within a Level 1 area, routers know how to reach all other routers in that area. The Level 2 routers know how to reach other area border routers and other Level 2 routers. Level 1/Level 2 routers straddle the boundary between two areas, routing traffic to and from the Level 2 backbone area. Level1/Level2 routers use the attached (ATT) bit signal Level 1 routers to set a default route to this Level1/Level2 router to connect to the Level 2 area.

In some instances, such as when you have two or more Level1/Level 2 routers in an area, you may want to control which Level1/Level2 router that the Level 1 routers use as the default route to the Level 2 area. You can configure which Level1/Level2 router sets the attached bit.

Each IS-IS instance in Inspur INOS supports either a single Level 1 or Level 2 area, or one of each. By default, all IS-IS instances automatically support Level 1 and Level 2 routing.

Figure 31 : IS-IS Network Divided into Areas



An autonomous system boundary router (ASBR) advertises external destinations throughout the IS-IS autonomous system. External routes are the routes redistributed into IS-IS from any other protocol.

NET and System ID

Each IS-IS instance has an associated network entity title (NET). The NET is comprised of the IS-IS system ID, which uniquely identifies this IS-IS instance in the area and the area ID. For example, if the NET is 47.0004.004d.0001.0001.0c11.1111.00, the system ID is 0000.0c11.1111.00 and the area is ID 47.0004.004d.0001.

Designated Intermediate System

IS-IS uses a designated intermediate system (DIS) in broadcast networks to prevent each router from forming unnecessary links with every other router on the broadcast network. IS-IS routers send LSPs to the DIS, which manages all the link-state information for the broadcast network. You can configure the IS-IS priority that IS-IS uses to select the DIS in an area.

9.2.2 IS-IS Authentication

You can configure authentication to control adjacencies and the exchange of LSPs. Routers that want to become neighbors must exchange the same password for their configured level of authentication. IS-IS blocks a router that does not have the correct password. You can configure IS-IS authentication globally or for an individual interface for Level 1, Level 2, or both Level 1/Level 2 routing.

IS-IS supports the following authentication methods:

- Clear text—All packets exchanged carry a cleartext 128-bit password.
- MD5 digest—All packets exchanged carry a message digest that is based on a 128-bit key.

To provide protection against passive attacks, IS-IS never sends the MD5 secret key as cleartext through the network. In addition, IS-IS includes a sequence number in each packet to protect against replay attacks.

You can use also keychains for hello and LSP authentication. See the *Inspur CN12700 Series INOS Security Configuration Guide*, for information on keychain management.

Mesh Groups

A mesh group is a set of interfaces in which all routers reachable over the interfaces have at least one link to every other router. Many links can fail without isolating one or more routers from the network.

In normal flooding, an interface receives a new LSP and floods the LSP out over all other interfaces on the router. With mesh groups, when an interface that is part of a mesh group receives a new LSP, the interface does not flood the new LSP over the other interfaces that are part of that mesh group.

You can also configure mesh groups in block mode for parallel links between routers. In this mode, all LSPs are blocked on that interface in a mesh group after the routers initially exchange their link-state information.

OverloadBit

IS-IS uses the overload bit to tell other routers not to use the local router to forward traffic but to continue routing traffic destined for that local router.

You may want to use the overload bit in these situations:

- The router is in a critical condition.
- Graceful introduction and removal of the router to/from the network.
- Other (administrative or traffic engineering) reasons such as waiting for BGP convergence.

9.2.3 Route Summarization

You can configure a summary aggregate address. Route summarization simplifies route tables by replacing a number of more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

If more specific routes are in the routing table, IS-IS advertises the summary address with a metric equal to the minimum metric of the more specific routes.

9.2.4 Route Redistribution

You can use IS-IS to redistribute static routes, routes learned by other IS-IS autonomous systems, or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into IS-IS. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on.

Whenever you redistribute routes into an IS-IS routing domain, Inspur INOS does not, by default, redistribute the default route into the IS-IS routing domain. You can generate a default route into IS-IS, which can be controlled by a route policy.

You also configure the default metric that is used for all imported routes into IS-IS.

9.2.5 Administrative Distance

The administrative distance is a rating of the trustworthiness of a routing information source. A higher value indicates a lower trust rating. The administrative distance is used to discriminate between routes learned from more than one routing protocol. The route with the lowest administrative distance is installed in the IP routing table.

You can configure the administrative distance for internal and external routes based on various match criteria for a given prefix. Routing protocols such as IS-IS configure the prefix into the Routing Information Base (RIB), along with the next hops based on these metrics. If multiple paths are available for a prefix, the routing protocol chooses the best path based on the cost to reach the next hop and the administrative distance. Beginning with Inspur INOS Release 8.4(1), you can specify that prefixes be considered based on specific routes. In prior releases, one administrative distance was sufficient for all internal routes.

9.2.6 Load Balancing

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the utilization of network segments and increases the effective network bandwidth.

Inspur INOS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the IS-IS route table and the unicast RIB. You can configure IS-IS to load balance traffic across some or all of those paths.

9.2.7 BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules. See the Inspur CN12700 Series INOS Interfaces Configuration Guide, for more information.

9.2.8 Virtualization Support

Inspur INOS supports multiple instances of the IS-IS protocol that runs on the same system. IS-IS supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). You can configure up to four IS-IS instances in a VDC.

By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. See the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

9.2.9 High Availability and Graceful Restart

Inspur INOS provides a multilevel high-availability architecture. IS-IS supports stateful restart, which is also referred to as non-stop routing (NSR). If IS-IS experiences problems, it attempts to restart from its previous run-time state. The neighbors would not register any neighbor event in this case. If the first restart is not successful and another problem occurs, IS-IS attempts a graceful restart as per RFC 3847. A graceful restart, or non-stop forwarding (NSF), allows IS-IS to remain in the data forwarding path through a process restart. When the restarting IS-IS interface is operational again, it rediscovers its neighbors, establishes adjacency, and starts sending its updates again. At this point, the NSF helpers recognize that the graceful restart has finished.

A stateful restart is used in the following scenarios:

- First recovery attempt after process experiences problems
- ISSU
- User-initiated switchover using the **system switchover** command

A graceful restart is used in the following scenarios:

- Second recovery attempt after the process experiences problems within a 4-minute interval
- Manual restart of the process using the **restart isis** command
- Active supervisor removal
- Active supervisor reload using the **reload module** *active-sup* command

9.2.10 Multiple IS-IS Instances

Inspur INOS supports a maximum of four instances of the IS-IS protocol that run on the same node. You cannot configure multiple instances over the same interface. Every instance uses the same system router ID.

9.3 Licensing Requirements for IS-IS

IS-IS requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *License and Copyright Information for Inspur INOS Software*.

9.4 Prerequisites for IS-IS

IS-IS has the following prerequisites:

- You must enable IS-IS.
- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

9.5 Guidelines and Limitations for IS-IS

IS-IS has the following configuration guidelines and limitations:

- You can configure a maximum of four IS-IS instances per VDC.
- Because the default reference bandwidth is different for Inspur INOS and Inspur IOS, the advertised tunnel IS-IS metric is different for these two operating systems.
- For the IS-IS Multitopology feature, one topology for IPv4 and one for IPv6 is supported.
- Unlike IOS, INOS-ISIS works even when there is a change in bandwidth. It causes an SPF and routes updates. This result in an excessive packet drop, but port P0 continues to be active.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

9.6 Default Settings for IS-IS

Table 16 : Default IS-IS Parameters

Parameters	**Default**
Administrative distance	115
Area level	Level-1-2
DIS priority	64
Graceful restart	Enabled
Hello multiplier	3
Hello padding	Enabled
Hello time	10 seconds
IS-IS feature	Disabled
LSP interval	33
LSP MTU	1492
Maximum LSP lifetime	1200 seconds
Maximum paths	4
Metric	40

Reference bandwidth	40 Gbps

9.7 Configuring IS-IS

9.7.1 IS-IS Configuration Modes

Router Configuration Mode Example

This example shows how to enter router configuration mode:

```
switch#: configure terminal
switch(config)# router  isis  isp
switch(config-router)#
```

Router Address Family Configuration Mode Example

This example shows how to enter router address family configuration mode:

```
switch(config)# router isis isp

switch(config-router)# address-family ipv4 unicast

switch(config-router-af)#
```

9.7.2 Enabling the IS-IS Feature

You must enable the IS-IS feature before you can configure IS-IS.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature isis**	Enables the IS-IS feature.
Step 3	(Optional) switch(config)# **show feature**	Displays enabled and disabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

9.7.3 Creating an IS-IS Instance

You can create an IS-IS instance and configure the area level for that instance.

You must remove any IS-IS commands that are configured in interface mode to completely remove all configuration for the IS-IS instance

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	switch(config-router)# **net** *network-entity-title*	Configures the NET for this IS-IS instance.
Step 4	(Optional) switch(config-router)#**is-type** {**level-1** | **level-2** | **level-1-2**}	Configures the area level for this IS-IS instance. The default is level-1-2.
Step 5	(Optional) switch(config)# **show isis** {**vrf** *vrf-name*} **process**	Displays a summary of IS-IS information for all IS-IS instances.
Step 6	(Optional) switch(config-router)# **distance** *value*	Sets the administrative distance for IS-IS. The range is from 1 to 255. The default is 115.
Step 7	(Optional) switch(config-router)#**log-adjacency-changes**	Sends a system message whenever an IS-IS neighbor changes the state.
Step 8	(Optional) switch(config-router)# **lsp-mtu** *size*	Sets the MTU for LSPs in this IS-IS instance. The range is from 128 to 4352 bytes. The default is 1492.
Step 9	(Optional) switch(config-router)# **maximum-paths** *number*	Configures the maximum number of equal-cost paths that IS-IS maintains in the route table. The range is from 1 to 16. The default is 4.
Step 10	(Optional) switch(config-router)# **reference-bandwidth** *bandwidth value* {**Mbps** | **Gbps**}	Sets the default reference bandwidth used for calculating the IS-IS cost metric. The range is from 1 to 4000 Gbps. The default is 40 Gbps.
Step 11	(Optional) switch(config-if)# **clear isis** [*instance-tag*] **adjacency** [***** | *system-id* | *interface*]	Clears neighbor statistics and removed adjacencies for this IS-IS instance.
Step 12	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to create an IS-IS instance in a level 2 area:

```
switch# configure terminal

switch(config)# router isis Enterprise

switch(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00

switch(config-router)# is-type level 2
switch(config-router)# copy running-config startup-config
```

9.7.4 Restarting an IS-IS Instance

You can restart an IS-IS instance. This action clears all neighbors for the instance.
To restart an IS-IS instance and remove all associated neighbors, use the following command:

Procedure

	Command or Action	Purpose
Step 1	switch(config)# **restart isis** *instance-tag*	Restarts the IS-IS instance and removes all neighbors.

9.7.5 Shutting Down IS-IS

You can shut down the IS-IS instance. This action disables this IS-IS instance and retains the configuration. To shut down the IS-IS instance, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **shutdown**	Disables the IS-IS instance.

9.7.6 Configuring IS-IS on an Interface

You can add an interface to an IS-IS instance.

Before you begin

You must enable IS-IS.
Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	(Optional) switch(config-if)# **medium** {**broadcast** | **p2p**}	Configures the broadcast or point-to-point mode for the interface. IS-IS inherits this mode.
Step 4	switch(config-if)# {**ip** | **ipv6**} **router isis** *instance-tag*	Associates this IPv4 or IPv6 interface with an IS-IS instance.
Step 5	(Optional) switch(config-if)# **show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*]	Displays IS-IS information for an interface.
Step 6	(Optional) switch(config-if)# **isis circuit-type** {**level-1** | **level-2** | **level-1-2**}	Sets the type of adjacency that this interface participates in. Use this command only for routers that participate in both Level 1 and Level 2 areas.
Step 7	(Optional) **isis metric** *value* {**level-1** | **level-2**}	Sets the IS-IS metric for this interface. The range is from 1 to 16777214. The default is 10.

Step 8	(Optional) switch(config-if)# **isis passive** *value* {**level-1** | **level-2** | **level-1-2**}	Prevents the interface from forming adjacencies but still advertises the prefix associated with the interface.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add Ethernet 1/2 interface to an IS-IS instance:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)# ip router isis Enterprise

switch(config-if)# copy running-config startup-config
```

9.7.7 Shutting Down IS-IS on an Interface

You can gracefully shut down IS-IS on an interface. This action removes all adjacencies and stops IS-IS traffic on this interface but preserves the IS-IS configuration.

To disable IS-IS on an interface, use the following command in interface configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **isis shutdown**	Disables IS-IS on this interface. The IS-IS interface configuration remains.

9.7.8 Configuring IS-IS Authentication in an Area

You can configure IS-IS to authenticate LSPs in an area.

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	switch(config-router)# **authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**}	Sets the authentication method used for a Level 1 or Level 2 area as cleartext or as an MD5 authentication digest.
Step 4	switch(config-router)# **authentication key-chain** *key* {**level-1** | **level-2**}	Configures the authentication key used for an IS-IS area-level authentication.
Step 5	(Optional) switch(config-router)# **authentication-check** {**level-1** | **level-2**}	Enables checking the authentication parameters in a received packet.

Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure cleartext authentication on an IS-IS instance:

```
switch# configure terminal

switch(config)# router isis Enterprise

switch(config-router)#        authentication-type cleartext     level-2
switch(config-router)# authentication    key-chain    ISISKey    level-2
switch(config-router)#    copy    running-config startup-config
```

9.7.9 Configuring IS-IS Authentication on an Interface

You can configure IS-IS to authenticate Hello packets on an interface.

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot*/*port*	Enters interface configuration mode.
Step 3	switch(config-if)# **isis authentication-type** {**cleartext** | **md5**} {**level-1** | **level-2**}	Sets the authentication type for IS-IS on this interface as cleartext or as an MD5 authentication digest.
Step 4	switch(config-if)# **isis authentication key-chain** *key* {**level-1** | **level-2**}	Configures the authentication key used for IS-IS on this interface.
Step 5	(Optional) **isis authentication-check** {**level-1** | **level-2**}	Enables checking the authentication parameters in a received packet.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure cleartext authentication on an IS-IS instance:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)#    isis    authentication-type cleartext    level-2
switch(config-if)#   isis authentication  key-chain  ISISKey
switch(config-if)# copy running-config startup-config
```

9.7.10 Configuring a Mesh Group

You can add an interface to a mesh group to limit the amount of LSP flooding for interfaces in that mesh group. You can optionally block all LSP flooding on an interface in a mesh group.

To add an interface to a mesh group, use the following command in interface configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# **isis mesh-group** {**blocked** | *mesh-id*}	Adds this interface to a mesh group. The range is from 1 to 4294967295.

9.7.11Configuring a Designated Intermediate System

You can configure a router to become the designated intermediate system (DIS) for a multiaccess network by setting the interface priority.

To configure the DIS, use the following command in interface configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# **isis priority** *number* {**level-1** | **level-2**}	Sets the priority for DIS selection. The range is from 0 to 127. The default is 64.

9.7.12 Configuring Dynamic Host Exchange

You can configure IS-IS to map between the system ID and the hostname for a router using dynamic host exchange.

To configure dynamic host exchange, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **hostname dynamic**	Enables dynamic host exchange.

9.7.13 Setting the Overload Bit

You can configure the router to signal other routers not to use this router as an intermediate hop in their shortest path first (SPF) calculations. You can optionally configure the overload bit temporarily on startup, until BGP converges.

In addition to setting the overload bit, you might also want to suppress certain types of IP prefix advertisements from LSPs for Level 1 or Level 2 traffic.

To set the overload bit, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **set-overload-bit** {**always** | **on-startup** {*seconds* | **wait-for bgp** *as-number*}} [**suppress** [**interlevel** | **external**]]	Sets the overload bit for IS-IS. The seconds range is from 5 to 86400.

9.7.14 Configuring the Attached Bit

You can configure the attached bit to control which Level 1/Level 2 router that the Level 1 routers use as the default route to the Level 2 area. If you disable setting the attached bit, the Level 1 routers do not use this Level 1/Level 2 router to reach the Level 2 area.

To configure the attached bit for a Level 1/Level 2 router, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# [**no**] **attached-bit**	Configures the Level 1/Level 2 router to set the attached bit. This feature is enabled by default.

9.7.15 Configuring the Transient Mode for Hello Padding

You can configure the transient mode for hello padding to pad hello packets when IS-IS establishes adjacency and remove that padding after IS-IS establishes adjacency.

To configure the mode for hello padding, use the following command in router configuration mode:

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# [**no**] **isis hello-padding**	Pads the hello packet to the full MTU. The default is enabled. Use the no form of this command to configure the transient mode of hello padding.

9.7.16 Configuring a Summary Address

You can create aggregate addresses that are represented in the routing table by a summary address. One summary address can include multiple groups of addresses for a given level. Inspur INOS advertises the smallest metric of all the more-specific routes.

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc c**ommand).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} {**unicast** | **multicast**}	Enters address family configuration mode.
Step 4	switch(config-router-af)# **summary-address** *ip-prefix/mask-len* {**level-1** | **level-2** | **level-1-2**}	Configures a summary address for an IS-IS area for IPv4 or IPv6 addresses.
Step 5	(Optional) switch(config-if)# **show isis** [**vrf***vrf-name*] {**ip** | **ipv6**} **summary-address** *ip-prefix* [**longer-prefixes**]	Displays IS-IS IPv4 or IPv6 summary address information.

Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure an IPv4 unicast summary address for IS-IS:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-router)#   address-family   ipv4 unicast
switch(config-router-af)# summary-address 192.0.2.0/24 level-2
switch(config-router-af)#  copy  running-config startup-config
```

9.7.17 Configuring Redistribution

You can configure IS-IS to accept routing information from another routing protocol and redistribute that information through the IS-IS network. You can optionally assign a default route for redistributed routes.

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters address family configuration mode.
Step 4	switch(config-router-af)# **redistribute** {**bgp** *as* | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static** | **direct**} **route-map** *map-name*	Redistributes routes from other protocols into IS-IS.
Step 5	(Optional) switch(config-router-af)# **default-information originate** [**always**] [**route-map** *map-name*]	Generates a default route into IS-IS.
Step 6	(Optional) switch(config-router-af)# **distribute** {**level-1** | **level-2**} **into** {**level-1** | **level-2**} {**route-map** *route-map* | **all**}	Redistributes routes from one IS-IS level to the other IS-IS level.
Step 7	(Optional) switch(config-router-af)# **show isis** [**vrf** *vrf-name*] {**ip** | **ipv6**} **route** *ip-prefix* [*detail* | **longer-prefixes** [**summary** | **detail**]]	Shows the IS-IS routes.
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to redistribute EIGRP into IS-IS:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-router)# address-family ipv4 unicast

switch(config-router-af)# redistribute eigrp 201 route-map ISISmap

switch(config-router-af)# copy running-config startup-config
```

9.7.18 Limiting the Number of Redistributed Routes

Route redistribution can add many routes to the IS-IS route table. You can configure a maximum limit to the number of routes accepted from external protocols. IS-IS provides the following options to configure redistributed route limits:

- Fixed limit—Logs a message when IS-IS reaches the configured maximum. IS-IS does not accept any more redistributed routes. You can optionally configure a threshold percentage of the maximum where IS-IS logs a warning when that threshold is passed.
- Warning only—Logs a warning only when IS-IS reaches the maximum. IS-IS continues to accept redistributed routes.
- Withdraw—Starts the timeout period when IS-IS reaches the maximum. After the timeout period, IS-IS requests all redistributed routes if the current number of redistributed routes is less than the maximum limit. If the current number of redistributed routes is at the maximum limit, IS-IS withdraws all redistributed routes. You must clear this condition before IS-IS accepts more redistributed routes. You can optionally configure the timeout period.

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router eigrp** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	switch(config-router)# **redistribute** {**bgp** *ip* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name*	Redistributes the selected protocol into IS-IS through the configured route map.

Step 4	switch(config-router)# **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries timeout*]]	Specifies a maximum number of prefixes that IS-IS distributes. The range is from 0 to 65536. You can optionally specify the following: • *threshold*—Percent of maximum prefixes that triggers a warning message. • **warning-only**—Logs an warning message when the maximum number of prefixes is exceeded. **withdraw**—Withdraws all redistributed routes. You can optionally try to retrieve the redistributed routes. The num-retries range is from 1 to 12. The timeout is 60 to 600 seconds. The default is 300 seconds. Use the **clear isis redistribution** command if all routes are withdrawn.
Step 5	(Optional) switch(config-router)# **show running-config isis**	Displays the IS-IS configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to limit the number of redistributed routes into IS-IS:

```
switch# configure terminal

switch(config)# router eigrp isis Enterprise

switch(config-router)#    redistribute    bgp    route-map FilterExternalBGP

switch(config-router)# redistribute maximum-prefix 1000 75
```

9.7.19 Configuring the Administrative Distance of Routes

You can set the administrative distance of routes added by IS-IS into the RIB.

Before you begin

You must enable IS-IS (see the "Enabling the IS-IS Feature" section on page 9-9). Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router isis** *instance-tag*
3. switch(config-router)# **table-map** *route-map-name* [**filter**]
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance and enters router configuration mode.
Step 3	switch(config-router)#**table-map***route-map-name*[**filter**]	Configures a table map with route map information. You can enter up to 63 alphanumeric characters for the map name. The **filter** keyword filters routes rejected by the route map and does not download them to the RIB.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

9.7.20 Disabling Strict Adjacency Mode

When both IPv4 and IPv6 address families are enabled, strict adjacency mode is enabled by default. In this mode, the device does not form an adjacency with any router that does not have both address families enabled. You can disable strict adjacency mode using the **no adjacency check** command.

Before you begin

You must enable IS-IS.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 3	switch(config-router)# **address-family ipv4 unicast**	Enters address family configuration mode.
Step 4	switch(config-router-af)# **no adjacency-check**	Disables strict adjacency mode for the IPv6 address family.
Step 5	switch(config-router-af)# **exit**	Exits address family configuration mode.
Step 6	switch(config-router-af)# **address-family ipv6 unicast**	Enters address family configuration mode.
Step 7	switch(config-router-af)# **no adjacency-check**	Disables strict adjacency mode for the IPv6 address family.
Step 8	(Optional) switch(config-router-af)# **show running-config isis**	Displays the IS-IS configuration.
Step 9	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to disable strict adjacency mode:

```
switch# configure terminal
```

```
switch(config)# router isis Enterprise
switch(config-router)#  address-family ip4 unicast
switch(config-router-af)# no adjacency-check

switch(config-router)# exit
switch(config-router-af)# address-family ip6 unicast

switch(config-router-af)# no adjacency-check
switch(config-router-af)# show running-config isis
switch(config-router-af)#   copy   running-config   startup-config
```

9.7.21 Configuring a Graceful Restart

You can configure a graceful restart for IS-IS.

Before you begin

You must enable IS-IS. Create the VDCs and VRFs.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS process with the configured name.
Step 3	switch(config-router)# **graceful-restart**	Enables a graceful restart and the graceful restart helper functionality. Enabled by default.
Step 4	switch(config-router)# **graceful-restart t3 manual** *time*	Configures the graceful restart T3 timer. The range is from 30 to 65535 seconds. The default is 60.
Step 5	(Optional) switch(config-router)# **show running-config isis**	Displays the IS-IS configuration.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable a graceful restart:

```
switch#        configure terminal

switch(config)# router  isis  Enterprise
switch(config-router)# graceful restart

switch(config-router)# copy running-config startup-config
```

9.7.22 Configuring Virtualization

You can configure multiple IS-IS instances in each VDC. You can also create multiple VRFs within each VDC and use the same or multiple IS-IS instances in each VRF. You assign an IS-IS interface to a VRF.

You must configure a NET for the configured VRF.

Before you begin

You must enable IS-IS. Create the VDCs and VRFs.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode.
Step 3	switch(config)# **exit**	Exits VRF configuration mode.
Step 4	switch(config)# **router isis** *instance-tag*	Creates a new IS-IS instance with the configured instance tag.
Step 5	(Optional) switch(config-router)# **vrf** *vrf-name*	Enters VRF configuration mode.
Step 6	switch(config-router-vrf)# **net** *network-entity-title*	Configures the NET for this IS-IS instance.
Step 7	switch(config-router-vrf)# **exit**	Exits router VRF configuration mode.
Step 8	switch(config)# **interface ethernet** *slot/port*	Enters interface configuration mode.
Step 9	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 10	switch(config-if)# **ip address** *ip-prefix/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 11	switch(config-if)# **ip router isis** *instance-tag*	Associates this IPv4 interface with an IS-IS instance.
Step 12	(Optional) switch(config-if)# **show isis** [**vrf** *vrf-name*] [*instance-tag*] **interface** [*interface-type slot/port*]	Displays IS-IS information for an interface. in a VRF.
Step 13	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to create a VRF and add an interface to the VRF:

```
switch#  configure  terminal

switch(config)#  vrf  context NewVRF

switch(config-vrf)# exit

switch(config)# router isis Enterprise

switch(config-router)# vrf NewVRF
switch(config-router-vrf)# net 47.0004.004d.0001.0001.0c11.1111.00
switch(config-router-vrf)# interface       ethernet 1/2
switch(config-if)#   vrf   member NewVRF

switch(config-if)#   ip   address 192.0.2.1/16

switch(config-if)# ip router isis Enterprise

switch(config-if)# copy running-config startup-config
```

9.7.23 Tuning IS-IS

You can tune IS-IS to match your network requirements.

You can use the following optional commands in router configuration mode to tune IS-IS:

Procedure

	Command or Action	Purpose
Step 1	switch(config-router)# **lsp-gen-interval** [**level-1** | **level-2**] *lsp-max-wait* [*lsp-initial-wait lsp-second-wait*]	Configures the IS-IS throttle for LSP generation. The optional parameters are as follows: • *lsp-max-wait*—The maximum wait between the trigger and LSP generation. The range is from 500 to 65535 milliseconds. • *lsp-initial-wait*—The initial wait between the trigger and LSP generation. The range is from 50 to 65535 milliseconds. • *lsp-second-wait*—The second wait used for LSP throttle during backoff. The range is from 50 to 65535 milliseconds.
Step 2	switch(config-router)# **max-lsp-lifetime** *lifetime*	Sets the maximum LSP lifetime in seconds. The range is from 1 to 65535. The default is 1200.
Step 3	switch(config-router)# **metric-style transition**	Enables IS-IS to generate and accept both narrow metric-style Type Length Value (TLV) objects and wide metric-style TLV objects. The default is disabled.
Step 4	switch(config-router)# **spf-interval** [**level-1** | **level-2**] *spf-max-wait* [*spf-initial-wait spf-second-wait*]	Configures the interval between LSA arrivals. The optional parameters are as follows: • *lsp-max-wait*—The maximum wait between the trigger and SPF computation. The range is from 500 to 65535 milliseconds. • *lsp-initial-wait*—The initial wait between the trigger and SPF computation. The range is from 50 to 65535 milliseconds. • *lsp-second-wait*—The second wait used for SPF computation during backoff. The range is from 50 to 65535 milliseconds.
Step 5	(Optional) switch(config-router-af)# **adjacency-check**	Performs an adjacency check to verify that an IS-IS instance forms an adjacency only with a remote IS-IS entity that supports the same address family. This command is enabled by default.

Step 6	(Optional) switch(config-if)# **isis csnp-interval** *seconds* [**level-1** | **level-2**]	Sets the complete sequence number PDU (CNSP) interval in seconds for IS-IS. The range is from 1 to 65535. The default is 10.
Step 7	(Optional) switch(config-if)# **isis hello-interval** *seconds* [**level-1** | **level-2**]	Sets the hello interval in seconds for IS-IS. The range is from 1 to 65535. The default is 10.
Step 8	(Optional) switch(config-if)# **isis hello-multiplier** *num* [**level-1** | **level-2**]	Specifies the number of IS-IS hello packets that a neighbor must miss before the router tears down an adjacency. The range is from 3 to 1000. The default is 3.
Step 9	(Optional) switch(config-if)# **isis lsp-interval** *milliseconds*	Sets the interval in milliseconds between LSPs sent on this interface during flooding. The range is from 10 to 65535. The default is 33.

9.8 Verifying the IS-IS Configuration

To display the IS-IS configuration, perform one of the following tasks:

Command	**Purpose**
show isis [*instance-tag*] **adjacency** [*interface*] [**detail** | **summary**] [**vrf** *vrf-name*]	Displays the IS-IS adjacencies. Use the **clear isis adjacency** command to clear these statistics.
show isis [*instance-tag*] **database** [**level-1 | level-2**] [**detail | summary**] [*LSP ID*] [{**ip** | **ipv6**} **prefix***ip-prefix*] | [**router-id** *router-id* | **adjacency** *node-id*] | [**zero-sequence**]} [**vrf** *vrf-name*]	Displays the IS-IS LSP database.
show isis [*instance-tag*] *hostname* [**vrf** *vrf-name*]	Displays the dynamic host exchange information.
show isis [*instance-tag*] **interface** [**brief** | *interface*] [**level-1 | level-2**] [**vrf***vrf-name*]	Displays the IS-IS interface information.
show isis [*instance-tag*] **mesh-group** [*mesh-id*] [**vrf***vrf-name*]	Displays the mesh group information.
show isis [*instance-tag*] **protocol** [**vrf** *vrf-name*]	Displays information about the IS-IS protocol.
show isis [*instance-tag*] {**ip** | **ipv6**} **redistribute route** [*ip-address* | **summary**] [[*ip-prefix* [**longer-prefixes** [**summary**]] [**vrf** *vrf-name*]	Displays the IS-IS route redistribution information.
show isis [*instance-tag*] {**ip** | **ipv6**} **route** [*ip-address* | **summary**] [*ip-prefix* [**longer-prefixes** [**summary**]] [**detail**] [**vrf** *vrf-name*]	Displays the IS-IS route table.
show isis [*instance-tag*] **rrm** [*interface*] [**vrf** *vrf-name*]	Displays the IS-IS interface retransmission information.
show isis [*instance-tag*] **srm** [*interface*] [**vrf** *vrf-name*]	Displays the IS-IS interface flooding information.
show isis [*instance-tag*] **ssn** [*interface*] [**vrf** *vrf-name*]	Displays the IS-IS interface PSNP information.
show isis [*instance-tag*] {**ip** | **ipv6**} **summary-address**] [*ip-address*] | [*ip-prefix*] [**vrf** *vrf-name*]	Displays the IS-IS summary address information.
show running-configuration isis	Displays the current running IS-IS configuration.

show tech-support isis [**detail**]	Displays the technical support details for IS-IS.

For detailed information about the fields in the output from these commands, see the *Inspur CN12700 Series INOS Unicast Routing Command Reference.*

9.9 Monitoring IS-IS

To display IS-IS statistics, use the following commands:

Command	**Purpose**
show isis [*instance-tag*] **adjacency** [*interface*] [**system-ID**] [**detail**] [**summary**] [**vrf** *vrf-name*]	Displays the IS-IS adjacency statistics.
show isis [*instance-tag*] **database** [**level-1 | level-2**] [**detail**] [**summary**] [*lsip*] {[**adjacency** *id* {**ip** | **ipv6**} **prefix** *prefix*] [**router-id** *id*] [*zero-sequence*]} [**vrf** *vrf-name*]	Displays the IS-IS database statistics.
show isis [*instance-tag*] **statistics** [*interface*] [**vrf** *vrf-name*]	Displays the IS-IS interface statistics.
show isis ip route-map statistics redistribute {**bgp** *id* | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** *vrf-name*]	Displays the IS-IS redistribution statistics.
show isis ip route-map statistics distribute {**level-1 | level-2**} **into** {**level-1 | level-2**} [**vrf** *vrf-name*]	Displays IS-IS distribution statistics for routes distributed between levels.
show isis [*instance-tag*] **spf-log** [**detail**] [**vrf** *vrf-name*]	Displays the IS-IS SPF calculation statistics.
show isis [*instance-tag*] **traffic** [*interface*] [**vrf** *vrf-name*]	Displays the IS-IS traffic statistics.

To clear IS-IS configuration statistics, perform one of the following tasks:

Command	**Purpose**
clear isis [*instance-tag*] **adjacency** [* | [*interface*] [**system-id** *id*]] [**vrf** *vrf-name*]	Clears the IS-IS adjacency statistics.
clear {**ip** | **ipv6**} **route map statistics** {**bgp** *id* | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} [**vrf** *vrf-name*]	Clears the IS-IS redistribution statistics
clear isis route-map statistics distribute {**level-1 | level-2**} **into** {**level-1 | level-2**} [**vrf** *vrf-name*]	Clears IS-IS distribution statistics for routes distributed between levels.
clear isis [*instance-tag*] **statistics** [* | *interface*] [**vrf** *vrf-name*]	Clears the IS-IS interface statistics.
clear isis [*instance-tag*] **traffic** [* | *interface*] [**vrf** *vrf-name*]	Clears the IS-IS traffic statistics.

9.10 Configuration Examples for IS-IS

The following example shows how to configure IS-IS:

```
router isis Enterprise is-type level-1
```

```
 net 49.0001.0000.0000.0003.00

 graceful-restart

 address-family
  ipv4      unicast
  default-
  information
  originate

interface
 ethernet
 2/1      ip
 address
 192.0.2.1/2
 4

 isis  circuit-
 type   level-1
 ip      router
 isis
 Enterprise
```

9.11 Related Documents forIS-IS

Related Topic	Document Title
IS-IS CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Command Reference*

9.12 Standards for IS-IS

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

9.13 Feature History forIS-IS

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 17 : Feature History for IS-IS

Feature Name	Release	Feature Information
IS-IS	8.4(1)	Added support for configuring the administrative distance of routes.
IS-IS	8.4(1)	Added the ability to configure all IS-IS interfaces as passive by default and then activate only those interfaces where adjacencies are desired.
IS-IS	8.4(1)	Added support for IPv6.
IS-IS	8.4(1)	Added the **no adjacency-check** command to disable strict adjacency mode.

IS-IS	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide*, for more information.
Graceful shutdown	8.4(1)	Added support to gracefully shut down an IS-IS instance or IS-IS on an interface but preserve the IS-IS configuration.
Limits on redistributed routes	8.4(1)	Added support for limiting the number of redistributed routes.
Transient mode for hello padding	8.4(1)	Added support to set or unset the hello padding mode.
Attached bit	8.4(1)	Added support to set or unset the attached bit.
IS-IS	8.4(1)	This feature was introduced.

CHAPTER 10 Configuring Basic BGP

This chapter contains the following sections:

- Finding Feature Information.
- Information About Basic BGP.
- Licensing Requirements for Basic BGP.
- Prerequisites for BGP.
- Guidelines and Limitations for BGP.
- Default Settings.
- CLI Configuration Modes.
- Configuring Basic BGP.
- Verifying the Basic BGP Configuration
- Monitoring BGP Statistics
- Configuration Examples for Basic BGP
- Related Documents for Basic BGP
- MIBs
- Feature History for BGP

10.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

10.2 Information About Basic BGP

Inspur INOS supports BGP version 4, which includes multiprotocol extensions that allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices.

BGP uses a path-vector routing algorithm to exchange routing information between BGP-enabled networking devices or BGP speakers. Based on this information, each BGP speaker determines a path to reach a particular destination while detecting and avoiding paths with routing loops. The routing information includes the actual route prefix for a destination, the path of autonomous systems to the destination, and additional path attributes.

BGP selects a single path, by default, as the best path to a destination host or network. Each path carries well-known mandatory, well-known discretionary, and optional transitive attributes that are used in BGP best-path analysis. You can influence BGP path selection by altering some of these attributes by configuring BGP policies.

BGP also supports load balancing or equal-cost multipath (ECMP).

For information on configuring BGP in an MPLS network, see the Inspur CN12700 Series INOS MPLS Configuration Guide.

10.2.1 BGP Autonomous Systems

An autonomous system (AS) is a network controlled by a single administration entity. An autonomous system forms a routing domain with one or more interior gateway protocols (IGPs) and a consistent set of routing policies. BGP supports 16-bit and 32-bit autonomous system numbers.

Separate BGP autonomous systems dynamically exchange routing information through external BGP (eBGP) peering sessions. BGP speakers within the same autonomous system can exchange routing information through internal BGP (iBGP) peering sessions.

4-Byte AS Number Support

BGP supports 2-byte or 4-byte AS numbers. Inspur INOS displays 4-byte AS numbers in plain-text notation (that is, as 32-bit integers). You can configure 4-byte AS numbers as either plain-text notation (for example, 1 to 4294967295) or AS.dot notation (for example, 1.0).

10.2.2 Administrative Distance

An administrative distance is a rating of the trustworthiness of a routing information source. By default, BGP uses the administrative distances shown in the table.

Table 18 : BGP Default Administrative Distances

Distance	**Default Value**	**Function**
External	20	Applied to routes learned from eBGP.
Internal	200	Applied to routes learned from iBGP.
Local	200	Applied to routes originatedby the router.

10.2.3 BGP Peers

A BGP speaker does not discover another BGP speaker automatically. You must configure the relationships between BGP speakers. A BGP peer is a BGP speaker that has an active TCP connection to another BGP speaker.

BGP Sessions

BGP uses TCP port 179 to create a TCP session with a peer. When a TCP connection is established between peers, each BGP peer initially exchanges all of its routes—the complete BGP routing table—with the other peer. After this initial exchange, the BGP peers send only incremental updates when a topology change occurs in the network or when a routing policy change occurs. In the periods of inactivity between these updates, peers exchange special messages called keepalives. The hold time is the maximum time limit that can elapse between receiving consecutive BGP update or keepalive messages.

Inspur INOS supports the following peer configuration options:

- Individual IPv4 or IPv4 address—BGP establishes a session with the BGP speaker that matches the remote address and AS number.
- IPv4 or IPv6 prefix peers for a single AS number—BGP establishes sessions with BGP speakers that match the prefix and the AS number.
- Dynamic AS number prefix peers—BGP establishes sessions with BGP speakers that match the prefix and an AS number from a list of configured AS numbers.

Dynamic AS Numbers for Prefix Peers

Inspur INOS accepts a range or list of AS numbers to establish BGP sessions. For example, if you configure BGP to use IPv4 prefix 192.0.2.0/8 and AS numbers 33, 66, and 99, BGP establishes a session with 192.0.2.1 with AS number 66 but rejects a session from 192.0.2.2 with AS number 50.

Inspur INOS does not associate prefix peers with dynamic AS numbers as either interior BGP (iBGP) or external BGP (eBGP) sessions until after the session is established.

10.2.4 BGP Router Identifier

To establish BGP sessions between peers, BGP must have a router ID, which is sent to BGP peers in the OPEN message when a BGP session is established. The BGP router ID is a 32-bit value that is often represented by an IPv4 address. You can configure the router ID. By default, Inspur INOS sets the router ID to the IPv4 address of a loopback interface on the router. If no loopback interface is configured on the router, the software chooses the highest IPv4 address configured to a physical interface on the router to represent the BGP router ID. The BGP router ID must be unique to the BGP peers in a network.

If BGP does not have a router ID, it cannot establish any peering sessions with BGP peers.

10.2.5 BGP Path Selection

Although BGP might receive advertisements for the same route from multiple sources, BGP selects only one path as the best path. BGP puts the selected path in the IP routing table and propagates the path to its peers.

The best-path algorithm runs each time that a path is added or withdrawn for a given network. The best-path algorithm also runs if you change the BGP configuration. BGP selects the best path from the set of valid paths available for a given network.

Beginning with Inspur INOS Release 8.4(1), the behavior of the BGP pre-best path point of insertion (POI) is changed. In this release, the INOS RPM, BGP, and HMM software uses a single cost community ID (either 128 for internal routes or 129 for external routes) to identify a BGP VPNv4 route as an EIGRP originated route.

Only the routes that have the pre-best path value set to cost community ID 128 or 129 are installed in the URIB along with the cost extcommunity. Any non-eigrp originated route carrying the above described cost community ID would be installed in URIB along with pre-best path cost community. As a result, URIB would use this cost to identify the better route between the route learnt through the iBGP and backdoor-EIGRP instead of the administrative distance.

- Inspur INOS implements the BGP best-path algorithm in the following steps:
- Compares two paths to determine which is better.
- Explores all paths and determines in which order to compare the paths to select the overall best path.
- Determines whether the old and new best paths differ enough so that the new best path should be used.

The path selection uses the BGP AS-path attribute. The AS-path attribute includes the list of autonomous system numbers (AS numbers) traversed in the advertised path. If you subdivide your BGP autonomous system into a collection or confederation of autonomous systems, the AS-path contains confederation segments that list these locally defined autonomous systems.

BGP Path Selection - Comparing Pairs of Paths

This first step in the BGP best-path algorithm compares two paths to determine which path is better. The following sequence describes the basic steps that Inspur INOS uses to compare two paths to determine the better path:

1. Inspur INOS chooses a valid path for comparison. (For example, a path that has an unreachable next hop is not valid.)
2. Inspur INOS chooses the path with the highest weight.
3. Inspur INOS chooses the path with the highest local preference.
4. If one of the paths is locally originated, Inspur INOS chooses that path.
5. Inspur INOS chooses the path with the shorter AS path.
6. Inspur INOS chooses the path with the lower origin. Interior Gateway Protocol (IGP) is considered lower than EGP.
7. Inspur INOS chooses the path with the lower multiexit discriminator (MED).

You can configure a number of options that affect whether or not this step is performed. In general, Inspur INOS compares the MED of both paths if the paths were received from peers in the same autonomous system; otherwise, Inspur INOS skips the MED comparison.

You can configure Inspur INOS to always perform the best-path algorithm MED comparison, regardless of the peer autonomous system in the paths. Otherwise, Inspur INOS performs a MED comparison that depends on the AS-path attributes of the two paths being compared:

1. If a path has no AS-path or the AS-path starts with an AS_SET, the path is internal and Inspur INOS compares the MED to other internal paths.
2. If the AS-path starts with an AS_SEQUENCE, the peer autonomous system is the first AS number in the sequence and Inspur INOS compares the MED to other paths that have the same peer autonomous system.
3. If the AS-path contains only confederation segments or starts with confederation segments

followed by an AS_SET, the path is internal and Inspur INOS compares the MED to other internal paths.

4. If the AS-path starts with confederation segments that are followed by an AS_SEQUENCE, the peer autonomous system is the first AS number in the AS_SEQUENCE and Inspur INOS compares the MED to other paths that have the same peer autonomous system.
5. If the non-deterministic MED comparison feature is enabled, the best-path algorithm uses the Inspur IOS style of MED comparison.

8. If one path is from an internal peer and the other path is from an external peer, Inspur INOS chooses the path from the external peer.
9. If the paths have different IGP metrics to their next-hop addresses, Inspur INOS chooses the path with the lower IGP metric.
10. Inspur INOS uses the path that was selected by the best-path algorithm the last time it was run.
11. If all path parameters in Step 1 through Step 9 are the same, and there is no current best path (for example, the current best path can be lost when the neighbor that offers the current best path goes down), then the route from the BGP router with the lowest router ID is chosen. If the path includes an originator attribute, Inspur INOS uses that attribute as the router ID to compare to; otherwise, Inspur INOS uses the router ID of the peer that sent the path. If the paths have different router IDs, Inspur INOS chooses the path with the lower router ID.
12. Inspur INOS selects the path with the shorter cluster length. If a path was not received with a cluster list attribute, the cluster length is 0.
13. Inspur INOS chooses the path received from the peer with the lower IP address. Locally generated paths (for example, redistributed paths) have a peer IP address of 0.

BGP Path Selection - Determining the Order of Comparisons

The second step of the BGP best-path algorithm implementation is to determine the order in which Inspur INOS compares the paths:

1. Inspur INOS partitions the paths into groups. Within each group, Inspur INOS compares the MED among all paths. Inspur INOS uses the same rule as in the section *Step 1—Comparing Pairs of Paths* to determine whether MED can be compared between any two paths. Typically, this comparison results in one group being chosen for each neighbor autonomous system. If you configure the **bgp bestpath med always** command, Inspur INOS chooses just one group that contains all the paths.
2. Inspur INOS determines the best path in each group by iterating through all paths in the group and keeping track of the best one so far. Inspur INOS compares each path with the temporary best path found so far and if the new path is better, it becomes the new temporary best path and Inspur INOS compares it with the next path in the group.
3. Inspur INOS forms a set of paths that contain the best path selected from each group in Step 2. Inspur INOS selects the overall best path from this set of paths by going through them as in Step 2.

BGP Path Selection - Determining the Best-Path Change Suppression

The next part of the implementation is to determine whether Inspur INOS uses the new best path or suppresses the new best path. The router can continue to use the existing best path if the new one is identical to the old path (if the router ID is the same). Inspur INOS continues to use the existing best path to avoid route changes in the network.

You can turn off the suppression feature by configuring the best-path algorithm to compare the router IDs. See the 〝Tuning the Best-Path Algorithm〞 section on page 11-10 for more information. If you configure this feature, the new best path is always preferred to the existing one.

You cannot suppress the best-path change if any of the following conditions occur:

- The existing best path is no longer valid.
- Either the existing or new best paths were received from internal (or confederation) peers or were locally generated (for example, by redistribution).
- The paths were received from the same peer (the paths have the same router ID).
- The paths have different weights, local preferences, origins, or IGP metrics to their next-hop addresses.
- The paths have different MEDs.

10.2.6 BGP and the Unicast RIB

BGP communicates with the unicast routing information base (unicast RIB) to store IPv4 routes in the unicast routing table. After selecting the best path, if BGP determines that the best path change needs to be reflected in the routing table, it sends a route update to the unicast RIB.

BGP receives route notifications regarding changes to its routes in the unicast RIB. It also receives route notifications about other protocol routes to support redistribution.

BGP also receives notifications from the unicast RIB regarding next-hop changes. BGP uses these notifications to keep track of the reachability and IGP metric to the next-hop addresses.

Whenever the next-hop reachability or IGP metrics in the unicast RIB change, BGP triggers a best-path recalculation for affected routes.

BGP communicates with the IPv6 unicast RIB to perform these operations for IPv6 routes.

10.2.7 BGP Prefix Independent Convergence

The BGP Prefix Independent Convergence (PIC) feature achieves subsecond convergence in the forwarding plane for BGP IP and Layer 3 VPN routes, when there are BGP next-hop network reachability failures.

BGP PIC has two categories:

- PIC core
- PIC edge

PIC core ensures fast convergence for BGP routes when there is a link or node failure in the core that causes a change in the IGP reachability to a remote BGP next-hop address.

PIC edge ensures fast convergence to a BGP backup path when an external (eBGP) edge link or an external neighbor node fails.

BGP PIC Feature Support Matrix

BGP PIC feature support matrix is shown in the table below:

BGP PIC	**IPv4 Unicast**	**IPv6 Unicast**	**VPNv4 (per prefix)**	**VPNv6 (per prefix)**	**VPNv4 (per VRF)**	**VPNv6 (per VRF)**
Core Unipath	Yes	Yes	No	No	Yes	No
Edge Unipath	Yes	Yes	No	No	No	No
Core with Multipath equal	Yes	Yes	No	No	Yes	No
Edge Multipath equal (multiple active ECMP, only one backup)	Yes	Yes	No	No	No	No

BGP PIC Core

The BGP PIC core feature is supported by Inspur INOS Release 8.4(1) and later. The feature allows for faster convergence for traffic destined to BGP prefixes that share the same remote next hop in case of a failure in the core of the network. Both MPLS and pure IP traffic can benefit from this feature. It is enabled by default and cannot be disabled.

IPv4, VPNv4, 6PE, and VPNv6 (6VPE) support PIC core with the following constraints:

- For both IP and MPLS core, convergence for internet routes is prefix-independent on the order of BGP next hops.

• With per-VRF label allocation, VPN route convergence is also prefix-independent on the order of BGP next hops. That is, when a path to a remote PE changes, the number of VRFs on that PE determines convergence.

• With per-prefix label allocation, route convergence is not prefix-independent. Convergence moves to the order of VPN routes that are advertised by a remote PE if a failure or change occurs in the reachability to that PE.

For additional considerations when using BGP PIC core in MPLS networks, see the *Inspur CN12700 Series INOS MPLS Configuration Guide.*

BGP PIC Edge

The BGP PIC for Edge feature improves BGP convergence after a network failure. This convergence is applicable to edge failures in an IP network. The BGP PIC Edge feature creates and stores a backup path in the routing information base (RIB) and forwarding information base (FIB) so that when a failure on an eBGP link to SP is detected (the primary path fails), the backup path can immediately take over, enabling fast fail over in the forwarding plane.

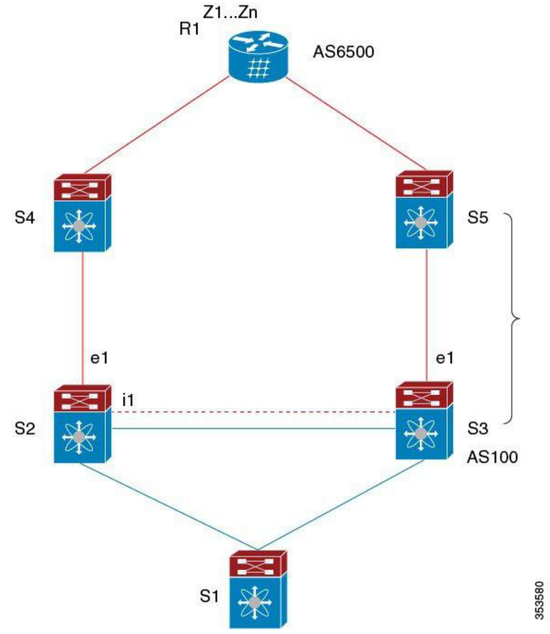
If BGP PIC edge is configured, BGP calculates an additional second best-path (the backup path) along with the primary best-path. BGP installs both best and backup paths for the prefixes with PIC support into the BGP RIB. BGP also downloads the backup path along with the RNH via APIs to the URIB, which then updates the FIB with the next hop marked as a backup. The backup path provides a fast reroute mechanism to counter a singular network failure.

This feature detects both the local interface failure and remote interface/link failure and triggers the use of the backup path.

BGP PIC Edge Unipath

A BGP PIC edge unipath topology is shown in the figure below:

Figure 32 : BGP PIC Edge Unipath



In the above figure:

- eBGP sessions are between S2-S4 and S3-S5
- iBGP session is between S2-S3
- Traffic from S1 uses S2 and uses the e1 interface to reach prefixes Z1..Zn.
- S2 has two paths to reach Z1...Zn
    - Primary path via S4

- Backup/alternate via S5

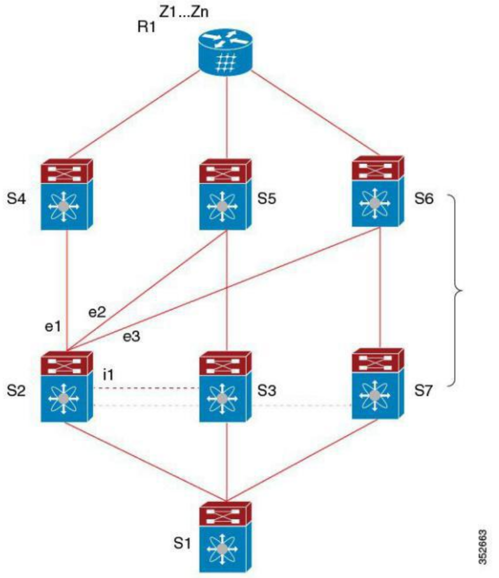
In this example, S3 advertises to S2 the prefixes Z1...Zn to reach with itself as the next hop. BGP on S2, with BGP PIC feature enabled, installs both bestpath (via S4) and backup path (via S3/S5) towards the AS6500 into the RIB and then the RIB downloads both routes to the FIB.

When the S2-S4 link goes down, the FIB on S2 detects the link failure. It automatically switches from the primary path to the backup/alternate and points to the new next hop S3. Traffic is quickly rerouted due to the local fast re-convergence in FIB. After learning the link failure event, BGP on S2 recomputes the bestpath (which is the previous backup path), removing the next hop S4 from RIB and reinstalling S3 as the primary next hop into RIB. It also computes a new backup/alternate path, if any, and notifies RIB. With the support of the BGP PIC feature, the FIB can switch to the available backup route instantly upon detection of link failure on the primary route without waiting for BGP to select new bestpath and converge, and achieve a fast reroute.

BGP PIC Edge with Multipaths

In the presence of Equal Cost Multipath (ECMP), none of the multipaths can be selected as the backup path when BGP PIC Edge support is enabled.

Figure 33 : BGP PIC Edge with Multipaths



In the above topology, there are six paths for a given prefix as follows:

- eBGP paths: e1, e2, e3
- iBGP paths: i1, i2, i3

The order of preference is e1 > e2 > e3 > i1 > i2 > i3. The potential multipath situations are:

No multipaths configured

- bestpath = e1
- multipath-set = []
- backup path = e2
- PIC behavior: When e1 fails, e2 is activated.

Two-way eBGP multipaths configured:

- bestpath = e1
- multipath-set = [e1, e2]
- backup path = e3

- PIC behavior: Active multipaths are mutually backed up. When all multipaths fail, e3 is activated.

Three-way eBGP multipaths configured:

- bestpath = e1
- multipath-set = [e1, e2, e3]
- backup path = i1
- PIC behavior: Active multipaths are mutually backed up. When all multipaths fail, i1 is activated.

Four-way eiBGP multipaths configured:

- bestpath = e1
- multipath-set = [e1, e2, e3, i1]
- backup path = i2
- PIC behavior: Active multipaths are mutually backed up. When all multipaths fail, i2 is activated.

10.2.8 BGP Virtualization

BGP supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

10.3 Licensing Requirements for Basic BGP

The following table shows the licensing requirements for this feature:

Product	License Requirements
Inspur INOS	BGP requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *Inspur INOS Licensing Guide*.

10.4 Prerequisites for BGP

BGP has the following prerequisites:

- You must enable BGP.
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must configure at least one IGP that is capable of recursive next-hop resolution.
- You must configure an address family under a neighbor for the BGP session establishment.

10.5 Guidelines and Limitations for BGP

BGP has the following configuration guidelines and limitations:

- Inspur INOS does not support "fast-external-fallover" for the multi-hop eBGP peering. The BGP differentiates the single-hop (directly connected) and the multi-hop eBGP neighbors using the **ebgp-multihop** command. When you use the **ebgp-multihop 2** command for an eBGP peer, the BGP treats it as multi-hop session and does not trigger the "fast-external-fallover". This is a known behaviour.
- The dynamic AS number prefix peer configuration overrides the individual AS number configuration inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.

- BGP sessions created through a dynamic AS number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.
- Use the maximum-prefix configuration option per peer to restrict the number of routes received and system resources used.
- Configure the update source to establish a session with BGP/eBGP multihop sessions.
- Specify a BGP policy if you configure redistribution.
- Define the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, you might experience BGP session flaps.
- The BGP minimum route advertisement interval (MRAI) value for all iBGP and eBGP sessions is zero and is not configurable.
- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur INOS Virtual Device Context Configuration Guide*).
- If you configure VRFs, install the Advanced Services license and enter the desired VRF.
- The BGP Prefix-Independent Convergence (PIC) Edge feature only supports IPv4 address family.
- Only one repair path (backup path) is supported with the BGP PIC Edge feature.

10.6 Default Settings

Table 19 : Default BGP Parameters

Parameters	**Default**
BGP feature	Disabled
Keep alive interval	60 seconds
Hold timer	180 seconds
BGP PIC core	Enabled
BGP PIC edge	Disabled
Auto-summary	Always disabled
Synchronization	Always disabled

10.7 CLI Configuration Modes

The following sections describe how to enter each of the CLI configuration modes for BGP. From a mode, you can enter the ? command to display the commands available in that mode.

10.7.1 Global Configuration Mode

Use global configuration mode to create a BGP process and configure advanced features such as AS confederation and route dampening.

This example shows how to enter router configuration mode:

```
switch# configuration
switch(config)# router   bgp   64496
switch(config-router)#
```

BGP supports Virtual Routing and Forwarding (VRF). You can configure BGP within the appropriate VRF if you are using VRFs in your network.

This example shows how to enter VRF configuration mode:

```
switch(config)# router   bgp   64497
switch(config-router)#  vrf  vrf_A
switch(config-router-vrf)#
```

10.7.2 Address Family Configuration Mode

You can optionally configure the address families that BGP supports. Use the **address-family** command in router configuration mode to configure features for an address family. Use the **address-family** command in neighbor configuration mode to configure the specific address family for the neighbor.

You must configure the address families if you are using route redistribution, address aggregation, load balancing, and other advanced features.

The following example shows how to enter address family configuration mode from the router configuration mode:

```
switch(config)# router bgp 64496

switch(config-router)# address-family ipv6 unicast

switch(config-router-af)#
```

The following example shows how to enter VRF address family configuration mode if you are using VRFs:

```
switch(config)# router bgp 64497

switch(config-router)# vrf vrf_A

switch(config-router-vrf)# address-family ipv6 unicast

switch(config-router-vrf-af)#
```

10.7.3 Neighbor Configuration Mode

Inspur INOS provides the neighbor configuration mode to configure BGP peers. You can use neighbor configuration mode to configure all parameters for a peer.

The following example shows how to enter neighbor configuration mode:

```
switch(config)# router bgp 64496

switch(config-router)# neighbor 192.0.2.1

switch(config-router-neighbor)#
```

The following example shows how to enter VRF neighbor configuration mode:

```
switch(config)# router bgp 64497

switch(config-router)# vrf vrf_A

switch(config-router-vrf)#neighbor 192.0.2.1
switch(config-router-vrf-neighbor)#
```

10.7.4 Neighbor Address Family Configuration Mode

An address family configuration submode inside the neighbor configuration submode is available for entering address family-specific neighbor configuration and enabling the address family for the neighbor. Use this mode for advanced features such as limiting the number of prefixes allowed for this neighbor and removing private AS numbers for eBGP.

The following example shows how to enter neighbor address family configuration mode:

```
switch(config)# router bgp 64496

switch(config-router # neighbor 192.0.2.1
switch(config-router-neighbor)# address-family ipv4 unicast

switch(config-router-neighbor-af)#
```

This example shows how to enter VRF neighbor address family configuration mode:

```
switch(config)# router bgp 64497

switch(config-router)# vrf vrf_A

switch(config-router-vrf)# neighbor 209.165.201.1

switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

10.8 Configuring Basic BGP

To configure a basic BGP, you must enable BGP and configure a BGP peer. Configuring a basic BGP network consists of a few required tasks and many optional tasks. You must configure a BGP routing process and BGP peers.

10.8.1 Enabling BGP

You must enable BGP before you can configure BGP.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature bgp**
3. (Optional) switch(config)# **show feature**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature bgp**	Enables BGP. Use the **no feature bgp** command to disable BGP and remove all associated configuration.
Step 3	(Optional) switch(config)# **show feature**	(Optional) Displays enabled and disabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

10.8.2 Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance. Inspur INOS supports 2-byte or 4-byte autonomous system (AS) numbers in plain-text notation or as.dot notation.

Before you begin

- You must enable BGP.
- BGP must be able to obtain a router ID (for example, a configured loopback address).
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp***autonomous-system-number*
3. switch(config-router)# **router-id** *ip-address*
4. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}
5. switch(config-router-af)# **network ip-prefix** [**route-map** *map-name*]
6. switch(config-router-af)# **show bgp all**
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp***autonomous-system-number*	Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. Use the **no** form of this command to disable this feature.
Step 3	switch(config-router)# **router-id** *ip-address*	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker.
Step 4	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters global address family configuration mode for the IP or VPN address family.
Step 5	switch(config-router-af)# **network ip-prefix** [**route-map** *map-name*]	(Optional) Specifies a network as local to this autonomous system and adds it to the BGP routing table. For exterior protocols, the network command controls which networks are advertised. Interior protocols use the **network** command to determine where to send updates.
Step 6	switch(config-router-af)# **show bgp all**	(Optional) Displays information about all BGP address families.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable BGP with the IPv4 unicast address family and manually add one network to advertise:

```
switch# configure terminal

switch(config)# router bgp 64496

switch(config-router)#   address-family   ipv4 unicast

switch(config-router-af)# network 192.0.2.0

switch(config-router-af)# copy running-config startup-config
```

10.8.3 Restarting a BGP Instance

You can restart a BGP instance and clear all peer sessions for the instance.

To restart a BGP instance and remove all associated peers, use the following command:

SUMMARY STEPS

1. switch(config)# **restart bgp** *instance-tag*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config)# **restart bgp** *instance-tag*	Restarts the BGP instance and resets or reestablishes all peering sessions.

10.8.4 Shutting Down BGP

You can shut down the BGP protocol and gracefully disable BGP and retain the configuration. To shut down BGP, use the following command in router configuration mode:

SUMMARY STEPS

1. switch(config-router)# **shutdown**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **shutdown**	Gracefully shuts down BGP.

10.8.5 Configuring BGP Peers

You can configure a BGP peer within a BGP process. Each BGP peer has an associated keepalive timer and hold timers. You can set these timers either globally or for each BGP peer. A peer configuration overrides a global configuration.

Before you begin

You must enable BGP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *autonomous-system-number*
3. switch(config-router)# **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*
4. switch(config-router-neighbor)# **description** *text*

5. switch(config-router-neighbor)# **timers** *keepalive-time hold-time*
6. switch(config-router-neighbor)# **shutdown**
7. switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}
8. switch(config-router-neighbor)# **weight** *value*
9. (Optional) switch(config-router-neighbor)# **show bgp** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**} **neighbors**
10. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 3	switch(config-router)# **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*	Configures the IPv4 or IPv6 address and AS number for a remote BGP peer. The ip-address format is x.x.x.x. The ipv6-address format is A:B::C:D.
Step 4	switch(config-router-neighbor)# **description** *text*	(Optional) Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.
Step 5	switch(config-router-neighbor)# **timers** *keepalive-time hold-time*	(Optional) Adds the keepalive and hold time BGP timer values for the neighbor. The range is from 0 to 3600 seconds. The default is 60 seconds for the keepalive time and 180 seconds for the hold time.
Step 6	switch(config-router-neighbor)# **shutdown**	(Optional). Administratively shuts down this BGP neighbor. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 7	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters neighbor address family configuration mode for the unicast IPv4 address family.
Step 8	switch(config-router-neighbor)# **weight** *value*	(Optional) Sets the default weight for routes from this neighbor. The range is from 0 to 65535. All routes learned from this neighbor have the assigned weight initially. The route with the highest weight is chosen as the preferred route when multiple routes are available to a particular network. The weights assigned with the **set weight route-map** command override the weights assigned with this command. If you specify a BGP peer policy template, all the members of the template inherit the characteristics configured with this command.
Step 9	(Optional) switch(config-router-neighbor)# **show bgp** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**} **neighbors**	(Optional) Displays information about BGP peers.
Step 10	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure a BGP peer:

```
switch# configure terminal

switch(config)# router bgp 64496

switch(config-router)#      neighbor      192.0.2.1   remote-as   64497
switch(config-router-neighbor)#    description     Peer     Router     B
switch(config-router-neighbor)#    address-family     ipv4     unicast
switch(config-router-neighbor)# weight 100

switch(config-router-neighbor-af)# copy running-config startup-config
```

10.8.6 Configuring AS-4 Dot Notation

You can configure 4-byte autonomous system (AS) numbers in asdot notation. The default value is asplain.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **as-format asdot**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **as-format asdot**	Configures the ASN notation to asdot.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example configures AS numbers in asdot notation.

```
switch # configure terminal

switch (config) # as-format asdot

switch (config) # copy running-config startup-config
```

10.8.7 Configuring Dynamic AS Numbers for Prefix Peers

You can configure multiple BGP peers within a BGP process. You can limit BGP session establishment to a single AS number or multiple AS numbers in a route map.

BGP sessions configured through dynamic AS numbers for prefix peers ignore the **ebgp-multihop** command and the **disable-connected-check** command.

You can change the list of AS numbers in the route map, but you must use the no neighbor command to change the route-map name. Changes to the AS numbers in the configured route map affect only new sessions.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *autonomous-system-number*
3. switch(config-router)# **neighbor** *prefix* **remote-as route-map** *map-name*
4. switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **neighbors**
5. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the AS number to the local BGP speaker. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.
Step 3	switch(config-router)# **neighbor** *prefix* **remote-as route-map** *map-name*	Configures the IPv4 or IPv6 prefix and a route map for the list of accepted AS numbers for the remote BGP peers. The prefix format for IPv4 is x.x.x.x/length. The length range is from 1 to 32. The prefix format for IPv6 is A:B::C:D/length. The length range is from 1 to 128. The *map-name* can be any case-sensitive, alphanumeric string up to 63 characters.
Step 4	switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **neighbors**	(Optional) Displays information about BGP peers.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure dynamic AS numbers for a prefix peer:

```
switch# configure terminal

switch(config)# route-map BGPPeers

switch(config-route-map)# match as-number 64496, 64501-64510
switch(config-route-map)#match as-number as-path-list List1, List2
switch(config-route-map)# exit

switch(config)# router bgp 64496

switch(config-router)# neighbor 192.0.2.0/8 remote-as route-map BGPPeers

switch(config-router-neighbor)# description Peer Router B
switch(config-router-neighbor)#     address-family     ipv4   unicast
switch(config-router-neighbor-af)#  copy  running-config startup-config
```

10.8.8 Configuring BGP PIC Edge

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Step 1 Enter configuration mode:
switch#**configure terminal**

Step 2 Enable BGP and assign the autonomous system number to the local BGP speaker:
switch(config)# **router bgp** *autonomous-system-number*

Step 3 Enter router address family configuration mode for the IPv4 unicast address family:
switch(config-router)# **address-family ipv4 unicast**

Step 4 Enable BGP to install the backup path to the routing table:
switch(config-router-af)# **additional-paths install backup**

Step 5 Exit router address family configuration mode:
switch(config-router-af)# **exit**

Example

This example shows how to configure the device to support BGP PIC Edge in IPv4 network:

```
interface Ethernet2/2 ip address 1.1.1.5/24 no shutdown

interface Ethernet2/3 ip address 2.2.2.5/24 no shutdown

router bgp 100

address-family ipv4 unicast additional-paths install backup

neighbor 1.1.1.6 remote-as 200 address-family ipv4 unicast

neighbor 2.2.2.6 remote-as 100 address-family ipv4 unicast
```

If BGP receives the same prefix (for example, 99.0.0.0/24) from the two neighbors 1.1.1.6 and 2.2.2.6, both paths will be installed in the URIB—one as the primary path and the other as the backup path.

BGP output:

```
switch(config)# show ip bgp 99.0.0.0/24

BGP  routing  table  information  for  VRF  default,  address
family IPv4 Unicast BGP routing table entry for 99.0.0.0/24,
version 4

Paths: (2 available, best #2)

Flags: (0x00001a) on xmit-list, is in urib, is best urib
route

  Path type: internal, path is valid, not best reason: Internal
  path, backup path AS-Path: 200 , path sourced external to AS

    2.2.2.6 (metric 0) from 2.2.2.6 (2.2.2.6)

      Origin IGP, MED not set, localpref 100, weight 0

  Advertised path-id 1
  Path  type:  external,  path  is
  valid, is best path AS-Path: 200 ,
  path sourced external to AS

    1.1.1.6 (metric 0) from 1.1.1.6 (99.0.0.1)

      Origin IGP, MED not set, localpref 100, weight 0

  Path-id          1
    advertised     to
    peers: 2.2.2.6
```

URIB output:

```
URIB output:

switch(config)# show ip route 99.0.0.0/24

IP Route Table for VRF
"default"  '*'  denotes
best   ucast   next-hop
'**' denotes best mcast
next-hop        '[x/y]'
denotes
[preference/metric]

'%<string>' in via output denotes VRF <string>

99.1.1.1/24, ubest/mbest: 1/0
    *via 1.1.1.6, [20/0], 14:34:51, bgp-100, external, tag 200

     via 2.2.2.6, [200/0], 14:34:51, bgp-100, internal, tag 200 (backup)
```

UFIB output:

```
switch# show forwarding route 123.1.1.0 detail
module 8

Prefix 123.1.1.0/24, No of paths: 1, Update time: Fri Feb  7 19:00:12 2014

   Vobj id: 141   orig_as: 65002   peer_as: 65100  rnh: 10.3.0.2

   10.4.0.2           Ethernet8/4         DMAC: 0018.bad8.4dfd

    packets: 2             bytes: 3484   Repair path    10.3.0.2

    Ethernet8/3 DMAC: 0018.bad8.4dfd

    packets: 0             bytes: 1
```

10.8.9 Clearing BGP Information

To clear BGP information, use the following commands:

Command	Purpose

clear bgp all {*neighbor* | * | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors from all address families. * clears all neighbors in all address families. The arguments are as follows: • *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all dampening [**vrf** *vrf-name*]	Clears route flap dampening networks in all address families. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all flap-statistics [**vrf** *vrf-name*]	Clears route flap statistics in all address families. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **dampening** [**vrf** *vrf-name*]	Clears route flap dampening networks in the selected address family. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **flap-statistics** [**vrf** *vrf-name*]	Clears route flap statistics in the selected address family. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.

clear bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {*neighbor* |* | *as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors from the selected address family. * clears all neighbors in the address family. The arguments are as follows: • *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp {**ip** {**unicast** | **multicast**}} {*neighbor* |* |*as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows: • *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-numbe*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp dampening [*ip-neighbor* |*ip-prefix*] [**vrf** *vrf-name*]	Clears route flap dampening in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

clear bgp flap-statistics [*ip-neighbor* |*ip-prefix*] [**vrf** *vrf-name*]	Clears route flap statistics in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp {**ip** {**unicast** | **multicast**}} {*neighbor* |* |*as-number* | **peer-template** *name* | *prefix*} [**vrf** *vrf-name*]	• *neighbor*—IPv4 or IPv6 address of a neighbor. • *as-number*— Autonomous system number. The AS number can be a 16-bit integer or a 32-bit integer in the form of higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format. • *name*—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • *prefix*—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp dampening [*ip-neighbor* |*ip-prefix*] [**vrf** *vrf-name*]	Clears route flap dampening in one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip mbgp flap-statistics [*ip-neighbor* |*ip-prefix*] [**vrf** *vrf-name*]	Clears route flap statistics one or more networks. The arguments are as follows: • *ip-neighbor*—IPv4 address of a neighbor. • *ip-prefix*—IPv4. All neighbors within that prefix are cleared. • *vrf-name*—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

10.9 Verifying the Basic BGP Configuration

To display the BGP configuration, perform one of the following tasks:

Command	Purpose
show bgp all [**summary**] [**vrf** *vrf-name*]	Displays the BGP information for all address families.
show bgp convergence [**vrf** *vrf-name*]	Displays the BGP information for all address families.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **community** [**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf***vrf-name*]	Displays the BGP routes that match a BGP community.
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP community list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **extcommunity** [**regexp** *expression* | [**generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **extcommunity-list** *list-name* [**exact-match**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* {**dampening** **dampened-paths** [**regexp** *expression*]} [**vrf***vrf-name*]	Displays the information for BGP route dampening. Use the **clear bgp dampening** command to clear the route flap dampening information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	Displays the BGP route history paths.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix* **filter-list** *list-name* [**vrf** *vrf-name*]	Displays the information for the BGP filter list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	Displays the information for the BGP route next hop.
show bgp paths	Displays the BGP path information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	Displays the BGP policy information. Use the **clear bgp policy** command to clear the policy information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the prefix list.

show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	Displays the BGP routes that match the AS_path regular expression.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the route map.
show bgp peer-policy *name* [**vrf** *vrf-name*]	Displays the information about BGP peer policies.
show bgp peer-session *name* [**vrf** *vrf-name*]	Displays the information about BGP peer sessions.
show bgp peer-template *name* [**vrf** *vrf-name*]	Displays the information about BGP peer templates. Use the **clear bgp peer-template** command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show {**ipv** | **ipv6**} **bgp options**	Displays the BGP status and configuration information. This command has multiple options. See the *Inspur CN12700 Series INOS Unicast Routing Command Reference*, for more information.
show {**ipv** | **ipv6**} **mbgp options**	Displays the BGP status and configuration information. This command has multiple options. See the *Inspur CN12700 Series INOS Unicast Routing Command Reference*, for more information.
show running-configuration bgp	Displays the current running BGP configuration.

10.10 Monitoring BGP Statistics

To display BGP statistics, use the following commands:

Command	**Purpose**
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	Displays the BGP route flap statistics. Use the **clear bgp flap-statistics** command to clear these statistics.
show bgp sessions [**vrf** *vrf-name*]	Displays the BGP sessions for all peers. Use the **clear bgp sessions** command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

10.11 Configuration Examples for Basic BGP

This example shows a basic BGP configuration:

```
switch (config) # feature bgp

switch (config) # router bgp 64496
switch (config-router) # neighbor 2001:ODB8:0:1::55 remote-as 64496
switch  (config-router)  #  address-family  ipv6 unicast
switch (config-router-af) # next-hop-self
```

This example shows a basic BGP configuration:

```
switch (config) # address-family
switch (config) # router bgp 64496
switch  (config-router)  #  address-family  ipv4 unicast
switch  (config-router)  #  network  1.1.10  mask 255.255.255.0
switch   (config-router)   #  neighbor  10.1.1.1 remote-as 64496
switch  (config-router)  #  address-family  ipv4 unicast
```

10.12 Related Documents for Basic BGP

Related Topics	**Document Title**
BGP CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
MPLS configuration	*Inspur CN12700 Series INOS MPLS Configuration Guide*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide, Release* 8.4(1)

10.13 MIBs

MIBs	**MIBs Link**
BGP4-MIB INSPUR-BGP4-MIB INSPUR-BGP-MIBv2	-

10.14 Feature History for BGP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 20 : Feature History for BGP

Feature Name	**Releases**	**Feature Information**
ECMP	8.4(1)	Added support for up to 64 paths to a destination. Supported on F3-Series I/O module.

BGP PIC Edge	8.4(1)	Introduced this feature.
BGP	8.4(1)	Added support for INSPUR-BGP-MIBv2
4-byte AS number	8.4(1)	Added the ability to configure 4-byte AS numbers in asdot notation.
BGP	8.4(1)	Added support for additional BGP paths.
BGP	8.4(1)	Added the ability to set the default weigh for routes from a neighbor using the **weight** command in the neighbor address family configuration mode.
BGP	8.4(1)	Added support for the BGP PIC core feature.
VPN address families	8.4(1)	Added support for VPN address families.
BFD	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide, Release* 8.4(1) for more information.
ISSU	8.4(1)	Lowered BGP minimum hold-time check to eight seconds.
IPv6	8.4(1)	Added support for IPv6.
4-Byte AS numbers	8.4(1)	Added support for 4-byte AS numbers in plaintext notation.
Conditional advertisement	8.4(1)	Added support for conditionally advertising BGP routes based on the existence of other routes in the BGP table.
Dynamic AS number for prefix peers	8.4(1)	Added support for a range of AS numbers for BGP prefix peer configuration.
BGP	8.4(1)	This feature was introduced.

CHAPTER 11 Configuring Advanced BGP

This chapter contains the following sections:

- Finding Feature Information.
- Information About Advanced BGP.
- Licensing Requirements for Advanced BGP.
- Prerequisites for Advanced BGP.
- Guidelines and Limitations for Advanced BGP.
- Default Settings.
- Configuring Advanced BGP.
- Verifying the Advanced BGP Configuration.
- Displaying Advanced BGP Statistics.
- Related Documents.
- RFCs.
- MIBs.
- Feature History for Advanced BGP .

11.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

11.2 Information About Advanced BGP

BGP is an interdomain routing protocol that provides loop-free routing between organizations or autonomous systems. Inspur INOS supports BGP version 4. BGP version 4 includes multiprotocol extensions that allow BGP to carry routing information for IP multicast routes and multiple Layer 3 protocol address families. BGP uses TCP as a reliable transport protocol to create TCP sessions with other BGP-enabled devices called BGP peers. When connecting to an external organization, the router creates external BGP (eBGP) peering sessions. BGP peers within the same organization exchange routing information through internal BGP (iBGP) peering sessions.

11.2.1 Peer Templates

BGP peer templates allow you to create blocks of common configuration that you can reuse across similar BGP peers. Each block allows you to define a set of attributes that a peer then inherits. You can choose to override some of the inherited attributes as well, making it a very flexible scheme for simplifying the repetitive nature of BGP configurations.

Inspur INOS implements three types of peer templates:

- The peer-session template defines BGP peer session attributes, such as the transport details, remote autonomous system number of the peer, and session timers. A peer-session template can also inherit attributes from another peer-session template (with locally defined attributes that override the attributes from an inherited peer-session).
- A peer-policy template defines the address-family dependent policy aspects for a peer including the inbound and outbound policy, filter-lists, and prefix-lists. A peer-policy template can inherit from a set of peer-policy templates. Inspur INOS evaluates these peer-policy templates in the order specified by the preference value in the inherit configuration. The lowest number is preferred over higher numbers.
- The peer template can inherit the peer-session and peer-policy templates to allow for simplified peer definitions. It is not mandatory to use a peer template but it can simplify the BGP configuration by providing reusable blocks of configuration.

11.2.2 Authentication

You can configure authentication for a BGP neighbor session. This authentication method adds an MD5 authentication digest to each TCP segment sent to the neighbor to protect BGP against unauthorized messages and TCP security attacks.

11.2.3 Route Policies and Resetting BGP Sessions

You can associate a route policy to a BGP peer. Route policies use route maps to control or modify the routes that BGP recognizes. You can configure a route policy for inbound or outbound route updates. The route policies can match on different criteria, such as a prefix or AS_path attribute, and selectively accept or deny the routes. Route policies can also modify the path attributes.

When you change a route policy applied to a BGP peer, you must reset the BGP sessions for that peer. Inspur INOS supports the following three mechanisms to reset BGP peering sessions:

- Hard reset—A hard reset tears down the specified peering sessions, including the TCP connection, and deletes routes coming from the specified peer. This option interrupts packet flow through the BGP network. Hard reset is disabled by default.
- Soft reconfiguration inbound—A soft reconfiguration inbound triggers routing updates for the specified peer without resetting the session. You can use this option if you change an inbound route policy. Soft reconfiguration inbound saves a copy of all routes received from the peer before processing the routes through the inbound route policy. If you change the inbound route policy, Inspur INOS passes these stored routes through the modified inbound route policy to update the route table without tearing down existing peering sessions. Soft reconfiguration inbound can use significant memory resources to store the unfiltered BGP routes. Soft reconfiguration inbound is disabled by default.
- Route Refresh—A route refresh updates the inbound routing tables dynamically by sending route refresh requests to supporting peers when you change an inbound route policy. The remote BGP peer responds with a new copy of its routes that the local BGP speaker processes with the modified route policy. Inspur INOS automatically sends an outbound route refresh of prefixes to the peer.
- BGP peers advertise the route refresh capability as part of the BGP capability negotiation when establishing the BGP peer session. Route refresh is the preferred option and enabled by default.

BGP also uses route maps for route redistribution, route aggregation, route dampening, and other features.

11.2.4 eBGP

External BGP (eBGP) allows you to connect BGP peers from different autonomous systems to exchange routing updates. Connecting to external networks enables traffic from your network to be forwarded to other networks and across the Internet.

You should use loopback interfaces for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface flap occurs when the interface is administratively brought up or down because of a failure or maintenance issue.

BGP Next Hop Unchanged

In an eBGP session, by default, the router changes the next-hop attribute of a BGP route to its own address when the router sends out a route. The BGP next-hop unchanged feature allows BGP to send an update to an eBGP multihop peer with the next-hop attribute unchanged.

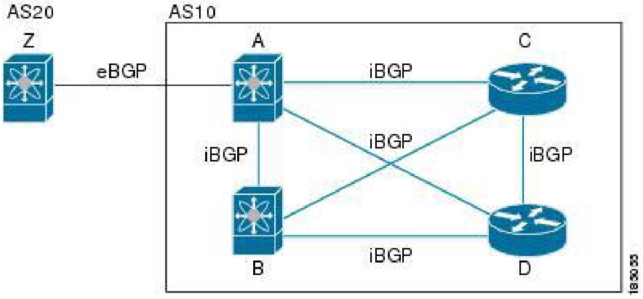
By default, BGP puts itself as the next hop when announcing to an eBGP peer. When you enter the **set ip next-hop unchanged** command for an outbound route map that is configured for an eBGP peer, it propagates the received next hop to the eBGP peer.

11.2.5 iBGP

Internal BGP (iBGP) allows you to connect BGP peers within the same autonomous system. You can use iBGP for multihomed BGP networks (networks that have more than one connection to the same external autonomous system).

The figure shows an iBGP network within a larger BGP network.

Figure 34 : iBGP Network



iBGP networks are fully meshed. Each iBGP peer has a direct connection to all other iBGP peers to prevent network loops.

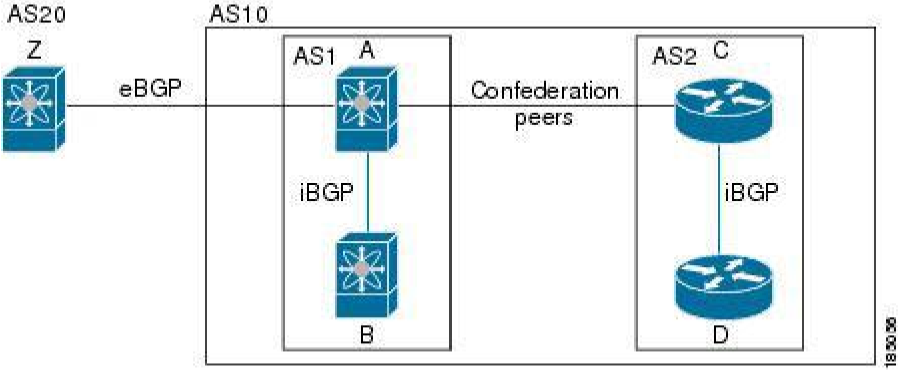
For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

AS Confederations

A fully meshed iBGP network becomes complex as the number of iBGP peers grows. You can reduce the iBGP mesh by dividing the autonomous system into multiple subautonomous systems and grouping them into a single confederation. A confederation is a group of iBGP peers that use the same autonomous system number to communicate to external networks. Each subautonomous system is fully meshed within itself and has a few connections to other subautonomous systems in the same confederation.

The figure shows the BGP network, split into two subautonomous systems and one confederation.

Figure 35 : AS Confederation



In this example, AS10 is split into two subautonomous systems, AS1 and AS2. Each subautonomous system is fully meshed, but there is only one link between the subautonomous systems. By using AS confederations, you can reduce the number of links compared to the fully meshed autonomous system.

Route Reflector

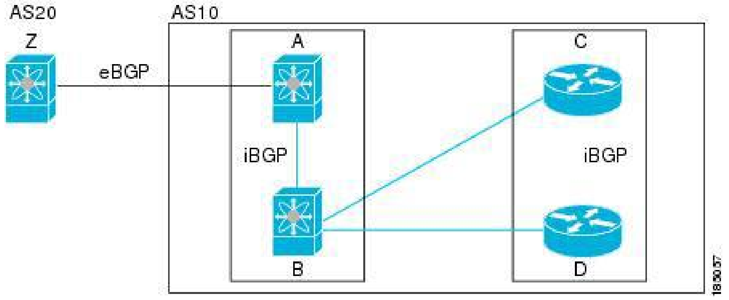
You can alternately reduce the iBGP mesh by using a route reflector configuration where route reflectors pass learned routes to neighbors so that all iBGP peers do not need to be fully meshed.

The figure below shows a simple iBGP configuration with four meshed iBGP speakers (routers A,B,C, and D.) Without these route reflectors, when router A receives a route from an external neighbor, it advertise the route to all three iBGP neighbors.

When you configure an iBGP peer to be a route reflector, it becomes responsible for passing iBGP learned routes to a set of iBGP neighbors.

In the figure, router B is the route reflector. When the route reflector receives routes advertised from router A, it advertises (reflects) the routes to routers C and D. Router A no longer has to advertise to both routers C and D.

Figure 36 : Route Reflector



The route reflector and its client peers form a cluster. You do not have to configure all iBGP peers to act as client peers of the route reflector. You must configure any nonclient peer as fully meshed to guarantee that complete BGP updates reach all peers.

11.2.6 Capabilities Negotiation

A BGP speaker can learn about BGP extensions that are supported by a peer by using the capabilities negotiation feature. Capabilities negotiation allows BGP to use only the set of features supported by both BGP peers on a link.

If a BGP peer does not support capabilities negotiation, Inspur INOS attempts a new session to the peer without capabilities negotiation if you have configured the address family as IPv4. Any other multiprotocol configuration (such as IPv6) requires capabilities negotiation.

11.2.7 Route Dampening

Route dampening is a BGP feature that minimizes the propagation of flapping routes across an internetwork. A route flaps when it alternates between the available and unavailable states in rapid succession.

For example, consider a network with three BGP autonomous systems: AS1, AS2, and AS3. Suppose that a route in AS1 flaps (it becomes unavailable). Without route dampening, AS1 sends a withdraw message to AS2. AS2 propagates the withdrawal message to AS3. When the flapping route reappears, AS1 sends an advertisement message to AS2, which sends the advertisement to AS3. If the route repeatedly becomes unavailable, and then available, AS1 sends many withdrawal and advertisement messages that propagate through the other autonomous systems.

Route dampening can minimize flapping. Suppose that the route flaps. AS2 (in which route dampening is enabled) assigns the route a penalty of 1000. AS2 continues to advertise the status of the route to neighbors. Each time that the route flaps, AS2 adds to the penalty value. When the route flaps so often that the penalty exceeds a configurable suppression limit, AS2 stops advertising the route, regardless of how many times that it flaps. The route is now dampened.

The penalty placed on the route decays until the reuse limit is reached. At that time, AS2 advertises the route again. When the reuse limit is at 50 percent, AS2 removes the dampening information for the route.

11.2.8 Load Sharing and Multipath

BGP can install multiple equal-cost eBGP or iBGP paths into the routing table to reach the same destination prefix. Traffic to the destination prefix is then shared across all the installed paths.

The BGP best-path algorithm considers the paths as equal-cost paths if the following attributes are identical:

- Weight
- Local preference
- AS_path
- Origin code
- Multi-exit discriminator (MED)
- IGP cost to the BGP next hop

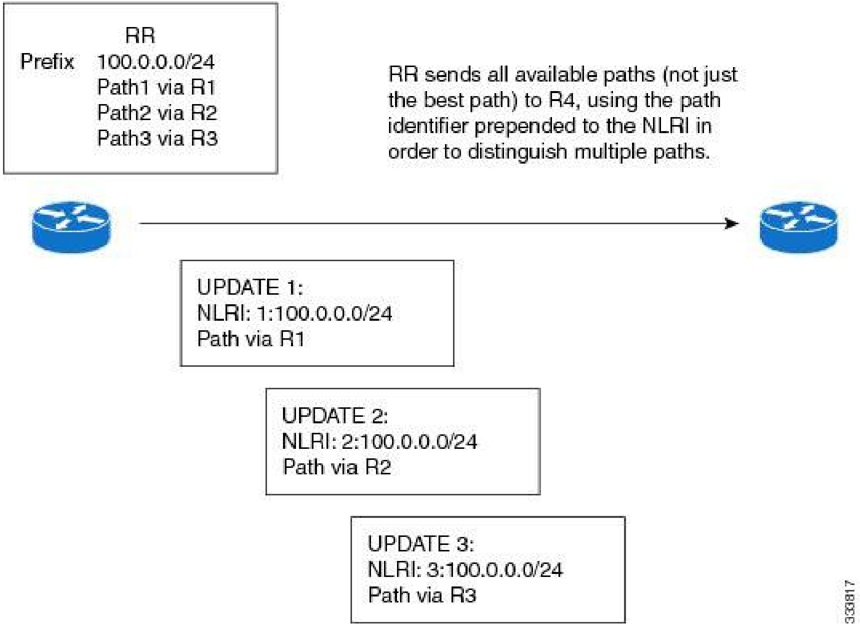
In Inspur INOS releases prior to 6.1, BGP selects only one of these multiple paths as the best path and advertises the path to the BGP peers. Beginning with Inspur INOS Release 8.4(1), BGP supports sending and receiving multiple paths per prefix and advertising such paths.

11.2.9 BGP Additional Paths

In Inspur INOS releases prior to 6.1, only one BGP best path is advertised, and the BGP speaker accepts only one path for a given prefix from a given peer. If a BGP speaker receives multiple paths for the same prefix within the same session, it uses the most recent advertisement.

Beginning with Inspur INOS Release 8.4(1), BGP supports the additional paths feature, which allows the BGP speaker to propagate and accept multiple paths for the same prefix without the new paths replacing any previous ones. This feature allows BGP speaker peers to negotiate whether they support advertising and receiving multiple paths per prefix and advertising such paths. A special 4-byte path ID is added to the network layer reachability information (NLRI) to differentiate multiple paths for the same prefix sent across a peer session. The following figure illustrates the BGP additional paths capability.

Figure 37 : BGP Route Advertisement with the Additional Paths Capability



11.2.10 Route Aggregation

You can configure aggregate addresses. Route aggregation simplifies route tables by replacing a number of more specific addresses with an address that represents all the specific addresses. For example, you can replace these three more specific addresses, 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one aggregate address, 10.1.0.0/16.

Aggregate prefixes are present in the BGP route table so that fewer routes are advertised.

Route aggregation can lead to forwarding loops. To avoid this problem, when BGP generates an advertisement for an aggregate address, it automatically installs a summary discard route for that aggregate address in the local routing table. BGP sets the administrative distance of the summary discard to 220 and sets the route type to discard. BGP does not use discard routes for next-hop resolution.

Summary entry is created in the BGP table when **aggregate-address** command is configured, though it will not be eligible for advertisement until a subset of the aggregate is found in the table.

11.2.11 BGP Conditional Advertisement

BGP conditional advertisement allows you to configure BGP to advertise or withdraw a route based on whether or not a prefix exists in the BGP table. This feature is useful, for example, in multihomed networks, in which you want BGP to advertise some prefixes to one of the providers only if information from the other provider is not present.

Consider an example network with three BGP autonomous systems: AS1, AS2, and AS3, where AS1 and AS3 connect to the Internet and to AS2. Without conditional advertisement, AS2 propagates all routes to both AS1 and AS3. With conditional advertisement, you can configure AS2 to advertise certain routes to AS3 only if routes from AS1 do not exist (if for example, the link to AS1 fails).

BGP conditional advertisement adds an exist or not-exist test to each route that matches the configured route map.

11.2.12 BGP Next-Hop Address Tracking

BGP monitors the next-hop address of installed routes to verify next-hop reachability and to select, install, and validate the BGP best path. BGP next-hop address tracking speeds up this next-hop reachability test by triggering the verification process when routes change in the Routing Information Base (RIB) that may affect BGP next-hop reachability.

BGP receives notifications from the RIB when the next-hop information changes (event-driven notifications). BGP is notified when any of the following events occurs:

- Next hop becomes unreachable.
- Next hop becomes reachable.
- Fully recursed Interior Gateway Protocol (IGP) metric to the next hop changes.
- First hop IP address or first hop interface changes.
- Next hop becomes connected.
- Next hop becomes unconnected.
- Next hop becomes a local address.
- Next hop becomes a nonlocal address.

Event notifications from the RIB are classified as critical and noncritical. Notifications for critical and noncritical events are sent in separate batches. However, a noncritical event is sent with the critical events if the noncritical event is pending and there is a request to read the critical events.

- Critical events are related to next-hop reachability, such as the loss of next hops resulting in a switchover to a different path. A change in the IGP metric for a next hop resulting in a switchover to a different path can also be considered a critical event.
- Non-critical events are related to next hops being added without affecting the best path or changing the IGP metric to a single next hop.

11.2.13 Route Redistribution

You can configure BGP to redistribute static routes or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into BGP. A route map allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on.

Prior to Inspur INOS Release 8.4(1), when you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an additional deny statement into the route map. Beginning with Inspur INOS Release 8.4(1), redistribution varies as follows:

- In a non-MPLS VPN scenario, iBGP is not redistributed to IGP by default.
- In an MPLS VPN scenario (route distinguisher configured under a VRF), iBGP is redistributed to IGP by default.

You can use route maps to override the default behavior in both scenarios, but be careful when doing so as incorrect use of route maps can result in network loops. The following examples show how to use route maps to change the default behavior.

You can change the default behavior for scenario 1 by modifying the route map as follows:

```
route-map    foo
   permit     10
```

```
                    match   route-
                    type internal

                 router ospf 1

                    redistribute bgp 100 route-map foo
```

Similarly, you can change the default behavior for scenario 2 by modifying the route map as follows:

```
route-map foo deny 10

  match   route-
type    internal
router ospf 1

   vrf bar

     redistribute bgp 100 route-map foo
```

The default route should be redistributed into BGP or advertised to peers only when **default-information originate** is configured for an Address Family where the command is supported.

BGP should withdraw the default route on removal of default-information originate if it was already advertised. Also, the redistributed path should be removed for the default route.

You can delete the redistributed path for default route using the following command: **no default-information originate**

11.2.14 BGP Support for Importing Routes from Default VRF

You can import IP prefixes from the global routing table (the default VRF) into any other VRF by using an import policy. The VRF import policy uses a route map to specify the prefixes to be imported into a VRF. The policy can import IPv4 and IPv6 unicast prefixes.

You can configure the maximum number of prefixes that can be imported from the default VRF.

11.2.15 BGP Support for Exporting Routes to Default VRF

You can export IP prefixes to the default VRF (global routing table) from any other VRF using an export policy. The VRF export policy leaks a VRF route into default VRF BGP table, which will then be installed in the IPv4/IPv6 routing table. The VRF export policy uses a route map to specify the prefixes to be exported to the default VRF. The policy can export IPv4 and IPv6 unicast prefixes.

You can configure the maximum number of prefixes that can be exported to the default VRF to prevent the routing table from being overloaded.

11.2.16 BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4 only. BFD is a detection protocol designed to provide fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules.

BFD for BGP is supported on eBGP peers and iBGP single-hop peers. Configure the update-source option in neighbor configuration mode for iBGP single-hop peers using BFD.

See the *Inspur CN12700 Series INOS Interfaces Configuration Guide* for more information.

11.2.17 Tuning BGP

You can modify the default behavior of BGP through BGP timers and by adjusting the best-path algorithm.

BGP Timers

BGP uses different types of timers for neighbor session and global protocol events. Each established session has a minimum of two timers for sending periodic keepalive messages and for timing out sessions when peer keepalives do not arrive within the expected time. In addition, there are other timers for handling specific features.

Typically, you configure these timers in seconds. The timers include a random adjustment so that the same timers on different BGP peers trigger at different times.

Tuning the Best-Path Algorithm

You can modify the default behavior of the best-path algorithm through optional configuration parameters, including changing how the algorithm handles the multi-exit discriminator (MED) attribute and the router ID.

11.2.18 Multiprotocol BGP

BGP on Inspur INOS supports multiple address families. Multiprotocol BGP (MP-BGP) carries different sets of routes depending on the address family. For example, BGP can carry one set of routes for IPv4 unicast routing, one set of routes for IPv4 multicast routing, and one set of routes for IPv6 multicast routing. You can use MP-BGP for reverse-path forwarding (RPF) checks in IP multicast networks.

Use the router address-family and neighbor address-family configuration modes to support multiprotocol BGP configurations. MP-BGP maintains separate RIBs for each configured address family, such as a unicast RIB and a multicast RIB for BGP.

A multiprotocol BGP network is backward compatible but BGP peers that do not support multiprotocol extensions cannot forward routing information, such as address family identifier information, that the multiprotocol extensions carry.

11.2.19 Graceful Restart and High Availability

Inspur INOS supports nonstop forwarding and graceful restart for BGP.

You can use nonstop forwarding (NSF) for BGP to forward data packets along known routes in the Forward Information Base (FIB) while the BGP routing protocol information is being restored following a failover. With NSF, BGP peers do not experience routing flaps. During a failover, the data traffic is forwarded through intelligent modules while the standby supervisor becomes active.

If a Inspur INOS router experiences a cold reboot, the network does not forward traffic to the router and removes the router from the network topology. In this scenario, BGP experiences a nongraceful restart and removes all routes. When Inspur INOS applies the startup configuration, BGP reestablishes peering sessions and relearns the routes.

A Inspur INOS router that has dual supervisors can experience a stateful supervisor switchover. During the switchover, BGP uses nonstop forwarding to forward traffic based on the information in the FIB, and the system is not removed from the network topology. A router whose neighbor is restarting is referred to as a "helper." After the switchover, a graceful restart operation begins. When it is in progress, both routers reestablish their neighbor relationship and exchange their BGP routes. The helper continues to forward prefixes pointing to the restarting peer, and the restarting router continues to forward traffic to peers even though those neighbor relationships are restarting. When the restarting router has all route updates from all BGP peers that are graceful restart capable, the graceful restart is complete, and BGP informs the neighbors that it is operational again.

When a router detects that a graceful restart operation is in progress, both routers exchange their topology tables. When the router has route updates from all BGP peers, it removes all the stale routes and runs the best-path algorithm on the updated routes.

After the switchover, Inspur INOS applies the running configuration, and BGP informs the neighbors that it is operational again.

For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.

With the additional BGP paths feature, if the number of paths advertised for a given prefix is the same before and after restart, the choice of path ID guarantees the final state and removal of stale paths. If fewer paths are advertised for a given prefix after a restart, stale paths can occur on the graceful restart helper peer.

Low Memory Handling

BGP reacts to low memory for the following conditions:

- Minor alert—BGP does not establish any new eBGP peers. BGP continues to establish new iBGP peers and confederate peers. Established peers remain, but reset peers are not re-established.

- Severe alert—BGP shuts down select established eBGP peers every two minutes until the memory alert becomes minor. For each eBGP peer, BGP calculates the ratio of total number of paths received to the number of paths selected as best paths. The peers with the highest ratio are selected to be shut down to reduce memory usage. You must clear a shutdown eBGP peer before you can bring the eBGP peer back up to avoid oscillation.
- Critical alert—BGP gracefully shuts down all the established peers. You must clear a shutdown BGP peer before you can bring the BGP peer back up.

11.2.20 ISSU

Inspur INOS supports in-service software upgrades (ISSU). ISSU allows you to upgrade software without impacting forwarding.

The following conditions are required to support ISSU:

- Graceful restart must be enabled (default)
- Keepalive and hold timers must not be smaller than their default values

If either of these requirements is not met, Inspur INOS issues a warning. You can proceed with the upgrade or downgrade, but service might be disrupted.

11.2.21 Virtualization Support

Inspur INOS supports multiple instances of BGP that run on the same system. BGP supports virtual routing and forwarding (VRF) instances that exist within virtual device contexts (VDCs). You can configure one BGP instance in a VDC, but you can have multiple VDCs on the system.

By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

11.3 Licensing Requirements for Advanced BGP

OSPFv3 requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *License and Copyright Information for Inspur INOS Software.*

11.4 Prerequisites for Advanced BGP

Advanced BGP has the following prerequisites:

- You must enable BGP.
- You should have a valid router ID configured on the system.
- You must have an AS number, either assigned by a Regional Internet Registry (RIR) or locally administered.
- You must have reachability (such as an interior gateway protocol [IGP], a static route, or a direct connection) to the peer that you are trying to make a neighbor relationship with.
- You must explicitly configure an address family under a neighbor for the BGP session establishment.

11.5 Guidelines and Limitations for Advanced BGP

Advanced BGP has the following configuration guidelines and limitations:

- The dynamic AS number prefix peer configuration overrides the individual AS number configuration inherited from a BGP template.
- If you configure a dynamic AS number for prefix peers in an AS confederation, BGP establishes sessions with only the AS numbers in the local confederation.
- BGP sessions created through a dynamic AS number prefix peer ignore any configured eBGP multihop time-to-live (TTL) value or a disabled check for directly connected peers.
- Configure a router ID for BGP to avoid automatic router ID changes and session flaps.

- Use the maximum-prefix configuration option per peer to restrict the number of routes received and system resources used.
- Configure the update source to establish a session with eBGP multihop sessions.
- Specify a BGP route map if you configure a redistribution.
- Configure the BGP router ID within a VRF.
- If you decrease the keepalive and hold timer values, the network might experience session flaps.

If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide, Release* 8.4(1)

- When you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an additional deny statement into the route map.
- Inspur INOS does not support multi-hop BFD. BFD for BGP has the following limitations:
  - BFD is supported only for BGP IPv4.
  - BFD is supported only for eBGP peers and iBGP single-hop peers.
  - To enable BFD for iBGP single-hop peers, you must configure the update-source option on the physical interface.
  - BFD is not supported for multi-hop iBGP peers and multi-hop eBGP peers.
- For single-hop iBGP peers with update-source configured under neighbor configuration mode, the peer supports fast external fall-over.
- The following guidelines and limitations apply to the remove-private-as command:
  - It applies only to eBGP peers.
  - It can be configured only in neighbor configuration mode and not in neighbor-address-family mode.
  - If the AS-path includes both private and public AS numbers, the private AS numbers are not removed.
  - If the AS-path contains the AS number of the eBGP neighbor, the private AS numbers are not removed.
  - Private AS numbers are removed only if all AS numbers in that AS-path belong to a private AS number range. Private AS numbers are not removed if a peer's AS number or a non-private AS number is found in the AS-path segment.
- BGP conditional route injection is available only for IPv4 and IPv6 unicast address families in all VRF instances.
- The **match interface** command is only supported for **redistribute** command **route-maps**.
- When sending a route advertisement to an iBGP peer, INOS sets the interface IP address through which the announced network is reachable for the peer as the next hop instead of preserving the original next hop of the non locally originated route.

This occurs with the 'network' statement and route 'redistribution' configurations in BGP.

The knobs 'set ip next-hop redist-unchanged' or 'set ipv6 next-hop redist-unchanged' available under route-map configuration mode helps to resolve this issue. These knobs are available from Inspur INOS Release 8.4(1) onwards.

11.6 Default Settings

Parameters	Default
BGP feature	Disabled
BGP additional paths	Disabled
Hold timer	180 seconds
Keep alive interval	60 seconds
Dynamic capability	Enabled

11.7 Configuring Advanced BGP

11.7.1 Configuring BGP Session Templates

You can use BGP session templates to simplify the BGP configuration for multiple BGP peers with similar configuration needs. BGP templates allow you to reuse common configuration blocks. You configure BGP templates first and then apply these templates to BGP peers.

With BGP session templates, you can configure session attributes such as inheritance, passwords, timers, and security.

A peer-session template can inherit from one other peer-session template. You can configure the second template to inherit from a third template. The first template also inherits this third template. This indirect inheritance can continue for up to seven peer-session templates.

Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *autonomous-system-number*
3. switch(config-router)# **template peer-session** *template-name*
4. switch(config-router-stmp)# **password** *number password*
5. switch(config-router-stmp)# **timers** *keepalive hold*
6. switch(config-router-stmp)# **exit**
7. switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
8. switch(config-router-neighbor)# **inherit peer-session** *template-name*
9. switch(config-router-neighbor)# **description** *text*
10. switch(config-router-neighbor)# **show bgp peer-session** *template-name*
11. switch(config-router-neighbor)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer-session** *template-name*	Enters peer-session template configuration mode.
Step 4	switch(config-router-stmp)# **password** *number password*	(Optional) Adds the clear text password test to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).
Step 5	switch(config-router-stmp)# **timers** *keepalive hold*	(Optional) Adds the BGP keepalive and holdtimer values to the peer-session template. The default keepalive interval is 60. The default hold time is 180.

Step 6	switch(config-router-stmp)# **exit**	Exits peer-session template configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	switch(config-router-neighbor)# **inherit peer-session** *template-name*	Applies a peer-session template to the peer.
Step 9	switch(config-router-neighbor)# **description** *text*	(Optional) Adds a description for the neighbor.
Step 10	switch(config-router-neighbor)# **show bgp peer-session** *template-name*	(Optional) Displays the peer-policy template.
Step 11	switch(config-router-neighbor)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

This example shows how to configure a BGP peer-session template and apply it to a BGP peer:

```
switch# configure terminal

switch(config)# router bgp 65535

switch(config-router)# template peer-session BaseSession

switch(config-router-stmp)# timers 30 90

switch(config-router-stmp)# exit

switch(config-router)#  neighbor  192.168.1.2  remote-as 65535

switch(config-router-neighbor)#   inherit   peer-session BaseSession
switch(config-router-neighbor)# description Peer Router A

switch(config-router-neighbor)#  address-family  ipv4 unicast

switch(config-router-neighbor)#  copy  running-config startup-config
```

11.7.2 Configuring BGP Peer-Policy Templates

You can configure a peer-policy template to define attributes for a particular address family. You assign a preference to each peer-policy template and these templates are inherited in the order specified, for up to five peer-policy templates in a neighbor address family.

Inspur INOS evaluates multiple peer policies for an address family using the preference value. The lowest preference value is evaluated first. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template.

Peer-policy templates can configure address family-specific attributes such as AS-path filter lists, prefix lists, route reflection, and soft reconfiguration.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *autonomous-system-number*
3. switch(config-router)# **template peer-policy** *template-name*
4. switch(config-router-ptmp)# **advertise-active-only**
5. switch(config-router-ptmp)# **maximum-prefix** *number*
6. switch(config-router-ptmp)# **exit**
7. switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
8. switch(config-router-neighbor)# **address-family**{**ipv4** | **ipv6** |**vpnv4** | **vpnv6**} {**multicast** | **unicast**}
9. switch(config-router-neighbor-af)# **inherit peer-policy** *template-namepreference*
10. switch(config-router-neighbor-af)# **show bgp peer-policy** *template-name*
11. switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer-policy** *template-name*	Creates a peer-policy template.
Step 4	switch(config-router-ptmp)# **advertise-active-only**	(Optional) Advertises only active routes to the peer.
Step 5	switch(config-router-ptmp)# **maximum-prefix** *number*	(Optional) Sets the maximum number of prefixes allowed from this peer.
Step 6	switch(config-router-ptmp)# **exit**	Exits peer-policy template configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	switch(config-router-neighbor)# **address-family**{**ipv4** | **ipv6** |**vpnv4** | **vpnv6**} {**multicast** | **unicast**}	Enters global address family configuration mode.
Step 9	switch(config-router-neighbor-af)# **inherit peer-policy** *template-namepreference*	Applies a peer-policy template to the peer address family configuration and assigns the preference value for this peer policy.
Step 10	switch(config-router-neighbor-af)# **show bgppeer-policy** *template-name*	(Optional) Displays the peer-policy template.
Step 11	switch(config-router-neighbor-af)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

This example shows how to configure a BGP peer-policy template and apply it to a BGP peer:

```
switch# configure terminal

switch(config)# router bgp 65535

switch(config-router)#   template   peer-session BasePolicy
switch(config-router-ptmp)# maximum-prefix 20

switch(config-router-ptmp)# exit
switch(config-router)# neighbor 192.168.1.1 remote-as 65535

switch(config-router-neighbor)#  address-family  ipv4 unicast

switch(config-router-neighbor-af)# inherit peer-policy BasePolicy

switch(config-router-neighbor-af)# copy running-config startup-config
```

11.7.3 Configuring BGP Peer Templates

You can configure BGP peer templates to combine session and policy attributes in one reusable configuration block. Peer templates can also inherit peer-session or peer-policy templates. Any attributes configured for the neighbor take priority over any attributes inherited by that neighbor from a BGP template. You configure only one peer template for a neighbor, but that peer template can inherit peer-session and peer-policy templates.

Peer templates support session and address family attributes, such as eBGP multihop time-to-live, maximum prefix, next-hop self, and timers.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *autonomous-system-number*
3. switch(config-router)# **template peer** *template-name*
4. switch(config-router-neighbor)# **inherit peer-session** *template-name*
5. switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**multicast** | **unicast**}
6. switch(config-router-neighbor-af)# **inherit peer** *template-name*
7. switch(config-router-neighbor-af)# **exit**
8. switch(config-router-neighbor)# **timers** *keepalive hold*
9. switch(config-router-neighbor)# **exit**
10. switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
11. switch(config-router-neighbor)# **inherit peer** *template-name*
12. switch(config-router-neighbor)# **timers** *keepalive hold*
13. switch(config-router-neighbor-af)# **show bgp peer-template** *template-name*
14. switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *autonomous-system-number*	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **template peer** *template-name*	Enter peer template configuration mode.
Step 4	switch(config-router-neighbor)# **inherit peer-session** *template-name*	(Optional) Inherits a peer-session template in the peer template.

Step 5	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**multicast** | **unicast**}	(Optional) Configures the global address family configuration mode.
Step 6	switch(config-router-neighbor-af)# **inherit peer** *template-name*	(Optional) Applies a peer template to the neighbor address family configuration.
Step 7	switch(config-router-neighbor-af)# **exit**	Exits BGP neighbor address family configuration mode.
Step 8	switch(config-router-neighbor)# **timers** *keepalive hold*	(Optional) Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 9	switch(config-router-neighbor)# **exit**	Exits BGP peer template configuration mode.
Step 10	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 11	switch(config-router-neighbor)# **inherit peer** *template-name*	Inherits the peer template.
Step 12	switch(config-router-neighbor)# **timers** *keepalive hold*	(Optional) Adds the BGP timer values to this neighbor. These values override the timer values in the peer template and the peer-session template.
Step 13	switch(config-router-neighbor-af)# **show bgp peer-template** *template-name*	(Optional) Displays the peer template.
Step 14	switch(config-router-neighbor-af)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

This example shows how to configure a BGP peer template and apply it to a BGP peer:

```
switch# configure terminal

switch(config)# router bgp 65535

switch(config-router)# template peer BasePeer

switch(config-router-neighbor)# inherit peer-session BaseSession

switch(config-router-neighbor)#  address-family  ipv4 unicast

switch(config-router-neighbor-af)#   inherit   peer-policy BasePolicy 1

switch(config-router-neighbor-af)# exit

switch(config-router-neighbor)# exit

switch(config-router)# neighbor 192.168.1.2 remote-as 65535
switch(config-router-neighbor)# inherit peer BasePeer
switch(config-router-neighbor)# copy running-config startup-config
```

11.7.4 Configuring Prefix Peering

BGP supports the definition of a set of peers using a prefix for both IPv4 and IPv6. This feature allows you to not have to add each neighbor to the configuration.

When defining a prefix peering, you must specify the remote AS number with the prefix. BGP accepts any peer that connects from that prefix and autonomous system if the prefix peering does not exceed the configured maximum peers allowed.

SUMMARY STEPS

1. (Optional) switch(config-router-neighbor)# **timers prefix-peer-timeout** *interval*
2. (Optional) switch(config-router-neighbor)# **timers prefix-peer-wait** *interval*
3. (Optional) switch(config-router-neighbor)# **maximum-peers** *value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch(config-router-neighbor)# **timers prefix-peer-timeout** *interval*	Configures the BGP prefix peering timeout value. When a BGP peer that is part of a prefix peering disconnects, the peer structures are held for a defined prefix peer timeout value which enables the peer to reset and reconnect without danger of being blocked. The timeout range is from 0 to 1200 seconds. The default value is 30.
Step 2	(Optional) switch(config-router-neighbor)# **timers prefix-peer-wait** *interval*	Configures the BGP prefix peering wait timer on a per-VRF basis or on the default VRF. You can use the **timers prefix-peer-wait** command to disable the peer prefix wait time so that there is no delay before BGP prefixes are inserted into the routing information base (RIB). The range of the *interval* is from 0 to 1200 seconds. The default value is 90. **Note** The timer is only applicable for BGP dynamic neighbors. It is only set when BGP is restarted or is coming up for the first time for the dynamic BGP neighbors.
Step 3	(Optional) switch(config-router-neighbor)# **maximum-peers** *value*	Configures the maximum number of peers for this prefix peering in neighbor configuration mode. The range is from 1 to 1000.

Example

This example shows how to configure a prefix peering that accepts up to 10 peers:

```
switch(config)# router bgp 65535

switch(config-router)#    timers    prefix-peer-timeout 120

switch(config-router)# neighbor 10.100.200.0/24 remote-as 65535
switch(config-router-neighbor)# maximum-peers 10

switch(config-router-neighbor)# address-family ipv4 unicast

switch(config-router-neighbor-af)#
```

11.7.5 Configuring BGP Authentication

You can configure BGP to authenticate route updates from peers using MD5 digests.

To configure BGP to use MD5 authentication, use the following command in neighbor configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor)# **password** {**0** | **3** | **7**} *string*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **password** {**0**|**3**|**7**} *string*	Configures an MD5 password (for authentication) for BGP neighbor sessions in neighbor configuration mode.

11.7.6 Resetting a BGP Session

If you modify a route policy for BGP, you must reset the associated BGP peer sessions. If the BGP peers do not support route refresh, you can configure a soft reconfiguration for inbound policy changes. Inspur INOS automatically attempts a soft reset for the session.

To configure soft reconfiguration inbound, use the following command in neighbor address-family configuration mode.

SUMMARY STEPS

1. switch(config-router-neighbor-af)# **soft-reconfiguration inbound**
2. switch# **clear bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**}

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# **soft-reconfiguration inbound**	This command in neighbor address-family configuration mode, enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 2	switch# **clear bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast** *ip-address* **soft** {**in** | **out**}	This command in any mode resets the BGP session without tearing down the TCP session.

11.7.7 Modifying the Next-Hop Address

You can modify the next-hop address used in a route advertisement in the following ways:

- Disable next-hop calculation and use the local BGP speaker address as the next-hop address.
- Set the next-hop address as a third-party address. Use this feature in situations where the original next-hop address is on the same subnet as the peer that the route is being sent to. Using this feature saves an extra hop during forwarding.

To modify the next-hop address, use the following commands in address-family configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor-af)# **next-hop-self**
2. switch(config-router-neighbor-af)# **next-hop-third-party**

DETAILED STEPS

	Command or Action	Purpose

Step 1	switch(config-router-neighbor-af)# **next-hop-self**	Uses the local BGP speaker address as the next-hop address in route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 2	switch(config-router-neighbor-af)# **next-hop-third-party**	Sets the next-hop address as a third-party address. Use this command for single-hop EBGP peers that do not have next-hop-self configured.

11.7.8 Configuring BGP Next-Hop Address Tracking

BGP next-hop address tracking is enabled by default and cannot be disabled.

You can modify the delay interval between RIB checks to increase the performance of BGP next-hop tracking.

To modify the BGP next-hop address tracking, use the following commands in address-family configuration mode:

SUMMARY STEPS

1. switch(config-router-af)# **nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds*
2. switch(config-router-af)# **nexthop route-map** *name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-af)# **nexthop trigger-delay** {**critical** | **non-critical**} *milliseconds*	Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to 4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000.
Step 2	switch(config-router-af)# **nexthop route-map** *name*	Specifies a route map to match the BGP next-hop addresses to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

11.7.9 Configuring Next-Hop Filtering

BGP next-hop filtering allows you to specify that when a next-hop address is checked with the RIB, the underlying route for that next-hop address is passed through the route map. If the route map rejects the route, the next-hop address is treated as unreachable.

BGP marks all next hops that are rejected by the route policy as invalid and does not calculate the best path for the routes that use the invalid next-hop address.

To configure BGP next-hop filtering, use the following command in address-family configuration mode:

SUMMARY STEPS

1. switch(config-router-af)# **nexthop route-map** *name*

DETAILED STEPS

	Command or Action	Purpose

Step 1	switch(config-router-af)# **nexthop route-map** *name*	Specifies a route map to match the BGP next-hop route to. The name can be any case-sensitive, alphanumeric string up to 63 characters.

11.7.10 Disabling Capabilities Negotiation

You can disable capabilities negotiations to interoperate with older BGP peers that do not support capabilities negotiation.

To disable capabilities negotiation, use the following command in neighbor configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor)# **dont-capability-negotiate**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **dont-capability-negotiate**	Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command.

11.7.11 Configuring BGP Additional Paths

Beginning with Inspur INOS Release 8.4(1), BGP supports sending and receiving multiple paths per prefix and advertising such paths.

Advertising the Capability of Sending and Receiving Additional Paths

You can configure BGP to advertise the capability of sending and receiving additional paths to and from the BGP peers.

SUMMARY STEPS

1. switch(config-router-neighbor-af)# [**no**]**capability additional paths send** [**disable**]
2. switch (config-router-neighbor-af)# [**no**]**capability additional paths receive** [**disable**]
3. switch(config-router-neighbor-af)# **show bgp neighbor**
4. switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# [**no**]**capability additional paths send** [**disable**]	Advertises the capability to send additional paths to the BGP peer. The disable option disables the advertising capability of sending additional paths. The no form of this command disables the capability of sending additional paths.
Step 2	switch (config-router-neighbor-af)# [**no**]**capability additional paths receive** [**disable**]	Advertises the capability to send additional paths to the BGP peer. The disable option disables the advertising capability of sending additional paths. The no form of this command disables the capability of sending additional paths.
Step 3	switch(config-router-neighbor-af)# **show bgp neighbor**	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.

Step 4	switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure BGP to advertise the capability to send and receive additional paths to the BGP peer:

```
switch(config)# router bgp 100

switch(config-router)#     neighbor     10.131.31.2     remote-as     100
switch(config-router-neighbor)#     address-family    ipv4 unicast

switch(config-router-neighbor-af)# capability additional-paths send

switch(config-router-neighbor-af)#capability   additional-paths receive

switch(config-router-neighbor-af)# show bgp neighbor

switch(config-router-neighbor-af)# copy running-config startup-config
```

Configuring the Sending and Receiving of Additional Paths

You can configure the capability of sending and receiving additional paths to and from the BGP peers.

SUMMARY STEPS

1. switch(config-router-neighbor-af)# [**no**]**additional-paths send**
2. switch (config-router-neighbor-af)# [**no**]**additional-paths receive** [**disable**]
3. switch(config-router-neighbor-af)# **show bgp neighbor**
4. switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# [**no**]**additional-paths send**	Enables the send capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The no form of this command disables the send capability.
Step 2	switch (config-router-neighbor-af)# [**no**]**additional-paths receive** [**disable**]	Enables the receive capability of additional paths for all of the neighbors under this address family for which the capability has not been disabled. The no form of this command disables the capability of sending additional paths.
Step 3	switch(config-router-neighbor-af)# **show bgp neighbor**	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 4	switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to enable the additional paths send and receive capability for neighbors under the specified address family for which this capability has not been disabled.:

```
switch(config)# router bgp 100

switch(config-router)#  neighbor  10.131.31.2 remote-as 100
switch(config-router-neighbor)# address-family ipv4 unicast

switch(config-router-neighbor-af)# additional-paths send

switch(config-router-neighbor-af)#additional-paths receive
switch(config-router-neighbor-af)#  show  bgp neighbor

switch(config-router-neighbor-af)#   copy   running-config startup-config
```

Configuring Advertised Paths

You can specify the paths that are advertised for BGP.

SUMMARY STEPS

1. switch(config-route-map)# [**no**]**set path-selection all advertise**
2. switch(config-route-map)# **show bgp neighbor** {**ipv4** | **ipv6**} **unicast***ip-address* | *ipv6-prefix* [**vrf***vrf-name*]
3. switch(config-route-map)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-route-map)# [**no**]**set path-selection all advertise**	Specifies that all paths be advertised for a given prefix. The no form of this command specifies that only the best path be advertised.
Step 2	switch(config-route-map)# **show bgp neighbor** {**ipv4** | **ipv6**} **unicast***ip-address* | *ipv6-prefix* [**vrf***vrf-name*]	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 3	switch(config-route-map)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to specify that all paths be advertised for the specified prefix:

```
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)#match   ip   address  prefeix-list   pl
switch(config-route-map)# show bgp ip4 unicast

switch(config-route-map)# copy running-config startup-config
```

Configuring Additional Path Selection

You can configure the capability of selecting additional paths for a prefix.

SUMMARY STEPS

1. switch(config-router-af)# [**no**]**additional-paths selection route-map***map-name*
2. switch(config-router-af)# **show bgp** {**ipv4** | **ipv6**} **unicast***ip-address* | *ipv6-prefix* [**vrf***vrf-name*]
3. (Optional) switch(config-router-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-af)# [**no**]**additional-paths selection route-map***map-name*	Specifies that all paths be advertised for a given prefix. The no form of this command specifies that only the best path be advertised.
Step 2	switch(config-router-af)# **show bgp** {**ipv4** | **ipv6**} **unicast***ip-address* | *ipv6-prefix* [**vrf***vrf-name*]	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.
Step 3	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to specify that all paths be advertised for the specified prefix:

```
switch(config)# router bgp 100

switch(config-router)# address-family ipv4 unicast

switch(config-router-af)#additional-paths selection route-map PATH_SELECTION_RMAP
switch(config-router-af)#  copy  running-config  startup-config
```

11.7.12 Configuring eBGP

Disabling eBGP Single-Hop Checking

You can configure eBGP to disable checking whether a single-hop eBGP peer is directly connected to the local router. Use this option for configuring a single-hop loopback eBGP session between directly connected switches.

To disable checking whether or not a single-hop eBGP peer is directly connected, use the following command in neighbor configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor)# **disable-connected-check**

DETAILED STEPS

	Command or Action	Purpose

Step 1	switch(config-router-neighbor)# **disable-connected-check**	Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.

Configuring eBGP Multihop

You can configure the eBGP time-to-live (TTL) value to support eBGP multihop. In some situations, an eBGP peer is not directly connected to another eBGP peer and requires multiple hops to reach the remote eBGP peer. You can configure the eBGP TTL value for a neighbor session to allow these multihop sessions.

To configure eBGP multihop, use the following command in neighbor configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor)# **ebgp-multihop** *ttl-value*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **ebgp-multihop** *ttl-value*	Configures the eBGP TTL value for eBGP multihop. The range is from 2 to 255. You must manually reset the BGP sessions after using this command.

Disabling a Fast External Fallover

By default, the Inspur CN12700 Series device supports fast external fallover for neighbors in all VRFs and address-families (IPv4 or IPv6).

Typically, when a BGP router loses connectivity to a directly connected eBGP peer, BGP triggers a fast external fallover by resetting the eBGP session to the peer. You can disable this fast external fallover to limit the instability caused by link flaps.

To disable fast external fallover, use the following command in router configuration mode:

SUMMARY STEPS

1. switch(config-router)# **no fast-external-fallover**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **no fast-external-fallover**	Disables a fast external fallover for eBGP peers. This command is enabled by default.

Limiting the AS-path Attribute

You can configure eBGP to discard routes that have a high number of AS numbers in the AS-path attribute.

To discard routes that have a high number of AS numbers in the AS-path attribute, use the following command in router configuration mode:

SUMMARY STEPS

1. switch(config-router)# **maxas-limit** *number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router)# **maxas-limit** *number*	Discards eBGP routes that have a number of AS-path segments that exceed the specified limit. The range is from 1 to 2000.

Configuring Local AS Support

The local-AS feature allows a router to appear to be a member of a second autonomous system (AS), in addition to its real AS. Local AS allows two ISPs to merge without modifying peering arrangements. Routers in the merged ISP become members of the new autonomous system but continue to use their old AS numbers for their customers.

This feature can only be used for true eBGP peers. You cannot use this feature for two peers that are members of different confederation sub-autonomous systems.

To configure eBGP local AS support, use the following command in neighbor configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor)# **local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor)# **local-as** *number* [**no-prepend** [**replace-as** [**dual-as**]]]	Configures eBGP to prepend the local AS number to the AS_PATH attribute. The AS number can be a 16-bit integer or a 32-bit integer in the form of a higher 16-bit decimal number and a lower 16-bit decimal number in xx.xx format.

Example

The local-AS feature under VRF configuration mode is supported for both IBGP and EBGP neighbor relationships.

This following example shows how to configure the feature for the IBGP neighbor 10.1.2.1:

```
router  bgp  65001 vrf BGP1

local-as 65002

address-family ipv4
unicast neighbor  10.1.2.1 remote-as 65002
```

The **local-as** command must be configured in the neighbor configuration mode for eBGP or a warning message is displayed stating that the local AS cannot be same as the remote AS. The following example shows how to configure the local-AS feature for eGBP:

```
router  bgp  65001
vrf BGP1

neighbor 20.1.2.1 remote-as 65003 local-as 65001
```

11.7.13 Configuring AS Confederations

To configure an AS confederation, you must specify a confederation identifier. To the outside world, the group of autonomous systems within the AS confederation look like a single autonomous system with the confederation identifier as the autonomous system number.

SUMMARY STEPS

1. switch(config-router)# **confederation identifier** *as-number*
2. switch(config-router)# **bgp confederation peers** *as-number* [*as-number2...*]

DETAILED STEPS

	Command or Action	Purpose

Step 1	switch(config-router)# **confederation identifier** *as-number*	In router configuration mode, this command configures a BGP confederation identifier. The command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 2	switch(config-router)# **bgp confederation peers** *as-number* [*as-number2...*]	In router configuration mode, this command configures the autonomous systems that belong to the AS confederation. The command specifies a list of autonomous systems that belong to the confederation and it triggers an automatic notification and session reset for the BGP neighbor sessions.

11.7.14 Configuring Route Reflector

You can configure iBGP peers as route reflector clients to the local BGP speaker, which acts as the route reflector. Together, a route reflector and its clients form a cluster. A cluster of clients usually has a single route reflector. In such instances, the cluster is identified by the router ID of the route reflector. To increase redundancy and avoid a single point of failure in the network, you can configure a cluster with more than one route reflector. You must configure all route reflectors in the cluster with the same 4-byte cluster ID so that a route reflector can recognize updates from route reflectors in the same cluster.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)# **cluster-id** *cluster-id*
4. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}
5. switch(config-router-af)# **client-to-client reflection**
6. switch(config-router-neighbor)# **exit**
7. switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
8. switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}
9. switch(config-router-neighbor-af)# **route-reflector-client**
10. (Optional) switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **neighbors**
11. (Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **cluster-id** *cluster-id*	Configures the local router as one of the route reflectors that serve the cluster. You specify a cluster ID to identify

		the cluster. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 4	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters router address family configuration mode for the specified address family.
Step 5	switch(config-router-af)# **client-to-client reflection**	(Optional) Configures client-to-client route reflection. This feature is enabled by default. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 6	switch(config-router-neighbor)# **exit**	Exits router address configuration mode.
Step 7	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 8	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters neighbor address family configuration mode for the unicast IPv4 address family.
Step 9	switch(config-router-neighbor-af)# **route-reflector-client**	Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 10	(Optional) switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} **neighbors**	Displays the BGP peers.
Step 11	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to configure the router as a route reflector and add one neighbor as a client:

```
switch(config)# router bgp 65535

switch(config-router)#     neighbor    192.0.2.10    remote-as   65535
switch(config-router-neighbor)#      address-family     ip      unicast
switch(config-router-neighbor-af)#              route-reflector-client
switch(config-router-neighbor-af)# copy running-config startup-config
```

11.7.15 Configuring Next-Hops on Reflected Routes Using an Outbound Route-Map

You can change the next-hop on reflected routes on a BGP route reflector using an outbound route-map. You can configure the outbound route-map to specify the peer's local address as the next-hop address.

Before you begin

You must enable BGP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

You must enter the **set next-hop** command to configure an address family specific next-hop address. For example, for the IPv6 address family, you must enter the **set ipv6 next-hop peer-address** command.

- When setting IPv4 next-hops using route-maps—If **set ip next-hop peer-address** matches the route-map, the next-hop is set to the peer's local address. If no next-hop is set in the route-map, the next-hop is set to the one stored in the path.
- When setting IPv6 next-hops using route-maps—If **set ipv6 next-hop peer-address** matches the route-map, the next hop is set as follows:

- For IPv6 peers, the next-hop is set to the peer's local IPv6 address.
- For IPv4 peers, if **update-source** is configured, the next-hop is set to the source interface's IPv6 address, if any. If no IPv6 address is configured, no next-hop is set
- For IPv4 peers, if **update-source** is not configured, the next-hop is set to the outgoing interface's IPv6 address, if any. If no IPv6 address is configured, no next-hop is set.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*
4. switch(config-router-neighbor)# **update-source** *interface number*
5. switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}
6. switch(config-router-neighbor-af)# **route-reflector-client**
7. switch(config-router-neighbor-af)# **route-map** *map-name* **out**
8. (Optional) switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** |**multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*]
9. (Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 4	switch(config-router-neighbor)# **update-source** *interface number*	(Optional) Specifies and updates the source of the BGP session.
Step 5	switch(config-router-neighbor)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters router address family configuration mode for the specified address family.
Step 6	switch(config-router-neighbor-af)# **route-reflector-client**	Configures the device as a BGP route reflector and configures the neighbor as its client. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 7	switch(config-router-neighbor-af)# **route-map** *map-name* **out**	Applies the configured BGP policy to outgoing routes.
Step 8	(Optional) switch(config-router-neighbor-af)# **show bgp** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [**ip-address** | **ipv6-prefix**] **route-map** *map-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the route map.
Step 9	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to configure the next-hop on reflected routes on a BGP route reflector using an outbound route-map:

```
switch(config)#   interface   loopback   300
switch(config-if)#  ip  address 192.0.2.11/32
switch(config-if)#ipv6      address       2001::a0c:1a65/64
switch(config-if)#ip  router  ospf1 area 0.0.0.0

switch(config-if)# exit

switch(config)#    route-map    setrrnh permit 10

switch(config-route-map)#  set  ip  next-hop peer-address

switch(config-route-map)# exit

switch(config)#    route-map    setrrnhv6 permit 10

switch(config-route-map)#  set  ipv6  next-hop peer-address

switch(config-route-map)# exit

switch(config)# router bgp 200

switch(config-router)#  neighbor  192.0.2.12 remote-as 200
switch(config-router-neighbor)# update-source loopback 300

switch(config-router-neighbor)# address-family ipv4 unicast

switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)#  route-map setrrnh out

switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 unicast

switch(config-router-neighbor-af)# route-reflector-client
switch(config-router-neighbor-af)#  route-map setrrnhv6 out
```

11.7.16 Configuring Route Dampening

You can configure route dampening to minimize route flaps propagating through your iBGP network.

To configure route dampening, use the following command in address-family or VRF address family configuration mode.

SUMMARY STEPS

1. switch (config-router-af)# **dampening** [*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch (config-router-af)# **dampening** [*half-life reuse-limit suppress-limit max-suppress-time* | **route-map** *map-name*}]	Disables capabilities negotiation. The parameter values are as follows: • half-life—The range is from 1 to 45 • reuse-limit—The range is from 1 to 20000. • suppress-limit—The range is from 1 to 20000. • max-suppress-time—The range is from 1 to 255

11.7.17 Configuring Load Sharing and ECMP

You can configure the maximum number of paths that BGP adds to the route table for equal-cost multipath load balancing.

To configure the maximum number of paths, use the following command in router address-family configuration mode:

Command	Purpose
switch(config-router-af)# **maximum-paths** [**ibgp**] *maxpaths*	Configures the maximum number of equal-cost paths for load sharing. The range is from 1 to 16. The default is 1. Starting from Inspur INOS Release 8.4(1), the range is from 1 to 64 on F3-Series I/O modules.

11.7.18 Configuring Maximum Prefixes

You can configure the maximum number of prefixes that BGP can receive from a BGP peer. If the number of prefixes exceeds this value, you can optionally configure BGP to generate a warning message or tear down the BGP session to the peer.

To configure the maximum allowed prefixes for a BGP peer, use the following command in neighbor address-family configuration mode:

SUMMARY STEPS

1. switch(config-router-neighbor-af)# **maximum-prefix** *maximum* [*threshold*] [**restart***time* | **warning-only**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-router-neighbor-af)# **maximum-prefix** *maximum* [*threshold*] [**restart***time* | **warning-only**]	Configure the maximum number of prefixes from a peer. The parameter ranges are as follows: • *maximum*—The range is from 1 to 300000. • *threshold*—The range is from 1 to 100 percent. The default is 75 percent. • *time*—The range is from 1 to 65535 minutes. This command triggers an automatic notification and session reset for the BGP neighbor sessions if the prefix limit is exceeded.

11.7.19 Configuring DynamicCapability

You can configure dynamic capability for a BGP peer.

To configure dynamic capability, use the following command in neighbor configuration mode:

Command	Purpose
switch(config-router-neighbor)# **dynamic-capability**	Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

11.7.20 Configuring Aggregate Addresses

You can configure aggregate address entries in the BGP route table.

To configure an aggregate address, use the following command in router address-family configuration mode:

Command	Purpose
aggregate-address *ip-prefix/length* [**as-set**] [**summary-only**] [**advertise-map** *map-name*] [**attribute-map** *map-name*] [**suppress-map** *map-name*]	Creates an aggregate address. The path advertised for this route is an autonomous system set that consists of all elements contained in all paths that are being summarized: • The **as-set** keyword generates autonomous system set path information and community information from contributing paths. • The **summary-only** keyword filters all more specific routes from updates. • The **advertise-map** *map-name* keyword and argument specify the route map used to select attribute information from selected routes. • The **attribute-map** *map-name* keyword and argument specify the route map used to select attribute information from the aggregate. • The **suppress-map** *map-name* keyword and argument conditionally filter more specific routes.

11.7.21 Unsuppressing the Advertisement of Aggregated Routes

You can configure BGP to advertise routes that are suppressed by the **aggregate-address** command.

To unsuppress the advertising of aggregated routes, use the following command in router neighbor address-family configuration mode:

SUMMARY STEPS

1. switch(config-route-neighbor-af)# **unsuppress-map** *map-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-route-neighbor-af)# **unsuppress-map** *map-name*	Advertises selective routes that are suppressed by the **aggregate-address** command.

11.7.21 Configuring BGP Conditional Route Injection

You can configure BGP conditional route injection to inject specific routes based on the administrative policy or traffic engineering information and control the packets being forwarded to these specific routes, which are injected into the BGP routing table only if the configured conditions are met. This feature allows you to improve the accuracy of common route aggregation by conditionally injecting or replacing less specific prefixes with more specific prefixes. Only prefixes that are equal to or more specific than the original prefix can be injected.

Before you begin

You must enable BGP

Ensure that you are in the correct VDC (or use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**
4. switch(config-router-af)# **inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**]
5. switch(config-router-af)# **exit**
6. switch(config-router)# **exit**
7. switch(config)# **ip prefix-list** *list-name* **seq** *sequence-number* **permit***network-length*
8. switch(config)# **route-map** *map-name* **permit** *sequence-number*
9. switch(config-route-map)# **match ip address prefix-list***prefix-list-name*
10. switch(config-route-map)# **match ip route-source prefix-list***prefix-list-name*
11. switch(config-route-map)# **exit**
12. switch(config)# **ip prefix-list** *list-name* **seq** *sequence-number* **permit** *network-length*
13. switch(config)# **route-map** *map-name* **permit** *sequence-number*
14. switch(config-route-map)# **set ip address prefix-list**
15. (Optional) switch(config-route-map)# **show bgp** {**ipv4** | **ipv6**} **unicast injected-routes**
16. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP configuration mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6**} **unicast**	Enters address family configuration mode.
Step 4	switch(config-router-af)# **inject-map** *inject-map-name* **exist-map** *exist-map-name* [**copy-attributes**]	Specifies the inject-map and exist-map routes for conditional route injection. These maps install one or more prefixes intoa BGP routing table. The *exist-map* route map specifies the prefixes that BGP tracks, and the *inject-map* route map defines the prefixes that are created and installed into the local BGP table. Use the **copy-attributes** keyword to specify that the injected route inherits the attributes of the aggregate route.
Step 5	switch(config-router-af)# **exit**	Exits address family configuration mode.
Step 6	switch(config-router)# **exit**	Exits BGP configuration mode.
Step 7	switch(config)# **ip prefix-list** *list-name* **seq** *sequence-number* **permit***network-length*	Configures a prefix list. Repeat this step for every prefix list to be created.

Step 8	switch(config)# **route-map** *map-name* **permit** *sequence-number*	Configures a route-map and enters route-map configuration mode.
Step 9	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	Specifies the aggregate route to which a more specific route will be injected.
Step 10	switch(config-route-map)# **match ip route-source prefix-list** *prefix-list-name*	Specifies the match conditions for the source fo the route.
Step 11	switch(config-route-map)# **exit**	Exits route-map configuration mode.
Step 12	switch(config)# **ip prefix-list** *list-name* **seq** *sequence-number* **permit** *network-length*	Configures a prefix list. Repeat this step for every prefix list to be created.
Step 13	switch(config)# **route-map** *map-name* **permit** *sequence-number*	Configures a route map and enters route-map configuration mode.
Step 14	switch(config-route-map)# **set ip address prefix-list**	Specifies the routes to be injected.
Step 15	(Optional) switch(config-route-map)# **show bgp** {**ipv4** | **ipv6**} **unicast injected-routes**	Displays injected routes in the routing table.
Step 16	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

11.7.21 Configuring BGP Conditional Advertisement

You can configure BGP conditional advertisement to limit the routes that BGP propagates. You define the following two route maps:

- Advertise map—Specifies the conditions that the route must match before BGP considers the conditional advertisement. This route map can contain any appropriate match statements.
- Exist map or nonexist map—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexist map defines the prefix that must not exist in the BGP table before BGP propagates a route that matches the advertise map. BGP processes only the permit statements in the prefix list match statements in these route maps.

If the route does not pass the condition, BGP withdraws the route if it exists in the BGP table.

Before you begin

You must enable BGP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)#**neighbor** *ip-address* **remote-as** *as-number*
4. switch(config-router-neighbor)# **address-family** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**}
5. switch(config-router-neighbor-af)# **advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*}
6. (Optional) switch(config-router-neighbor-af)# **show ip bgp neighbor**
7. (Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)#**neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	switch(config-router-neighbor)# **address-family** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**}	Enters address family configuration mode.
Step 5	switch(config-router-neighbor-af)# **advertise-map** *adv-map* {**exist-map** *exist-rmap*|**non-exist-map** *nonexist-rmap*}	Configures BGP to conditionally advertise routes based on the two configured route maps: • **adv-map**—Specifies a route map with match statements that the route must pass before BGP passes the route to the next route map. The adv-map is a case-sensitive, alphanumeric string up to 63 characters. • **exist-rmap**—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must match a prefix in the prefix list before BGP advertises the route. The exist-rmap is a case-sensitive, alphanumeric string up to 63 characters. • **nonexist-rmap**—Specifies a route map with match statements for a prefix list. A prefix in the BGP table must not match a prefix in the prefix list before BGP advertises the route. The nonexist-rmap is a case-sensitive, alphanumeric string up to 63 characters.
Step 6	(Optional) switch(config-router-neighbor-af)# **show ip bgp neighbor**	Displays information about BGP and the configured conditional advertisement route maps.
Step 7	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to configure BGP conditional advertisement:

```
switch# configure terminal

switch(config)# router bgp 65535

switch(config-router)# neighbor 192.0.2.2 remote-as 65537

switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#  advertise-map  advertise  exist-map exist

switch(config-router-neighbor-af)# exit
switch(config-router-neighbor) # exit
switch(config-router)# exit
switch(config)# route-map advertise

switch(config-route-map)# match as-path pathList

switch(config-route-map)# exit

switch(config)# route-map exit

switch(config-route-map)#  match  ip  address  prefix-list plist

switch(config-route-map)# exit
switch(config)# ip prefix-list plist permit 209.165.201.0/27
```

11.7.22 Configuring Route Redistribution

You can configure BGP to accept routing information from another routing protocol and redistribute that information through the BGP network. Optionally, you can assign a default metric for redistributed routes.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}
4. switch(config-router-af)# **redistribute** {**direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
5. (Optional) switch(config-router-af)# **default-metric** *value*
6. (Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**}	Enters address family configuration mode.
Step 4	switch(config-router-af)# **redistribute** {**direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*	Redistributes routes from other protocols into BGP.
Step 5	(Optional) switch(config-router-af)# **default-metric** *value*	Generates a default metric into BGP.
Step 6	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to redistribute EIGRP into BGP:

```
switch# configure terminal

switch(config)# router bgp 65535

switch(config-router)#   address-family   ipv4 unicast

switch(config-router-af)# redistribute eigrp 201 route-map Eigrpmap

switch(config-router-af)# copy running-config startup-config
```

11.7.23 Advertising the Default Route

You can configure BGP to advertise the default route (network 0.0.0.0).

Before you begin

You must enable BGP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **route-map allow permit**
3. switch(config-route-map)# **exit**
4. switch(config)# **ip route** *ip-address network-mask* **null** *null-interface-number*
5. switch(config)# **router bgp** *as-number*
6. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**
7. switch(config-router-af)# **default-information originate**
8. switch(config-router-af)# **redistribute static route-map allow**
9. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **route-map allow permit**	Enters router map configuration mode and defines the conditions for redistributing routes
Step 3	switch(config-route-map)# **exit**	Exits router map configuration mode.
Step 4	switch(config)# **ip route** *ip-address network-mask* **null** *null-interface-number*	Configures the IP address.
Step 5	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the AS number to the local BGP speaker.
Step 6	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**	Enters address family configuration mode.
Step 7	switch(config-router-af)# **default-information originate**	Advertises the default route.
Step 8	switch(config-router-af)# **redistribute static route-map allow**	Redistributes the default route.

Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

11.7.24 Configuring Route Import from Default VRF to any other VRF

Perform the following steps to import routes from default VRF to any other non-default VRF.

Before you begin

- Enable BGP.
- Ensure that you are in the correct VDC.

Step 1 Enter the global configuration mode: switch#configure terminal

Step 2 Enable BGP: switch(config)#feature bgp

Step 3 Create a new VRF and enter VRF configuration mode:
switch(config)#**vrf context** *vrf-name*

Step 4 Enter the IPv4 / IPv6 unicast address family configuration mode:
switch(config-vrf)# **address-family {ipv4 | ipv6}** unicast

Step 5 Configure an import policy for a VRF to import prefixes from the default VRF:
switch(config-vrf-af)# **import vrf default** *[prefix-limit]* **map** *route-map*

prefix-limit limits the number of routes that can be imported. Default value is 1000.

route-map specifies the route-map to be imported and can be case-sensitive, alphanumeric string up to 63 characters.

11.7.25 Configuring Route Export from BGP VRF to Default VRF

Perform the following steps to export routes from non-default VRF to Default VRF.

Before you begin

- Enable BGP.
- Ensure that you are in the correct VDC.

Step 1 Enter the global configuration mode: switch#**configure terminal**

Step 2 Enable BGP: switch(config)#**feature bgp**

Step 3 Create a new VRF and enter VRF configuration mode:
switch(config)#**vrf context** *vrf-name*

Step 4 Enter the IPv4 / IPv6 unicast address family configuration mode:
switch(config-vrf)# **address-family {ipv4 | ipv6}** unicast

Step 5 Export IPv4 or IPv6 prefixes from non-default VRF to default VRF, filtered by route-map.:
switch(config-vrf-af)# **export vrf default** *[prefix-limit]* **map** *route-map*

prefix-limit limits the number of routes that can be exported, in order to avoid the global table being overloaded. Default value is 1000.

route-map can be case-sensitive, alphanumeric string up to 63 characters. It specifies the route-map.

If the route map does not exist, the command will be accepted but processed at a later time when the route map is created.

Example

The following example shows how to export the route map, BgpMap, to default VRF, and verify the configuration.

```
switch# configure terminal
switch(config)# feature bgp
switch(config)# vrf context vpn1
switch(config-vrf)# address-family ipv4 unicast

switch(config-vrf-af)#    export    vrf default 3 map BgpMap

switch(config-vrf-af)# exit

switch(config)#show bgp process vrf vpn1
Information   regarding
configured  VRFs:  BGP
Information   for   VRF
vpn1
VRF Id                       : 3
VRF state                    : UP
Router-ID                    : 20.0.0.1
Configured Router-ID         : 0.0.0.0
Confed-ID                    : 0
Cluster-ID                   : 0.0.0.0
No.  of  configured
peers      : 2 No.
of  pending  config
peers  :  0  No.  of
established
peers      : 2

VRF RD                       : 100:1

    Information  for  address  family  IPv4  Unicast  in  VRF
    vpn1
    Table Id                 : 3
    Table state              : UP
    Peers      Active-peers    Routes     Paths      Networks   Aggregates
    1          1               6          6          0          0

    Redistribution

        static, route-map allow

    Export RT list: 100:1
    1000:1
    Import RT list: 100:1
    Label mode: per-prefix Aggregate label: 492287

    Import default limit     :

    1000   Import   default
    prefix   count   :   2
    Import default map       : allow
```

```
Export default limit       :

1000   Export   default
prefix   count   :   3
Export default map         : allow
```

11.7.26 Configuring Multiprotocol BGP

You can configure MP-BGP to support multiple address families, including IPv4 and IPv6 unicast and multicast routes.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)#**neighbor** *ip-address* **remote-as** *as-number*
4. switch(config-router-neighbor)# **address-family** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**}
5. (Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	switch(config-router)#**neighbor** *ip-address* **remote-as** *as-number*	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	switch(config-router-neighbor)# **address-family** {**ipv4**|**ipv6**|**vpnv4**|**vpnv6**} {**unicast**|**multicast**}	Enters address family configuration mode.
Step 5	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to enable advertising and receiving IPv4 and IPv6 routes for multicast RPF for a neighbor:

```
switch# configure terminal
switch(config)# interface ethernet 2/1
switch(config-if)#  ipv6  address 2001:0DB8::1
switch(config-if)#   router   bgp 65535
switch(config-router)#   neighbor   192.168.1.2 remote-as 35537
switch(config-router-neighbor)# address-family ipv4 multicast
switch(config-router-neighbor-af)# exit
switch(config-router-neighbor)# address-family ipv6 multicast
switch(config-router-neighbor-af)#   copy   running-config startup-config
```

11.7.27 Configuring Policy-Based Administrative Distance

You can configure a distance for external BGP (eBGP) and internal BGP (iBGP) routes that match a policy described in the configured route map. The distance configured in the route map is downloaded to the unicast RIB along with the matching routes. BGP uses the best path to determine the administrative distance when downloading next hops in the unicast RIB table. If there is no match or a deny clause in the policy, BGP uses the distance configured in the distance command or the default distance for routes.

The policy-based administrative distance feature is useful when there are two or more different routes to the same destination from two different routing protocols.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip prefix-list** *name* **seq** *number* **permit***prefix-length*
3. switch(config)# **route-map** *map-tag* **permit** *sequence-number*
4. switch(config-route-map)# **match ip address prefix-list***prefix-list-name*
5. switch(config-route-map)# **set distance** *<value1> <value2> <value3>*
6. switch(config-route-map)# **exit**
7. switch(config)# **router bgp** *as-number*
8. switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**
9. switch(config-router-af)# **table-map** *map-name*
10. (Optional) switch(config-router-af)# **show forwarding distribution**
11. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip prefix-list** *name* **seq** *number* **permit** *prefix-length*	Creates a prefix list to match IP packets or routes with the permit keyword.
Step 3	switch(config)# **route-map** *map-tag* **permit** *sequence-number*	Creates a route map and enters route-map configuration mode with the permit keyword. If the match criteria for the route is met in the policy, the packet is policy routed.
Step 4	switch(config-route-map)# **match ip address prefix-list** *prefix-list-name*	Matches IPv4 network routes based on a prefix list. The prefix-list name can be any alphanumeric string up to 63 characters.

Step 5	switch(config-route-map)# **set distance** *<value1> <value2> <value3>*	Specifies the administrative distance for interior BGP (iBGP) or exterior BGP (eBGP) routes and BGP routes originated in the local autonomous system. The range is from 1 to 255. After you enter the value for the external administrative distance, you must enter the value for the administrative distance for the internal routes or/and the value for the administrative distance for the local routes depending on your requirement; so that the internal/local routes are also considered in the route administration.
Step 6	switch(config-route-map)# **exit**	Exits route-map configuration mode.
Step 7	switch(config)# **router bgp** *as-number*	Enters BGP mode and assigns the AS number to the local BGP speaker.
Step 8	switch(config-router)# **address-family** {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **unicast**	Enters address family configuration mode.
Step 9	switch(config-router-af)# **table-map** *map-name*	Configures the selective administrative distance for a route map for BGP routes before forwarding them to the RIB table. The table-map name can be any alphanumeric string up to 63 characters. **Note** You can also configure the **table-map** command under the VRF address-family configuration mode.
Step 10	(Optional) switch(config-router-af)# **show forwarding distribution**	Displays forwarding information distribution.
Step 11	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

11.7.28 Tuning BGP

You can tune BGP characteristics through a series of optional parameters.

SUMMARY STEPS

1. switch(config-router)# **bestpath** [**always-compare-med** | **as-path multipath-relax** | **compare-routerid** | **cost-community ignore** | **med** {**confed** | **missing-as-worst** | **non-deterministic**}]
2. switch(config-router)# **enforce-first-as**
3. switch(config-router)# **log-neighbor-changes**
4. switch(config-router)# **router-id** *id*
5. switch(config-router)# **timers** [**bestpath-delay** *delay* | **bgp** *keepalive holdtime* | **prefix-peer-timeout** *timeout*]
6. switch(config-router-af)# **distance** *ebgp-distance ibgp-distance local-distance*
7. switch(config-router-neighbor)# **description** *string*
8. switch(config-router-neighbor)# **low-memory exempt**
9. switch(config-router-neighbor)# **transport connection-mode passive remove-private-as**
10. switch(config-router-neighbor)# **update-source** *interface-type number*
11. switch(config-router-neighbor)# **suppress-inactive**
12. switch(config-router-neighbor)# **default-originate** [**route-map** *map-name*]
13. switch(config-router-neighbor)# **filter-list** *list-name* {**in**|**out**}
14. switch(config-router-neighbor)# **prefix-list** *list-name* {**in**|**out**}
15. switch(config-router-neighbor)# **send-community**
16. switch(config-router-neighbor)# **send-community extended**

DETAILED STEPS

	Command or Action	Purpose
Step 1	Required: switch(config-router)# **bestpath** [**always-compare-med** | **as-path multipath-relax** | **compare-routerid** | **cost-community ignore** | **med** {**confed** | **missing-as-worst** | **non-deterministic**}]	Modifies the best-path algorithm. The optional parameters are as follows: • **always-compare-med**—Compares MED on paths from different autonomous systems. • **as-path multipath-relax**—Allows load sharing across the providers with different (but equal-length) AS paths. Without this option, the AS paths must be identical for load sharing. • **compare-routerid**—Compares the router IDs for identical eBGP paths. • **cost-community ignore**—Ignores the cost community for BGP best-path calculations. For more information on the BGP cost community, see the "Configuring MPLS Layer 3 VPN Load Balancing" chapter of the Inspur CN12700 Series INOS MPLS Configuration Guide. • **med confed**—Forces bestpath to do a MED comparison only between paths originated within a confederation. • **med missing-as-worst**—Treats a missing MED as the highest MED. • **med non-deterministic**—Does not always pick the best MED path from among the paths from the same autonomous system.
Step 2	switch(config-router)# **enforce-first-as**	Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.
Step 3	switch(config-router)# **log-neighbor-changes**	Generates a system message when a neighbor changes state.
Step 4	switch(config-router)# **router-id** *id*	Manually configures the router ID for this BGP speaker.
Step 5	switch(config-router)#**timers** [**bestpath-delay** *delay* | **bgp** *keepalive holdtime* | **prefix-peer-timeout** *timeout*]	Sets the BGP timer values. The optional parameters are as follows: • **delay**—Initial best-path timeout value after a restart. The range is from 0 to 3600 seconds. The default value is 300. • **keepalive**—BGP session keepalive time. The range is from 0 to 3600 seconds. The default value is 60. • **holdtime**—BGP session hold time.The range is from 0 to 3600 seconds. The default value is 180. • **timeout**—Prefix peer timeout value. The range is from 0 to 1200 seconds. The default value is 30.

Step 6	switch(config-router-af)# **distance** *ebgp-distance ibgp-distance local-distance*	Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows: • **ebgp-distance**—20. • **ibgp-distance**—200. • **local-distance**—220. Local-distance is the administrative distance used for aggregate discard routes when they are installed in the RIB.
Step 7	switch(config-router-neighbor)# **description** *string*	Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.
Step 8	switch(config-router-neighbor)# **low-memory exempt**	Exempts this BGP neighbor from a possible shutdown due to a low memory condition.
Step 9	switch(config-router-neighbor)# **transport connection-mode passive**	Allows a passive connection setup only. This BGP speaker does not initiate a TCP connection to a BGP peer. You must manually reset the BGP sessions after configuring this command.
Step 10	**remove-private-as**	Removes private AS numbers from outbound route updates to an eBGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions. **Note** See the “Guidelines and Limitations for Advanced BGP” section for more information on this command.
Step 11	switch(config-router-neighbor)# **update-source** *interface-type number*	Configures the BGP speaker to use the source IP address of the configured interface for BGP sessions to the peer. This command triggers an automatic notification and session reset for the BGP neighbor sessions. Single-hop iBGP peers support fast external failover when **update-source** is configured.
Step 12	switch(config-router-neighbor)# **suppress-inactive**	Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 13	switch(config-router-neighbor)# **default-originate** [**route-map** *map-name*]	Generates a default route to the BGP peer.
Step 14	switch(config-router-neighbor)# **filter-list** *list-name* {**in**|**out**}	Applies an AS path filter list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 15	switch(config-router-neighbor)# **prefix-list** *list-name* {**in**|**out**}	Applies a prefix list to this BGP peer for inbound or outbound route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 16	switch(config-router-neighbor)# **send-community**	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
Step 17	switch(config-router-neighbor)# **send-community extended**	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

11.7.29 Configuring a Graceful Restart

You can configure a graceful restart and enable the graceful restart helper feature for BGP.

Before you begin

- You must enable BGP. Create the VDCs and VRFs.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router bgp** *as-number*
3. switch(config-router)# **graceful-restart**
4. switch(config-router)# **graceful-restart** {**restart-time** *time*|**stalepath-time** *time*}
5. switch(config-router)# **graceful-restart-helper**
6. (Optional) switch(config-router)# **show running-config bgp**
7. (Optional) switch(config-router)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **router bgp** *as-number*	Creates a new BGP process with the configured autonomous system number.
Step 3	switch(config-router)# **graceful-restart**	Enables a graceful restart and the graceful restart helper functionality. This command is enabled by default. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 4	switch(config-router)# **graceful-restart** {**restart-time** *time*|**stalepath-time** *time*}	Configures the graceful restart timers. The optional parameters are as follows: • **restart-time**—Maximum time for a restart sent to the BGP peer. The range is from 1 to 3600 seconds. The default is 120. • **stalepath-time**—Maximum time that BGP keeps the stale routes from the restarting BGP peer. The range is from 1 to 3600 seconds. The default is 300. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 5	switch(config-router)# **graceful-restart-helper**	Enables the graceful restart helper functionality. Use this command if you have disabled graceful restart but you still want to enable graceful restart helper functionality. This command triggers an automatic notification and session reset for the BGP neighbor sessions.
Step 6	(Optional) switch(config-router)# **show running-config bgp**	Displays the BGP configuration.
Step 7	(Optional) switch(config-router)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to enable a graceful restart:

```
switch#  configure  terminal
switch(config)#         router         bgp        65535
switch(config-router)# graceful-restart

switch(config-router)#    copy    running-config startup-config
```

11.7.30 Configuring Virtualization

You can configure one BGP process in each VDC. You can create multiple VRFs within each VDC and use the same BGP process in each VRF.

Before you begin

- You must enable BGP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*
3. switch(config-vrf)# **exit**
4. switch(config)# **router bgp** *as-number*
5. switch(config-router)# **vrf** *vrf-name*
6. switch(config-router-vrf)# **neighbor** *ip-address* **remote-as** *as-number*
7. (Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a new VRF and enters VRF configuration mode.
Step 3	switch(config-vrf)# **exit**	Exits VRF configuration mode.
Step 4	switch(config)# **router bgp** *as-number*	Creates a new BGP process with the configured autonomous system number.
Step 5	switch(config-router)# **vrf** *vrf-name*	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
Step 6	switch(config-router-vrf)#**neighbor** *ip-address* **remote-as** *as-number*	Configures the IP address and AS number for a remote BGP peer.
Step 7	(Optional) switch(config-router-neighbor-af)# **copy running-config startup-config**	Saves this configuration change.

Example

This example shows how to create a VRF and configure the router ID in the VRF:

```
switch# configure terminal

switch(config)# vrf context NewVRF
switch(config-vrf)# exit
switch(config)# router bgp 65535
switch(config-router)# vrf NewVRF
switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535
switch(config-router-vrf-neighbor)# copy running-config startup-config
```

11.8 Verifying the Advanced BGP Configuration

To display the BGP configuration, perform one of the following tasks:

Command	Purpose
show bgp all [**summary**] [**vrf** *vrf-name*]	Displays the BGP information for all address families.
show bgp convergence vrf *vrf-name*	Displays the BGP information for all address families.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community** {**regexp** *expression* | [**community**] [**no-advertise**] [**no-export**] [**no-export-subconfed**]} [**vrf** vrf-name]	Displays the BGP routes that match a BGP community.
show bgp [**vrf** *vrf-name*] {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **community-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP community list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity** {**regexp** *expression* | **generic** [**non-transitive** | **transitive**] *aa4:nn* [**exact-match**]} [**vrf** *vrf-name*]	Displays the BGP routes that match a BGP extended community.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **extcommunity-list** *list-name* [**exact-match**]} [**vrf** vrf-name]	Displays the BGP routes that match a BGP extended community list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**dampening dampened-paths** [**regexp** *expression*]} [**vrf** vrf-name]	Displays the information for BGP route dampening. Use the **clear bgp dampening** command to clear the route flap dampening information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **history-paths** [**regexp** *expression*] [**vrf** *vrf-name*]	Displays the BGP route history paths.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **filter-list** *list-name* [**vrf** *vrf-name*]	Displays the information for the BGP filter list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **neighbors** [*ip-address* | *ipv6-prefix*] [**vrf** *vrf-name*]	Displays the information for BGP peers. Use the **clear bgp neighbors** command to clear these neighbors.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] {**nexthop** | **nexthop-database**} [**vrf** *vrf-name*]	show bgp {ipv4 | ipv6 | vpnv4 | vpnv6} {unicast | multicast} [ip-address | ipv6-prefix] {nexthop | nexthop-database} [vrfvrf-name]

show bgp paths	Displays the BGP path information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **policy** *name* [**vrf** *vrf-name*]	Displays the BGP policy information. Use the **clear bgp policy** command to clear the policy information.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **prefix-list** *list-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the prefix list.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **received-paths** [**vrf** *vrf-name*]	Displays the BGP paths stored for soft reconfiguration.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **regexp** *expression* [**vrf** *vrf-name*]	Displays the BGP routes that match the AS_path regular expression.
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **route-map** *map-name* [**vrf** *vrf-name*]	Displays the BGP routes that match the route map.
show bgp peer-policy *name* [**vrf** *vrf-name*]	Displays the information about BGP peer policies.
show bgp peer-session *name* [**vrf** *vrf-name*]	Displays the information about BGP peer sessions.
show bgp peer-template *name* [**vrf** *vrf-name*]	Displays the information about BGP peer templates. Use the **clear bgp peer-template** command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **bgp** *options*	Displays the BGP status and configuration information. This command has multiple options. See the *Inspur CN12700 Series INOS Unicast Routing Command Reference*, for more information.
show {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} **mbgp** *options*	Displays the BGP status and configuration information. This command has multiple options. See the *Inspur CN12700 Series INOS Unicast Routing Command Reference*, for more information.
show running-configuration bgp	Displays the current running BGP configuration.

11.9 Displaying Advanced BGP Statistics

To display advanced BGP statistics, use the following commands:

Command	Purpose
show bgp {**ipv4** | **ipv6** | **vpnv4** | **vpnv6**} {**unicast** | **multicast**} [*ip-address* | *ipv6-prefix*] **flap-statistics** [**vrf** *vrf-name*]	Displays the BGP route flap statistics. Use the **clear bgp flap-statistics** command to clear these statistics.
show bgp sessions [**vrf** *vrf-name*]	Displays the BGP sessions for all peers. Use the clear bgp sessions command to clear these statistics.
show bgp statistics	Displays the BGP statistics.

11.10 Related Documents

Related Topic	Document Title
BGP CLI commands	**Inspur CN12700 Series INOS Unicast Routing Command Reference**
VDCs and VRFs	**Inspur CN12700 Series INOS Virtual Device Context Configuration Guide**

11.11 RFCs

RFC	Title
RFC 2918	-

11.12 MIBs

MIBs	MIBs Link
BGP4-MIB - INSPUR-BGP4-MIB INSPUR-BGP-MIBv2	

11.13 Feature History for Advanced BGP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 21 : Feature History for Advanced BGP

Feature Name	Release	Feature Information
ECMP	8.4(1)	Added support for up to 64 paths to a destination. Supported on F3-Series I/O modules.
BGP	8.4(1)	Added support for exporting routes to Default VRF

BGP	8.4(1)	Added support for INSPUR-BGP-MIBv2
BGP	8.4(1)	Added support for RFC 5549
BGP Next Hop Unchanged	8.4(1)	Introduced this feature.
BGP	8.4(1)	Added BFD support for the IPv6 address family.
BGP	8.4(1)	Added the ability to configure BGP to advertise the default route and introduced the **default-information originate** command.
BGP	8.4(1)	Added the ability to advertise routes that are suppressed by the **aggregate-address** command.
Policy-based administrative distance	8.4(1)	Introduced this feature.
BGP conditional route injection	8.4(1)	Introduced this feature.
BGP AS-path multipath relax	8.4(1)	Added the **as-path multipath-relax** option to the **bestpath** command.
BGP outbound route-maps	8.4(1)	Added support for setting next-hops on reflected routes using an outbound route-map.
BGP cost community ignore	8.4(1)	Added the **cost-community ignore** option to the **bestpath** command.
VPN address families	8.4(1)	Added support for VPN address families.
BGP	8.4(1)	No change from Release 5.0.
BFD	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide* for more information.
ISSU	8.4(1)	Lowered the BGP minimum hold-time check to eight seconds.
Next-hop addressing	8.4(1)	Added support for the BGP next-hop address tracking and filtering.
4-Byte AS numbers	8.4(1)	Added support for 4-byte AS numbers in plaintext notation.
Conditional advertisement	8.4(1)	Added support for conditionally advertising BGP routes based on the existence of other routes in the BGP table.
Dynamic AS number for prefix peers	8.4(1)	Added support for a range of AS numbers for the BGP prefix peer configuration.
BGP	8.4(1)	This feature was introduced.

CHAPTER 12 Configuring RIP

This chapter contains the following sections:

- Finding Feature Information.
- Information About RIP.
- Licensing Requirements for RIP.
- Prerequisites for RIP.
- Guidelines and Limitations for RIP.
- Default Settings for RIP Parameters.
- Configuring RIP.
- Verifying the RIP Configuration.
- Displaying RIP Statistics.
- Configuration Examples for RIP.
- Related Documents for RIP.
- Standards for RIP.
- Feature History for RIP.

12.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

12.2 Information About RIP

RIP uses User Datagram Protocol (UDP) data packets to exchange routing information in small internetworks. RIPv2 supports IPv4. RIPv2 uses an optional authentication feature supported by the RIPv2 protocol.

RIP uses the following two message types:

- Request—Sent to the multicast address 224.0.0.9 to request route updates from other RIP-enabled routers.
- Response—Sent every 30 seconds by default. The router also sends response messages after it receives a request message. The response message contains the entire RIP route table. RIP sends multiple response packets for a request if the RIP routing table cannot fit in one response packet.

RIP uses a hop count for the routing metric. The hop count is the number of routers that a packet can traverse before reaching its destination. A directly connected network has a metric of 1; an unreachable network has a metric of 16. This small range of metrics makes RIP an unsuitable routing protocol for large networks.

12.2.1 RIPv2 Authentication

You can configure authentication on RIP messages to prevent unauthorized or invalid routing updates in your network.Inspur INOSsupports a simple password or an MD5 authentication digest.

You can configure the RIP authentication per interface by using key-chain management for the authentication keys. Key-chain management allows you to control changes to the authentication keys used by an MD5 authentication digest or simple text password authentication. See the Inspur CN12700 Series INOS Security Configuration Guide, for more details about creating key-chains.

To use an MD5 authentication digest, you configure a password that is shared at the local router and all remote RIP neighbors. Inspur INOS creates an MD5 one-way message digest based on the message itself and the encrypted password and sends this digest with the RIP message (Request or Response). The receiving RIP neighbor validates the digest by using the same encrypted password. If the message has not changed, the calculation is identical and the RIP message is considered valid.

An MD5 authentication digest also includes a sequence number with each RIP message to ensure that no message is replayed in the network.

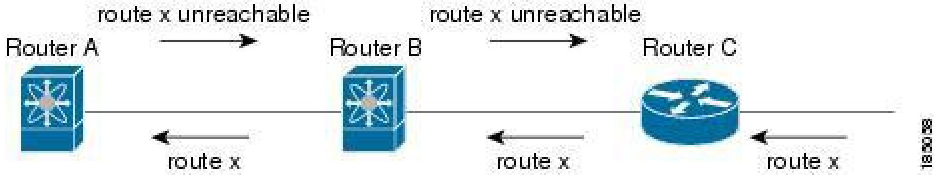
12.2.2 Split Horizon

You can use split horizon to ensure that RIP never advertises a route out of the interface where it was learned.

Split horizon is a method that controls the sending of RIP update and query packets. When you enable split horizon on an interface, Inspur INOS does not send update packets for destinations that were learned from this interface. Controlling update packets in this manner reduces the possibility of routing loops.

You can use split horizon with poison reverse to configure an interface to advertise routes learned by RIP as unreachable over the interface that learned the routes.

Figure 38 : Sample RIP Network with Split Horizon Poison Reverse Enabled



Router C learns about route X and advertises that route to Router B. Router B in turn advertises route X to Router A, but sends a route X unreachable update back to Router C.

By default, split horizon is enabled on all interfaces.

12.2.3 Route Filtering

You can configure a route policy on a RIP-enabled interface to filter the RIP updates. Inspur INOS updates the route table with only those routes that the route policy allows.

12.2.4 Route Summarization

You can configure multiple summary aggregate addresses for a specified interface. Route summarization simplifies route tables by replacing a number of more-specific addresses with an address that represents all the specific addresses. For example, you can replace 10.1.1.0/24, 10.1.2.0/24, and 10.1.3.0/24 with one summary address, 10.1.0.0/16.

If more specific routes are in the routing table, RIP advertises the summary address from the interface with a metric equal to the maximum metric of the more specific routes.

12.2.5 Route Redistribution

You can use RIP to redistribute static routes or routes from other protocols. You must configure a route map with the redistribution to control which routes are passed into RIP. A route policy allows you to filter routes based on attributes such as the destination, origination protocol, route type, route tag, and so on.

Whenever you redistribute routes into a RIP routing domain, Inspur INOS does not, by default, redistribute the default route into the RIP routing domain. You can generate a default route into RIP, which can be controlled by a route policy.

You also configure the default metric that is used for all imported routes into RIP.

12.2.6 Load Balancing

You can use load balancing to allow a router to distribute traffic over all the router network ports that are the same distance from the destination address. Load balancing increases the usage of network segments and increases effective network bandwidth.

Inspur INOS supports the Equal Cost Multiple Paths (ECMP) feature with up to 16 equal-cost paths in the RIP route table and the unicast RIB. You can configure RIP to load balance traffic across some or all of those paths.

12.2.7 High Availability for RIP

Inspur INOS supports stateless restarts for RIP. After a reboot or supervisor switchover, Inspur INOS applies the running configuration and RIP immediately sends request packets to repopulate its routing table.

12.2.8 Virtualization Support

Inspur INOS supports multiple instances of the RIP protocol that run on the same system. RIP supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs).

You can configure up to four RIP instances on a VDC. By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

See the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

12.3 Licensing Requirements forRIP

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the *Inspur INOS licensing scheme, see the Inspur INOS Licensing Guide.*

12.4 Prerequisites for RIP

- You must enable RIP.
- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur INOS Virtual Device Context Configuration Guide*).

12.5 Guidelines and Limitations for RIP

- Inspur INOS does not support RIPv1. if Inspur INOS receives a RIPv1 packet, it logs a message and drops the packet.
- Inspur INOS does not establish adjacencies with RIPv1 routers.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

12.6 Default Settings for RIP Parameters

Default RIP Parameters

Parameters	Default
Maximum paths for load balancing	8
RIP feature	Disabled
Split horizon	Enabled

12.7 Configuring RIP

12.7.1 Enabling RIP

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] feature rip**
3. (Optional) switch(config)# **copy running-config startup-config**
4. (Optional) switch(config)# **show feature**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature rip**	Enables the RIP feature. Use the **no** form of this command to disable this feature.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	(Optional) switch(config)# **show feature**	Displays enables and disabled features.

Example

The following example enables RIP:

```
switch          #          configure          terminal

switch(config)# feature rip

switch(config)#  copy  running-config  startup-config
```

12.7.2 Creating a RIP Instance

You can create a RIP instance and configure the address family for that instance.

Before you begin

You must enable RIP.

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# [**no**] **router rip** *instance-tag*	Creates a new RIP instance with the configured instance-tag. Use the **no** form of this command to disable this feature. **Note** You must also remove any RIP commands configured in interface mode.
Step 3	switch(config-router)# **address-family ipv4 unicast**	Configures the address family for this RIP instance and enters address-family configuration mode.
Step 4	(Optional) switch(config-router-af)# **show ip rip** [**instance** *instance-tag*] [**vrf** *vrf-name*]	Displays a summary of RIP information for all RIP instances.
Step 5	(Optional) switch(config-router-af)# **distance** *value*	Sets the administrative distance for RIP, in address-family configuration mode. The range is from 1 to 255.
Step 6	(Optional) switch(config-router-af)# **maximum-paths** *number*	Configures the maximum number of equal-cost paths that RIP maintains in the route table, in address-family configuration mode. The range is from 1 to 16.
Step 7	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example creates a RIP instance for IPv4 and sets the number of equal-cost paths for load balancing:

```
switch# configure terminal
switch(config)# router rip Enterprise
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# max-paths 10
switch(config-router-af)#  copy  running-config  startup-config
```

12.7.3 Restarting a RIP Instance

You can restart a RIP instance and remove all associated neighbors for the instance.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **restart rip** *instance-tag*	Restarts the RIP instance and removes all neighbors.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example restarts a RIP instance:

```
switch # configure terminal

switch(config)# restart rip Enterprise

switch(config)#  copy  running-config  startup-config
```

12.7.4 Configuring RIP on an Interface

Before you begin

- You must enable RIP.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot*/*port*
3. switch(config-if)# **ip router rip** *instance-tag*
4. (Optional) switch(config-if)# **copy running-config startup-config**
5. (Optional) switch(config-if)# **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*] [**detail**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot*/*port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip router rip** *instance-tag*	Associates this interface with a RIP instance.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	(Optional) switch(config-if)# **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*] [**detail**]	Displays RIP information for an interface.

Example

The following example configures RIP on an Ethernet interface:

```
switch  #  configure  terminal

switch(config)# interface ethernet 1/2
switch(config-if)# no switchport
switch(config-if)#  ip router rip Enterprise

switch(config-if)# show ip rip Enterprise ethernet 1/2

switch(config-if)# copy running-config startup-config
```

12.7.5 Configuring RIP Authentication

You can configure authentication for RIP packets on an interface.

Before you begin

- You must enable RIP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).
- Configure a keychain if necessary before enabling authentication. For details about implementing key chains, see the *Inspur CN12700 Series INOS Security Configuration Guide.*

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **ip rip authentication mode** {**text** | **md5**}
4. switch(config-if)# **ip rip authentication keychain** *key*
5. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip rip authentication mode** {**text** | **md5**}	Sets the authentication type for RIP on this interface as cleartext or MD5 authentication digest.
Step 4	switch(config-if)# **ip rip authentication keychain** *key*	Configures the authentication key used for RIP on this interface.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example creates a key chain and configures MD5 authentication on a RIP interface:

```
switch# configure terminal
switch(config)#  key chain RIPKey

switch(config)# key-string myrip

switch(config)# accept-lifetime 00:00:00 Jan 01 2000 infinite

switch(config)# send-lifetime 00:00:00 Jan 01 2000 infinite

switch(config)# interface ethernet 1/2

switch(config-if)#  ip  rip  authentication mode  md5
switch(config-if)#  ip  rip authentication  keychain RIPKey
switch(config-if)#  copy  running-config startup-config
```

12.7.6 Configuring a Passive Interface

You can configure a RIP interface to receive routes but not send route updates by setting the interfaces to passive mode. You can configure a RIP interface in passive mode in the interface configuration mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot*/*port*
3. switch(config-if)# **ip rip passive-interface**
4. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot*/*port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip rip passive-interface**	Sets the interface into passive mode.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example configures a RIP interface in passive mode:

```
switch  #  configure  terminal
switch(config)#          interface   ethernet    1/2
switch(config-if)# ip rip passive-interface

switch(config-if)# copy running-config startup-config
```

12.7.7 Configuring Split Horizon with Poison Reverse

You can configure an interface to advertise routes learned by RIP as unreachable over the interface that learned the routes by enabling poison reverse. You can configure split horizon with poison reverse on an interface using the interface configuration mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot*/*port*
3. switch(config-if)# **ip rip poison-reverse**
4. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot*/*port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip rip poison-reverse**	Enables split horizon with poison reverse. Split horizon with poison reverse is disabled by default.

Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example restarts a RIP instance:

```
switch # configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip rip poison-reverse
switch(config-if)# copy running-config startup-config
```

12.7.8 Configuring Route Summarization

You can create aggregate addresses that are represented in the routing table by a summary address. Inspur INOS advertises the summary address metric that is the smallest metric of all the more-specific routes. To configure a summary address on an interface, use the interface configuration mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **ip rip summary-address** *ip-prefix/mask-len*
4. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip rip summary-address** *ip-prefix/mask-len*	Configured a summary address for RIP for IPv4 addresses.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example restarts a RIP instance:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)# ip router rip summary-address 192.0.2.0/24

switch(config-if)# copy running-config startup-config
```

12.7.9 Configuring Route Redistribution

You can configure RIP to accept routing information from another routing protocol and redistribute that information through the RIP network. Redistributed routes can optionally be assigned a default route.

Before you begin

- You must enable RIP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).
- Configure a route map before configuring redistribution.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router rip** *instance-tag*
3. switch(config-router)# **address-family ipv4 unicast**
4. switch(config-router-af)# **redistribute** {**bgp** *as* | **direct** | **eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
5. (Optional) switch(config-router-af)# **default-information originate** [**always**] [**route-map** *map-name*]
6. (Optional) switch(config-router-af)# **default-metric** *value*
7. (Optional) switch(config-router-af)# **copy running-config startup-config**
8. (Optional) switch(config-router-af)# **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**]] [**vrf** *vrf-name*] [**summary**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router rip** *instance-tag*	Creates a new RIP instance with the configured instance-tag.
Step 3	switch(config-router)# **address-family ipv4 unicast**	Enters address family configuration mode.
Step 4	switch(config-router-af)# **redistribute** {**bgp** *as* | **direct** | **eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*	Redistributes routes from other protocols into RIP.
Step 5	(Optional) switch(config-router-af)# **default-information originate** [**always**] [**route-map** *map-name*]	Generates a default route into RIP, optionally controlled by a route map.
Step 6	(Optional) switch(config-router-af)# **default-metric** *value*	Sets the default metric for all redistributed routes. The range is from 1 to 15. The default is 1.
Step 7	(Optional) switch(config-router-af)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config-router-af)# **show ip rip route** [*ip-prefix* [**longer-prefixes** | **shorter-prefixes**]] [**vrf** *vrf-name*] [**summary**]	Shows the routes in RIP.

Example

The following example shows how to redistribute EIGRP into RIP:

```
switch# configure terminal

switch(config)# router rip Enterprise

switch(config-router)# address-family ipv4 unicast

switch(config-router-af)# redistribute eigrp 201 route-map RIPmap

switch(config-router-af)#  copy  running-config  startup-config
```

12.7.10 Configuring Inspur INOS RIP for Compatibility with Inspur IOS RIP

Beginning with Inspur INOS Release 8.4(1), you can configure Inspur INOS RIP to behave like Inspur IOS RIP in the way that routes are advertised and processed.

Directly connected routes are treated with cost 1 in Inspur INOS RIP and with cost 0 in Inspur IOS RIP. When routes are advertised in Inspur INOS RIP, the receiving device adds a minimum cost of +1 to all received routes and installs the routes in its routing table. In Inspur IOS RIP, this cost increment is done on the sending router, and the receiving router installs the routes without any modification. This difference in behavior can cause issues when both Inspur INOS and Inspur IOS devices are working together. You can prevent these compatibility issues by configuring Inspur INOS RIP to advertise and process routes like Inspur IOS RIP

Before you begin

- You must enable RIP.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **router rip** *instance-tag*
3. switch(config-router)# [**no**] **metric direct 0**
4. (Optional) switch(config-router)# **show running-config rip**
5. (Optional) switch(config-router)# **copy running-config startup config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router rip** *instance-tag*	Creates a new RIP instance with the configured instance-tag. You can enter 100, 201, or up to 20 alphanumeric chapters for the instance tag.
Step 3	switch(config-router)# [**no**] **metric direct 0**	Configures all directly connected routes with cost 0 instead of the default of cost 1 in order to make Inspur INOS RIP compatible with Inspur IOS RIP in the way that routes are advertised and processed. **Note** This command must be configured on all Inspur INOS devices that are present in any RIP network that also contains Inspur IOS devices.
Step 4	(Optional) switch(config-router)# **show running-config rip**	Displays the current running RIP configuration.
Step 5	(Optional) switch(config-router)# **copy running-config startup config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration

Example

The following example shows how to disable INOS RIP compatibility with Inspur IOS RIP by returning all direct routes from cost 0 to cost 1:

```
switch#       configure      terminal
switch(config)#   router   rip 100
switch(config-router)# no metric direct 0
switch(config-router)# show running-config rip
switch(config-router)#    copy    running-config startup-config
```

12.7.11 Configuring Virtualization

You can configure multiple RIP instances in each VDC. You can also create multiple VRFs within each VDC and use the same or multiple RIP instances in each VRF. You assign a RIP interface to a VRF.

Before you begin

- You must enable RIP.
- Create the VDCs.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf** *vrf-name*
3. switch(config-vrf)# **exit**
4. switch(config)# **router rip** *instance-tag*
5. switch(config-router)# **vrf** *vrf-name*
6. (Optional) switch(config-router-vrf)# **address-family ipv4 unicast**
7. (Optional) switch(-router-vrf-af)# **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*
8. switch(config-router-vrf-af)# **interface ethernet** *slot*/*port*
9. switch(config-if)# **no switchport**
10. switch(config-if)# **vrf member** *vrf-name*
11. switch(config-if)# **ip-address** *ip-prefix*/*length*
12. switch(config-if)# **ip router rip** *instance-tag*
13. (Optional) switch(config-if)# **copy running-config startup-config**
14. (Optional) switch(config-if)# **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf** *vrf-name*	Creates a new VRF.
Step 3	switch(config-vrf)# **exit**	Exits VRF configuration mode.
Step 4	switch(config)# **router rip** *instance-tag*	Creates a new RIP instance with the configured instance tag.
Step 5	switch(config-router)# **vrf** *vrf-name*	Creates a new VRF and enters VRF configuration mode.
Step 6	(Optional) switch(config-router-vrf)# **address-family ipv4 unicast**	(Optional) Configures the VRF address family for this RIP instance.

Step 7	(Optional) switch(-router-vrf-af)# **redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} **route-map** *map-name*	-
Step 8	switch(config-router-vrf-af)#**interfaceethernet***slot*/*port*	Enters interface configuration mode.
Step 9	switch(config-if)# **no switchport**	Configures the interface as a Layer 3 routed interface.
Step 10	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 11	switch(config-if)# **ip-address** *ip-prefix*/*length*	Configures an IP address for this interface. You must perform this step after you assign this interface to a VRF.
Step 12	switch(config-if)# **ip router rip** *instance-tag*	Associates this interface with a RIP instance.
Step 13	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 14	(Optional) switch(config-if)# **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot*/*port*] [**vrf** *vrf-name*]	Displays RIP information for an interface in a VRF.

Example
The following example shows how to create a VRF and add an interface to the VRF:

```
switch# configure terminal

switch(config)# vrf context RemoteOfficeVRF

switch(config-vrf)# exit

switch(config)# router rip Enterprise

switch(config-router)# vrf RemoteOfficeVRF

switch(config-router-vrf)#  address-family  ipv4 unicast

switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap

switch(config-router-vrf-af)#    interface ethernet 1/2
switch(config-if)# vrf member RemoteOfficeVRF

switch(config-if)#ip address 192.0.2.1/16
switch(config-if)#ip router rip Enterprise
switch(config-if)#copy running-config startup-config
```

12.7.12 Tuning RIP

You can tune RIP to match your network requirements. RIP uses several timers that determine the frequency of routing updates, the length of time before a route becomes invalid, and other parameters. You can adjust these timers to tune routing protocol performance to better suit your internet work needs.

SUMMARY STEPS

1. (Optional) switch(config-router-af)# **timers basic** *update timeout holddown garbage-collection*
2. switch(config-router-af)# **exit**
3. switch(config-router)# **exit**
4. switch(config)# **interface** *type number*
5. (Optional) switch(config-if)# **ip rip metric-offset** *value*
6. (Optional) switch(config-if)# **ip rip route-filter** {**prefix-list** *list-name* | **route-map** *map-name* | [**in** | **out**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	(Optional) switch(config-router-af)# **timers basic** *update timeout holddown garbage-collection*	**Note** This is set in the address-family configuration mode. Sets the RIP timers in seconds. The parameters are as follows: • *update*—The range is from 5 to any positive integer. The default is 30. • *timeout*—The time that Inspur INOS waits before declaring a route as invalid. If Inspur INOS does not receive route update information for this route before the timeout interval ends, Inspur INOS declares the route as invalid. The range is from 1 to any positive integer. The default is 180. • *holddown*—The time during which Inspur INOS ignores better route information for an invalid route. The range is from 0 to any positive integer. The default is 180. • *garbage-collection*—The time from when Inspur INOS marks a route as invalid until Inspur INOS removes the route from the routing table. The range is from 1 to any positive integer. The default is 120.
Step 2	switch(config-router-af)# **exit**	Exits address-family configuration mode.
Step 3	switch(config-router)# **exit**	Exits router configuration mode.
Step 4	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 5	(Optional) switch(config-if)# **ip rip metric-offset** *value*	**Note** This is set in the interface configuration mode. Adds a value to the metric for every router received on this interface. The range is from 1 to 15. The default is 1.
Step 6	(Optional) switch(config-if)# **ip rip route-filter** {**prefix-list** *list-name*	**route-map** *map-name*	[**in**	**out**]	**Note** This is set in the interface configuration mode. Specifies a route map to filter incoming or outgoing RIP updates.

Example

The following optional examples show how to tune RIP:

```
switch(config-router-af)#  timers  basic  40  120 120 100

switch(config-router-af)#    exit
switch(config-router)#       exit
switch(config)#exit

switch(config)# interface ethernet 1/2

switch(config-if)# ip rip metric-offset 10

switch(config-if)# ip rip route-filter route-map InputMap in
```

12.8 Verifying the RIP Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show ip rip instance [*instance-tag*] [**vrf** *vrf-name*]	Displays the status for an instance of RIP.
show ip rip [**instance** *instance-tag*] **interface** *slot/port* **detail** [**vrf** *vrf-name*]	Displays the RIP status for an interface
show ip rip [**instance** *instance-tag*] **neighbor** [*interface-type number*] [**vrf** *vrf-name*]	Displays the RIP neighbor table
show ip rip [**instance** *instance-tag*] **route** [*ip-prefix/length* [**longer-prefixes** | **shorter-prefixes**]] [**summary**] [**vrf** *vrf-name*]	Displays the RIP route table
show running-configuration rip	Displays the current running RIP configuration.

12.9 Displaying RIP Statistics

Use one of the following commands to display RIP statistics:

Command	Purpose
show ip rip [**instance** *instance-tag*] **policy statistics redistribute** {**bgp** *as* | **direct** | {**eigrp** | **isis** | **ospf** | **ospfv3** | **rip**} *instance-tag* | **static**} [**vrf** *vrf-name*]	Displays the RIP policy status. Use the **clear ip rip policy** command to clear policy statistics.
show ip rip [**instance** *instance-tag*] **statistics** *interface-type number*] [**vrf** *vrf-name*]	Displays the RIP statistics. Use the **clear ip rip statistics** command to clearRIP statistics.

Use the **clear ip rip policy** command to clear policy statistics. Use the **clear ip rip statistics** command to clear RIP statistics.

12.10 Configuration Examples for RIP

This example creates the Enterprise RIP instance in a VRF and adds Ethernet interface 1/2 to this RIP instance. The example also configures authentication for Ethernet interface 1/2 and redistributes EIGRP into this RIP domain.

```
vrf context NewVRF
!
feature rip
router rip Enterprise vrf NewVRF
    address-family ip unicast
     redistribute  eigrp  201
     route-map  RIPmap  max-
     paths 10
!
interface ethernet 1/2 vrf NewVRF
ip address 192.0.2.1/16 ip router rip Enterprise
  ip rip authentication mode md5
  ip rip authentication keychain RIPKey
```

12.11 Related Documents for RIP

Related Topic	**Document Title**
RIP CLI	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*

12.12 Standards forRIP

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

12.13 Feature History for RIP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	**Releases**	**Feature Information**
RIP	8.4(1)	Added the ability to configure Inspur INOS RIP to be behaviorally compatible with Inspur IOS RIP in the way that routes are advertised and processed.
RIP	8.4(1)	This feature was introduced.

CHAPTER 13 Configuring Static Routing

This chapter contains the following sections:

- Finding Feature Information.
- Information About Static Routing.
- Licensing Requirements for Static Routing.
- Prerequisites for Static Routing.
- Guidelines and Limitations for Static Routing.
- Default Settings for Static Routing Parameters.
- Configuring Static Routing.
- Verifying the Static Routing Configuration.
- Related Documents for Static Routing.
- Feature History for Static Routing.

13.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

13.2 Information About Static Routing

Routers forward packets using either route information from route table entries that you manually configure or the route information that is calculated using dynamic routing algorithms.

Static routes, which define explicit paths between two routers, cannot be automatically updated; you must manually reconfigure static routes when network changes occur. Static routes use less bandwidth than dynamic routes. No CPU cycles are used to calculate and analyze routing updates.

You can supplement dynamic routes with static routes where appropriate. You can redistribute static routes into dynamic routing algorithms but you cannot redistribute routing information calculated by dynamic routing algorithms into the static routing table.

You should use static routes in environments where network traffic is predictable and where the network design is simple. You should not use static routes in large, constantly changing networks because static routes cannot react to network changes. Most networks use dynamic routes to communicate between routers but might have one or two static routes configured for special cases. Static routes are also useful for specifying a gateway of last resort (a default router to which all unroutable packets are sent).

13.2.1 Administrative Distance

An administrative distance is the metric used by routers to choose the best path when there are two or more routes to the same destination from two different routing protocols. An administrative distance guides the selection of one routing protocol (or static route) over another, when more than one protocol adds the same route to the unicast routing table. Each routing protocol is prioritized in order of most to least reliable using an administrative distance value.

Static routes have a default administrative distance of 1. A router prefers a static route to a dynamic route because the router considers a route with a low number to be the shortest. If you want a dynamic route to override a static route, you can specify an administrative distance for the static route. For example, if you have two dynamic routes with an administrative distance of 120, you would specify an administrative distance that is greater than 120 for the static route if you want the dynamic route to override the static route.

13.2.2 Directly Connected Static Routes

You must specify only the output interface (the interface on which all packets are sent to the destination network) in a directly connected static route. The router assumes the destination is directly attached to the output interface and the packet destination is used as the next-hop address. The next hop can be an interface, only for point-to-point interfaces. For broadcast interfaces, the next hop must be an IPv4or IPv6 address.

13.2.3 Fully Specified Static Routes

You must specify either the output interface (the interface on which all packets are sent to the destination network) or the next-hop address in a fully specified static route. You can use a fully specified static route when the output interface is a multi-access interface and you need to identify the next-hop address. The next-hop address must be directly attached to the specified output interface.

13.2.4 Floating Static Routes

A floating static route is a static route that the router uses to back up a dynamic route. You must configure a floating static route with a higher administrative distance than the dynamic route that it backs up. In this instance, the router prefers a dynamic route to a floating static route. You can use a floating static route as a replacement if the dynamic route is lost.

13.2.5 Remote Next-Hops for Static Routes

You can specify the next-hop address of a neighboring router which is not directly connected to the router for static routes with remote (non-directly attached) next-hops. If a static route has remote next-hops during data-forwarding, the next-hops are recursively used in the unicast routing table to identify the corresponding directly attached next-hop(s) that have reachability to the remote next-hops.

13.2.6 Reliable Static Routing Backup Using Object Tracking Deployment

You can configure Inspur INOS to initiate a backup connection from an alternative port if the circuit to the primary gateway is interrupted. You can ensure reliable deployment backups in the case of certain catastrophic events, such as an Internet circuit failure or peer device failure.

Reliable static routing backup using object tracking can determine the state of the primary connection without having to enable a dynamic routing protocol. It also provides a reliable backup solution that can be used for critical circuits that must not go down without automatically engaging a backup circuit.

In a typical scenario, the primary interface of the remote router forwards traffic from the remote LAN to the main office. If the router loses the connection to the main office, the status of the tracked object changes from up to down. When this change occurs, the router removes the routing table entry for the primary interface and installs the preconfigured floating static route on the secondary interface. The router’s secondary interface then forwards traffic to the preconfigured destination. The backup circuit can be configured to use the Internet. When the state of the tracked object changes from down to up, the router reinstalls the routing table entry for the primary interface and removes the floating static route for the secondary interface.

IP Service Level Agreements

This feature uses IP service level agreements (IP SLAs), a network monitoring feature set, to generate ICMP pings to monitor the state of the connection to the primary gateway. An IP SLA is configured to ping a target, such as a publicly routable IP address or a target inside the corporate network. The pings are routed from the primary interface only. A track object is created to monitor the status of the IP SLA configuration. The track object informs the client, the static route, if a state change occurs. The preconfigured floating static route on the secondary interface is installed when the state changes from up to down.

For more information on IP SLAs, see the *Inspur CN12700 Series INOS IP SLAs Configuration Guide*.

13.2.7 BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol designed to provide fast forwarding-path failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules. See the , for more information.

13.2.8 Virtualization Support

Static routes support virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

13.3 Licensing Requirements for Static Routing

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

13.4 Prerequisites for Static Routing

If the next-hop address for a static route is unreachable, the static route will not be added to the unicast routing table.

13.5 Guidelines and Limitations for Static Routing

- You can specify an interface as the next-hop address for a static route only for point-to-point interfaces such as generic routing encapsulation (GRE) tunnels.
- Starting from Inspur INOS Release 8.4(1), static IPv6 route with next-hop as the VxLAN route is supported.
- The forward referencing of static routes is not supported for track objects.
- Starting from Inspur INOS Release 8.4(1), IPv6 static routes with next-hops that are learnt over a VXLAN tunnel can be added to the Unicast Routing Information Base (URIB). This feature was supported on IPv4 since Inspur INOS Release 8.4(1).
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

13.6 Default Settings for Static Routing Parameters

Default Static Routing Parameters

Parameters	Default
Administrative distance	1
RIP feature	Disabled

13.7 Configuring Static Routing

13.7.1 Configuring a Static Route for IPv4

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ip route** {*ip-prefix* | *ip-addr/ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value*] [*pref*]
3. (Optional) switch(config)# **copy running-config startup-config**
4. (Optional) switch(config)# **show ip static-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip route** {*ip-prefix* | *ip-addr/ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value*] [*pref*]	Configures a static route and the interface for this static route. Use ? to display a list of supported interfaces. You can specify a null interface by using null 0. You can optionally configure the next-hop address. The preference value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	(Optional) switch(config)# **show ip static-route**	Displays information about static routes.

Configuration Example

Configuring a Static Route for a null interface.

```
switch# configure terminal

switch(config)# ip static-route 1.1.1.1/32 null 0

switch(config)#  copy  running-config  startup-config
```

Use the **no ip static-route** command to remove the static route.

13.7.2 Configuring a Static Route for IPv6

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **ipv6 route** *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*nh-prefix* [*interface*] | *link-local-nh-prefix* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]

3. (Optional) switch(config)# **copy running-config startup-config**
4. (Optional) switch(config)# **show ipv6 static-route**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ipv6 route** *ip6-prefix* {*nh-prefix*| *link-local-nh-prefix*} | {*nh-prefix* [*interface*]| *link-local-nh-prefix* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]	Configures a static route and the interface for this static route. Use ? to display a list of supported interfaces. You can specify a null interface by using null 0. You can optionally configure the next-hop address. The preference value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 4	(Optional) switch(config)# **show ipv6 static-route**	Displays information about static routes.

Example

The following example configures a static route for IPv6:

```
switch# configure terminal

switch(config)# ipv6 route 2001:0DB8::/48 6::6 null 0
```

13.7.3 Configuring a Static Route over a VLAN

You can configure a static route without next hop support over a VLAN, also known as a switch virtual switch (SVI).

Before you begin

Ensure that the access port is part of the VLAN.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature interface-vlan**
3. switch(config)# **interface vlan** *vlan-id*
4. switch(config-if)# **ip address** *ip-addr/length*
5. switch(config-if)# **ip route** *ip-addr/length vlan-id*
6. (Optional) switch(config-if)# **ip route** *ip-addr/length vlan-id next-hop-ip-address*
7. (Optional) switch(config-if)# **show ip route**
8. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature interface-vlan**	Enables VLAN interface mode.

Step 3	switch(config)# **interface vlan** *vlan-id*	Creates a switch virtual inteface (SVI) and enters interface configuration mode. The range for the *vlan-id* argument is from 1 to 4094, except for the VLANs reserved for the internal switch.
Step 4	switch(config-if)# **ip address** *ip-addr/length*	Configures an IP address for the VLAN.
Step 5	switch(config-if)# **ip route** *ip-addr/length vlan-id*	Adds an interface static route without a next hop on the SVI. The IP address is the address that is configured on the interface that is connected to the switch.
Step 6	(Optional) switch(config-if)# **ip route** *ip-addr/length vlan-id next-hop-ip-address*	Configures explicit next hop address when you set up a /32 static route over an interface VLAN. The IP address is the address that is configured on the interface that is connected to the switch.
Step 7	(Optional) switch(config-if)# **show ip route**	Displays routes from the Unicast Route Information Base (URIB).
Step 8	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure a static route without a next hop over an SVI:

```
switch#   configure   terminal

switch(config)# feature interface-vlan
swicth(config)# interface vlan 10
switch(config-if)# ip address 192.0.2.1/8

switch(config-if)# ip route 209.165.200.224/27 vlan 10 <===209,165.200.224
is the IP address of the interface that is configured on the interface that
is directly connected to the switch.

switch(config-if)# copy running-config startup-config
```

This example shows how to configure an explicit next hop when you set up a /32 static route over an interface VLAN:

```
switch#        configure terminal
switch(config)#  feature interface-vlan
swicth(config)# interface vlan 10

switch(config-if)# ip address 209.165.202.128/27

switch(config-if)#  ip  route  209.165.202.130/32  vlan  10 209.165.202.130

switch(config-if)# copy running-config startup-config
```

What to do next

Use the **no ip static-route** command to remove the static route.

13.7.4 Configuring Reliable Static Routing Backup Using Object Tracking

You can configure Inspur INOS to use Internet Control Message Protocol (ICMP) pings to identify when a connection goes down and initiate a backup connection from any alternative port.

Before you begin

- Configure both a primary interface and a backup interface to used for reliable static routing backup.
- Configure an IP SLA with policy-based routing object tracking to be used for reliable static routing backup.
- Configure a routing policy for static routing to be used for reliable static routing backup.
- Create a track object to be associated with the static route using the **track** *object-id* **interface** command
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# {**ip** | **ipv6**} *route ip-prefix ip-mask ip-addr* **track** *object-number*
3. switch(config)# **show** {**ip** | **ipv6**} **static-route track-table**
4. switch(config)# **show track** *track-number*
5. switch(config)# {**ip** | **ipv6**} **route** *network-number network-mask* {*ip-address* | *interface*} [*distance*] [**name** *name*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip** | **ipv6**} *route ip-prefix ip-mask ip-addr* **track** *object-number*	Configures a static route associated with the track object. The object-number argument specifies that the static route is installed only if the configured track object is up.
Step 3	switch(config)# **show** {**ip** | **ipv6**} **static-route track-table**	Displays information about the IPv4 or IPv6 static-route track table.
Step 4	switch(config)# **show track** *track-number*	Displays information about a specific tracked object.
Step 5	switch(config)# {**ip** | **ipv6**} **route** *network-number network-mask* {*ip-address* | *interface*} [*distance*] [**name** *name*]	Configures a floating IPv4 or IPv6 static route on the secondary interface. The network prefix and mask length must be the same as the static route previously configured for the primary interface associated with a track object. The floating static route should have a higher value of preference than the route associated with the track object.

13.7.5 Configuring Virtualization for IPv4

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*

3. switch(config-vrf)# **ip route** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop* | *nh-prefix* | *interface* [*sub-intf-separtor sub-intf-num*] *next-hop* } [**tag** *tag-value*] [*pref*]
4. (Optional) switch(config-vrf)# **copy running-config startup-config**
5. (Optional) switch(config-vrf)# **show ip static-route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a VRF and enters VRF configuration mode.
Step 3	switch(config-vrf)# **ip route** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop* | *nh-prefix* | *interface* [*sub-intf-separtor sub-intf-num*] *next-hop* } [**tag** *tag-value*] [*pref*]	Configures a static route and the interface for this static route. Use ? to display a list of supported interfaces. You can specify a null interface by using null 0. You can optionally configure the next-hop address. The preference value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 4	(Optional) switch(config-vrf)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	(Optional) switch(config-vrf)# **show ip static-route vrf** *vrf-name*	Displays information on static routes.

Example

The following example configures VRF for IPv4.

```
switch # configure terminal

switch(config)# vrf context StaticVrf

switch(config-vrf)#  ip  route  192.0.2.0/8  ethernet  1/2 10.0.0.2

switch(config-vrf)# show running-config startup-config
```

13.7.6 Configuring Virtualization for IPv6

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **vrf context** *vrf-name*
3. switch(config-vrf)# **ipv6 route** *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix* } | {*next-hop* | *link-local-net-hop* | *interface* [*sub-intf-separtor sub-intf-num*] *next-hop* } [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]
4. (Optional) switch(config-vrf)# **copy running-config startup-config**
5. (Optional) switch(config-vrf)# **show ipv6 static-route vrf** *vrf-name*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *vrf-name*	Creates a VRF and enters VRF configuration mode.
Step 3	switch(config-vrf)# **ipv6 route** *ip6-prefix* {*nh-prefix*| *link-local-nh-prefix* } | {*next-hop* | *link-local-net-hop* | *interface*[*sub-intf-separtor sub-intf-num*] *next-hop*} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*]	Configures a static route and the interface for this static route. Use ? to display a list of supported interfaces. You can specify a null interface by using null 0. You can optionally configure the next-hop address. The preference value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 4	(Optional) switch(config-vrf)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	(Optional) switch(config-vrf)# **show ipv6 static-route vrf** *vrf-name*	Displays information on static routes.

Example

The following example configures virtualization for IPv6:

```
switch # configure terminal

switch(config)# vrf context StaticVrf

switch(config-vrf)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1 2b11::2f01:4c

switch(config-vrf)# copy running-config startup-config
```

13.8 Verifying the Static Routing Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show {**ip** | **ipv6**} **static-route**	Displays the configured static routes.
show ipv6 static-route vrf *vrf-name*	Displays static route information for each VRF.
show {**ip** | **ipv6**} **static-route track-table**	Displays information about the IPv4 or IPv6 static-route track table.
show track *track-number*	Displays information about a specific tracked object.

13.9 Related Documents for Static Routing

Related Topic	Document Title
Static Routing CLI	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*

13.10 Feature History for Static Routing

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	**Releases**	**Feature Information**
Static IPv6 Route	8.4(1)	Added support for static IPv6 route with next-hop as the VxLAN route.
Static Route over VLAN	8.4(1)	This feature was introduced.
Reliable static routing backup using object tracking	8.4(1)	This feature was introduced.
Static routing	8.4(1)	Updated for F3 Series modules.
Layer 3 routing using a mixed chassis	8.4(1)	This feature was introduced.
Static routing	8.4(1)	Added the **name** option to the **ip route** command.
BFD	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide*, for more information.
Static routing	8.4(1)	This feature was introduced.

CHAPTER 14 Configuring the Interoperability of Modules for Unicast Routing

This chapter contains the following sections:

- Finding Feature Information.
- Configuring the Interoperability of Modules for Unicast Routing.
- Information About the Interoperability of Modules for Unicast Routing.
- Licensing Requirements for the Interoperability of Modules for Unicast Routing.
- Guidelines and Limitations for the Interoperability of Modules for Unicast Routing.
- Configuring the Interoperability of Modules for Unicast Routing.
- Verifying the Configuration for the Interoperability of Modules for Unicast Routing.
- Configuration Examples for the Interoperability of Modules for Unicast Routing.
- Related Documents for the Interoperability of Modules for Unicast Routing.
- Feature History for the Interoperability of Modules for Unicast Routing.

14.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

14.2 Configuring the Interoperability of Modules for Unicast Routing

This chapter describes how to configure the interoperability of F3 Series module for unicast routing on the Inspur INOS device.

14.3 Information About the Interoperability of Modules for Unicast Routing

A mixed chassis is a Inspur CN12700 Series chassis that contains at least one F3Series module. Because the F3 Series module processes only Layer 2 traffic, you must configure it to pass Layer 3 traffic through the chassis.

14.4 Licensing Requirements for the Interoperability of Modules for Unicast Routing

The interoperability of modules for unicast routing requires no license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide.*

14.4.1 Configuring the Interoperability of Modules for Unicast Routing

To configure a Layer 3 gateway in a mixed chassis, you use the proxy routing functionality. You enable routing on a specific VLAN by configuring a VLAN interface, and the system automatically provides load-balanced routing functionality. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide* for more information about Layer 3 routing and VLAN interfaces.

Before you begin

You must configure a VLAN interface for each VLAN on the F3 Series module that you want to use with the proxy-routing functionality in a mixed chassis.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **hardware proxy layer-3 routing** {**use** | **exclude**} {**module** *mod-number* | **interface** *slot/port*} [**module-type f1**]
3. (Optional) switch(config)# **show hardware proxy layer-3 detail**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **hardware proxy layer-3 routing** {**use** | **exclude**} {**module** *mod-number* | **interface** *slot/port*} [**module-type f1**]	Configures specific modules and physical interfaces on the M Series module to provide the proxy routing on the F3 Series module.
Step 3	(Optional) switch(config)# **show hardware proxy layer-3 detail**	Displays information about the proxy Layer 3 functionality.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

14.5 Verifying the Configuration for the Interoperability of Modules for Unicast Routing

To display the interoperability of modules for unicast routing configuration, perform one of the following tasks:

Command	Purpose
show hardware proxy layer-3 counters {**brief** | **detail**}	Displays the number of packets sent by F3 Series modules to each M Series module for proxy forwarding. **Note** Enter the **clear hardware proxy layer-3 counters** command to clear the counters.
show hardware proxy layer-3 detail	Displays information about proxy routing from an F3 Series module to an M Series module in a chassis that contains both types of modules.

14.6 Configuration Examples for the Interoperability of Modules for Unicast Routing

This example shows how to specify physical interfaces on M Series modules to perform proxy routing on F3 Series modules in a mixed chassis:

```
switch# configure terminal

switch(config)# hardware proxy layer-3 routing use module 1, 7

switch(config)# show hardware proxy layer-3 detail
```

14.7 Related Documents for the Interoperability of Modules for Unicast Routing

Related Topic	Document Title
Interoperability of modules for unicast routing CLI	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*

14.8 Feature History for the Interoperability of Modules for Unicast Routing

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 22 : Feature History for the Interoperability of Modules for Unicast Routing

Feature Name	Release	Feature Information
Interoperability of modules for unicast routing	8.4(1)	Added support for F3 Series modules.
Interoperability of modules for unicast routing	8.4(1)	This feature was introduced.

CHAPTER 15 Configuring Layer 3 Virtualization

This chapter contains the following sections:

- Finding Feature Information.
- Information About Layer 3 Virtualization.
- Licensing Requirements for VRFs.
- Prerequisites for VRF.
- Guidelines and Limitations for VRF.
- Default Settings for VRF.
- Configuring VRFs.
- Verifying the VRF Configuration.
- Configuration Examples for VRF.
- Related Documents for VRF.
- Standards for VRF.
- Feature History for VRF.

15.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

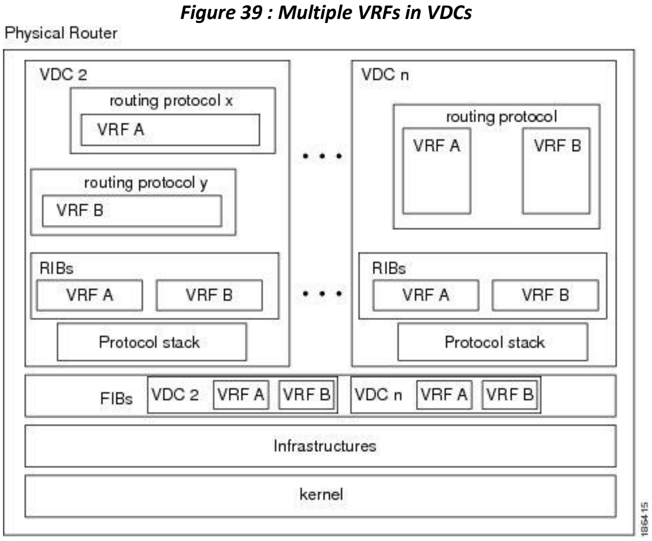
15.2 Information About Layer 3 Virtualization

Inspur INOS supports a hierarchy of virtualization that can divide the physical system resources into multiple virtual device contexts (VDCs). Each VDC acts as a standalone device with both Layer 2 and Layer 3 services available. You can configure up to 4 VDCs, including the default VDC. See the Inspur CN12700 Series INOS Virtual Device Context Configuration Guide, Release 8.4(1), for more information on VDCs.

Inspur INOS further virtualizes each VDC to support virtual routing and forwarding instances (VRFs). You can configure multiple VRFs in a VDC. Each VRF contains a separate address space with unicast and multicast route tables for IPv4 and IPv6 and makes routing decisions independent of any other VRF.

The figure shows multiple independent VRFs in two different VDCs.

Figure 39 : Multiple VRFs in VDCs



A VRF name is local to a VDC, so you can configure two VRFs with the same name if the VRFs exist in different VDCs. In Figure 14-1, VRF A in VDC 2 is independent of VRF B and VRF A in VDC n.

Each router has a default VRF and a management VRF. All Layer 3 interfaces and routing protocols exist in the default VRF until you assign them to another VRF. The mgmt0 interface exists in the management VRF and is shared among multiple VDCs. Each VDC has a unique IP address for the mgmt0 interface (see the Inspur CN12700 Series INOS Fundamentals Configuration Guide, Release 8.4(1).

Management VRF

- The management VRF is for management purposes only.
- Only the mgmt 0 interface can be in the management VRF.
- The mgmt 0 interface cannot be assigned to another VRF.
- The mgmt 0 interface is shared among multiple VDCs.
- No routing protocols can run in the management VRF (static only).

Default VRF

- All Layer 3 interfaces exist in the default VRF until they are assigned to another VRF.
- Routing protocols run in the default VRF context unless another VRF context is specified.
- The default VRF uses the default routing context for all show commands.
- The default VRF is similar to the global routing table concept in Inspur IOS.

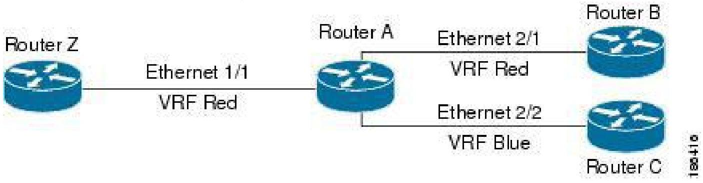
15.2.1 VRF and Routing

All unicast and multicast routing protocols support VRFs. When you configure a routing protocol in a VRF, you set routing parameters for the VRF that are independent of routing parameters in another VRF for the same routing protocol instance.

You can assign interfaces and route protocols to a VRF to create virtual Layer 3 networks. An interface exists in only one VRF. Figure 9-1 shows one physical network split into two virtual networks with two VRFs.

Routers Z, A, and B exist in VRF Red and form one address domain. These routers share route updates that do not include router C because router C is configured in a different VRF.

Figure 40 : VRFs in a Network



By default, Inspur INOS uses the VRF of the incoming interface to select which routing table to use for a route lookup. You can configure a route policy to modify this behavior and set the VRF that Inspur INOS uses for incoming packets.

Inspur INOS supports route leaking (import or export) between VRFs, both in VRF lite and MPLS VPN scenarios. VRF lite does not require an MPLS license for route leaking. For more information on route leaking, see the *Inspur CN12700 Series INOS MPLS Configuration Guide.*

15.2.2 VRF-Aware Services

A fundamental feature of the Inspur INOS architecture is that every IP-based feature is VRF aware.

The following VRF-aware services can select a particular VRF to reach a remote server or to filter information based on the selected VRF:

- AAA
- Call Home
- DNS
- GLBP

- HSRP
- HTTP
- NetFlow
- NTP
- RADIUS
- Ping and Traceroute
- SSH
- SNMP
- Syslog
- TACACS+
- TFTP
- VRRP
- XML

See the appropriate configuration guide for each service for more information on configuring VRF support in that service.

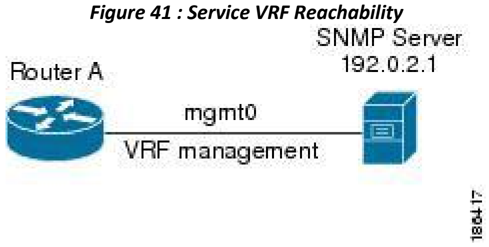
Reachability

Filtering

Reachability indicates which VRF contains the routing information necessary to get to the server providing the service. For example, you can configure an SNMP server that is reachable on the management VRF. When you configure that server address on the router, you also configure which VRF that Inspur INOS must use to reach the server.

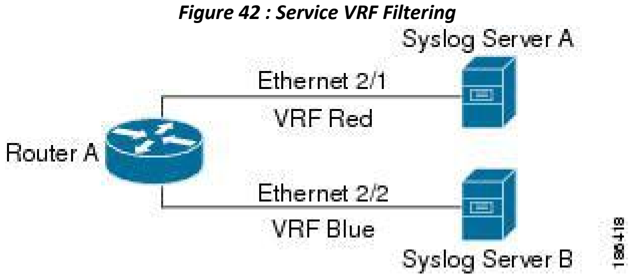
Th figure shows an SNMP server that is reachable over the management VRF. You configure router A to use the management VRF for SNMP server host 192.0.2.1.

Figure 41 : Service VRF Reachability



Filtering allows you to limit the type of information that goes to a VRF-aware service based on the VRF. For example, you can configure a syslog server to support a particular VRF. The figure shows two syslog servers with each server supporting one VRF. syslog server A is configured in VRF Red, so Inspur INOS sends only system messages generated in VRF Red to syslog server A.

Figure 42 : Service VRF Filtering

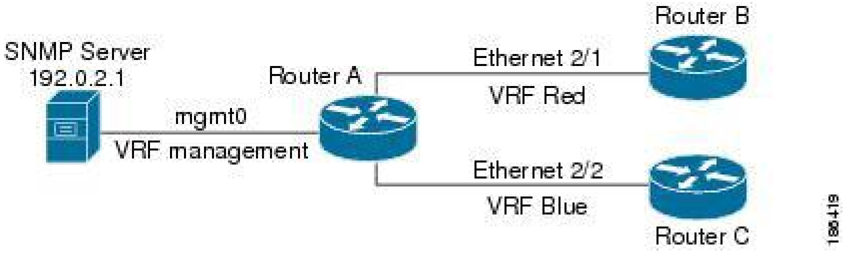


Combining Reachability and Filtering

You can combine reachability and filtering for VRF-aware services. You configure the VRF that Inspur INOS uses to connect to that service as well as the VRF that the service supports. If you configure a service in the default VRF, you can optionally configure the service to support all VRFs.

The figure shows an SNMP server that is reachable on the management VRF. You can configure the SNMP server to support only the SNMP notifications from VRF Red, for example.

Figure 43 : Service VRF Reachability Filtering



15.3 Licensing Requirements for VRFs

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the Inspur INOS Licensing Guide.

15.4 Prerequisites forVRF

You must install the Advanced Services license to use VDCs besides the default VDC.

15.5 Guidelines and Limitations for VRF

- When you make an interface a member of an existing VRF, Inspur INOS removes all Layer 3 configurations. You should configure all Layer 3 parameters after adding an interface to a VRF.
- You should add the mgmt0 interface to the management VRF and configure the mgmt0 IP address and other parameters after you add it to the management VRF.
- If you configure an interface for a VRF before the VRF exists, the interface is operationally down until you create the VRF.
- Inspur INOS creates the default and management VRFs by default. You should make the mgmt0 interface a member of the management VRF.
- The write **erase boot** command does not remove the management VRF configurations. You must use the **write erase** command and then the write **erase boot** command.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

15.6 Default Settings for VRF

Parameters	**Default**
Configured VRFs	Default, management
routing context	Default VRF

15.7 Configuring VRFs

15.7.1 Creating a VRF

Commands available in global configuration mode are also available in VRF configuration mode.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **vrf context** *name*	Creates a new VRF and enters VRF configuration mode. The name can be any case-sensitive, alphanumeric string up to 32 characters.
Step 3	(Optional) switch(config-vrf)# **ip route** {*ip-prefix* | *ip-addr ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]	Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The preference value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 4	(Optional) switch(config-vrf)# **show vrf** [*vrf-name*]	Displays VRF information.
Step 5	switch(config-vrf)# **exit**	Exists the current configuration mode.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a VRF and add a static route to the VRF:

```
switch# configure terminal

switch(config)# vrf context Enterprise

switch(config-vrf)#   ip   route   192.0.2.0/8 ethernet 1/2

switch(config-vrf)# exit

switch(config)#  copy  running-config  startup-config
```

15.7.2 Assigning VRF Membership to an Interface

Before you begin

- Ensure that you are in the correct VDC or use **switchto vdc** command).
- Assign the IP address for an interface after you have configured the interface for a VRF

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrf member** *vrf-name*	Adds this interface to a VRF.
Step 4	switch(config-if)# **ip address** *ip-prefix/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.

Step 5	(Optional) switch(config-vrf)# **show vrf** *vrf-name* **interface** *interface-type number*	Displays VRF information.
Step 6	switch(config-vrf)# **exit**	Exits the current configuration mode.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add an interface to the VRF:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)#       vrf       member   RemoteOfficeVRF
switch(config-if)#  ip address   192.0.2.1/16
switch(config-if)#   copy   running-config   startup-config
```

15.7.3 Configuring VRF Parameters for a Routing Protocol

You can associate a routing protocol with one or more VRFs. See the appropriate chapter for information on how to configure VRFs for routing protocol. This section uses OSPFv2 as an example protocol for the detailed configuration steps.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **router osfp** *instance tag*	Creates a new OSFPv2 instance with the configured instance tag.
Step 3	switch(config-router)# **vrf** *vrf-name*	Enters VRF configuration mode.
Step 4	switch(config-router-vrf)# **maximum-paths** *paths*	(Optional) Configures the maximum number of equal OSPFv2 paths to a destination in the route table for this VRF. Used for load balancing.
Step 5	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 6	switch(config-if)# **ip address** *ip-prefix/length*	Assigns this interface to the OSPFv2 instance and area configured.
Step 7	switch(config-if)# **ip address** *ip-prefix/length*	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 8	switch(config-if)# **ip router ospf area area-id** *instance-tag* **area** *area-id*	Assigns this interface to the OSPFv2 instance and area configured.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add an interface to the VRF:

```
switch# configure terminal

switch(config)# vrf context RemoteOfficeVRF

switch(config-vrf)# exit

switch(config)# router ospf 201

switch(config-router)#  vrf RemoteOfficeVRF
switch(config-router-vrf)# maximum-paths 4

switch(config-router-vrf)# interface      ethernet      1/2
switch(config-if)#   vrf   member RemoteOfficeVRF

switch(config-if)#   ip   address 192.0.2.1/16

switch(config-if)#  ip  router  ospf 201 area 0

switch(config-if)# exit

switch(config)# copy running-config startup-config
```

15.7.4 Configuring VRF Aware Service

You can configure a VRF-aware service for reachability and filtering. See the “VRF-Aware Services” section for links to the appropriate chapter or configuration guide for information on how to configure the service for VRFs. This section uses SNMP and IP domain lists as example services for the detailed configuration steps.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **snmp-server host** *ip-address*[**filter-vrf** *vrf-name*] [**use-vrf** *vrf-name*]	Configures a global SNMP server and configures the VRF that Inspur INOS uses to reach the service. Use the **filter-vrf** keyword to filter information from the selected VRF to this server.
Step 3	switch(config)# **vrf context** *vrf-name*	Creates a new VRF.
Step 4	switch(config-vrf)# **ip domain-list** *domain-name* [**all-vrfs**] [**use-vrf** *vrf-name]*]	Configures the domain list in the VRF and optionally configures the VRF that Inspur INOS uses to reach the domain name listed.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to send SNMP information for all VRFs to SNMP host 192.0.2.1, reachable on VRF Red:

```
switch# configure terminal

switch(config)# snmp-server host 192.0.2.1 for-all-vrfs use-vrf Red

switch(config)#  copy  running-config  startup-config
```

This example shows how to filter SNMP information for VRF Blue to SNMP host 192.0.2.12, reachable on VRF Red:

```
switch# configure terminal

switch(config)# vrf context Blue

switch(config-vrf)# snmp-server host 192.0.2.12 use-vrf Red

switch(config)#  copy  running-config  startup-config
```

15.7.5 Setting the VRF Scope

You can set the VRF scope for all EXEC commands (for example, show commands). This automatically restricts the scope of the output of EXEC commands to the configured VRF. You can override this scope by using the VRF keywords available for some EXEC commands.

To set the VRF scope, use the following command in EXEC mode:

Procedure

	Command or Action	Purpose
Step 1	switch# **routing-context vrf** *vrf-name*	Sets the routing context for all EXEC commands. Default routing context is the default VRF. **Note** To return to the default VRF scope, use the following command in EXEC mode: **routing-context vrf default** Sets the default routing context.

15.8 Verifying the VRF Configuration

To display VRF configuration information, perform one of the following tasks:

SUMMARY STEPS

1. **show vrf** [*vrf-name*]
2. **show vrf** [*vrf-name*] **detail**
3. **show vrf** [*vrf-name*] [**interface** *interface-typeslot/port*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	**show vrf** [*vrf-name*]	Displays the information for all or one VRF.
Step 2	**show vrf** [*vrf-name*] **detail**	Displays detailed information for all or one VRF.
Step 3	**show vrf** [*vrf-name*] [**interface** *interface-typeslot/port*]	Displays the VRF status for an interface.

15.9 Configuration Examples for VRF

This example shows how to configure VRF Red, add an SNMP server to that VRF, and add an instance of OSPF to VRF Red:

```
configure terminal vrf context Red

snmp-server         host
192.0.2.12  use-vrf  Red
router ospf 201

vrf Red

interface ethernet 1/2 vrf member Red

ip address 192.0.2.1/16 ip router ospf 201 area
0
```

This example shows how to configure VRF Red and Blue, add an instance of OSPF to each VRF, and create an SNMP context for each OSPF instance in each VRF:

```
configure terminal

!Create  the  VRFs  vrf  context  Red  vrf  context
Blue vrf context Green

!Create the OSPF instances and associate them with a single VRF or
multiple VRFs (recommended)

feature ospf router ospf Lab vrf Red

!

router ospf Production vrf Blue

router-id 1.1.1.1 vrf Green

router-id 2.2.2.2

!Configure  one  interface  to  use  ospf
Lab on VRF Red interface ethernet 1/2

vrf member Red

ip address 192.0.2.1/16 ip router ospf Lab area
0 no shutdown

!Configure another interface to use ospf Production
on VRF Blue interface ethernet 10/2

vrf member Blue

ip address 192.0.2.1/16

ip    router    ospf
Production  area  0  no
shutdown

!

interface ethernet 10/3 vrf member Green

ip address 192.0.2.1/16
ip    router    ospf
Production  area  0  no
shutdown

!configure the SNMP server

snmp-server  user  admin  network-admin  auth
md5  nbv-12345  snmp-server  community  public
ro

!Create the SNMP contexts for each VRF

snmp-server context lab instance Lab vrf Red

snmp-server context production instance Production vrf Blue

!
```

Use the SNMP context lab to access the OSPF-MIB values for the OSPF instance Lab in VRF Red in this example. Use the SNMP context lab to access the OSPF-MIB values for the OSPF instance Lab in VRF Red in this example.

15.10 Related Documents for VRF

Related Topic	Document Title
VRF CLI	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VRFs	*Inspur CN12700 Series INOS Fundamentals Configuration Guide* *Inspur CN12700 Series INOS MPLS Configuration Guide* *Inspur CN12700 Series INOS System Management Configuration Guide*
VDCs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*

15.11 Standards for VRF

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

15.12 Feature History for VRF

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 23 : Feature History for VRF

Feature Name	Release	Feature Information
VRF	8.4(1)	This feature was introduced.

CHAPTER 16 Managing the Unicast RIB and FIB

This chapter describes how to manage routes in the unicast Routing Information Base (RIB) and the Forwarding Information Base (FIB) on the Inspur INOS device.

- Finding Feature Information.
- Information About the Unicast RIB and FIB.
- Licensing Requirements for the Unicast RIB and FIB.
- Guidelines and Limitations for the Unicast RIB and FIB.
- Default Settings for the Unicast RIB and FIB.
- Managing the Unicast RIB and FIB.
- Verifying the Unicast RIB and FIB.
- Related Documents for the Unicast RIB and FIB.
- Feature History for the Unicast RIB and FIB.

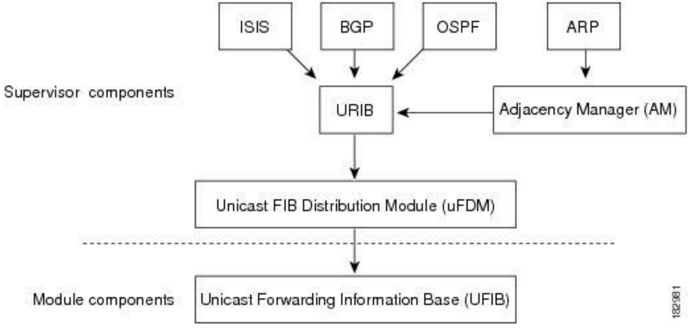
16.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

16.2 Information About the Unicast RIB and FIB

The unicast RIB (IPv4 RIB and IPv6 RIB) and FIB are part of the Inspur INOS forwarding architecture,

Figure 44 : Inspur INOS Forwarding Architecture



The unicast RIB exists on the active supervisor. It maintains the routing table with directly connected routes, static routes, and routes learned from dynamic unicast routing protocols. The unicast RIB also collects adjacency information from sources such as the Address Resolution Protocol (ARP). The unicast RIB determines the best next hop for a given route and populates the unicast forwarding information bases (FIBs) on the modules by using the services of the unicast FIB distribution module (FDM).

Each dynamic routing protocol must update the unicast RIB for any route that has timed out. The unicast RIB then deletes that route and recalculates the best next hop for that route (if an alternate path is available).

16.2.1 Layer 3 Consistency Checker

In rare instances, an inconsistency can occur between the unicast RIB and the FIB on each module. Inspur INOS supports the Layer 3 consistency checker. This feature detects inconsistencies between the unicast IPv4 RIB on the supervisor module and the FIB on each interface module. Inconsistencies include the following:

- Missing prefix
- Extra prefix

- Wrong next-hop address
- Incorrect Layer 2 rewrite string in the ARP or neighbor discovery (ND) cache

The Layer 3 consistency checker compares the FIB entries to the latest adjacency information from the Adjacency Manager (AM) and logs any inconsistencies. The consistency checker then compares the unicast RIB prefixes to the module FIB and logs any inconsistencies.

16.2.2 Dynamic TCAM Allocation

Dynamic TCAM allocation reallocates unused TCAM blocks on F3 Series non-XL modules to an adjacent region when all existing blocks in that region are full. Dynamic TCAM allocation allows more flexibility in the number of routes that the FIB can allocate for a route type.

Inspur INOS divides the FIB to support multiple address families. The FIB TCAM for F3 Series non-XL modules has 128K physical entries.

Table 24 : Default FIB TCAM Allocation

Region	Default Number of Routes	Number of TCAM Blocks	Entry Size
IPv4 Unicast Routes	56,000	7	72 bits
IPv4 Multicast Routes or IPv6 Unicast Routes	32,000	8	144 bits
IPv6 Multicast Routes	2,000	1	288 bits

16.2.3 Maximum TCAM Entries and FIB Scale Limits

The FIB TCAM entries are system wide resources that are shared across virtual device contexts (VDC) configured on the module. Table 16-2 describes the supported maximum FIB scale entries on the CN12700 system configuration per route-type.

Table 25 : Maximum Supported TCAM Entries and FIB Scale Limits

Module Type in a VDC	Maximum TCAM Physical Entries in a VDC	Maximum Supported IPv4 Unicast Routes	Maximum Supported IPv4 Multicast Routes	Maximum Supported IPv6 Unicast Routes	Maximum Supported IPv6 Multicast Routes
Only non-XL modules in a VDC	128,000	112,000	32,000 mroutes	56,000 routes	2,000 routes
Only XL modules in a VDC	900,000	900,000	32,000 mroutes	350,000 routes	2,000 routes
Mix of XL/non-XL modules in the same VDC	128,000	112,000	32,000 mroutes	56,000 routes	2,000 routes
Only F3 Series modules in a VDC[2]	32,000	32,768	16,384 mroutes	16,384 routes	8,192 routes

[2] Utilization may vary based on the sequence of routes being added and on the mix of unicast and multicast routes.

You must install the Scalable Services License (see the Inspur INOS Licensing Guide) and configure the higher shared memory sizes (see the Inspur CN12700 Series INOS Virtual Device Context Configuration Guide, Release 8.4(1) for the routing table to enable the higher FIB scale on the XL modules. See the Inspur CN12700 Series Hardware Installation and Reference Guide for more information on the XL modules.

When you install the Scalable Services license, you may see the following system message:

```
"2011  Mar  30  12:38:13  switch  %PLTFM_CONFIG-4-XL_LICENSE_MIX_NOTIFY:
Mixed  use  of  non-XL  with  XL  modules  in  the  same  VDC  may  limit  common
resources to non-XL capacity."
```

This message occurs if you install the Scalable Services license in a system with non-XL modules or when non-XL modules come on line after you install this license.

The unicast RIB and FIB support virtual routing and forwarding (VRF) instances. VRF exists within VDCs. By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

16.3 Licensing Requirements for the Unicast RIB and FIB

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a compete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*

16.4 Guidelines and Limitations for the Unicast RIB and FIB

When you plan your configuration, consider the following:

- To enable higher FIB sizes for XL modules, you must install the Scalable Services license and configure higher shared memory sizes.

16.5 Default Settings for the Unicast RIB and FIB

Table 26 : Default Unicast RIB and FIB Parameters

Parameters	**Default**
Dynamic TCAM allocation	Enabled by default and cannot be disabled.

16.5.1 Displaying Module FIB Information

The following show commands can be entered in any mode.

Procedure

	Command or Action	**Purpose**
Step 1	switch# **show forwarding** {**ipv4**|**ipv6**} **adjacency module** *slot*	Displays the adjacency information for IPv4 or IPv6.
Step 2	switch# **show forwarding** {**ipv4** | **ipv6**} **route module***slot*	Displays the route table for IPv4 or IPv6.

16.5.2 Configuring Load Sharing in the Unicast FIB

Dynamic routing protocols such as Open Shortest Path First (OSPF) support load balancing with equal-cost multipath (ECMP). The routing protocol determines its best routes based on the metrics configured for the protocol and installs up to the protocol-configured maximum paths in the unicast RIB. The unicast RIB compares the administrative distances of all routing protocol paths in the RIB and selects a best path set from all of the path sets installed by the routing protocols. The unicast RIB installs this best path set into the FIB for use by the forwarding plane.

The forwarding plane uses a load-sharing algorithm to select one of the installed paths in the FIB to use for a given data packet.

You can globally configure the following load-sharing settings:

• Load-share mode—Selects the best path based on the destination address and port or the source and the destination address and port.

• Universal ID—Sets the random seed for the hash algorithm. You do not need to configure the Universal ID. Inspur INOS chooses the Universal ID if you do not configure it.

To configure the unicast FIB load-sharing algorithm, use the following command in global configuration mode:

Command	**Purpose**
switch(config)# **ip load-sharing address** {**destination port destination** | **source-destination** [**port source-destination**]} [**universal-id** *seed*] [**gtp-teid**] [**rotate** *rotate*] [**concatenation**]	Configures the unicast FIB load-sharing algorithm for data traffic. • The **universal-id** option sets the random seed for the hash algorithm and shifts the flow from one link to another. You do not need to configure the universal ID. Inspur INOS chooses the Universal ID if you do not configure it. The universal-id range is from 1 to 4294967295. • The **rotate** option causes the hash algorithm to rotate the link picking selection so that it does not continually choose the same link across all nodes in the network. It does so by influencing the bit pattern for the hash algorithm. This option shifts the flow from one link to another and load balances the already load-balanced (polarized) traffic from the first ECMP level across multiple links. If you specify a rotate value, the 64-bit stream is interpreted starting from that bit position in a cyclic rotation. The rotate range is from 1 to 63, and the default is 32. **Note** With multi-tier Layer 3 topology, polarization is possible. To avoid polarization, use a different rotate bit at each tier of the topology. **Note** To configure a rotation value for port channels, use the port-channel load-balance src-dst ip-l4port rotate rotate command. For more information on this command, see the *Inspur CN12700 Series INOS Interfaces Configuration Guide*.

	• For packets with GTP header, **gtp-teid** specifies that 32-bit TEID value has to be considered for the path calculation. The concatenation option ties together the hash tag values for ECMP and the hash tag values for port channels in order to use a stronger 64-bit hash. If you do not use this option, you can control ECMP load-balancing and port-channel load-balancing independently. The default is disabled.

To display the unicast FIB load-sharing algorithm, use the following command in any mode:

Command	Purpose
switch(config)# **show ip load-sharing**	Displays the unicast FIB load-sharing algorithm for data traffic.

To display the route that the unicast RIB and FIB use for a particular source address and destination address, use the following command in any mode:

Command	Purpose
switch# **show routing hash** *source-addr dest-addr* [*source-port dest-port*] [**vrf** *vrf-name*]	Displays the route that the unicast RIB FIB use for a source and destination address pair. The source address and destination address format is x.x.x.x. The source port and destination port range is from 1 to 65535. The VRF name can be any case-sensitive, alphanumeric string up to 64 characters.

This example shows the route selected for a source/destination pair:

```
switch# show routing hash 10.0.0.5 30.0.0.2

   Load-share parameters used for software forwarding:

   load-share  mode:  address  source-destination  port
   source-destination Universal-id seed: 0xe05e2e85

   Hash for VRF "default"

   Hashing to path *20.0.0.2 (hash: 0x0e), for route:
```

16.5.3 Configuring Per-Packet Load Sharing

You can use per-packet load sharing to evenly distribute data traffic in an IP network over multiple equal-cost connections. Per-packet load sharing allows the router to send successive data packets over paths on a packet-by-packet basis rather than on a per-flow basis.

Per-packet load sharing uses the round-robin method to determine which path each packet takes to the destination. With per-packet load sharing enabled on interfaces, the router sends one packet for destination1 over the first path, the second packet for (the same) destination1 over the second path, and so on. Per-packet load sharing ensures balancing over multiple links.

Use per-packet load sharing to ensure that a path for a single source-destination pair does not get overloaded. If most of the traffic passing through parallel links is for a single pair, per-destination load sharing will overload a single link while other links will have very little traffic. Enabling per-packet load sharing allows you to use alternate paths to the same busy destination.

You configure per-packet load sharing on the input interface. This configuration determines the output interface that Inspur INOS chooses for the packet.

For example, if you have ECMP paths on two output interfaces, Inspur INOS uses the following load-sharing methods for input packets on Ethernet 1/1:

- Per-packet load sharing if you configure per-packet load sharing on Ethernet 1/1.
- Per-flow load sharing.

The configurations for the other interfaces have no effect on the load-sharing method used for Ethernet 1/1 in this example.

Procedure

	Command or Action	Purpose
Step 1	switch(config-if)# **ip load-sharing per-packet**	Configures per-packet load sharing on an interface.

16.5.4 Checking Routes in the Unicast FIB

You can configure a policy in the Embedded Event Manager (EEM) to check for inconsistent, missing, or failed routes in the unicast Forwarding Information Base (FIB).

Before you begin

- You must have network-admin or vdc-admin user privileges to configure EEM.
- Ensure that you are in the correct virtual device context (VDC) or use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **event manager applet** *applet-name*	Registers the applet with EEM and enters applet configuration mode. The *applet-name* argument is a unique identifier for this policy. The maximum range is 29 alphanumeric, case-sensitive characters. **Note** You can use the **no** form of this command to the EEM policy configuration.
Step 3	(Optional) switch(config-applet)# **description** *description*	Configures a descriptive string for the policy. The maximum range is 80 alphanumeric characters. Enclose the string in quotation marks.
Step 4	switch(config-applet)# [**no**] **event fib route** {**inconsistent** | **missing** | **failure**}	Configures an event statement for the policy.

		• The **inconsistent** keyword triggers an event if the route or adjacency programming is changed in the hardware configuration. • The **missing** keyword triggers an event if the route is deleted in the unicast FIB. • The **failure** keyword triggers an event if a route fails to be inserted in the unicast FIB. **Note** You can use the **no** form of this command to remove an event statement from an EEM policy.
Step 5	switch(config-applet)# [**no**] **action** *number* [**.***number2*] *action-statement*	Configures an action statement to describe the action triggered by a policy. Repeat this step to create multiple action statements. • Each policy can have multiple action statements. If no action is associated with a policy, EEM observes events but takes no actions. • The *number***.***number2* argument is a label for the action statement. • The format for the label is number, as in 1, or number.number2, as in 1.0. You must separate the two numbers with a period (.). • The range for the *number* argument is from 0 to any number up to 16 digits in length. • The range for the *number2* argument is from 0 to 9. • Only predefined keywords are supported for the *action-statement* argument. For information, see the *Inspur CN12700 Series INOS System Management Configuration Guide*. • The **event-default** keyword executes the default action for the associated event. The default action for the ternary content addressable memory (TCAM) usage event is to log the event details. • You can configure a different action statement, such as **action 1.0 snmp-trap strdata "inconsistent route"** to send an SNMP trap, for this event. **Note** You can use the **no** form of this command to delete the action statement from an EEM policy.
Step 6	(Optional) switch(config-applet)# **show event manager policy-state** *applet-name*	Displays information about the status of the specified event policy.

Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config)# **show event manager events action-log policy** *applet-name*	Displays the event action log for the specified EEM policy.

16.5.5 Displaying Routing and Adjacency Information

Procedure

	Command or Action	Purpose
Step 1	switch# **show** {**ip** | **ipv6**} **route** [*route-type* | **interface** *int-type number* | **next-hop**]	Displays the unicast route table. The route-type argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the **?** command to see the supported interfaces.
Step 2	switch# **show** {**ip** | **ipv6**} **adjacency** [*prefix* | *interface-type number* [**summary**] | **non-best**] [**detail**] [**vrf** *vrf-id*]	Displays the adjacency table. The argument ranges are as follows: • *prefix*—Any IPv4 or IPv6 prefix address. • *interface-type number*—Use the ? command to see the supported interfaces. • *vrf-id*—Any case-sensitive, alphanumeric string up to 64 characters.
Step 3	switch# **show** {**ip** | **ipv6**} **routing** [*route-type* | **interface** *int-type number* | **next-hop** | **recursive-next-hop** | **summary** | **updated** {**since** | **until**} *time*]	Displays the unicast route table. The route-type argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the **?** command to see the supported interfaces.

Example

The following example displays the unicast route table:

```
switch# show ip route

IP Route Table for Context "default"

'*' denotes best ucast next-hop       '**'   denotes   best
mcast next-hop '[x/y]' denotes [preference/metric]



0.0.0.0/0, 1 ucast next-hops, 0 mcast next-hops

*via  10.1.1.1,  mgmt0,  [1/0],  5d21h,
static 0.0.0.0/32, 1 ucast next-hops,
0 mcast next-hops

*via  Null0,  [220/0],  1w6d,  local,
discard

10.1.0.0/22, 1 ucast next-hops, 0 mcast next-hops, attached

*via 10.1.1.55, mgmt0, [0/0], 5d21h,
direct

10.1.0.0/32,  1  ucast  next-hops,  0  mcast  next-hops,
attached
```

```
   *via 10.1.0.0, Null0, [0/0], 5d21h, local

10.1.1.1/32,  1  ucast  next-hops,  0  mcast  next-hops,
attached

   *via 10.1.1.1, mgmt0, [2/0], 5d16h, am

10.1.1.55/32,  1  ucast  next-hops,  0  mcast  next-hops,
attached

   *via 10.1.1.55, mgmt0, [0/0], 5d21h, local

10.1.1.253/32,  1  ucast  next-hops,  0  mcast  next-hops,
attached

   *via 10.1.1.253, mgmt0, [2/0], 5d20h, am

10.1.3.255/32, 1 ucast next-hops, 0 mcast next-hops, attached
   *via  10.1.3.255,  mgmt0,  [0/0],  5d21h,
local  255.255.255.255/32,  1  ucast  next-
hops, 0 mcast next-hops

   *via Eth Inband Port, [0/0], 1w6d, local
```

The following example shows the adjacency information:

```
switch# show ip adjacency

IP  Adjacency  Table  for
context   default   Total
number of entries: 2
Address         Age       MAC Address     Pref Source     Interface       Best
10.1.1.1        02:20:54  00e0.b06a.71eb  50   arp        mgmt0           Yes
10.1.1.253      00:06:27  0014.5e0b.81d1  50   arp        mgmt0           Yes
```

16.5.6 Triggering the Layer 3 Consistency Checker

You can manually trigger the Layer 3 consistency checker.

To manually trigger the Layer 3 consistency checker, use the following commands in global configuration mode:

Command	**Purpose**
test forwarding [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}]	Starts a Layer 3 consistency check. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters. The *slot* range is from 1 to 10.

To stop the Layer 3 consistency checker, use the following commands in global configuration mode:

Command	**Purpose**
test forwarding [**ipv4** | **ipv6**] [**unicast**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot*| **all**}] **stop**	Stops a Layer 3 consistency check. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters. The *slot* range is from 1 to 10.

To display the Layer 3 inconsistencies, use the following commands in any mode:

Command	**Purpose**
show forwarding [**ipv4** | **ipv6**] **inconsistency** [**vrf** *vrf-name*] [**module** {*slot* | **all**}]	Displays the results of a Layer 3 consistency check. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters. The *slot* range is from 1 to 10.

16.5.7 Clearing Forwarding Information in the FIB

You can clear one or more entries in the FIB. Clearing a FIB entry does not affect the unicast RIB.

⚠

Caution The **clear forwarding** command disrupts forwarding on the device.

Procedure

	Command or Action	Purpose
Step 1	switch# **clear forwarding** {**ipv4** | **ipv6**} **route** {* | *prefix*} [**vrf** *vrf-name*] **module** {*slot* | **all**}	Clears one or more entries from the FIB. The route options are: • *—all routes. • *prefix*—Any IP or IPv6 prefix. The *vrf-name* can be a case-sensitive, alphanumeric string up to 64 characters. The *slot* range is from 1 to 10.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to clear one or more entries from the FIB:

```
switch(config)# clear forwarding ipv4 route *
module 1

switch(config-if)# copy running-config startup-
config
```

16.5.8 Configuring Maximum Routes for the Unicast RIB

You can configure the maximum number of routes allowed in the routing table.

Before you begin

Ensure that you are in the default VDC (or use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch (config)# **vrf context** *vrf name*	Creates a VRF and enters VRF configuration mode.
Step 3	switch (config-vrf)# **ip4 unicast**	Enters address family configuration mode.
Step 4	switch (config vrf-af-ipv4)# **maximum routes** *max routes* [*threshold* [**reinstall** *threshold*] | **warning-only**]	Configures the maximum number of routes allowed in the routing table. You can optionally specify the following:

		• *threshold*—Percentage of maximum routes that triggers a warning message. The range is from 1 to 100. • **warning-only**—Logs a warning message when the maximum number of routes is exceeded. • **reinstall** *threshold*—Reinstalls routes that previously exceeded the maximum route limit and were rejected and specifies the threshold value at which to reinstall them. The threshold range is from 1 to 100.
Step 5	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure maximum routes for the unicast RIB:

```
switch#  configure terminal
switch(config)# vrf  context  Red
switch(config-vrf)#        ipv4 unicast

switch(config-vrf-af-ipv4)# maximum routes 250 90

switch(config-vrf-af-ipv4)#  copy  running-config  startup-config
```

16.5.9 Estimating Memory Requirements for Routes

You can estimate the memory that will be used by a number of routes and next-hop addresses.

Procedure

	Command or Action	Purpose
Step 1	switch# **show routing** {**ipv6**} **memory estimate routes** *num-routes* **next-hops** *num-nexthops*	Displays the memory requirements for routes. The *num-routes* range is from 1000 to 1000000. The *num-nexthops* range is from 1 to 16.
Step 2	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to estimate the memory requirements for routes:

```
switch# show routing memory estimate routes 5000 next-hops

switch(config-if)# copy running-config startup-config
```

16.5.10 Clearing Routes in the Unicast RIB

You can estimate the memory to be used by a number of routes and next-hop addresses.

⚠

Caution The * keyword is severely disruptive to routing.

Procedure

	Command or Action	Purpose
Step 1	**clear** {**ip** | **ip4** | **ipv6**} **route** {* | *prefix/length*} [**next hop interface**] [**vrf** *vrf-name*] **module** {*slot* | **all**}	Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are: • *****—All routes. • **route**— An individual IP or IPv6 route. • *prefix/length*— Any IP or IPv6 prefix. • *next-hop*—The next-hop address. • *interface*—The interface to reach the next-hop address. The *vrf-name* can be an case-sensitive, alphanumeric string up to 32 64characters.
Step 2	**clear routing** [**multicast** | **unicast**]{**ip** |**ip4** | **ipv6**} {*| {*route* |*prefix/length*} [*next-hop interface*]} [**vrf** *vrf-name*]	Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are: • *****—All routes. • **route**— An individual IP or IPv6 route. • *prefix/length*— Any IP or IPv6 prefix. • *next-hop*—The next-hop address. • *interface*—The interface to reach the next-hop address. The *vrf-name* can be any case-sensitive, alphanumeric string up to 64 characters.
Step 3	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

16.5.11 Monitoring TCAM Utilization

You can configure a policy in the Embedded Event Manager (EEM) to monitor ternary content addressable memory (TCAM) utilization on Inspur CN12700 F3-Series Ethernet modules.

Before you begin

- You must have network-admin or vdc-admin user privileges to configure EEM.
- Ensure that you are in the correct virtual device context (VDC) or use the **switchto vdc** command.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# [**no**] **event manager applet** *applet-name*	Registers the applet with EEM and enters applet configuration mode. The *applet-name* argument is a unique identifier for this policy. The maximum range is 29 alphanumeric, case-sensitive characters. **Note** You can use the **no** form of this command to the EEM policy configuration.
Step 3	(Optional) switch(config-applet)# **description** *description*	Configures a descriptive string for the policy. The maximum range is 80 alphanumeric characters. Enclose the string in quotation marks.
Step 4	switch(config-applet)# [**no**] **event fib resource tcam usage**	Configures an event statement for the policy. This command triggers an event each time that the TCAM utilization percentage becomes a multiple of 5, in either direction. **Note** You can use the **no** form of this command to remove an event statement from an EEM policy.
Step 5	switch(config-applet)# [**no**] **action** *number* [*.number2*] *action-statement*	Configures an action statement to describe the action triggered by a policy. Repeat this step to create multiple action statements. • Each policy can have multiple action statements. If no action is associated with a policy, EEM observes events but takes no actions. • The *number***.***number2* argument is a label for the action statement. • The format for the label is number, as in 1, or number.number2, as in 1.0. You must separate the two numbers with a period (.). • The range for the *number* argument is from 0 to any number up to 16 digits in length. • The range for the *number2* argument is from 0 to 9. • Only predefined keywords are supported for the *action-statement* argument. For information, see the *Inspur CN12700 Series INOS System Management Configuration Guide*.

	Command or Action	Purpose
		• The **event-default** keyword executes the default action for the associated event. The default action for the TCAM usage event is to log the event details. • You can configure a different action statement, such as **action 1.0 snmp-trap strdata "TCAM usage percent"** to send an SNMP trap, for this event. **Note** You can use the **no** form of this command to delete the action statement from an EEM policy.
Step 6	(Optional) switch(config-applet)# **show event manager policy-state** *applet-name*	Displays information about the policy state, including thresholds.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config)# **show event manager events action-log policy** *applet-name*	Displays the event action log for the specified EEM policy.

16.6 Verifying the Unicast RIB and FIB

To display advanced BGP statistics, use the following commands:

Command	Purpose
show forwarding adjacency	Displays the adjacency table on a module.
show forwarding distribution {**clients** | **fib-state**}	Displays the FIB distribution information.
show forwarding interfaces module *slot*	Displays the FIB information for a module.
show forwarding {**ip** | **ipv4** | **ipv6**} **route**	Displays routes in the FIB.
show {**ip** |**ipv4** | **ipv6**} **adjacency**}	Displays the adjacency table.
show {**ip** | **ipv6**} **route**}	Displays the IPv4 or IPv6 routes from the unicast RIB.
show routing	Displays routes from the unicast RIB.

16.7 Related Documents for the Unicast RIB and FIB

Feature Name	Feature Information
Unicast RIB and FIB CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*

16.8 Feature History for the Unicast RIB and FIB

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 27 : Feature History for the Unicast RIB and FIB

Feature Name	Release	Feature Information
Load Sharing in Unicast FIB	8.4(1)	Added support for GTP headers.
Unicast FIB	8.4(1)	Added the ability to check for inconsistent, missing, or failed routes in the unicast FIB.
TCAM utilization	8.4(1)	Added the ability to monitor TCAM utilization on F3 Series modules.
Unicast RIB	8.4(1)	Added the optional keyword **longer-prefixes** [**detail**] to the **show routing** command to display specific routes for a particular prefix.
Maximum routes	8.4(1)	Added support to configure the maximum number of routes allowed in the routing table.
TCAM Size for XL Modules	8.4(1)	Added support for larger TCAM and FIB sizes with XL modules.
Dynamic TCAM allocation	8.4(1)	Enabled by default and cannot be disabled.
IPv6 forwarding inconsistency checker	8.4(1)	Added support to check for inconsistencies in the IPv6 forwarding table.
Dynamic TCAM allocation	8.4(1)	Added support for dynamically allocating TCAM blocks in the FIB.
Per-packet load sharing	8.4(1)	Added support to load balance per packet on an interface.
Unicast RIB and FIB	8.4(1)	Added support to clear individual routes in unicast RIB and FIB.
Unicast RIB and FIB	8.4(1)	This feature was introduced.

CHAPTER 17 Configuring Route Policy Manager

This chapter contains the following sections:

- Finding Feature Information.
- Information About Route Policy Manager.
- Licensing Requirements for Route Policy Manager .
- Prerequisites for Route Policy Manager .
- Guidelines and Limitations.
- Default Settings for Route Policy Manager Parameters.
- Configuring Route Policy Manager.
- Configuration Examples for Route Policy Manager.
- Related Documents for Route Policy Manager.
- Standards for Route Policy Manager.
- Feature History for Route Policy Manager.

17.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

17.2 Information About Route Policy Manager

Route Policy Manager supports route maps and IP prefix lists. These features are used for route redistribution. A prefix list contains one or more IPv4 network prefixes and the associated prefix length values. You can use a prefix list by itself in features such as Border Gateway Protocol (BGP) templates, route filtering, or redistribution of routes that are exchanged between routing domains.

Route maps can apply to both routes and IP packets. Route filtering and redistribution pass a route through a route map.

17.2.1 Prefix Lists

17.2.2 MAC Lists

You can use prefix lists to permit or deny an address or range of addresses. Filtering by a prefix list involves matching the prefixes of routes or packets with the prefixes listed in the prefix list. An implicit deny is assumed if a given prefix does not match any entries in a prefix list.

You can configure multiple entries in a prefix list and permit or deny the prefixes that match the entry. Each entry has an associated sequence number that you can configure. If you do not configure a sequence number, Inspur INOS assigns a sequence number automatically. Inspur INOS evaluates prefix lists starting with the lowest sequence number. Inspur INOS processes the first successful match for a given prefix. Once a match occurs, Inspur INOS processes the permit or deny statement and does not evaluate the rest of the prefix list.

Prefix Lists in Inspur INOS support only one of the following addresses at a time:

- source
- destination
- group address

You can use MAC lists to permit or deny a MAC address or range of addresses. A MAC list consists of a list of MAC addresses and optional MAC masks. A MAC mask is a wild-card mask that is logically AND-ed with the

MAC address when the route map matches on the MAC list entry. Filtering by a MAC list involves matching the MAC address of packets with the MAC addresses listed in the MAC list. An implicit deny is assumed if a given MAC address does not match any entries in a MAC list.

You can configure multiple entries in a MAC list and permit or deny the MAC addresses that match the entry. Each entry has an associated sequence number that you can configure. If you do not configure a sequence number, Inspur INOS assigns a sequence number automatically. Inspur INOS evaluates MAC lists starting with the lowest sequence number. Inspur INOS processes the first successful match for a given MAC address. Once a match occurs, Inspur INOS processes the permit or deny statement and does not evaluate the rest of the MAC list.

17.2.3 Route Maps

You can use route maps for route redistribution or policy-based routing. Route map entries consist of a list of match and set criteria. The match criteria specify match conditions for incoming routes or packets, and the set criteria specify the action taken if the match criteria are met.

You can configure multiple entries in the same route map. These entries contain the same route map name and are differentiated by a sequence number.

You create a route map with one or more route map entries arranged by the sequence number under a unique route map name. The route map entry has the following parameters:

- Sequence number
- Permission—permit or deny
- Match criteria
- Set changes

The IPv4 and the IPv6-based matches on the same route map sequence number is not supported in Inspur CN12700 Series.

By default, a route map processes routes or IP packets in a linear fashion, that is, starting from the lowest sequence number. You can configure the route map to process in a different order using the **continue** statement, which allows you to determine which route map entry to process next.

Match Criteria

You can use a variety of criteria to match a route or IP packet in a route map. Some criteria, such as BGP community lists, are applicable only to a specific routing protocol, while other criteria, such as the IP source or the destination address, can be used for any route or IP packet.

When Inspur INOS processes a route or packet through a route map, it compares the route or packet to each of the match statements configured. If the route or packet matches the configured criteria, Inspur INOS processes it based on the permit or deny configuration for that match entry in the route map and any set criteria configured.

The match categories and parameters are as follows:

- IP access lists—(For policy-based routing only). Match based on source or destination IP address, protocol, or QoS parameters.
- BGP parameters—Match based on AS numbers, AS-path, community attributes, or extended community attributes.
- Prefix lists—Match based on an address or range of addresses.
- Multicast parameters—Match based on rendezvous point, groups, or sources.
- Other parameters—Match based on IP next-hop address or packet length.

Set Changes

Once a route or packet matches an entry in a route map, the route or packet can be changed based on one or more configured set statements.

The set changes are as follows:

- BGP parameters—Change the AS-path, tag, community, extended community, dampening, local preference, origin, or weight attributes.
- Metrics—Change the route-metric, the route-tag, or the route-type.

- Policy-based routing only—Change the interface or the default next-hop address.
- Other parameters—Change the forwarding address or the IP next-hop address.

Access Lists

IP access lists can match the packet to a number of IP packet fields such as the following:

- Source or destination IPv4 or IPv6 address
- Protocol
- Precedence
- ToS

You can use ACLs in a route map for policy-based routing only. See the *Inspur CN12700 Series INOS Security Configuration Guide* for more information on ACLs.

AS Numbers for BGP

You can configure a list of AS numbers to match against BGP peers. If a BGP peer matches an AS number in the list and matches the other BGP peer configuration, BGP creates a session. If the BGP peer does not match an AS number in the list, BGP ignores the peer. You can configure the AS numbers as a list, a range of AS numbers, or you can use an AS-path list to compare the AS numbers against a regular expression.

AS-path Lists for BGP

You can configure an AS-path list to filter inbound or outbound BGP route updates. If the route update contains an AS-path attribute that matches an entry in the AS-path list, the router processes the route based on the permit or deny condition configured. You can configure AS-path lists within a route map.

You can configure multiple AS-path entries in an AS-path list by using the same AS-path list name. The router processes the first entry that matches.

Community Lists for BGP

You can filter BGP route updates based on the BGP community attribute by using community lists in a route map. You can match the community attribute based on a community list, and you can set the community attribute using a route map.

A community list contains one or more community attributes. If you configure more than one community attribute in the same community list entry, the BGP route must match all community attributes listed to be considered a match.

You can also configure multiple community attributes as individual entries in the community list by using the same community list name. In this case, the router processes the first community attribute that matches the BGP route, using the permit or deny configuration for that entry.

You can configure community attributes in the community list in one of the following formats:

- A named community attribute, such as **internet** or **no-export.**
- In aa:nn format, where the first two bytes represent the two-byte AS number and the last two bytes represent a user-defined network number.
- A regular expression.

See the *Inspur CN12700 Series INOS Unicast Routing Command Reference for more* information on regular expressions.

Extended Community Lists for BGP

Extended community lists support 4-byte AS numbers. You can configure community attributes in the extended community list in one of the following formats:

- In aa4:nn format, where the first four bytes represent the four-byte AS number and the last two bytes represent a a user-defined network number.
- A regular expression.

See the *Inspur CN12700 Series INOS Unicast Routing Command Reference* for more information on regular expressions.

Inspur INOS supports generic specific extended community lists, which provide similar functionality to regular community lists for four-byte AS numbers. You can configure generic specific extended community lists with the following properties:

- Transitive—BGP propagates the community attributes across autonomous systems.
- Nontransitive—BGP removes community attributes before propagating the route to another autonomous system.

17.2.4 Route Redistribution and Route Maps

You can use route maps to control the redistribution of routes between routing domains. Route maps match on the attributes of the routes to redistribute only those routes that pass the match criteria. The route map can also modify the route attributes during this redistribution using the set changes.

The router matches redistributed routes against each route map entry. If there are multiple match statements, the route must pass all of the match criteria. If a route passes the match criteria defined in a route map entry, the actions defined in the entry are executed. If the route does not match the criteria, the router compares the route against subsequent route map entries. Route processing continues until a match is made or the route is processed by all entries in the route map with no match.

Each ACL ends with an implicit deny statement, by design convention; there is no similar convention for route-maps. If the end of a route-map is reached during matching attempts, the result depends on the specific application of the route-map. Fortunately, route-maps that are applied to redistribution behave the same way as ACLs: if the route does not match any clause in a route-map, then the route redistribution is denied, as if the route-map contained a deny statement at the end.

17.2.5 Route Map Support Matrix for Routing Protocols

The following tables include the configurable match and set statements for routing protocols on Inspur CN12700 Series switches running the latest shipping release.

The following legend applies to the tables:

- Yes—The statement is supported for the protocol.
- No—The statement is not supported for the protocol.
- If a statement does not apply for the protocol, there is an em dash (—) in the column next to the statement.
- Where clarification is required, information is added in the appropriate row/column.

Table 28 : SET Route Map Statements by Protocol

SET Route Map Statement	OSPF Redistribution	EIGRP Redistribution	ISIS Redistribution	RIP Redistribution	BGP Redistribution
Forwarding-address	Yes	—	—	—	—
Standard/Extended Community	—	—	—	—	Standard community only
Site of Origin (SOO)	—	—	—	—	No
Routing Protocol Metric	Yes	Yes	Yes	Yes	Yes
Routing Protocol Metric Type	Yes	—	No	—	—

Route Tag	Yes	Yes	No	Yes	—
NSSA Only	Yes	—	—	—	—
Orgin	—	—	—	—	Yes
Level	—	—	Yes	—	—
Weight	—	—	—	—	Yes

Table 29 : SET Route Map Statements by Protocol

SET Route Map Statement	BGP Neighbor	BGP Table Map	OSPF Table Map	EIGRP Table Map	ISIS Table Map	EIGRP Distribute List
Standard/Extended Community	Yes	No	—	—	—	—
Standard/Extended Community-List Deletion	Yes	No	—	—	—	—
Site of Origin (SOO)	No	—	—	—	—	—
Routing Protocol Metric	No	No	—	—	—	Yes
Routing Protocol Metric Type	Yes	No	—	—	—	—
IPv4 Next Hop	Yes	—	—	—	—	—
IPv6 Next Hop	Yes	—	—	—	—	—
IPv4 Prefix list	Yes	—	—	—	—	—
IPv6 Prefix list	Yes	—	—	—	—	—
Interface	No	—	—	—	—	—
Route Tag	—	—	—	—	—	Yes
AS PATH	Yes	No	—	—	—	—
Orgin	Yes	No	—	—	—	—
All Path Advertisement	Yes	No	—	—	—	—
Distance	—	Yes	Yes	Yes	Yes	—
Dampening	No	No	—	—	—	—
Level	No	No	—	—	—	—
Weight	Yes	Yes	No	—	—	—

Table 30 : SET Route Map Statements by Protocol

MATCH Route Map Statement	OSPF Redistribution	EIGRP Redistribution	ISIS Redistribution	RIP Redistribution	BGP Redistribution
Community List	OSPFv2 only	Yes	yes	Yes	—
Ext Community List	OSPFv2 only	Yes	—	Yes	—
Interface	Yes	Yes	Yes	Yes	Yes
IPv4 Next Hop	Yes	Yes	Yes	Yes	Yes
IPv6 Next Hop	Yes	Yes	Yes	No	Yes
Metric	Yes	Yes	Yes	Yes	Yes
Route Type	Yes	Yes	Yes	Yes	Yes
Tag	Yes	Yes	Yes	Yes	Yes
IPv6 Prefix List	Yes	Yes	Yes	No	Yes
IPv4 Prefix list	Yes	Yes	Yes	Yes	Yes
IP ACL	No	No	No	No	No
Source Protocol	Yes	Yes	Yes	Yes	—
AS Path	No	No	No	No	—
AS Number	No	No	No	No	—

Table 31 : MATCH Route Map Statements by Protocol

MATCH Route Map Statement	BGP Neighbor	BGP Table Map	OSPF Table Map	EIGRP Table Map	ISIS Table Map	EIGRP Distribute List
Community List	Yes	Yes	—	—	—	—
Ext Community List	Yes	Yes	—	—	—	—
Interface	—	No	Yes	Yes	Yes	—
IPv4 Next Hop	Yes	Yes	Yes	Yes	Yes	Yes
IPv6 Next Hop	Yes	Yes	Yes	Yes	Yes	Yes

Metric	Yes	Yes	Yes	No	No	No
Route Type	Yes	Yes	Yes	Yes	Yes	No
Tag	—	Yes	Yes	Yes	No	Yes
IPv6 Prefix List	Yes	Yes	Yes	Yes	Yes	Yes
IPv4 Prefix list	Yes	Yes	Yes	Yes	Yes	Yes
IP ACL	No	No	No	No	No	No
AS Path	Yes	Yes	—	—	—	—
AS Number	Yes	No	—	—	—	—
IPv4 Route Source	—	—	Yes	—	—	—

17.2.6 Policy-Based Routing

You can use policy-based routing to forward a packet to a specified next-hop address based on the source of the packet or other fields in the packet header.

17.3 Licensing Requirements for Route Policy Manager

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide.*

17.4 Prerequisites for Route Policy Manager

If you configure VDCs, install the appropriate license and enter the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide* for configuration information and the *Inspur INOS Licensing Guide for licensing information*).

17.5 Guidelines and Limitations

- An empty route map denies all the routes.
- An empty prefix list permits all the routes.
- Without any match statement in a route-map entry, the permission (permit or deny) of the route-map entry decides the result for all the routes or packets.
- If referred policies (for example, prefix lists) within a match statement of a route-map entry return either a no-match or a deny-match, Inspur INOS fails the match statement and processes the next route-map entry.
- When you change a route map, Inspur INOS holds all the changes until you exit from the route-map configuration submode. Inspur INOS then sends all the changes to the protocol clients to take effect.
- Because you can use a route map before you define it, verify that all your route maps exist when you finish a configuration change.
- You can view the route-map usage for redistribution and filtering. Each individual routing protocol provides a way to display these statistics.

- When you redistribute BGP to IGP, iBGP is redistributed as well. To override this behavior, you must insert an additional deny statement into the route map.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

17.6 Default Settings for Route Policy Manager Parameters

Default Route Policy Manager Parameters

Parameters	**Default**
Route Policy Manager	Enabled
Administrative distance	115

17.7 Configuring Route Policy Manager

17.7.1 Configuring IP Prefix Lists

IP prefix lists match the IP packet or route against a list of prefixes and prefix lengths. You can create an IP prefix list for IPv4 and create an IPv6 prefix list for IPv6.

You can configure the prefix list entry to match the prefix length exactly, or to match any prefix with a length that matches the configured range of prefix lengths.

Use the **ge** and lt keywords to create a range of possible prefix lengths. The incoming packet or route matches the prefix list if the prefix matches and if the prefix length is greater than or equal to the **ge** keyword value (if configured) and less than or equal to the lt keyword value (if configured).

Procedure

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	(Optional) switch(config)# {**ip** | **ipv6**} **prefix-list** *name* **description** *string*	Adds an information string about the prefix list.
Step 3	switch(config)# **ip prefix-list** *name* [**seq** *number*] [{**permit** | **deny**} *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le** *prefix-length*]}]	Creates an IPv4 prefix list or adds a prefix to an existing prefix list. The prefix length is matched as follows: • **eq**—Matches the exact *prefix length*. • **ge**—Matches a prefix length that is equal to or greater than the configured *prefix length*. • **le**—Matches a prefix length that is equal to or less than the configured *prefix length*.

Step 4	switch(config)# **ipv6 prefix-list** *name* [**seq** *number*] [{**permit** | **deny**} *prefix* {[**eq** *prefix-length*] | [**ge** *prefix-length*] [**le***prefix-length*]}]	Creates an IPv6 prefix list or adds a prefix to an existing prefix list. The prefix length is matched as follows: • **eq**—Matches the exact *prefix length*. • **ge**—Matches a prefix length that is equal to or greater than the configured *prefix length*. • **le**—Matches a prefix length that is equal to or less than the configured *prefix length*.
Step 5	(Optional) switch(config)# **show** {**ip** | **ipv6**} **prefix-list** *name*	Displays information about prefix lists.
Step 6	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an IPv4 prefix list with two entries and apply the prefix list to a BGP neighbor:

```
switch# configure terminal

switch(config)#  ip  prefix-list  allowprefix  seq  10  permit 192.0.2.0/24 eq 24
switch(config)# ip prefix-list allowprefix seq 20  permit  209.165.201.0/27  eq  27
switch(config)#  router  bgp 65536:20

switch(config-router)#    neighbor    192.0.2.1/16 remote-as  65535:20
switch(config-router-neighbor)# address-family  ipv4  unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in
```

17.7.2 Configuring MAC Lists

You can configure a MAC list to permit or deny a range of MAC addresses.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **mac-list** *name* [**seq** *number*] {**permit** | **deny**} *mac-address* [*mac-mask*]	Creates a MAC list or adds a MAC address to an existing MAC list. The *seq* range is from 1 to 4294967294. The *mac-mask* specifies the portion of the MAC address to match against and is in MAC address format.
Step 3	(Optional) switch(config)# **show mac-list** *name*	Displays information about MAC lists.
Step 4	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a MAC list and copy the running configuration to the startup configuration:

```
switch# configure terminal
```

```
switch(config)# mac-list AllowMac seq 1 permit 0022.5579.a4c1 ffff.ffff.0000

switch# copy running-config startup-config
```

17.7.3 Configuring AS-path Lists

You can specify an AS-path list filter on both inbound and outbound BGP routes. Each filter is an access list based on regular expressions. If the regular expression matches the representation of the AS-path attribute of the route as an ASCII string, the permit or deny condition applies.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip as-path access-list** *name* {**deny** | **permit**} *expression*	Creates a BGP AS-path list using a regular expression.
Step 3	(Optional) switch(config)# **show** {**ip** | **ipv6**} **as-path-access-list** *name*	Displays information about as-path access lists.
Step 4	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create an AS-path list with two entries and apply the AS path list to a BGP neighbor:

```
switch# configure terminal

switch(config)#   ip   as-path   access-list AllowAS permit 64510

switch(config)#   ip   as-path   access-list AllowAS permit 64496

switch(config)# copy running-config startup-config

switch(config)# router bgp 65536:20

switch(config-router)#   neighbor   192.0.2.1/16 remote-as 65535:20

switch(config-router-neighbor)#   address-family ipv4 unicast

switch(config-router-neighbor-af)#   filter-list AllowAS in
```

17.7.4 Configuring Community Lists

You can use community lists to filter BGP routes based on the community attribute. The community number consists of a 4-byte value in the aa:nn format. The first two bytes represent the autonomous system number, and the last two bytes represent a user-defined network number.

When you configure multiple values in the same community list statement, all community values must match to satisfy the community list filter. When you configure multiple values in separate community list statements, the first list that matches a condition is processed.

Use community lists in a match statement to filter BGP routes based on the community attribute.

Procedure

	Command or Action	Purpose

Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip community-list standard** *list-name* {**deny** | **permit**} [*community-list*] [**internet**] [**local-AS**] [**no-advertise**] [**no-export**]	Creates a standard BGP community list. The *list-name* can be any case-sensitive, alphanumeric string up to 63 characters. The *community-list* can be one or more communities in the *aa:nn* format. Do not perform this step if you need to create an expanded BGP community list.
Step 3	switch(config)# **ip community-list expanded** *list-name* {**deny** | **permit**} *expression* **Example:**	Creates an expanded BGP community list using a regular expression. Do not perform this step if you need to create a standard BGP community list.
Step 4	(Optional) switch(config)# **show ip community list** *name*	Displays information about community lists.
Step 5	(Optional) switch# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a standard community list with two entries:

```
switch# configure terminal

switch(config)#  ip  community-list  standard  BGPCommunity  permit
no-advertise  65536:20
switch(config)#  ip  community-list  standard BGPCommunity  permit
local-AS  no-export
switch(config)#  copy  running-config startup-config
```

17.7.5 Configuring Extended Community Lists

You can use extended community lists to filter BGP routes based on the community attribute. The community number consists of a 6-byte value in the aa4:nn format. The first four bytes represent the autonomous system number, and the last two bytes represent a user-defined network number.

When you configure multiple values in the same extended community list statement, all extended community values must match to satisfy the extended community list filter. When you configure multiple values in separate extended community list statements, the first list that matches a condition is processed.

Use extended community lists in a match statement to filter BGP routes based on the extended community attribute.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **ip extcommunity-list standard** *list-name* {**deny**|**permit**} **4bytegeneric** {**transitive**|**nontransitive**} *community1* [*community2...*]	Creates a standard BGP extended community list. The *community* can be one or more extended communities in the *aa4:nn* format. Do not perform this step if you need to create an expanded BGP extended community list.

Step 3	switch(config)#**ip extcommunity-list expanded** *list-name* {**deny** | **permit**} *expression*	Creates an expanded BGP extended community list using a regular expression. Do not perform this step if you need to create a standard BGP extended community list.
Step 4	(Optional) switch(config)# **show ip extcommunity list** *name*	Displays information about community lists.
Step 5	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to create a generic specific extended community list:

```
switch# configure terminal

switch(config)#  ip  extcommunity-list  standard  test1  permit  4bytegeneric
 transitive 65536:40 65536:60

switch(config)# copy running-config startup-config
```

17.7.6 Optional Match Parameters for Route Maps

You can configure the following optional match parameters for route maps in route-map configuration mode:

Command	**Purpose**
switch(config-route-map)# **match as-path** *name* [*name...*]	Matches against one or more AS-path lists. Create the AS-path list with the **ip as-path access-list** command.
switch(config-route-map)# **match as-number** {*number* [,*number...*] | **as-path-list name** [*name...*]}	Matches against one or more AS numbers or AS-path lists. Create the AS-path list with the **ip as-path access-list** command. The number range is from 1 to 65535. The AS-path list name can be any case-sensitive, alphanumeric string up to 63 characters.
switch(config-route-map)# **match community** *name* [*name...*][**exact-match**]	Matches against one or more community lists. Create the community list with the **ip community-list** command.
switch(config-route-map)# **match extcommunity** *name* [*name...*][**exact-match**]	Matches against one or more extended community lists. Create the community list with the **ip extcommunity-list** command.
switch(config-route-map)# **match interface** *interface-type number* [*interface-type number...*]	Matches any routes that have their next hop out one of the configured interfaces. Use **?** to find a list of supported interface types.
switch(config-route-map)# **match ip address prefix-list** *name* [*name...*]	Matches against one or more IPv4 prefix lists. Use the **ip prefix-list** command to create the prefix list.
switch(config-route-map)# **match ipv6 address prefix-list** *name* [*name...*]	Matches against one or more IPv6 prefix lists. Use the **ipv6 prefix-list** command to create the prefix list.
switch(config-route-map)# **match ip multicast** [**source** *ipsource*] [[**group** *ipgroup*] [**rp** *iprp*]]	Matches an IPv4 multicast packet based on the multicast source, group, or rendezvous point.

switch(config-route-map)# **match ipv6 multicast** [**source** *ipsource*] [[**group** *ipgroup*] [**rp** *iprp*]]	Matches an IPv6 multicast packet based on the multicast source, group, or rendezvous point.
switch(config-route-map)# **match ip next-hop prefix-list** *name* [*name...*]	Matches the IPv4 next-hop address of a route to one or more IP prefix lists. Use the **ip prefix-list** command to create the prefix list.
switch(config-route-map)# **match ipv6 next-hop prefix-list** *name* [*name...*]	Matches the IPv6 next-hop address of a route to one or more IP prefix lists. Use the **ipv6 prefix-list** command to create the prefix list.
switch(config-route-map)# **match ip route-source prefix-list** *name* [*name...*]	Matches the IPv4 route source address of a route to one or more IP prefix lists. Use the **ip prefix-list** command to create the prefix list.
switch(config-route-map)# **match ipv6 route-source prefix-list** *name* [*name...*]	Matches the IPv6 route-source address of a route to one or more IP prefix lists. Use the **ipv6 prefix-list** command to create the prefix list.
switch(config-route-map)# **match mac-list** *name* [*name...*]	Matches against one or more MAC lists. Use the **mac-list** command to create the MAC list.
switch(config-route-map)# **match metric** *value* [**+-** *deviation.*] [*value..*]	Matches the route metric against one or more metric values or value ranges. Use +- deviation argument to set a metric range. The route map matches any route metric that falls the range: *value* - *deviation* to *value* + *deviation.*
switch(config-route-map)# **match route-type** *route-type*	Matches against a type of route. The *route-type* can be one or more of the following: • external • inter-area • internal • intra-area • level-1 • level-2 • local • nssa-external • type-1 • type-2
switch(config-route-map)# **match tag** *tagid* [*tagid...*]	Matches a route against one or more tags for filtering or redistribution.
switch(config-route-map)# **match vlan** *vlan-id* [*vlan-range*]	Matches against a VLAN.

17.7.7 Optional Set Parameters for Route Maps

You can configure the following optional set parameters for route maps in route-map configuration mode:

Command	**Purpose**
switch(config-route-map)# **set as-path** {**tag** | **prepend** {**last-as** *number* | *as-1* [*as-2...*]}}	Modifies an AS-path attribute for a BGP route. You can prepend the configured number of last AS numbers or a string of particular AS-path values (*as-1 as-2...as-n*).
switch(config-route-map)# **set comm-list** *name* **delete**	Removes communities from the community attribute of an inbound or outbound BGP route update. Use the **ip community-list** command to create the community list.
switch(config-route-map)# **set community** {**none** | **additive** | **local-AS** | **no-advertise** | **no-export** | *community-1* [*community-2...*]}	Sets the community attribute for a BGP route update. **Note** When you use both the **set community** and **set comm-list delete** commands in the same sequence of a route map attribute, the deletion operation is performed before the set operation. **Note** Use the **send-community** command in BGP neighbor address family configuration mode to propagate BGP community attributes to BGP peers.
switch(config-route-map)# **set dampening** *halflife reuse suppress duration*	Sets the following BGP route dampening parameters: • *halflife*—The range is from 1 to 45 minutes. The default is 15. • *reuse*—The range is from is 1 to 20000 seconds. The default is 750. • *suppress*—The range is from is 1 to 20000. The default is 2000. • *duration*—The range is from is 1 to 255 minutes. The default is 60.
switch(config-route-map)# **set distance** *value*	Sets the administrative distance of routes for OSPFv2 or OSPFv3. The range is from 1 to 255.
switch(config-route-map)# **set extcomm-list** *name* **delete**	Removes communities from the extended community attribute of an inbound or outbound BGP route update. Use the **ip extcommunity-list** command to create the extended community list.

switch(config-route-map)# **set extcommunity 4byteas-generic** {**transitive** | **nontransitive**} {**none** | **additive**] *community-1* [*community-2...*]}	Sets the extended community attribute for a BGP route update. Note When you use both the **set extcommunity** and **set extcomm-list delete** commands in the same sequence of a route map attribute, the deletion operation is performed before the set operation. Note Use the **send-community** command in BGP neighbor address family configuration mode to propagate BGP extended community attributes to BGP peers.
switch(config-route-map)# **set extcommunity cost** *community-id1 cost* [**igp** | **pre-bestpath**] [**community-id2...**]}	Sets the cost community attribute for a BGP route update. This attribute allows you to customize the BGP best path selection process for a local autonomous system or confederation. The *community-id* range is from 0 to 255. The *cost* range is from 0 to 4294967295. The path with the lowest cost is preferred. For paths with equal cost, the path with the lowest community ID is preferred. The **igp** keyword compares the cost after the IGP cost comparison. The **pre-bestpath** keyword compares before all other steps in the bestpath algorithm.
switch(config-route-map)# **set extcommunity rt** *community-1* [**additive**] [*community-2...*]}	Sets the extended community route target attribute for a BGP route update. The *community* value can be a 2-byte AS number:4-byte network number, a 4-byte AS number:2-byte network number, or an IP address:2-byte network number. Use the **additive** keyword to add a route target to an existing extended community route target attribute.
switch(config-route-map)# **set forwarding-address**	Sets the forwarding address for OSPF.
switch(config-route-map)# **set level** {**backbone** | **level-1** | **level-1-2** | **level-2**}	Sets what area to import routes to for IS-IS. The options for IS-IS are level-1, level-1-2, or level-2. The default is level-1.
switch(config-route-map)# **set local-preference***value*	Sets the BGP local preference value. The range is from 0 to 4294967295.
switch(config-route-map)# **set metric** [+ | **-**]*bandwidth-metric*	Adds or subtracts from the existing metric value. The metric is in Kb/s. The range is from 0 to 4294967295.

switch(config-route-map)# **set metric** *bandwidth* [*delay reliability load mtu*]	Sets the route metric values. Metrics are as follows: • *metric0*—Bandwidth in Kb/s. The range is from 0 to 4294967295. • *metric1*—Delay in 10-microsecond units. • *metric2*—Reliability. The range is from 0 to 255 (100 percent reliable). • *metric3*—Loading. The range is from 1 to 200 (100 percent loaded). • *metric4*—MTU of the path. The range is from 1 to 4294967295.
switch(config-route-map)# **set metric-type** {**external** | **internal** | **type-1** | **type-2**}	Sets the metric type for the destination routing protocol. The options are as follows: • **external**—IS-IS external metric • **internal**— IGP metric as the MED for BGP • **type-1**—OSPF external type 1 metric • **type-2**—OSPF external type 2 metric The **set metric-type internal** command affects an outgoing policy and an eBGP neighbor only. If you configure both the **metric** and **metric-type internal** commands in the same BGP peer outgoing policy, then Inspur INOS ignores the **metric-type internal** command.
switch(config-route-map)# **set nssa-only**	Sets Type-7 LSA generated on ASBR with no P bit set. This prevents Type-7 to Type-5 LSA translation in OSPF.
switch(config-route-map)# **set origin** {**egp** *as-number* | **igp** | **incomplete**}	Sets the BGP origin attribute. The EGP *as-number* range is from 0 to 65535.
switch(config-route-map)# **set tag** *name*	Sets the tag value for the destination routing protocol. The *name* parameter is an unsigned integer.
switch(config-route-map)# **set weight** *count*	Sets the weight for the BGP route. The range is from 0 to 65535.

17.7.8 Verifying the Route Policy Manager Configuration

Use one of the following commands to verify the configuration:

Command	**Purpose**
show ip community-list [*name*]	Displays information about a community list.
show ip extcommunity-list [*name*]	Displays information about an extended community list.

show [**ip** | **ipv6**] **prefix-list** [*name*]	Displays information about an IPv4 or IPv6 prefix list.
show route-map [*name*]	Displays information about a route map.

17.8 Configuration Examples for Route Policy Manager

This example shows how to use an address family to configure Route Policy Manager so that any unicast and multicast routes from neighbor 209.0.2.1 are accepted if they match prefix-list AllowPrefix:

```
router bgp 64496

neighbor  209.0.2.1
 remote-as    64497
 address-family
 ipv4 unicast

  route-map filterBGP in

route-map filterBGP

 match ip address prefix-list AllowPrefix

ip prefix-list AllowPrefix 10 permit 192.0.2.0/24

ip prefix-list AllowPrefix 20 permit 209.165.201.0/27
```

17.9 Related Documents for Route Policy Manager

Related Topic	Document Title
Route Policy Manager CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*

17.10 Standards for Route Policy Manager

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

17.11 Feature History for Route Policy Manager

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
Multiple match statements under table-map	8.4(1)	Added support for multiple match statements under table-map.
Route map support matrix	8.4(1)	Added the route map support matrix for routing protocols.

Match interfaces	8.4(1)	Added support for null interfaces to the **match interface** command. Added support for the following set and match statements in a route map for the EIGRP distribute list: • Set routing protocol metric • Set route tag • Match tag
Route policy manager	8.4(1)	Added support for the **set distance** command and for the **inter-area** and **intra-area** options for the **match route-type** command.
MPLS set clauses	8.4(1)	Added support for **set extcommunity cost**, **set extcommunity rt**, and **set nssa-only** commands.
MAC lists , metric, and VLANs	8.4(1)	Added support for the **match mac-list**, **match metric**, and **match vlan** commands.
Extended community lists	8.4(1)	Added support for generic specific extended community lists.
Match interfaces	8.4(1)	Added support to match a list of interfaces in a route map.
Match AS numbers	8.4(1)	Added support to match a range of AS numbers in a route map.
Route policy manager	8.4(1)	This feature was introduced.

CHAPTER 18 Configuring Policy-Based Routing

This chapter contains the following sections:

- Finding Feature Information.
- Information About Policy Based Routing.
- Licensing Requirements for Policy-Based Routing.
- Prerequisites for Policy-Based Routing.
- Guidelines and Limitations for Policy-Based Routing.
- Default Settings for Policy-Based Routing.
- Configuring Policy-Based Routing.
- Verifying the Policy-Based Routing Configuration.
- Configuration Examples for Policy Based-Routing.
- Related Documents for Policy-Based Routing.
- Standards for Policy-Based Routing.
- Feature History for Policy-Based Routing.

18.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

18.2 Information About Policy Based Routing

Policy-based routing allows you to configure a defined policy for IPv4 and IPv6 traffic flows, lessening reliance on routes derived from routing protocols. All packets received on an interface with policy-based routing enabled are passed through enhanced packet filters or route maps. The route maps dictate the policy, determining where to forward packets.

Route maps are composed of match and set statements that you can mark as permit or deny. You can interpret the statements as follows:

- If the packets match any route map statements, all the set statements are applied. One of these actions involves choosing the next hop.
- If a statement is marked as deny, the packets that meet the match criteria are sent back through the normal forwarding channels and destination-based routing is performed.
- If the statement is marked as permit and the packets do not match any route-map statements, the packets are sent back through the normal forwarding channels and destination-based routing is performed.

Policy-based routing includes the following features:

- Source-based routing—Routes traffic that originates from different sets of users through different connections across the policy routers.
- Quality of Service (QoS)—Differentiates traffic by setting the precedence or type of service (ToS) values in the IP packet headers at the periphery of the network and leveraging queuing mechanisms to prioritize traffic in the core or backbone of the network (see the *Inspur CN12700 Series INOS Quality of Service Configuration Guide*).
- Load sharing—Distributes traffic among multiple paths based on the traffic characteristics.

18.2.1 Policy Route Maps

Each entry in a route map contains a combination of match and set statements. The match statements define the criteria for whether appropriate packets meet the particular policy (that is, the conditions to be met). The set clauses explain how the packets should be routed once they have met the match criteria.

You can mark the route-map statements as permit or deny. If the statement is marked as a deny, the packets that meet the match criteria are sent back through the normal forwarding channels (destination-based routing is performed). If the statement is marked as permit and the packets meet the match criteria, all the set clauses are applied. If the statement is marked as permit and the packets do not meet the match criteria, those packets are also forwarded through the normal routing channel.

18.2.2 Set Criteria for Policy-Based Routing

The set criteria in a route map is evaluated in the order listed in the route map. Set criteria specific to route maps used for policy-based routing are as follows:

1. List of interfaces through which the packets can be routed—If more than one interface is specified, the first interface that is found to be up is used for forwarding the packets.
2. List of specified IP addresses—The IP address can specify the adjacent next-hop router in the path toward the destination to which the packets should be forwarded. The first IP address associated with a currently up connected interface is used to route the packets.
3. List of default interfaces—If there is no explicit route available to the destination address of the packet being considered for policy routing, the route map routes it to the first up interface in the list of specified default interfaces.
4. List of default next-hop IP addresses—Route to the interface or the next-hop address specified by this set statement only if there is no explicit route for the destination address of the packet in the routing table.

If the packets do not meet any of the defined match criteria, those packets are routed through the normal destination-based routing process.

Local Policy Routing

Local policy routing allows you to apply a route map to local (device-generated) traffic. All packets originating on the device that are not normally policy routed are subject to local policy routing.

18.2.3 Route Map Support Matrix for Policy-Based Routing

The following tables include the configurable match and set statements for policy-based routing on Inspur CN12700 Series switches running the latest shipping release.

The following legend applies to the tables:

- Yes—The statement is supported for policy-based routing.
- No—The statement is not supported for policy-based routing.
- If a statement does not apply for policy-based routing, there is an em dash (—) in the column next to the statement.
- Where clarification is required, information is added in the appropriate row/column.

Table 32 : SET Route Map Statements for Policy-Based Routing

SET Route Map Statement	Policy-Based Routing (PBR)
IPv4 Next Hop	Yes
IPv6 Next Hop	Yes
Default IPv4 Next Hop	Yes

Default IPv6 Next Hop	Yes
IPv4 Next Hop Verify Availability	Yes
IPv6 Next Hop Verify Availability	Yes
Default IPv4 Next Hop Verify Availability	Yes
Default IPv6 Next Hop Verify Availability	Yes
Interface null0	Yes
VRF	Yes
IPv4 Precedence	Yes
IPv6 Precedence	Yes
Interface, GRE Ethernet	No

Table 33 : MATCH Route Map Statements for Policy-Based Routing

MATCH Route Map Statement	**Policy-Based Routing (PBR)**
Tag	Yes
Packet Length	Yes
VLAN ID	Yes
MAC ACL	Yes
IPv4 Prefix List	No
IPv6 Prefix List	No
IP ACL	Yes

18.3 Licensing Requirements for Policy-Based Routing

Policy-based routing requires an Enterprise Services license. For a complete explanation of the Inspur INOS licensing scheme and how to obtain and apply licenses, see the *License and Copyright Information for Inspur INOS Software.*

18.4 Prerequisites for Policy-Based Routing

Policy-based routing has the following prerequisites:

- Install the correct license.
- You must enable policy-based routing.
- Assign an IP address on the interface and bring the interface up before you apply a route map on the interface for policy-based routing.
- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the Inspur CN12700 Series INOS Virtual Device Context Configuration Guide).

18.5 Guidelines and Limitations for Policy-Based Routing

Policy-based routing has the following configuration guidelines and limitations:

- Inspur INOS uses recursive next hops. You do not need to enter any commands for recursive next hops like you do for Inspur IOS.
- A policy-based routing route map can have only one match or set statement per route-map statement.
- A **match** command cannot refer to more than one ACL in a route map used for policy-based routing.
- The same route map can be shared among different interfaces for policy-based routing as long as the interfaces belong to the same virtual routing and forwarding (VRF) instance.
- Prior to Inspur INOS Release 8.4(1) setting a tunnel interface or an IP address via a tunnel interface as a next hop in a policy-based routing policy is not supported. Applying policy-based routing or **ip policy route-map** on tunnel interfaces is also not supported. From Inspur INOS Release 8.4(1) onwards GRE next hop is supported on policy-based routing.
- Policy-based routing is not supported with inbound traffic on FEX ports.
- Using a prefix-list as a match criteria is not supported. Do not use a prefix-list in a policy-based routing route-map.
- Beginning with Inspur INOS Release 8.4(1), policy-based routing and WCCPv2 are supported on the same interface. However, policy-based routing with statistics and WCCPv2 is supported on the same interface only if bank chaining is disabled.
- Beginning with Inspur INOS Release 8.4(1), you can configure the device to support deny access control entries (ACEs) in a sequence for the following sequence-based features: VACLs and QoS. For more information, see the “Configuring VLAN ACLs” chapter in the *Inspur CN12700 Series INOS Security Configuration Guide*.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.
- PBR marks the next-hop as down even when the next hop and the corresponding tracks are up. This issue is due to the RPM that does not effectively process the tracks.

Currently the object tracking manager (OTM) does not support forward referencing for track objects. Track objects must be created in the OTM before they are used in any configuration.

Perform the following steps to configure the track objects with RPM or PBR so that the PBR next-hop issue does not occur:

1. Create the track object in OTM using the **track** <*object id*> command.
2. Use the configured track object in a route map using the **set ip next-hop verify-availability** <*ip1*>
3. **track** <*object id*> command.
4. Apply the route map to an interface using the **ip policy route-map** <*map-name*> command.

18.6 Default Settings for Policy-Based Routing

Table 34 : Default Policy-Based Routing Parameters

Parameters	Default
Policy-based routing	Disabled

18.7 Configuring Policy-Based Routing

18.7.1 Enabling the Policy-Based Routing

You must enable the policy-based routing feature before you can configure a route policy.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature pbr**	Enables the policy-based routing feature. Use the **no feature pbr** command to disable the policy-based routing feature and remove all associated configuration.
Step 3	(Optional) switch(config)# **show feature**	Displays enabled and disabled features.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

18.7.2 Configuring a Route Policy

You can use route maps in policy-based routing to assign routing policies to the inbound interface. Inspur INOS routes the packet as soon as it finds a next hop and an interface.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip policy route-map** *map-name*	Assigns a route map for IPv4 policy-based routing to the interface.
Step 4	switch(config-if)# **ipv6 policy route-map** *map-name*	Assigns a route map for IPv6 policy-based routing to the interface.
Step 5	(Optional) switch(config-route-map)# **end**	Exits route-map configuration mode and enters the privileged executive mode.
Step 6	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to add a route map to an interface:

```
switch# configure terminal

switch(config)# interface ethernet 1/2
switch(config-if)#   ip   policy   route-map   Testmap
switch(config)# exit

switch(config)#  copy  running-config  startup-config
```

You can configure the following optional match parameters for route maps in route-map configuration mode:

Command	Purpose
match ip address access-list-name *name [name...]* Example: `switch(config-route-map)# match ip address access-list-name ACL1`	Matches an IPv4 address against one or more IP access control lists (ACLs). This command is used for policy-based routing and is ignored by route filtering or redistribution.
match ipv6 address access-list-name *name [name...]* Example: `switch(config-route-map)# match ipv6 address access-list-name ACLv6`	Matches an IPv6 address against one or more IPv6 ACLs. This command is used for policy-based routing and is ignored by route filtering or redistribution.
match length *min max* Example: `switch(config-route-map)# match length 64 1500`	Matches against the length of the packet. This command is used for policy-based routing.
match mac-list *maclist [...maclist]*	Matches against a list of MAC addresses. This command is used for policy-based routing.
match metric *metric-value [+- deviation-number] [...metric-value [+- deviation-number]]* [+- *deviation-number*] [... *metric-value* +- *deviation-number*]] Example: `switch(config-route-map)# match metric 10`	Matches against the routing protocol metric. This command is used for policy-based routing.
match vlan *vlan-range* `switch(config-route-map)# match vlan 64`	Matches against the VLAN ID of the packet. This command is used for policy-based routing.

You can configure the following optional set parameters for route maps in route-map configuration mode:

Command	Purpose

set ip next-hop *address2* [*address2...*] [**load-share**| **peer-address** | **unchanged** | **verify-availability**]
Example:
`switch(config-route-map)#`
`set ip next-hop 192.0.2.1`
	Sets the IPv4 next-hop address for policy-based routing. This command uses the first valid next-hop address if multiple addresses are configured. This can done with next-hop tracking only.
	• Use the optional **load-share** keyword to load balance traffic across a maximum of 16 next-hop addresses.
	• Use the optional **peer-address** keyword to the next hop to be the Border Gateway Protocol (BGP) peering address.
	• Use the optional **unchanged** keyword to specifiy that the next-hop attribute in the BGP update to the eBGP peer is unmodified.
	• Use the optional **verify-availability** keyword to verify the reachability of the tracked object.

set ip default next-hop *address2* [*address2...*] [**load-share** | **verify-availability**]
Example:
`switch(config-route-map)#`
`set ip default next-hop 192.0.2.2`
	Sets the IPv4 next-hop address for policy-based routing when there is no explicit route to a destination. This command uses the first valid next-hop address if multiple addresses are configured. This can done with next-hop tracking only.
	• Use the optional **load-share** keyword to load balance traffic across a maximum of 16 next-hop addresses.
	• Use the optional **verify-availability** keyword to verify the reachability of the tracked object.
	Note For software-forwarded traffic, the route that is present in the unicast routing table (of the VRF in which packet was received) for packet-specified destination takes preference over what is specified in **set ip default next-hop** command, when there is condition match. Even if there is a default route present in the VRF, that default route overrides what is set in the command. This applies to software-forwarded traffic only.

set ipv6 next-hop *address2* [*address2...*][**load-share** | **peer-address** | **unchanged** | **verify-availability**]
Example:
`switch(config-route-map)#`
`set ipv6 next-hop 2001:0DB8::1`
	Sets the IPv6 next-hop address for policy-based routing. This command uses the first valid next-hop address if multiple addresses are configured. This can done with next-hop tracking only.
	• Use the optional **load-share** keyword to load balance traffic across a maximum of 16 next-hop addresses.
	• Use the optional **peer-address** keyword to the next hop to be the Border Gateway Protocol (BGP) peering address.
	• Use the optional **unchanged** keyword to specifiy that the next-hop attribute in the BGP update to the eBGP peer is unmodified.
	• Use the optional **verify-availability** keyword to verify the reachability of the tracked object.

set ipv6 default next-hop *address2* [*address2...*] [**load-share** | **verify-availability**]
Example:
`switch(config-route-map)#`
`set ipv6 default next-hop 2001:0DB8::2`
	Sets the IPv6 next-hop address for policy-based routing when there is no explicit route to a destination. This command uses the first valid next-hop address if multiple addresses are configured. This can done with next-hop tracking only.
	• Use the optional **load-share** keyword to load balance traffic across a maximum of 16 next-hop addresses.
	• Use the optional **verify-availability** keyword to verify the reachability of the tracked object.

set ip precedence *precedence-value*
Example:
`switch(config-route-map)#`
`set ip precedence highv4`
	Sets the precedence value in the IPv4 packet header.

set ipv6 precedence *precedence-value*
Example:
`switch(config-route-map)#`
`set ipv6 precedence highv6`
	Sets the precedence value in the IPv6 packet header.

set ipv6 precedence address prefix-list *prefix-list-name*
Example:
`switch(config-route-map)#`
`set ipv6 precedence address prefix-list acl1`
	Sets the IPv6 map routes to be injected.

set interface {*null10* | *tunnel-te*}
Example:
`switch(config-route-map)#`
`set interface null0`
	Sets the interface used for routing. Use the **null0** interface to drop packets. Use the **tunnel-te** interface to forward packets on the MPLS TE tunnel.

set vrf *vrf-name* Example: `switch(config-route-map)# set vrf MainVRF`	Sets the VRF for next-hop resolution.

18.7.3 Configuring Local Policy Routing

You can enable local policy routing for packets generated by the device and specify which route map the device should use.

Procedure

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# {**ip** | **ipv6**} **local policy route-map** *map-name*	Configures IPv4 or IPv6 local policy route maps for packets generated by the device.
Step 3	(Optional) **show** {**ip** | **ipv6**} **local policy**	Displays the route map used for IPv4 or IPv6 local policy routing.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

18.7.4 Configuring a Deny ACE

Beginning with Inspur INOS Release 8.4(1) you can configure the device to support deny access control entries (ACEs) in a sequence for the following sequence-based features: VACL and Quality of service (QoS).

When deny ACEs are enabled, the traffic that matches a deny ACE (an ACL rule with the **deny** keyword) in a class-map-acl is recursively matched against subsequent class-map-acls until it hits a permit ACE.

Before you begin

Ensure that you are in the default or admin VDC.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **hardware access-list allow deny ace**
3. (Optional) **show running-config aclmgr**
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# [**no**] **hardware access-list allow deny ace**	Enables deny ACEs in a sequence. The **no** form of the command disables deny ACEs.
Step 3	(Optional) **show running-config aclmgr**	Displays the ACL configuration.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

18.8 Verifying the Policy-Based Routing Configuration

To display policy-based routing configuration information, perform one of the following tasks:

Command	Purpose
show [**ip** | **ipv6**] **policy** [*name*]	Displays information about an IPv4 or IPv6 policy.
show {**ip** | **ipv6**} **local policy** [**vrf** *name*]	Displays the route map used for IPv4 or IPv6 local policy routing.
show route-map [*name*] **pbr-statistics**	Displays policy statistics.

Use the **route-map** *map-name* **pbr-statistics** to enable policy statistics. Use the **clear route-map** *map-name* **pbr-statistics** to clear these policy statistics.

18.9 Configuration Examples for Policy Based-Routing

This example shows how to configure a simple route policy on an interface:

```
feature pbr

ip access-list pbr-sample

  permit tcp host 10.1.1.1 host 192.168.2.1 eq 80

!

route-map pbr-sample

 match      ip
 address  pbr-
 sample    set
 ip   next-hop
 192.168.1.1

!

route-map pbr-sample pbr-statistics

 interface ethernet 1/2

  ip policy route-map pbr-sample
```

The following output verifies this configuration:

```
CN12700# show route-map pbr-sample

route-map       pbr-sample,
 permit, sequence 10 Match
 clauses:
   ip   address   (access-
 lists):  pbr-sample  Set
 clauses:

   ip next-hop 192.168.1.1

CN12700# show route-map pbr-sample pbr-statistics

route-map      pbr-sample,
 permit,    sequence    10
 Policy  routing  matches:
 84 packets

Default routing: 233 packets
```

18.9.1 Configuration Example for Local Policy Routing

The following example sends packets with a destination IP address matching that allowed by extended access list 131 to the router at IP address 172.30.3.20:

```
ip local policy route-map xyz

!

route-map xyz

match ip address 131

set ip next-hop 172.30.3.20
```

18.10 Related Documents for Policy-Based Routing

Related Topic	Document Title
Policy-based routing CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS Virtual Device Context Command Reference*

18.11 Standards for Policy-Based Routing

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

18.12 Feature History for Policy-Based Routing

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 35 : Feature History for Policy-Based Routing

Feature Name	Release	Feature Information
Route map support matrix	8.4(1)	Added the route map support matrix for policy-based routing.

Policy-based routing	8.4(1)	Added support for deny access control entries (ACEs) in a sequence for the following sequence-based features: VACLs, policy-based routing, and QoS.
Policy-based routing	8.4(1)	Added support for policy-based routing and WCCPv2 on the same interface if bank chaining is disabled.
Interfaces	8.4(1)	Added support for **set interface route-map** command.
IPv6 policies	8.4(1)	Added support for IPv6 policies.
Policy-based routing	8.4(1)	This feature was introduced.

CHAPTER 19 Configuring GLBP

This chapter contains the following sections:

- Finding Feature Information.
- Information About GLBP.
- Licensing Requirements for GLBP.
- Prerequisites for GLBP.
- Guidelines and Limitations for GLBP.
- Default Settings for GLBP.
- Configuring GLBP.
- Verifying the GLBP Configuration.
- Configuration Examples for GLBP.
- Related Documents for GLBP.
- Standards for GLBP.
- Feature History for GLBP.

19.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

19.2 Information About GLBP

Gateway Load Balancing Protocol (GLBP) provides path redundancy for IP by sharing protocol and Media Access Control (MAC) addresses between redundant gateways. Additionally, GLBP allows a group of Layer 3 routers to share the load of the default gateway on a LAN. A GLBP router can automatically assume the forwarding function of another router in the group if the other router fails.

GLBP provides automatic gateway backup for IP hosts configured with a single default gateway on an IEEE

LAN. Multiple routers on the LAN combine to offer a single virtual first-hop IP gateway while sharing the IP packet forwarding load. Other routers on the LAN might act as redundant GLBP gateways that become active if any of the existing forwarding gateways fail.

GLBP performs a similar function to the Hot Standby Redundancy Protocol (HSRP) and the Virtual Router Redundancy Protocol (VRRP). HSRP and VRRP allow multiple routers to participate in a virtual group configured with a virtual IP address. These protocols elect one member as the active router to forward packets to the virtual IP address for the group. The other routers in the group are redundant until the active router fails.

GLBP performs an additional load balancing function that the other protocols do not provide. GLBP load balances over multiple routers (gateways) using a single virtual IP address and multiple virtual MAC addresses. GLBP shares the forwarding load among all routers in a GLBP group instead of allowing a single router to handle the whole load while the other routers remain idle. You configure each host with the same virtual IP address, and all routers in the virtual group participate in forwarding packets. GLBP members communicate between each other using periodic hello messages.

19.2.1 GLBP Active Virtual Gateway

GLBP prioritizes gateways to elect an active virtual gateway (AVG). If multiple gateways have the same priority, the gateway with the highest real IP address becomes the AVG. The AVG assigns a virtual MAC address to each member of the GLBP group. Each member is the active virtual forwarder (AVF) for its assigned virtual MAC address, forwarding packets sent to its assigned virtual MAC address.

The AVG also answers Address Resolution Protocol (ARP) requests for the virtual IP address. Load sharing is achieved when the AVG replies to the ARP requests with different virtual MAC addresses.

19.2.2 GLBP Virtual MAC Address Assignment

The AVG assigns the virtual MAC addresses to each member of the group. The group members request a virtual MAC address after they discover the AVG through hello messages. The AVG assigns the next MAC address based on the load-balancing algorithm selected. A gateway that is assigned with a virtual MAC address by the AVG is the primary virtual forwarder. The other members of the GLBP group that learn the virtual MAC addresses from hello messages are secondary virtual forwarders.

19.2.3 GLBP Virtual Gateway Redundancy

GLBP provides virtual gateway redundancy. A member in a group can be in the active, standby, or listen state. GLBP uses a priority algorithm to elect one gateway as the AVG and elect another gateway as the standby virtual gateway. The remaining gateways go into the listen state. You can configure the GLBP priority on each gateway. If the GLBP priority is identical on multiple gateways, GLBP uses the gateway with the highest IP address as the AVG.

If an AVG fails, the standby virtual gateway assumes responsibility for the virtual IP address. GLBP elects a new standby virtual gateway from the gateways in the listen state.

19.2.4 GLBP Virtual Forwarder Redundancy

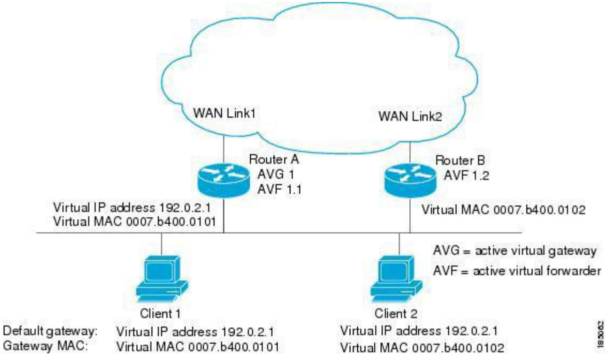
GLBP provides virtual forwarder redundancy. Virtual forwarder redundancy is similar to virtual gateway redundancy with an active virtual forwarder (AVF). If the AVF fails, a secondary virtual forwarder in the listen state assumes responsibility for the virtual MAC address. This secondary virtual forwarder is also a primary virtual forwarder for a different virtual MAC address. GLBP migrates hosts away from the old virtual MAC address of the failed AVF, using the following two timers:

- Redirect timer—Specifies the interval during which the AVG continues to redirect hosts to the old virtual MAC address. When the redirect time expires, the AVG stops using the old virtual MAC address in ARP replies, although the secondary virtual forwarder continues to forward packets that were sent to the old virtual MAC address.
- Secondary hold timer—Specifies the interval during which the virtual MAC address is valid. When the secondary hold time expires, GLBP removes the virtual MAC address from all gateways in the GLBP group and load balances the traffic over the remaining AVFs. The expired virtual MAC address becomes eligible for reassignment by the AVG.

GLBP uses hello messages to communicate the current state of the timers.

In the figure, router A is the AVG for a GLBP group and is responsible for the virtual IP address 192.0.2.1. Router A is also an AVF for the virtual MAC address 0007.b400.0101. Router B is a member of the same GLBP group and is designated as the AVF for the virtual MAC address 0007.b400.0102. Client 1 has a default gateway IP address of 192.0.2.1, the virtual IP address, and a gateway MAC address of 0007.b400.0101 that points to router A. Client 2 shares the same default gateway IP address but receives the gateway MAC address 0007.b400.0102 because router B is sharing the traffic load with router A.

Figure 45 : GLBP Topology


WAN Link1
WAN Link2
Router A
AVG 1
AVF 1.1
Router B
AVF 1.2
Virtual IP address 192.0.2.1
Virtual MAC 0007.b400.0101
Virtual MAC 0007.b400.0102
AVG = active virtual gateway
AVF = active virtual forwarder
Client 1
Client 2
Default gateway:
Gateway MAC:
Virtual IP address 192.0.2.1
Virtual MAC 0007.b400.0101
Virtual IP address 192.0.2.1
Virtual MAC 0007.b400.0102
185062

If router A becomes unavailable, client 1 does not lose access to the WAN because router B assumes responsibility for forwarding packets sent to the virtual MAC address of router A and for responding to packets sent to its own virtual MAC address. Router B also assumes the role of the AVG for the entire GLBP group. Communication for the GLBP members continues despite the failure of a router in the GLBP group.

19.2.5 GLBP Authentication

GLBP has three authentication types:

- MD5 authentication
- Plain text authentication
- No authentication

MD5 authentication provides greater security than plain text authentication. MD5 authentication allows each GLBP group member to use a secret key to generate a keyed MD5 hash that is part of the outgoing packet. At the receiving end, a keyed hash of an incoming packet is generated. If the hash within the incoming packet does not match the generated hash, the packet is ignored. The key for the MD5 hash can either be given directly in the configuration using a key string or supplied indirectly through a key chain.

You can also choose to use a simple password in plain text to authenticate GLBP packets, or choose no authentication for GLBP.

GLBP rejects packets in any of the following cases:

- The authentication schemes differ on the router and in the incoming packet.
- MD5 digests differ on the router and in the incoming packet.
- Text authentication strings differ on the router and in the incoming packet.

19.2.6 GLBP Load Balancing and Tracking

You can configure the following load-balancing methods for GLBP:

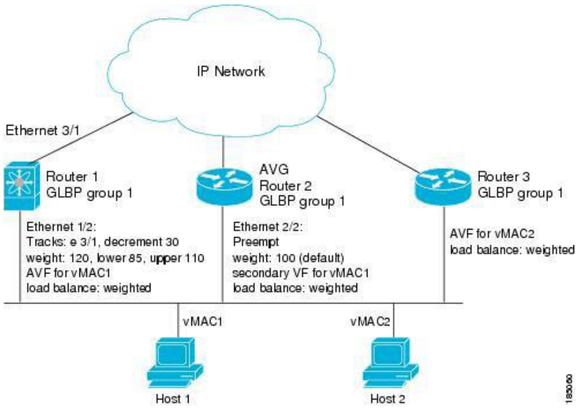
- Round-robin—GLBP cycles through the virtual MAC addresses sent in ARP replies, load balancing the traffic across all the AVFs.
- Weighted—AVG uses the advertised weight for an AVF to decide the load directed to the AVF. A higher weight means that the AVG directs more traffic to the AVF.
- Host dependent—GLBP uses the MAC address of the host to determine which virtual MAC address to direct the host to use. This algorithm guarantees that a host gets the same virtual MAC address if the number of virtual forwarders does not change.

The default for IPv4 networks is round-robin. You can disable all load balancing for GLBP on an interface. If you do not configure load balancing, the AVG handles all traffic for the hosts while the other GLBP group members are in standby or listen mode.

You can configure GLBP to track an interface or routes and enable the secondary virtual forwarder to take over if a tracked link goes down. GLBP tracking uses weighted load-balancing to determine whether a GLBP group member acts as an AVF. You must configure the initial weighting values and optional thresholds to enable or disable this group member as an AVF. You can also configure the interface to track and the value that reduces the interface's weighting if the interface goes down. When the GLBP group weighting drops below the lower threshold, the member is no longer an AVF and a secondary virtual forwarder takes over.

When the weighting rises above the upper threshold, the member can resume its role as an AVF.

Figure 46 : GLBP Object Tracking and Weighting



In the figure, the Ethernet 1/2 interface on router 1 is the gateway for host 1 (the AVF for virtual MAC address, vMAC1), while Ethernet 2/2 on router 2 acts as a secondary virtual forwarder for Host 1. Ethernet 1/2 tracks Ethernet 3/1, which is the network connection for router 1. If Ethernet 3/1 goes down, the weighting for Ethernet 1/2 drops to 90. Ethernet 2/2 on router 2 preempts Ethernet 1/2 and takes over as AVF because it has the default weighting of 100 and is configured to preempt the AVF.

19.2.7 High Availability and Extended Nonstop Forwarding

GLBP supports high availability through stateful restarts and stateful switchovers. A stateful restart occurs when the GLBP process fails and is restarted. A stateful switchover occurs when the active supervisor switches to the standby supervisor. Inspur INOS applies the run-time configuration after the switchover.

If GLBP hold timers are configured for short time periods, these timers might expire during a controlled switchover or in-service software upgrade (ISSU). GLBP supports extended non-stop forwarding (NSF) to temporarily extend these GLBP hold timers during a controlled switchover or ISSU.

With extended NSF configured, GLBP sends hello messages with the extended timers. GLBP peers update their hold timers with these new values. The extended timers prevent unnecessary GLBP state changes during the switchover or ISSU. After the switchover or ISSU event, GLBP restores the hold timers to their original configured values. If the switchover fails, GLBP restores the hold timers after the extended hold timer values expire.

19.2.9 Virtualization Support

GLBP supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

If you change the VRF membership of an interface, Inspur INOS removes all Layer 3 configuration, including GLBP.

For more information, see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide.*

19.3 Licensing Requirements for GLBP

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide*.

19.4 Prerequisites for GLBP

GLBP has the following prerequisites:

- Globally enable the GLBP feature.
- You can only configure GLBP on Layer 3 interfaces (see the *Inspur CN12700 Series INOS Interfaces*

Configuration Guide, and the *Interfaces Configuration Guide*, Inspur DCNM for LAN).

- If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

19.5 Guidelines and Limitations for GLBP

GLBP has the following configuration guidelines and limitations:

- You should configure all customization options for GLBP on all GLBP member gateways before enabling a GLBP group by configuring a virtual IP address.
- You must configure an IP address for the interface that you configure GLBP on and enable that interface before GLBP becomes active.
- The GLBP virtual IP address must be in the same subnet as the interface IP address.
- We recommend that you do not configure more than one first-hop redundancy protocol on the same interface.
- Inspur INOS removes all Layer 3 configuration on an interface when you change the VDC, interface VRF membership, port channel membership, or when you change the port mode to Layer 2.
- Inspur INOS does not support GLBP group configuration on interface secondary subnets.
- Inspur INOS does not support GLBP for IPv6.
- The GLBP does not support gratuitous ARP by design.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

19.6 Default Settings forGLBP

Table 36 : Default GLBP Parameters

Parameters	**Default**
Authentication	No authentication
Extended hold timer	10 seconds
Forwarder preemption delay	30 seconds
Forwarder timeout	14400 seconds
Hello timer	3 seconds
Hold timer	10 seconds
GLBP feature	Disabled
Load balancing	Round robin
Preemption	Disabled
Priority	100
Redirect timer	600 seconds
Weighting	100

19.7 Configuring GLBP

19.7.1 Enabling GLBP

You must enable GLBP before you can configure and enable any GLBP groups.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# [**no**] **feature glbp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# [**no**] **feature glbp**	Enables GLBP.

Example

```
switch# configure terminal

switch(config)# feature glbp

switch(config)# copy running-config startup-config
```

19.7.2 Configuring GLBP Authentication

You can configure GLBP to authenticate the protocol using cleartext or an MD5 digest. MD5 authentication uses a key chain (see the Inspur CN12700 Series INOS Security Configuration Guide).

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command).
Enable GLBP.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **ip** *ip-address/length*
4. switch(config-if)# **glbp** *group-number*
5. switch(config-if-glbp)# **authentication text** *string*
6. switch(config-if-glbp)# **authentication md5** {**key-chain** *key-chain* | **key-string** {*text* | **encrypted** *text*}}
7. switch(config-if-glbp)# **ip** [*ip-address* [**secondary**]]
8. (Optional) switch(config-if-glbp)# **show glbp** [**group** *group-number*]
9. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip** *ip-address/length*	Configures the IPv4 address for the interface.
Step 4	switch(config-if)# **glbp** *group-number*	Creates a GLBP group and enters GLBP configuration mode. The range is from 0 to 1024.
Step 5	switch(config-if-glbp)# **authentication text** *string*	Configures cleartext authentication for GLBP on this interface.

Step 6	switch(config-if-glbp)# **authentication md5** {**key-chain** *key-chain* | **key-string** {*text* | **encrypted** *text*}}	Configures MD5 authentication for GLBP on this interface.
Step 7	switch(config-if-glbp)# **ip** [*ip-address* [**secondary**]]	Enables GLBP on an interface and identifies the primary IP address of the virtual gateway. After you identify a primary IP address, you can use the **glbp** *group* **ip** command again with the secondary keyword to indicate additional IP addresses supported by this group. If you only use the **ip** keyword, GLBP learns the virtual IP address from the neighbors.
Step 8	(Optional) switch(config-if-glbp)# **show glbp** [**group** *group-number*]	Displays GLBP information.
Step 9	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

This example shows how to configure MD5 authentication for GLBP on Ethernet 1/2 after creating the key chain:

```
switch#configure terminal
switch(config)#key chainglbp-keys
switch(config-keychain)# key 0

switch(config-keychain-key)#    key-string    7 zqdest

switch(config-keychain-key) accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep
12 2008

switch(config-keychain-key)  send-lifetime  00:00:00  Jun  01  2008  23:59:59  Aug
12 2008

switch(config-keychain-key) key 1
switch(config-keychain-key) key-string 7 uaeqdyito

switch(config-keychain-key) accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec
12 2008

switch(config-keychain-key)  send-lifetime  00:00:00  Sep  12  2008  23:59:59  Nov
12 2008

switch(config)# interface ethernet 1/2

switch(config-if)# glbp 1

switch(config-if-glbp)# authenticate md5 key-chain glbp-keys

switch(config-if-glbp)#   copy   running-config startup-config
```

19.7.3 Configuring GLBP Load Balancing

You can configure GLBP to use load balancing based on round-robin, weighted, or host-dependent methods.

SUMMARY STEPS

1. switch(config-if)# **glbp** *group-number*
2. switch(config-if-glbp)# **load-balancing** [**host-dependent** | **round-robin** | **weighted**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-if)# **glbp** *group-number*	Creates a GLBP group and enters GLBP configuration mode. The range is from 0 to 1024.
Step 2	switch(config-if-glbp)# **load-balancing** [**host-dependent** | **round-robin** | **weighted**]	Sets the GLBP load-balancing method. The default is round-robin.

Example

This example shows how to configure load balancing for GLBP:

```
switch(config-if)# glbp 1

switch(config-if-glbp)# load-balancing weighted
```

19.7.4 Configuring GLBP Weighting and Tracking

You can configure GLBP weighting values and object tracking to work with the GLBP weighted load-balancing method.

You can optionally configure the interface to preempt an AVF if the interface was originally assigned with the virtual MAC address or if this interface has a higher weight than the AVF.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Enable GLBP.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *object-id* **interface** *interface-type number* {**ip routing** | **line-protocol**}
3. switch(config)# **track** *object-id* **ip route** *ip-prefix/length* **reachability**
4. switch(config)# **interface** *interface-type slot/por*
5. switch(config-if)# **ip** *ip-address/length*
6. switch(config-if)# **glbp** *group-number*
7. switch(config-if-glbp)# **weighting** *maximum* [**lower** *lower*] [**upper** *upper*]
8. switch(config-if-glbp)# **weighting-track** *object-number* [**decrement** *value*]
9. (Optional) switch(config-if-glbp)# **forwarder preempt** [**delay minimum** *seconds*]
10. switch(config-if-glbp)# **ip** [*ip-address* [**secondary**]]
11. (Optional) switch(config-if-glbp)# **show glbp** *interface-type number*
12. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **track** *object-id* **interface** *interface-type number* {**ip routing** | **line-protocol**}	Configures the interface that this GLBP interface tracks. Changes in the state of the interface affect the priority of this GLBP interface as follows: • You configure the interface and corresponding object number that you use with the **track** command in GLBP configuration mode. • The **line-protocol** keyword tracks whether the interface is up. The **ip** keyword also checks that IP routing is enabled on the interface and an IP address is configured.
Step 3	switch(config)# **track** *object-id* **ip route** *ip-prefix/length* **reachability**	Creates a tracked object for a route and enters tracking configuration mode. The *object-id* range is from 1 to 500.
Step 4	switch(config)# **interface** *interface-type slot/por*	Enters interface configuration mode.
Step 5	switch(config-if)# **ip** *ip-address/length*	Configures the IPv4 address for the interface.
Step 6	switch(config-if)# **glbp** *group-number*	Creates a GLBP group and enters GLBP configuration mode.
Step 7	switch(config-if-glbp)# **weighting** *maximum* [**lower** *lower*] [**upper** *upper*]	Specifies the initial weighting value and the upper and lower thresholds for a GLBP gateway. The maximum range is from 1 to 254. The default weighting value is 100. The lower range is from 1 to 253. The upper range is from 1 to 254.
Step 8	switch(config-if-glbp)# **weighting-track** *object-number* [**decrement** *value*]	Specifies an object to be tracked that affects the weighting of a GLBP gateway. The value argument specifies a reduction in the weighting of a GLBP gateway when a tracked object fails. The range is from 1 to 255.
Step 9	(Optional) switch(config-if-glbp)# **forwarder preempt** [**delay minimum** *seconds*]	Configures the router to take over as AVF for a GLBP group if the current AVF for a GLBP group falls below its low weighting threshold. The range is from 0 to 3600 seconds. This command is enabled by default with a delay of 30 seconds.
Step 10	switch(config-if-glbp)# **ip** [*ip-address* [**secondary**]]	Enables GLBP on an interface and identifies the primary IP address of the virtual gateway. After you identify a primary IP address, you can use the **glbp** *group* **ip** command again with the **secondary** keyword to indicate additional IP addresses supported by this group. If you only use the **ip** keyword, GLBP learns the virtual IP address from the neighbors.
Step 11	(Optional) switch(config-if-glbp)# **show glbp** *interface-type number*	Displays GLBP information for an interface.
Step 12	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure GLBP weighting and tracking on Ethernet 1/2:

```
switch# configure terminal

switch(config)# track 2 interface ethernet 2/2 ip routing

switch(config)# interface ethernet 1/2

switch(config-if)# glbp 1

switch(config-if-glbp) weighting 110 lower 95 upper 105

switch(config-if-glbp) weighting track 2 decrement 20

switch(config-if-glbp)# copy running-config startup-config
```

19.7.5 Customizing GLBP

Customizing the behavior of GLBP is optional. Be aware that as soon as you enable a GLBP group by configuring a virtual IP address, that group is operational. If you enable a GLBP group before you customize GLBP, the router could take over control of the group and become the AVG before you finish customizing the feature. If you plan to customize GLBP, you should do so before enabling GLBP.

SUMMARY STEPS

1. switch(config-if-glbp)# **glbp** *group-number*
2. switch(config-if-glbp)# **timers** [**msec**] *hellotime* [**msec**] *holdtime*
3. switch(config-if-glbp)# **timers redirect** *redirecttimeout*
4. switch(config-if-glbp)# **priority** *level*
5. switch(config-if-glbp)# **preempt** [**delay minimum** *seconds*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-if-glbp)# **glbp** *group-number*	Creates a GLBP group and enters GLBP configuration mode.
Step 2	switch(config-if-glbp)# **timers** [**msec**] *hellotime* [**msec**] *holdtime*	Configures the following hello and hold times for this GLBP member: • *hellotime*—The interval between successive hello packets sent by the AVG in a GLBP group. The range is from 1 to 60 seconds or from 250 to 60000 milliseconds. The default value is 3 seconds. • *holdtime*—The interval before the virtual gateway and virtual forwarder information in the hello packet is considered invalid. The range is from 2 to 180 seconds or from 1020 to 180000 milliseconds. The default is 10 seconds. The optional **msec** keyword specifies that the argument is expressed in milliseconds, instead of the default seconds.

Step 3	switch(config-if-glbp)# **timers redirect** *redirect timeout*	Configures the following timers: • *redirect*—The time interval in seconds during which the AVG continues to redirect clients to an AVF. The range is from 0 to 3600 seconds. The default is 600 seconds. • *timeout*—The interval in seconds before a secondary virtual forwarder becomes invalid. The range is from 610 to 64800 seconds. The default is 14,440 seconds.
Step 4	switch(config-if-glbp)# **priority** *level*	Sets the priority level used to select the AVG in a GLBP group. The range is from 1 to 255. The default is 100.
Step 5	switch(config-if-glbp)# **preempt** [**delay minimum** *seconds*]	Configures the router to take over as AVG for a GLBP group if it has a higher priority than the current AVG. This command is disabled by default. Use the optional **delay minimum** keywords and the *seconds* argument to specify a minimum delay interval in seconds before preemption of the AVG takes place. The seconds range is from 0 to 3600 seconds. The minimum delay default is 3600 seconds.

Example

The following example shows how to customize GLBP:

```
switch(config-if)# glbp 1

switch(config-if-glbp)# timers 5 18

switch(config-if-glbp)#  timers  redirect  600 7200

switch(config-if-glbp)# priority 254

switch(config-if-glbp)#  preempt  delay  minimum 60
```

19.7.6 Configuring Extended Hold Timers for GLBP

You can configure GLBP to use extended hold timers to support extended NSF during a controlled (graceful) switchover or ISSU, including software upgrades and supervisor switchovers. You should configure extended hold timers on all GLBP gateways.

Use the **show glbp** command to display the extended hold time.

SUMMARY STEPS

1. switch(config-if)# **glbp** *group-number*
2. switch(config)# **glbp timers extended-hold** [*timer*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-if)# **glbp** *group-number* **Example:**	Creates a GLBP group and enters GLBP configuration mode.

Step 2	switch(config)# **glbp timers extended-hold** [*timer*]	Sets the GLBP extended hold timer, in seconds. The timer range is from 10 to 255. The default is 10.

Example

The following example shows how to configure extended hold timers for GLBP:

```
switch(config-if)# glbp 1

switch(config)# glbp timers extended-hold 30
```

19.7.7 Enabling a GLBP Group

You can configure the virtual IP address on an interface to enable the GLBP group. You must configure each gateway in the GLBP group with the same group number. The GLBP member can learn all other required parameters from another GLBP member.

Before you begin

Ensure that you are in the correct VDC (or use the **switchto vdc** command). Enable GLBP.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/por*
3. switch(config-if)# **ip** *ip-address/length*
4. switch(config-if)# **glbp** *group-number*
5. switch(config-if-glbp)# **ip** [*ip-address* [**secondary**]]
6. switch(config-if-glbp)# **show glbp** [**group** *group-number*] [**brief**]
7. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/por*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip** *ip-address/length*	Configures the IPv4 address for the interface.
Step 4	switch(config-if)# **glbp** *group-number*	Creates a GLBP group and enters GLBP configuration mode.
Step 5	switch(config-if-glbp)# **ip** [*ip-address* [**secondary**]]	Enables GLBP on an interface and identifies the virtual IP address. The virtual IP should be in the same subnet as the interface IP address. After you identify a virtual IP address, you can use the **glbp** *group* **ip** command again with the **secondary** keyword to indicate additional IP addresses supported by this group. If you only use the **ip** keyword, GLBP learns the virtual IP address from the neighbors.
Step 6	switch(config-if-glbp)# **show glbp** [**group** *group-number*] [**brief**]	Displays a brief summary of GLBP information.
Step 7	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example
This example shows how to enable GLBP on Ethernet 1/2:

```
switch# configure terminal
switch(config)#   interface  ethernet               1/2
switch(config-if)# glbp 1
switch(config-if-glbp)# ip 192.0.2.10
```

19.8 Verifying the GLBP Configuration

To display GLBP configuration information, perform one of the following tasks:

Command	Purpose
show glbp [**group** *group-number*]	Displays the GLBP status for all or one group.
show glbp capability	Displays the GLBP capability for all or one group.
show glbp interface *interface-type slot/port*	Displays the GLBP status for an interface.
show glbp interface *interface-type slot/port* [**active**] [**disabled**] [**init**] [**listen**] [**standby**]	Displays the GLBP status for a group or interface for virtual forwarders in the selected state.
show glbp interface *interface-type slot/port* [**active**] [**disabled**] [**init**] [**listen**] [**standby**] **brief**	Displays a brief summary of the GLBP status for a group or interface for virtual forwarders in the selected state.

19.9 Configuration Examples for GLBP

The following example shows how to enable GLBP on an interface, with MD5 authentication, interface tracking, and weighted load balancing:

```
key chain glbp-keys key 0

   key-string 7 zqdest

   accept-lifetime  00:00:00  Jun  01  2008  23:59:59  Sep  12
   2008

   send-lifetime 00:00:00 Jun 01 2008 23:59:59 Aug 12 2008

  key 1

   key-string 7 uaeqdyito

   accept-lifetime  00:00:00  Aug  12  2008  23:59:59  Dec  12
   2008

   send-lifetime   00:00:00   Sep   12   2008
23:59:59 Nov 12 2008 feature glbp

track  2  interface
ethernet   2/2   ip
interface  ethernet
1/2
```

```
ip   address   192.0.2.2/8
glbp 1

  authentication  md5  key-
  chain          glbp-keys
  weighting  110  lower  95
  upper 105

  weighting track 2 decrement 20

  ip 192.0.2.10

 no shutdown
```

19.10 Related Documents for GLBP

Related Topic	Document Title
IS-IS CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
VDCs and VRFs	*Inspur CN12700 Series INOS High Availability and Redundancy Guide*

19.11 Standards for GLBP

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

19.12 Feature History for GLBP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Table 37 : Feature History for GLBP

Feature Name	Release	Feature Information
GLBP	8.4(1)	This feature was introduced.

CHAPTER 20 Configuring HSRP

This chapter contains the following sections:

- Finding Feature Information.
- Information About HSRP.
- Licensing Requirements for HSRP.
- Prerequisites for HSRP.
- Guidelines and Limitations for HSRP.
- Default Settings for HSRP Parameters.
- Configuring HSRP.
- Verifying the HSRP Configuration.
- Configuration Examples for HSRP.
- Related Documents for HSRP.
- MIBs.
- Feature History for HSRP, on page 502

20.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

20.2 Information About HSRP

HSRP is a first-hop redundancy protocol (FHRP) that allows a transparent failover of the first-hop IP router. HSRP provides first-hop routing redundancy for IP hosts on Ethernet networks configured with a default router IP address. You use HSRP in a group of routers for selecting an active router and a standby router. In a group of routers, the active router is the router that routes packets; the standby router is the router that takes over when the active router fails or when preset conditions are met.

Many host implementations do not support any dynamic router discovery mechanisms but can be configured with a default router. Running a dynamic router discovery mechanism on every host is not practical for many reasons, including administrative overhead, processing overhead, and security issues. HSRP provides failover services to these hosts.

When you use HSRP, you configure the HSRP virtual IP address as the host default router (instead of the IP address of the actual router). The virtual IP address is an IPv4 or IPv6 address that is shared among a group of routers that run HSRP.

When you configure HSRP on a network segment, you provide a virtual MAC address and a virtual IP address for the HSRP group. You configure the same virtual address on each HSRP-enabled interface in the group. You also configure a unique IP address and MAC address on each interface that acts as the real address. HSRP selects one of these interfaces to be the active router. The active router receives and routes packets destined for the virtual MAC address of the group.

HSRP detects when the designated active router fails. At that point, a selected standby router assumes control of the virtual MAC and IP addresses of the HSRP group. HSRP also selects a new standby router at that time.

HSRP uses a priority designator to determine which HSRP-configured interface becomes the default active router. To configure an interface as the active router, you assign it with a priority that is higher than the priority of all the other HSRP-configured interfaces in the group. The default priority is 100, so if you configure just one interface with a higher priority, that interface becomes the default active router.

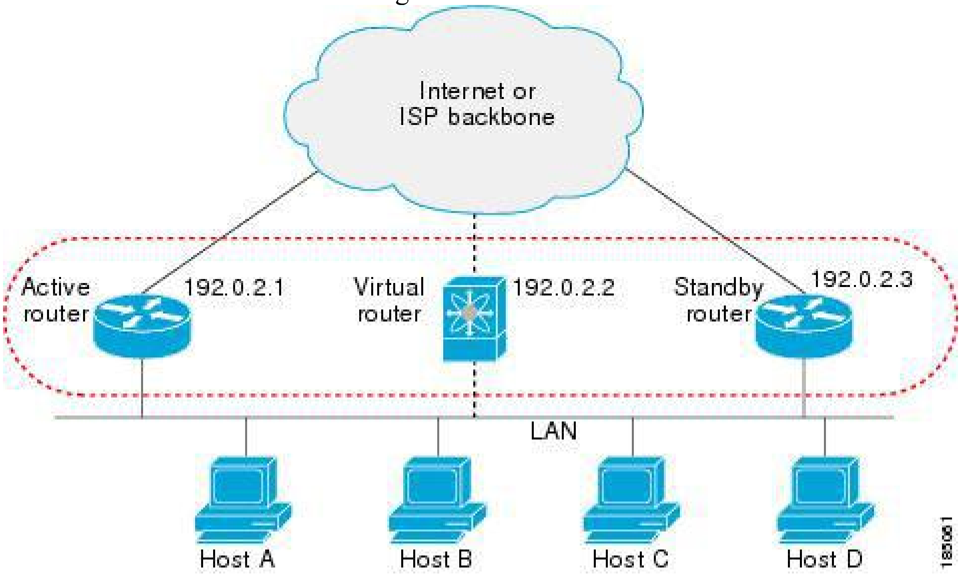
Interfaces that run HSRP send and receive multicast User Datagram Protocol (UDP)-based hello messages to detect a failure and to designate active and standby routers. When the active router fails to send a hello message within a configurable period of time, the standby router with the highest priority becomes the active router. The transition of packet forwarding functions between the active and standby router is completely transparent to all hosts on the network.

You can configure multiple HSRP groups on an interface.

A network configured for HSRP. By sharing a virtual MAC address and a virtual IP address, two or more interfaces can act as a single virtual router.

Figure 47 : HSRP Topology with Two Enabled Routers

The following figure shows a network configured for HSRP. by sharing a virtual MAC address and a virtual IP address, two or more interfaces can act as a single virtual router.



The virtual router does not physically exist but represents the common default router for interfaces that are configured to provide backup to each other. You do not need to configure the hosts on the LAN with the IP address of the active router. Instead, you configure them with the IP address of the virtual router (virtual IP address) as their default router. If the active router fails to send a hello message within the configurable period of time, the standby router takes over, responds to the virtual addresses, and becomes the active router, assuming the active router duties. From the host perspective, the virtual router remains the same.

Packets received on a routed port destined for the HSRP virtual IP address terminate on the local router, regardless of whether that router is the active HSRP router or the standby HSRP router. This includes ping and Telnet traffic. Packets received on a Layer 2 (VLAN) interface destined for the HSRP virtual IP address terminate on the active router.

20.2.1 HSRP for IPv4

HSRP routers communicate with each other by exchanging HSRP hello packets. These packets are sent to the destination IP multicast address 224.0.0.2 (reserved multicast address used to communicate to all routers) on UDP port 1985. The active router sources hello packets from its configured IP address and the HSRP virtual MAC address while the standby router sources hellos from its configured IP address and the interface MAC address, which might be the burned-in address (BIA). The BIA is the last six bytes of the MAC address that is assigned by the manufacturer of the network interface card (NIC).

Because hosts are configured with their default router as the HSRP virtual IP address, hosts must communicate with the MAC address associated with the HSRP virtual IP address. This MAC address is a virtual MAC address, 0000.0C07.ACxy, where xy is the HSRP group number in hexadecimal based on the respective interface. For example, HSRP group 1 uses the HSRP virtual MAC address of 0000.0C07.AC01. Hosts on the adjoining LAN segment use the normal Address Resolution Protocol (ARP) process to resolve the associated MAC addresses.

HSRP version 2 uses the new IP multicast address 224.0.0.102 to send hello packets instead of the multicast address of 224.0.0.2, which is used by version 1. HSRP version 2 permits an expanded group number range of 0 to 4095 and uses a new MAC address range of 0000.0C9F.F000 to 0000.0C9F.FFFF.

20.2.2 HSRP for IPv6

IPv6 hosts learn of available IPv6 routers through IPv6 neighbor discovery (ND) router advertisement (RA) messages. These messages are multicast periodically, or might be solicited by hosts, but the time delay for detecting when a default route is down might be 30 seconds or more. HSRP for IPv6 provides a much faster switchover to an alternate default router than the IPv6 ND protocol provides, less than a second if the milliseconds timers are used. HSRP for IPv6 provides a virtual first hop for IPv6 hosts.

When you configure an IPv6 interface for HSRP, the periodic RAs for the interface link-local address stop after IPv6 ND sends a final RA with a router lifetime of zero. No restrictions occur for the interface IPv6 link-local address. Other protocols continue to receive and send packets to this address.

IPv6 ND sends periodic RAs for the HSRP virtual IPv6 link-local address when the HSRP group is active. These RAs stop after a final RA is sent with a router lifetime of 0 when the HSRP group leaves the active state. HSRP uses the virtual MAC address for active HSRP group messages only (hello, coup, and redesign).

HSRP for IPv6 uses the following parameters:

- HSRP version 2
- UDP port 2029
- Virtual MAC address range from 0005.73A0.0000 through 0005.73A0.0FFF
- Multicast link-local IP destination address of FF02::66
- Hop limit set to 255

HSRP for IPv6 Addresses

An HSRP IPv6 group has a virtual MAC address that is derived from the HSRP group number and a virtual IPv6 link-local address that is derived, by default, from the HSRP virtual MAC address. The default virtual MAC address for an HSRP IPv6 group is always used to form the virtual IPv6 link-local address, regardless of the actual virtual MAC address used by the group.

The following table shows the MAC and IP addresses used for IPv6 neighbor discovery packets and HSRP packets.

Table 38 : HSRP and IPv6 ND Addresses

Packet	MAC Source Address	IPv6sourceAddress	IPv6 Destination Address	Link-layer Address Option
Neighbor solicitation (NS)	Interface MAC address	Interface IPv6 address	—	Interface MAC address
Router solicitation (RS)	Interface MAC address	Interface IPv6 address	—	Interface MAC address
Neighbor advertisement (NA)	Interface MAC address	Interface IPv6 address	Virtual IPv6 address	HSRP virtual MAC address
Route advertisement (RA)	Interface MAC address	Interface IPv6 address	—	HSRP virtual MAC address
HSRP (inactive)	Interface MAC address	Interface IPv6 address	—	—
HSRP (active)	Virtual MAC address	Interface IPv6 address	—	—

HSRP does not add IPv6 link-local addresses to the Unicast Routing Information Base (URIB). There are also no secondary virtual IP addresses for link-local addresses.

For global unicast addresses, HSRP adds the virtual IPv6 address to the URIB and IPv6, but does not register the virtual IPv6 addresses to ICMPv6. ICMPv6 redirects are not supported for HSRP IPv6 groups.

20.2.3 Multiple Group Optimization for HSRP

Beginning with Inspur INOS Release 8.4(1), HSRP supports multiple group optimization (MGO). MGO optimizes performance and bandwidth when multiple HSRP groups are configured on many subinterfaces. MGO requires only one HSRP group, known as the master group, on the physical interface for the purpose of electing active and standby routers.

You can create other HSRP groups on subinterfaces of the physical interface or a different interface, such as an SVI interface, and link these to the master HSRP group. These groups are known as slave groups. Slave groups follow their master group state so that they do not participate in any HSRP election mechanisms.

Master groups send hello messages at their configured rates. Slave groups send hello messages at a reduced rate, which is called the mac-refresh interval rate. This process is required so that the slave groups can send out periodic messages in order to refresh MAC addresses in switches and learning bridges.

20.2.4 HSRP Versions

Inspur INOS supports HSRP version 1 by default. You can configure an interface to use HSRP version 2. HSRP version 2 has the following enhancements to HSRP version 1:

Expands the group number range. HSRP version 1 supports group numbers from 0 to 255. HSRP version 2 supports group numbers from 0 to 4095.

For IPv4, uses the IPv4 multicast address 224.0.0.102 or the IPv6 multicast address FF02::66 to send hello packets instead of the multicast address of 224.0.0.2, which is used by HSRP version 1.

Uses the MAC address range from 0000.0C9F.F000 to 0000.0C9F.FFFF for IPv4 and 0005.73A0.0000 through 0005.73A0.0FFF for IPv6 addresses. HSRP version 1 uses the MAC address range 0000.0C07.AC00 to 0000.0C07.ACFF.

Adds support for MD5 authentication.

When you change the HSRP version, Inspur INOS reinitializes the group because it now has a new virtual MAC address.

HSRP version 2 has a different packet format than HSRP version 1. The packet format uses a type-length-value (TLV) format. HSRP version 2 packets received by an HSRP version 1 router are ignored.

20.2.5 HSRP Authentication

HSRP message digest 5 (MD5) algorithm authentication protects against HSRP-spoofing software and uses the industry-standard MD5 algorithm for improved reliability and security. HSRP includes the IPv4 or IPv6 address in the authentication TLVs.

20.2.6 HSRP Messages

Routers that are configured with HSRP exchange the following three types of multicast messages:

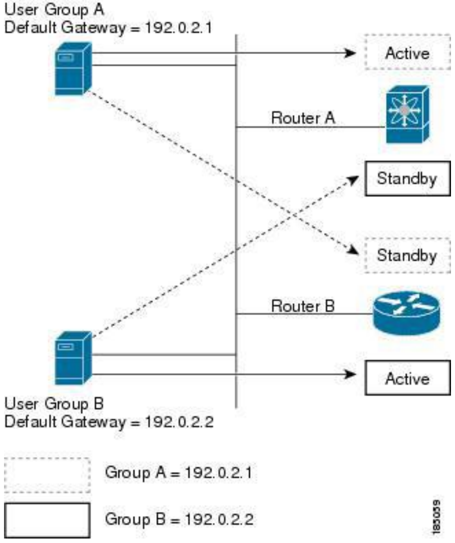
- Hello—The hello message conveys the HSRP priority and state information of the router to other HSRP routers.
- Coup—When a standby router wants to assume the function of the active router, it sends a coup message.
- Resign—A router that is the active router sends this message when it is about to shut down or when a router that has a higher priority sends a hello or coup message.

20.2.7 HSRP Load Sharing

HSRP allows you to configure multiple groups on an interface. You can configure two overlapping IPv4 HSRP groups to load share traffic from the connected hosts while providing the default router redundancy expected from HSRP. The following figure shows an example of a load-sharing HSRP IPv4 configuration.

Figure 48 : HSRP Load Sharing

The figure shows two routers A and B and two HSRP groups. Router A is the active router for group A but is the standby router for group B. Similarly, router B is the active router for group B and the standby router for group A. If both routers remain active, HSRP load balances the traffic from the hosts across both routers. If either router fails, the remaining router continues to process traffic for both hosts.



20.2.8 Object Tracking and HSRP

You can use object tracking to modify the priority of an HSRP interface based on the operational state of another interface. Object tracking allows you to route to a standby router if the interface to the main network fails.

Two objects that you can track are the line protocol state of an interface or the reachability of an IP route. If the specified object goes down, Inspur INOS reduces the HSRP priority by the configured amount.

20.2.9 vPC and HSRP

HSRP interoperates with virtual port channels (vPCs). vPCs allow links that are physically connected to two different Inspur CN devices to appear as a single port channel by a third device. See the *Inspur CN12700 Series INOS Layer 2 Switching Configuration Guide*, for more information on vPCs.

vPC forwards traffic through both the active HSRP router and the standby HSRP router.

vPC Peer Gateway and HSRP

Some third-party devices can ignore the HSRP virtual MAC address and instead use the source MAC address of an HSRP router. in a vPC environment, the packets using this source MAC address may be sent across the vPC peer link, causing a potential dropped packet. Configure the vPC peer gateway to enable the HSRP routers to directly handle packets sent to the local vPC peer MAC address and the remote vPC peer MAC address, as well as the HSRP virtual MAC address. See the *Inspur CN12700 Series INOS Layer 2 Switching Configuration Guide,* for more information on the vPC peer gateway.

For mixed-chassis configurations where the vPC peer link is configured on an F-series module, configure the vPC peer gateway exclude option to exclude the Layer 3 backup route that traverses the vPC peer link. See the *Inspur CN12700 Series INOS Layer 2 Switching Configuration Guide,* for more information on the vPC peer gateway exclude option.

20.2.10 FabricPath Anycast HSRP

Inspur INOS Release 8.4(1) and later releases facilitate further scalability at the spine layer by providing support for more than two nodes. You can create an anycast bundle, which is an association between a set of VLANs and an anycast switch ID. The set of VLANs or the HSRP group elects an active router and a standby router. The remaining routers in the group are in listen state.

All of the HSRP routers that have a configured anycast switch ID advertise the ID through FabricPath IS-IS. The active HSRP router is the only router that uses the anycast switch ID in its hello packets. The leaf switches learn that the anycast switch ID is reachable by all of the routers in the group.

All of the first hop gateways at the spine layer need to function in active-active forwarding mode. IP packets are received by any of the spine switches with the destination set as the gateway MAC address, and these packets are terminated and locally forwarded.

20.2.11 BFD

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol that provides fast-forwarding and path-failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide*, for more information.

20.2.12 High Availability and Extended Nonstop Forwarding

HSRP supports stateful restarts and stateful switchovers. A stateful restart occurs when the HSRP process fails and is restarted. A stateful switchover occurs when the active supervisor switches to the standby supervisor. Inspur INOS applies the run-time configuration after the switchover.

If HSRP hold timers are configured for short time periods, these timers might expire during a controlled switchover or in-service software upgrade (ISSU). Ping to a virtual IP is also unreachable during this timer expiry period. HSRP supports extended non-stop forwarding (NSF) to temporarily extend these HSRP hold timers during a controlled switchover or ISSU.

With extended NSF configured, HSRP sends hello messages with the extended timers. HSRP peers update their hold timers with these new values. The extended timers prevent unnecessary HSRP state changes during the switchover or ISSU. After the switchover or ISSU event, HSRP restores the hold timers to their original configured values. If the switchover fails, HSRP restores the hold timers after the extended hold timer values expire.

20.2.13 Virtualization Support

HSRP supports virtual routing and forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF.

If you change the VRF membership of an interface, Inspur INOS removes all Layer 3 configuration, including HSRP.

20.2.14 HSRP VIP

Starting with Inspur INOS Release 8.4(1) the Hot Standby Router Protocol (HSRP) Virtual IP (VIP) feature provides support for an HSRP Virtual IP configuration to be in a different subnet than that of the interface subnet. This feature is supported only for IPv4 address and not for IPv6. The following are the enhancements:

・Enhance ARP to source with VIP from Supervisor Engine (SUP) for hosts, when the hosts in VIP subnet are referenced by static route to VLAN configuration.

・Support periodic ARP synchronization to vPC peer if the HSRP VIP feature is enabled.

・Allow VIP address as the Layer 3 source address and gateway address for all communications with a Dynamic Host Configuration Protocol (DHCP) server.

・Enhance DHCP relay agent to relay DHCP packets with VIP address as source address instead of SVI IP address.

The following is an example for VIP subnet address configuration wherein the VIP address is not configured in the same subnet of the interface IP subnet.

```
switch#   configure   terminal
switch(config)# feature  hsrp
switch(config)#       feature interface-vlan
switch(config)# interface vlan 2

switch(config-if)#  ip address 192.0.2.1/24

switch(config-if)#  hsrp 2

switch(config-if-hsrp)#  ip 209.165.201.1/24
```

The following is an example for VIP address mismatch. Here the VIP address is not in the same subnet of the interface IP subnet.

```
switch#     configure   terminal
switch(config)#    feature  hsrp
switch(config)#             feature    interface-vlan
switch(config)# interface vlan 2
switch(config-if)#     ip     address     192.0.2.1/24
switch(config-if)#  hsrp 2
switch(config-if-hsrp)#  ip 209.165.201.1



!ERROR:  Invalid  IP  address(Mismatch  with  IP
subnet)!
```

The following is an example for VIP address mismatch. Here the VIP subnet address is configured along with VIP address in the same subnet of the interface IP subnet.

```
switch#   configure   terminal
switch(config)#  feature  hsrp

switch(config)# feature interface-vlan
switch(config)# interface vlan 2
switch(config-if)#  ip address 192.0.2.1/24

switch(config-if)#  hsrp 2

switch(config-if-hsrp)#  ip 192.0.2.10/24



!ERROR:  Invalid  IP  address(Mismatch  with  IP
subnet)!
```

20.3 Licensing Requirements for HSRP

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the *Inspur INOS Licensing Guide.*

20.4 Prerequisites for HSRP

You must enable the HSRP feature in a device before you can configure and enable any HSRP groups.

If you configure VDCs, install the Advanced Services license and enter the desired VDC (see the *Inspur INOS Virtual Device Context Configuration Guide*).

20.5 Guidelines and Limitations for HSRP

HSRP has the following configuration guidelines and limitations:

- You must configure an IP address for the interface on which you configure HSRP and enable that interface before HSRP becomes active.
- You must configure HSRP version 2 when you configure an IPv6 interface for HSRP.
- For IPv4, the virtual IP address must be in the same subnet as the interface IP address.
- The value of the first 2 digits of a type **7 key string** configured by using the key-string 7 text-string command has to be between 0 and 15. For example, you can configure 07372b557e2c1a as the key string value in which case the sum value of the first 2 digits will be 7. But, you cannot configure 85782916342021 as the key string value because the value of the first 2 digits will be 85. We recommend unconfiguring any type 7 key strings that do not adhere to this value or to configure a type 0 string.
- We recommend that you do not configure more than one first-hop redundancy protocol on the same interface.
- HSRP version 2 does not interoperate with HSRP version 1. An interface cannot operate both version 1 and version 2 because both versions are mutually exclusive. However, the different versions can be run on different physical interfaces of the same router.
- You cannot change from version 2 to version 1 if you have configured groups above the group number range allowed for version 1 (0 to 255).
- HSRP for IPv4 is supported with BFD. HSRP for IPv6 is not supported with BFD.
- Inspur INOS removes all Layer 3 configurations on an interface when you change the interface VRF membership, port channel membership, or when you change the port mode to Layer 2.
- If you configure virtual MAC addresses with vPC, you must configure the same virtual MAC address on both vPC peers.
- For mixed-chassis configurations where the vPC peer link is configured on an F-series module, configure the vPC peer gateway exclude option to exclude the Layer 3 backup route that traverses the vPC peer link.
- You cannot use the HSRP MAC address burned-in option on a VLAN interface that is a vPC member.
- If you have not configured authentication, the **show hsrp** command displays the following string:

```
Authentication text "Inspur"
```

- The following is the default behavior of HSRP as defined in RFC 2281:

```
If no authentication data is configured, the RECOMMENDED default value
is 0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00.
```

- Anycast HSRP does not support BFD.
- HSRP for MGO has the following limitations:
  - Master groups and slave groups are not restricted to the same interface.
  - HSRP for MGO supports only HSRP version 2.
  - Master and slave groups must have the same address types.
  - Configuring an HSRP group as a slave group clears the group's other configurations, such as its virtual IP address, without notification, so you must enter the **follow** command before you enter the **ip***ip-address* command.
  - Bidirectional forwarding (BFD) is not applicable to slave groups.
  - HSRP for MGO supports both IPv4 and IPv6 interfaces and works for all Layer 3 interfaces on which a regular HSRP group works.
  - An HSRP group cannot be configured as both a master and slave group at the same time.

20.6 Default Settings for HSRP Parameters

Default HSRP Parameters

Parameters	Default
HSRP	Disabled

Authentication	Enabled as text for version 1, with Inspur as the password
HSRP version	Version 1
Preemption	Disabled
Priority	100
Virtual MAC address	Derived from HSRP group number

20.7 Configuring HSRP

20.7.1 Enabling HSRP

You must globally enable HSRP before you can configure and enable any HSRP groups.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] feature hsrp**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **[no] feature hsrp**	Enables the HSRP feature. Use the **no** form of this command to disable this feature. You can use this command to enable or disable the HSRP feature and remove all associated configurations in a VDC in the global configuration mode.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example enables HSRP:

```
switch # configure terminal

switch(config)# feature hsrp

switch(config)# copy running-config startup-config
```

20.7.2 Configuring the HSRP Version

You can configure the HSRP version. If you change the version for existing groups, Inspur INOS reinitializes HSRP for those groups because the virtual MAC address changes. The HSRP version applies to all groups on the interface

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type number*
3. switch(config-if)# **hsrp version {1 | 2}**
4. (Optional) switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 3	switch(config-if)# **hsrp version {1 | 2}**	Confirms the HSRP version. Version 1 is the default.
Step 4	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure an HSRP version:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# hsrp version 2
switch(config-if)# copy running-config startup-config
```

20.7.3 Configuring an HSRP Group for IPv4

You can configure an HSRP group on an IPv4 interface and configure the virtual IP address and virtual MAC address for the HSRP group.

Before you begin

- You must enable HSRP.
- Inspur INOS enables an HSRP group once you configure the virtual IP address on any member interface in the group. You must configure HSRP attributes such as authentication, timers, and priority before you enable the HSRP group.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type number*
3. switch(config-if)# **ip** *ip-address/length*
4. switch(config-if)# **hsrp** *group-number* [**ipv4**]
5. switch(config-if-hsrp)# **ip** [*ip-address* [**secondary**]]
6. switch(config-if-hsrp)# **exit**
7. switch(config-if)# **no shutdown**

8. (Optional) switch(config-if)# **copy running-config startup-config**
9. (Optional) switch(config-if)# **show hsrp** [**group** *group-number*] [**ipv4**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip** *ip-address/length*	Configures the IPv4 address of the interface.
Step 4	switch(config-if)# **hsrp** *group-number* [**ipv4**]	Creates an HSRP group and enters hsrp configuration mode. The range for HSRP version 1 is from 0 to 255. The range is for HSRP version 2 is from 0 to 4095. The default value is 0.
Step 5	switch(config-if-hsrp)# **ip** [*ip-address* [**secondary**]]	Configures the virtual IP address for the HSRP group and enables the group. This address should be in the same subnet as the IPv4 address of the interface.
Step 6	switch(config-if-hsrp)# **exit**	Exits HSRP configuration mode.
Step 7	switch(config-if)# **no shutdown**	Enables the interface.
Step 8	(Optional) switch(config-if)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 9	(Optional) switch(config-if)# **show hsrp** [**group** *group-number*] [**ipv4**]	Displays HSRP information.

Example

The following example shows how to configure an HSRP group on Ethernet 1/2:

```
switch# configure terminal

switch(config)# interface ethernet 1/2



switch(config-if)# ip 192.0.2.2/8

switch(config-if)# hsrp 2

switch(config-if-hsrp)#  ip  192.0.2.1
switch(config-if-hsrp)#          exit
switch(config-if)# no shutdown

switch(config-if)# copy running-config startup-config
```

20.7.4 Configuring an HSRP Group for IPv6

You can configure an HSRP group on an IPv6 interface and configure the virtual IP address and virtual MAC address for the HSRP group. When you configure an HSRP group for IPv6, HSRP generates a link-local address from the link-local prefix. HSRP also generates a modified EUI-64 format interface identifier in which the EUI-64 interface identifier is created from the relevant HSRP virtual MAC address. There are no HSRP IPv6 secondary addresses.

Before you begin

• You must enable HSRP.

• Ensure that you have enabled HSRP version 2 on the interface where you want to configure an IPv6 HSRP group.

• Ensure that you have configured HSRP attributes such as authentication, timers, and priority before you enable the HSRP group.

• Confirm that you are in the correct VDC. To change the VDC, use the switchto vdc command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type number*
3. switch(config-if)# **ipv6 address** *ipv6-address/length*
4. switch(config-if)# **hsrp version 2**
5. switch(config-if)# **hsrp** *group-number* [**ipv6**]
6. switch(config-if-hsrp)# **ip** [*ipv6-address* [**secondary**]]
7. switch(config-if-hsrp)# **ip autoconfig**
8. switch(config-if-hsrp)# **no shutdown**
9. switch(config-if-hsrp)# **copy running-config startup-config**
10. (Optional) switch(config-if-hsrp)# **show hsrp** [**group** *group-number*] [**ipv6**]

DETAILED STEPS

	Command or Action	Purpose
Step 1		
Step 2	switch# **configure terminal**	Enters global configuration mode.
Step 3	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 4	switch(config-if)# **ipv6 address** *ipv6-address/length*	Configures the IPv6 address of the interface.
Step 5	switch(config-if)# **hsrp version 2**	Configures the group for HSRP version 2.
Step 6	switch(config-if)# **hsrp** *group-number* [**ipv6**]	Creates an IPv6 HSRP group and enters hsrp configuration mode. The range for HSRP version 2 is from 0 to 4095. The range is for HSRP version 2 is from 0 to 4095. The default value is 0.
Step 7	switch(config-if-hsrp)# **ip** [*ipv6-address* [**secondary**]]	Configures the virtual IPv6 address for the HSRP group and enables the group.
Step 8	switch(config-if-hsrp)# **ip autoconfig**	Autoconfigures the virtual IPv6 address for the HSRP group from the calculated link-local virtual IPv6 address and enables the group.
Step 9	switch(config-if-hsrp)# **no shutdown**	Enables the interface.
Step 10	switch(config-if-hsrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 11	(Optional) switch(config-if-hsrp)# **show hsrp** [**group** *group-number*] [**ipv6**]	Displays HSRP information.

Example

The following example shows how to configure an HSRP group on Ethernet 1/2:

```
switch# configure terminal

switch(config)# interface ethernet 3/2

switch(config-if)# ipv6 address 2001:0DB8:0001:0001:/64

switch(config-if)# hsrp    version    2
switch(config-if)# hsrp     2     ipv6
switch(config)#   ip 2001:DB8::1
switch(config-if-hsrp)#          exit
switch(config-if-hsrp)# no shutdown

switch(config-if-hsrp)# copy running-config startup-config
```

20.7.5 Configuring an HSRP Master Group Task

You can configure HSRP for MGO to optimize performance when scaling by configuring master and slave groups. Slave groups follow the master group state, which minimizes the number of hello messages that are sent. Inspur INOS enables an HSRP group once you configure its virtual IP address.

We recommend that you configure master groups on the same parent interface as their slave groups to allow the slave groups to have the same redundancy requirements as the master group. If a failure occurs on the master link, all the slave groups are brought down as well, even if the links on which they are configured remain up.

Before you begin

- Ensure that you have enabled the HSRP feature.
- Configure HSRP attributes such as authentication, timers, and priority before you enable an HSRP group as a master group.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type/number*
3. switch(config-if)# **ip address***ip-address/length*
4. switch(config-if)# **hsrp version 2**
5. switch(config-if)# **hsrp** *group-number* [**ipv6**]
6. switch(config-if-hsrp)# **name** [*master-group-name*]
7. switch(config-if-hsrp)# **ip** [*ip-address* [**secondary**]]
8. switch(config-if-hsrp)# **exit**
9. switch(config-if)# **no shutdown**
10. switch(config-if)# **show hsrp** [**brief**] [**group** *group-number*] [**ipv4**] [**ipv6**]
11. switch(config-if)# **show hsrp mgo** [**name** *name*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **interface** *type/number*	Enters interface configuration mode and configures an interface type.
Step 3	switch(config-if)# **ip address***ip-address/length*	Configures the IP address of the interface.
Step 4	switch(config-if)# **hsrp version 2**	Configures the HSRP version. Because MGO supports only HSRP version 2, you must set the HSRP version to version 2. Version 1 is the default.
Step 5	switch(config-if)# **hsrp** *group-number* [**ipv6**]	Creates an HSRP group and enters HSRP configuration mode. The range for the HSRP group number is from 0 to 4095. The no form of this command removes the group.
Step 6	switch(config-if-hsrp)# **name** [*master-group-name*]	Specifies a master group name. The name command changes a regular HSRP group into a master group. If you do not specify a name, a unique name is automatically generated. The no form of this command returns the master group to a regular HSRP group.
Step 7	switch(config-if-hsrp)# **ip** [*ip-address* [**secondary**]]	Configures the virtual IP address for the HSRP group and enables the master group.
Step 8	switch(config-if-hsrp)# **exit**	Exits the HSRP configuration mode.
Step 9	switch(config-if)# **no shutdown**	Enables the interface.
Step 10	switch(config-if)# **show hsrp** [**brief**] [**group** *group-number*] [**ipv4**] [**ipv6**]	(Optional) Displays HSRP information.
Step 11	switch(config-if)# **show hsrp mgo** [**name** *name*] [**brief**]	(Optional) Displays the relationships between HSRP groups that are in use for MGO and their slave sessions. The name keyword restricts the output to the session with a matching configured name. The brief keyword provides a summary of each MGO session with the associated slave sessions.

Example

The following example shows how to configure an HSRP master group on Ethernet interface 1/1:

```
switch#  configure  terminal
switch(config)# interface ethernet 1/1
switch(config-if)# ip address 11.0.0.1/24
switch(config-if)# hsrp version 2
switch(cofig-if)# hsrp 11
switch(config-if-hsrp)# name master1
switch(config-if-hsrp)# ip 11.0.0.100
switch(config-if-hsrp)# exit
switch(config-if)# no shutdown
switch(config-if)# show hsrp group 11
switch(config-if)# show hsrp mgo name master1
```

20.7.6 Configuring an HSRP Slave Group

If a failure occurs in a slave link that belongs to a different interface than the master group, the slave group is brought down, regardless of the state of the group it is following.

You can configure HSRP for MGO to optimize performance when scaling by configuring master and slave groups. Slave groups follow the master group state, which minimizes the number of hello messages that are sent. Inspur INOS enables an HSRP group once you configure its virtual IP address.

We recommend that you configure master groups on the same parent interface as their slave groups to allow the slave groups to have the same redundancy requirements as the master group. If a failure occurs on the master link, all the slave groups are brought down as well, even if the links on which they are configured remain up.

Before you begin

- Ensure that you have enabled the HSRP feature.
- Configure HSRP attributes such as authentication, timers, and priority before you enable an HSRP group as a master group.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type/number*
3. switch(config-if)# **ip address***ip-address/length*
4. switch(config-if)# **hsrp version 2**
5. switch(config-if)# **hsrp mac refresh** *seconds*
6. switch(config-if)# **hsrp** *group-number* [**ipv6**]
7. switch(config-if-hsrp)# **follow** [*master-group-name*]
8. switch(config-if-hsrp)# **ip** [*ip-address*]
9. switch(config-if-hsrp)# **exit**
10. switch(config-if)# **no shutdown**
11. switch(config-if)# **show hsrp** [**brief**] [**group** *group-number*] [**ipv4**] [**ipv6**]
12. switch(config-if)# **show hsrp mgo** [**name** *name*] [**brief**]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type/number*	Enters interface configuration mode and configures an interface type.
Step 3	switch(config-if)# **ip address***ip-address/length*	Configures the IP address of the interface.
Step 4	switch(config-if)# **hsrp version 2**	Configures the HSRP version. Because MGO supports only HSRP version 2, you must set the HSRP version to version 2. Version 1 is the default.

Step 5	switch(config-if)# **hsrp mac refresh** *seconds*	(Optional) Configures the MAC refresh interval for the HSRP slave group. You can use this command to minimize the number of hello messages that are sent out and reduce HSRP protocol overheads and CPU utilization when multiple subinterfaces are configured. This command is not available for individual subinterfaces. It applies to all groups on all subinterfaces. The default is 60 seconds. The range is from 0 to 10000.
Step 6	switch(config-if)# **hsrp** *group-number* [**ipv6**]	Creates an HSRP group and enters HSRP configuration mode. The range for the HSRP group number is from 0 to 4095. The no form of this command removes the group.
Step 7	switch(config-if-hsrp)# **follow** [*master-group-name*]	Configures a regular HSRP group as a slave group. Configuring an HSRP group as a slave group clears the group's other configurations, such as its virtual IP address without notification, so you must enter the follow command before you enter the ip ip-address command. Slave groups may forward reference master group names that are undefined. The no form of this command returns the slave group to a regular HSRP group.
Step 8	switch(config-if-hsrp)# **ip** [*ip-address*]	Configures the virtual IP address for the HSRP group and enables the slave group.
Step 9	switch(config-if-hsrp)# **exit**	Exits the HSRP configuration mode.
Step 10	switch(config-if)# **no shutdown**	Enables the interface.
Step 11	switch(config-if)# **show hsrp** [**brief**] [**group** *group-number*] [**ipv4**] [**ipv6**]	(Optional) Displays HSRP information.
Step 12	switch(config-if)# **show hsrp mgo** [**name** *name*] [**brief**]	(Optional) Displays the relationships between HSRP groups that are in use for MGO and their slave sessions. The name keyword restricts the output to the session with a matching configured name. The brief keyword provides a summary of each MGO session with the associated slave sessions.

Example

The following example shows how to configure an HSRP slave group on Ethernet interface 1/2:

```
switch# configure terminal
switch(config)#  interface ethernet 1/2
switch(config-if)# ip address 12.0.0.1/24
switch(config-if)# hsrp version 2
switch(cofig-if)# hsrp 12
switch(config-if-hsrp)# follow master1

switch(config-if-hsrp)#  ip 12.0.0.100

switch(config-if-hsrp)# exit

switch(config-if)#    no shutdown

switch(config-if)#  show hsrp group 11

switch(config-if)# show hsrp mgo name master1
```

20.7.7 Configuring the HSRP Virtual MAC Address Manually

You can override the default virtual MAC address that HSRP derives from the configured group number. You must configure the same virtual MAC address on both vPC peers of a vPC link.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type number*
3. switch(config-if)# **hsrp** *group-number* [**ipv4**]
4. switch(config-if-hsrp)# **mac-address** *string*
5. switch(config-if-hsrp)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 3	switch(config-if)# **hsrp** *group-number* [**ipv4**]	Creates an HSRP group and enters hsrp configuration mode. The range for HSRP version 1 is from 0 to 255. The range is for HSRP version 2 is from 0 to 4095. The default value is 0.
Step 4	switch(config-if-hsrp)# **mac-address** *string*	Configures the virtual MAC address for an HSRP group. The string uses the standard MAC address format (xxxx.xxxx.xxxx).
Step 5	switch(config-if-hsrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example configures the HSRP virtual MAC address manually:

```
switch# configure termnial

switch(config)#   interface ethernet               1/2

switch(config-if)# hsrp 2

switch(config-if-hsrp)# mac-address 5000.1000.1060
switch(config-if-hsrp)#   copy   running-config startup-config
```

20.7.8 Configuring the HSRP Virtual MAC Address Using Burned-in MAC Address

You can override the default virtual MAC address that HSRP derives from the configured group number. You must configure the same virtual MAC address on both vPC peers of a vPC link.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type number*
3. switch(config-if)# **hsrp use-bia** [**scope interface**]
4. switch(config-if)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 3	switch(config-if)# **hsrp use-bia** [**scope interface**]	Configures HSRP to use the burned-in MAC address of the interface for the HSRP virtual MAC address. Optionally, you can configure HSRP to use the burned-in MAC address for all groups on this interface by using the scope interface keyword. **Note** Proxy ARP breaks when HSRP is configured with use-bia command. A standby router cannot cover for the lost proxy ARP database of the failed router. When the use-bia option is configured, the ARP process on the HSRP active device mistakenly sees the HSRP group as the standby device because of the lack of virtual address that it looks for. As a result, both the HSRP active and the standby devices suppress ARP replies to proxy ARP requests.
Step 4	switch(config-if)# **copy running-config startup-config**	(Optional) Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration

Example

The following example configures the HSRP virtual MAC address manually:

```
switch   #   configure terminal
switch(config)# interface  ethernet  1/2
switch(config-if)# hsrp use-bia

switch(config-if)# copy running-config startup-config
```

20.7.9 Authenticating HSRP

You can configure HSRP to authenticate the protocol using cleartext or MD5 digest authentication. MD5 authentication uses a key chain. For more details, see the *Inspur CN12700 Series INOS Security Configuration Guide.*

Before you begin

- You must enable HSRP.
- You must configure the same authentication and keys on all members of the HSRP group.
- Ensure that you have created the key chain if you are using MD5 authentication.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **hsrp** *group-number* [**ipv4** | **ipv6**]
4. switch(config-if-hsrp)# **authentication** {**text** *string* | **md5** {**key-chain** *key-chain* | **key-string** {**0** | **7**} *text* [**timeout** *seconds*]}}
5. switch(config-if-hsrp)# **copy running-config startup-config**
6. (Optional) switch(config-if-hsrp)# **show hsrp** [**group** *group-number*]

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **hsrp** *group-number* [**ipv4**	**ipv6**]	Creates an HSRP group and enters HSRP configuration mode.
Step 4	switch(config-if-hsrp)# **authentication** {**text** *string*	**md5** {**key-chain** *key-chain*	**key-string** {**0**	**7**} *text* [**timeout** *seconds*]}}	Configures cleartext authentication for HSRP on this interface by using the **authentication text** command, or you can configure MD5 authentication for HSRP on this interface using the **authentication md5** command. If you configure MD5 authentication, you can use a key chain or key string. If you use a key string, you can optionally set the timeout for when HSRP only accepts a new key. The range is from 0 to 32767 seconds.
Step 5	switch(config-if-hsrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 6	(Optional) switch(config-if-hsrp)# **show hsrp** [**group** *group-number*]	Displays HSRP information.

Example

The following example shows how to configure MD5 authentication for HSRP on Ethernet 1/2 after creating the key chain:

```
switch# configure terminal

switch  (config)#  interface ethernet              1/2
```

```
switch(config)#      key      chain     hsrp-keys
switch(config-keychain)# key 0

switch(config-keychain-key)#    key-string    7 zqdest

switch(config-keychain-key) accept-lifetime 00:00:00 Jun 01 2010 23:59:59 Sep
12 2010

switch(config-keychain-key)  send-lifetime  00:00:00  Jun  01  2010  23:59:59  Aug
12 2010

switch(config-keychain-key) key 1

switch(config-keychain-key)key-string 7 uaeqdyito
switch(config-keychain-key) accept-lifetime 00:00:00 Aug 12 2010 23:59:59 Dec
12 2010

switch(config-keychain-key)  send-lifetime  00:00:00  Sep  12  2010  23:59:59  Nov
12 2010

switch(config-keychain-key)# interface ethernet 1/2



switch(config-if)# hsrp 2

switch(config-if-hsrp)# authenticate md5 key-chain hsrp-keys

switch(config-if-hsrp)#   copy   running-config startup-config
```

20.7.10 Configuring HSRP Object Tracking

You can configure an HSRP group to adjust its priority based on the availability of other interfaces or routes. The priority of a device can change dynamically if it has been configured for object tracking and the object that is being tracked goes down.

The tracking process periodically polls the tracked objects and notes any value change. The value change triggers HSRP to recalculate the priority. The HSRP interface with the higher priority becomes the active router if you configure the HSRP interface for preemption.

SUMMARY STEPS

1. switch# **configure terminal**
2. Perform one of the following tasks:
3. switch(config)# **interface** *interface-type slot/port*
4. switch(config-if)# **hsrp** *group-number* [**ipv4** | **ipv6**]
5. switch(config-if-hsrp)# **priority** [*value*]
6. switch(config-if-hsrp)# **track** *object-number* [**decrement** *value*]
7. switch(config-if-hsrp)# **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
8. switch(config-if-hsrp)# **copy running-config startup-config**
9. (Optional) switch(config-if-hsrp)# **show hsrp interface** *interface-type number*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	Perform one of the following tasks:	

Step 3	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 4	switch(config-if)# **hsrp** *group-number* [**ipv4** | **ipv6**]	Creates an HSRP group and enters hsrp configuration mode.
Step 5	switch(config-if-hsrp)# **priority** [*value*]	Sets the priority level used to select the active router in an HSRP group. The range is from 0 to 255. The default is 100.
Step 6	switch(config-if-hsrp)# **track** *object-number* [**decrement** *value*]	Specifies an object to be tracked that affects the weighting of an HSRP interface. The value argument specifies a reduction in the priority of an HSRP interface when a tracked object fails. The range is from 1 to 255. The default is 10.
Step 7	switch(config-if-hsrp)# **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]	Configures the router to take over as the active router for an HSRP group if it has a higher priority than the current active router. This command is disabled by default. The range is from 0 to 3600 seconds.
Step 8	switch(config-if-hsrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 9	(Optional) switch(config-if-hsrp)# **show hsrp interface** *interface-type number*	Displays HSRP information for an interface.

Example

The following example shows how to configure HSRP object tracking on Ethernet 1/2:

```
switch# configure terminal

switch(config)#  track  1  interface  ethernet  2/2  line-protocol

switch(config)# interface ethernet 1/2


switch(config-if)# hsrp 2

switch(config-if-hsrp)# track 1 decrement 20

switch(config-if-hsrp)# copy running-config startup-config
```

20.7.11 Configuring the HSRP Priority

You can configure the HSRP priority on an interface. HSRP uses the priority to determine which HSRP group member acts as the active router. If you configure HSRP on a vPC-enabled interface, you can optionally configure the upper and lower threshold values to control when to fail over to the vPC trunk. If the standby router priority falls below the lower threshold, HSRP sends all standby router traffic across the vPC trunk to forward through the active HSRP router. HSRP maintains this scenario until the standby HSRP router priority increases above the upper threshold.

For IPv6 HSRP groups, if all group members have the same priority, HSRP selects the active router based on the IPv6 link-local address.

For IPv4 HSRP groups, HSRP selects the active router based on the interface IP address when the priority is same.

After Inspur INOS Release 8.4(1), even if the HSRP peer has a higher source interface IP address than the existing HSRP active peer and if preemption is enabled, the HSRP peer that has the same priority as the existing HSRP active peer does not preempt the existing HSRP active peer in the network.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type number*
3. switch(config-if)# **hsrp** *group-number* [**ipv4**]
4. switch(config-if-hsrp)# **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]
5. switch(config-if-hsrp)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type number*	Enters interface configuration mode.
Step 3	switch(config-if)# **hsrp** *group-number* [**ipv4**]	Creates an HSRP group and enters hsrp configuration mode. The range for HSRP version 1 is from 0 to 255. The range is for HSRP version 2 is from 0 to 4095. The default value is 0.
Step 4	switch(config-if-hsrp)# **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]	Sets the priority level used to select the active router in an HSRP group in interface configuration mode. The level range is from 0 to 255. The default is 100. Optionally, sets the upper and lower threshold values used by vPC to determine when to fail over to the vPC trunk. The lower-value range is from 1 to 255. The default is 1. The upper-value range is from 1 to 255. The default is 255.
Step 5	switch(config-if-hsrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example configures the HSRP prioritylevel to 60 and the upper and lower threshold values used by vPC:

```
switch#       configure terminal
switch(config)# interface  ethernet  1/2
switch(config-if)# hsrp 2

switch(config-if-hsrp)#  priority  60  forwarding-threshold  lower  40
upper 50

switch(config-if-hsrp)# copy running-config startup-config
```

20.7.12 Customizing HSRP in HSRP Configuration Mode

You can optionally customize the behavior of HSRP. Be aware that as soon as you enable an HSRP group by configuring a virtual IP address, that group is now operational. If you first enable an HSRP group before customizing HSRP, the router could take control over the group and become the active router before you finish customizing the feature. If you plan to customize HSRP, you should do so before you enable the HSRP group. To customize HSRP, use the following commands in HSRP configuration mode.

SUMMARY STEPS

1. switch(config-if-hsrp)# **name** *string*
2. switch(config-if-hsrp)# **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]
3. switch(config-if-hsrp)# **timers** [**msec**] *hellotime* [**msec**] *holdtime*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch(config-if-hsrp)# **name** *string*	Specifies the IP redundancy name for an HSRP group. The string is from 1 to 255 characters. The default string has the following format: hsrp-interface short-name group-id. For example, hsrp-Eth2/1-1.
Step 2	switch(config-if-hsrp)# **preempt** [**delay** [**minimum** *seconds*] [**reload** *seconds*] [**sync** *seconds*]]	Configures the router to take over as an active router for an HSRP group if it has a higher priority than the current active router. This command is disabled by default. The range is from 0 to 3600 seconds.
Step 3	switch(config-if-hsrp)# **timers** [**msec**] *hellotime* [**msec**] *holdtime*	Configures the hello and hold time for this HSRP member as follows: • hellotime—The interval between successive hello packets sent. The range is from 1 to 254 seconds. • holdtime—The interval before the information in the hello packet is considered invalid. The range is from 3 to 255. The optional msec keyword specifies that the argument is expressed in milliseconds instead of the default seconds. The timer ranges for milliseconds are as follows: • hellotime—The interval between successive hello packets sent. The range is from 255 to 999 milliseconds. • holdtime—The interval before the information in the hello packet is considered invalid. The range is from 750 to 3000 milliseconds.

Example

The following example shows how to customize HSRP in HSRP configuration mode:

```
switch(config-if-hsrp)#  name  HSRP-1
switch(config-if-hsrp)#         preempt   delay    minimum    60
switch(config-if-hsrp)# timers 5 18
```

20.7.13 Customizing HSRP in Interface Configuration Mode

You can optionally customize the behavior of HSRP. Be aware that as soon as you enable an HSRP group by configuring a virtual IP address, that group is now operational. If you first enable an HSRP group before customizing HSRP, the router could take control over the group and become the active router before you finish customizing the feature. If you plan to customize HSRP, you should do so before you enable the HSRP group. To customize HSRP, use the following commands in interface configuration mode.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot*/*port*
3. switch(config-if)# **hsrp delay minimum** *seconds*
4. switch(config-if)# **hsrp delay reload** *seconds*

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot*/*port*	Enters interface configuration mode.
Step 3	switch(config-if)# **hsrp delay minimum** *seconds*	Specifies the minimum amount of time that HSRP waits after a group is enabled before participating in the group. The range is from 0 to 10000 seconds. The default is 0.
Step 4	switch(config-if)# **hsrp delay reload** *seconds*	Specifies the minimum amount of time that HSRP waits after reload before participating in the group. The range is from 0 to 10000 seconds. The default is 0.

Example

The following example shows how to customize HSRP in interface configuration mode:

```
switch # configure terminal
switch(config)#  interface ethernet 1/2
switch(config-if)#   hsrp delay minimum 30
switch(config-if)#   hsrp delay reload 30
```

20.7.14 Configuring Extended Hold Timers for HSRP

You can configure HSRP to use extended hold timers to support extended NSF during a controlled (graceful) switchover or ISSU, including software upgrades and supervisor switchovers. You should configure extended hold timers on all HSRP routers.

You must configure extended hold timers on all HSRP routers if you configure extended hold timers. If you configure a nondefault hold timer, you should configure the same value on all HSRP routers when you configure HSRP extended hold timers.

HSRP extended hold timers are not applied if you configure millisecond hello and hold timers for HSRPv1. This statement does not apply to HSRPv2.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **hsrp timers extended-hold** [*timer*]
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **hsrp timers extended-hold** [*timer*]	Sets the HSRP extended hold timer, in seconds, for both IPv4 and IPv6 groups. The timer range is from 10 to 255. The default is 10. **Note** Use the **show hsrp** command or the **show running-config hsrp** command to display the extended hold time.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure extended hold timers for HSRP:

```
switch # configure terminal

switch(config)# hsrp timers extended-hold

switch(config)#  copy  running-config  startup-config
```

20.8 Verifying the HSRP Configuration

Use one of the following commands to verify the configuration:

Command	**Purpose**
show hsrp [**group** *group-number*]	Displays the HSRP status for all groups or one group.
show hsrp delay [**interface** *interface-type slot/port*]	Displays the HSRP delay value for all interfaces or one interface.
show hsrp [**interface** *interface-type slot/port*]	Displays the HSRP status for an interface.
show hsrp [**group** *group-number*] [**interface** *interface-type slot/port*] [**active**] [**all**] [**init**] [**learn**] [**listen**] [**speak**] [**standby**]	Displays the HSRP status for a group or interface for virtual forwarders in the active, init, learn, listen, or standby state. Use the all keyword to see all states, including disabled.
show hsrp [**group** *group-number*] [**interface** *interface-type slot/port*] **active**] [**all**] [**init**] [**learn**] [**listen**] [**speak**] [**standby**] **brief**	Displays a brief summary of the HSRP status for a group or interface for virtual forwarders in the active, init, learn, listen, or standby state. Use the all keyword to see all states, including disabled.
show hsrp mgo [**name***name*] **[brief]**	Displays the relationships between HSRP groups that are in use for MGO and their slave sessions.

20.9 Configuration Examples for HSRP

This example shows how to enable HSRP on an interface with MD5 authentication and interface tracking:

```
key chain hsrp-keys key 0

   key-string 7 zqdest

   accept-lifetime 00:00:00 Jun 01 2008 23:59:59 Sep 12 2008

   send-lifetime  00:00:00  Jun  01  2008  23:59:59
   Aug 12 2008

  key 1
```

```
 key-string 7 uaeqdyito

 accept-lifetime 00:00:00 Aug 12 2008 23:59:59 Dec 12 2008

 send-lifetime 00:00:00 Sep 12 2008 23:59:59
Nov 12 2008 feature hsrp

track   2   interface
ethernet    2/2    ip
interface    ethernet
1/2

 ip address 192.0.2.2/8 hsrp 1

  authenticate  md5  key-
  chain hsrp-keys priority
  90

  track 2 decrement 20

  ip 192.0.2.10

no shutdown
```

This example shows how to configure the HSRP priority on an interface:

```
interface vlan 1

hsrp 0

    preempt

    priority 100 forwarding-threshold lower 80 upper 90

    ip 192.0.2.2

    track 1 decrement 30
```

20.10 Related Documents for HSRP

Related Topic	Document Title
Configuring the Gateway Load Balancing protocol	Configuring GLBP
Configuring the Virtual Router Redundancy protocol	Configuring VRRP
HSRP CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
Configuring high availability	*Inspur CN12700 Series INOS High Availability and Redundancy Guide*

20.11 MIBs

Related Topic	Document Title
INSPUR-HSRP-MIB	-

20.12 Feature History for HSRP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
MGO	8.4(1)	This feature was introduced.

FabricPath anycast HSRP	8.4(1)	This feature was introduced.
BFD	8.4(1)	Added support for BFD. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide* for more information.
IPv6	8.4(1)	Added support to IPv6.
Object track lists	8.4(1)	Added support for object track lists.
Extended hold timers	8.4(1)	Added support for extended hold timers for extended NSF support.
INSPUR-HSRP-MIB	8.4(1)	Added support for INSPUR-HSRP-MIB
Priority thresholds	8.4(1)	Added support for vPC threshold values on HSRP priority.
HSRP	8.4(1)	This feature was introduced.

CHAPTER 21 Configuring VRRP

This chapter contains the following sections:

- Finding Feature Information
- Information About VRRP
- Licensing Requirements for VRRP
- Guidelines and Limitations for VRRP
- Default Settings for VRRP Parameters
- Configuring VRRP
- Verifying the VRRP Configuration
- Monitoring VRRP Statistics
- Configuration Example for VRRP
- Related Documents for VRRP
- Feature History for VRRP

21.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

21.2 Information About VRRP

VRRP allows for transparent failover at the first-hop IP router by configuring a group of routers to share a virtual IP address. VRRP selects a master router in that group to handle all packets for the virtual IP address. The remaining routers are in standby and take over if the master router fails.

21.2.1 VRRP Operation

A LAN client can determine which router should be the first hop to a particular remote destination by using a dynamic process or static configuration. Examples of dynamic router discovery are as follows:

Proxy ARP—The client uses Address Resolution Protocol (ARP) to get the destination it wants to reach, and a router responds to the ARP request with its own MAC address.

Routing protocol—The client listens to dynamic routing protocol updates (for example, from Routing Information Protocol [RIP]) and forms its own routing table.

ICMP Router Discovery Protocol (IRDP) client—The client runs an Internet Control Message Protocol (ICMP) router discovery client.

The disadvantage to dynamic discovery protocols is that they incur some configuration and processing overhead on the LAN client. Also, if a router fails, the process of switching to another router can be slow.

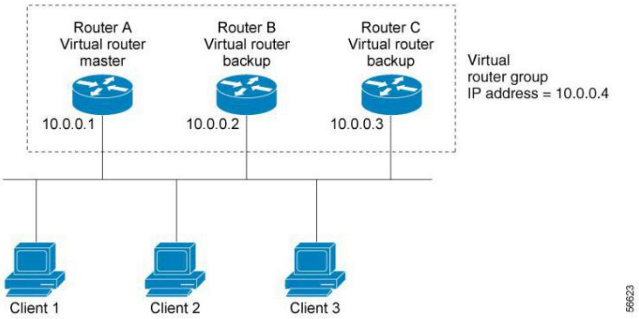
An alternative to dynamic discovery protocols is to statically configure a default router on the client. Although, this approach simplifies client configuration and processing, it creates a single point of failure. If the default gateway fails, the LAN client is limited to communicating only on the local IP network segment and is cut off from the rest of the network.

VRRP can solve the static configuration problem by enabling a group of routers (a VRRP group) to share a single virtual IP address.

You can then configure the LAN clients with the virtual IP address as their default gateway.

Figure 49 : Basic VRRP Topology

This image shows a basic VLAN topology where routers A, B, and C form a VRRP group. The IP address of the VRRP group must be different to the address that was configured for the Ethernet interface of Router A (10.0.0.1).



Because the virtual IP address uses the IP address of the physical Ethernet interface of Router A, Router A is the master (also known as the IP address owner). As the master, Router A owns the virtual IP address of the VRRP group and forwards packets sent to this IP address. Clients 1 through 3 are configured with the default gateway IP address of 10.0.0.1.

Routers B and C function as backups. If the master fails, the backup router with the highest priority becomes the master and takes over the virtual IP address to provide uninterrupted service for the LAN hosts. When router A recovers, it becomes the master again.

21.2.2 VRRP Benefits

The benefits of VRRP are as follows:

- Redundancy—Enables you to configure multiple routers as the default gateway router, which reduces the possibility of a single point of failure in a network.
- Load sharing—Allows traffic to and from LAN clients to be shared by multiple routers. The traffic load is shared more equitably among available routers.
- Multiple VRRP groups—Supports up to 255 VRRP groups on a router physical interface if the platform supports multiple MAC addresses. Multiple VRRP groups enable you to implement redundancy and load sharing in your LAN topology.
- Multiple IP addresses—Allows you to manage multiple IP addresses, including secondary IP addresses. If you have multiple subnets configured on an Ethernet interface, you can configure VRRP on each subnet.
- Preemption—Enables you to preempt a backup router that has taken over for a failing master with a higher priority backup router that has become available.
- Advertisement protocol—Uses a dedicated Internet Assigned Numbers Authority (IANA) standard multicast address (224.0.0.18) for VRRP advertisements. This addressing scheme minimizes the number of routers that must service the multicasts and allows test equipment to accurately identify VRRP packets on a segment. IANA has assigned the IP protocol number 112 to VRRP.
- VRRP tracking—Ensures that the best VRRP router is the master for the group by altering VRRP priorities based on interface states.

- The benefits of VRRPv3 are as follows:
  - Interoperability in multi-vendor environments.
  - Support for the IPv4 and IPv6 address families.
  - Improved scalability through the use of VRRS pathways.

21.2.3 Multiple VRRP Groups

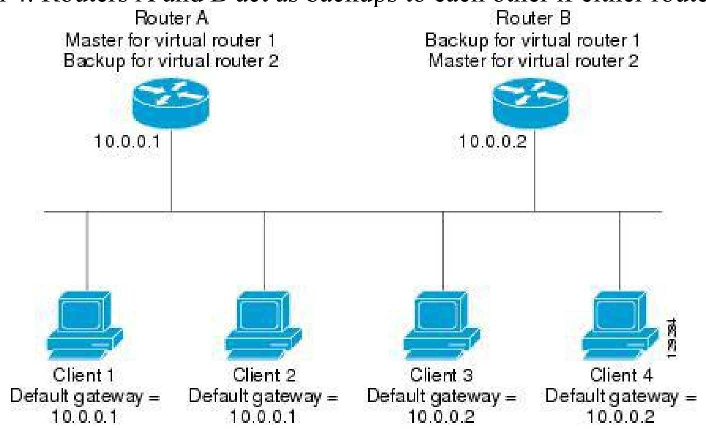
You can configure up to 255 VRRP groups on a physical interface. The number of VRRP groups that a router interface can support depends on the following factors:

- Router processing capability
- Router memory capability

In a topology where multiple VRRP groups are configured on a router interface, the interface can act as a master for one VRRP group and as a backup for one or more other VRRP groups.

Figure 50 : Load Sharing and Redundancy VRRP Topology

This image shows a LAN topology in which VRRP is configured so that Routers A and B share the traffic to and from clients 1 through 4. Routers A and B act as backups to each other if either router fails.



This topology contains two virtual IP addresses for two VRRP groups that overlap. For VRRP group 1, Router A is the owner of IP address 10.0.0.1 and is the master. Router B is the backup to Router A. Clients 1 and 2 are configured with the default gateway IP address of 10.0.0.1.

For VRRP group 2, Router B is the owner of IP address 10.0.0.2 and is the master. Router A is the backup to router B. Clients 3 and 4 are configured with the default gateway IP address of 10.0.0.2.

21.2.4 VRRP Router Priority and Preemption

An important aspect of the VRRP redundancy scheme is the VRRP router priority because the priority determines the role that each VRRP router plays and what happens if the master router fails.

If a VRRP router owns the virtual IP address and the IP address of the physical interface, this router functions as the master. The priority of the master is 255.

Priority also determines if a VRRP router functions as a backup router and the order of ascendancy to becoming a master if the master fails.

For example, if Router A, the master in a LAN topology, fails, VRRP must determine if backups B or C should take over. If you configure Router B with priority 101 and Router C with the default priority of 100, VRRP selects Router B to become the master because it has the higher priority. If you configure routers B and C with the default priority of 100, VRRP selects the backup with the higher IP address to become the master.

VRRP uses preemption to determine what happens after a VRRP backup router becomes the master. With preemption enabled by default, VRRP switches to a backup if that backup comes online with a priority higher than the new master. For example, if Router A is the master and fails, VRRP selects Router B (next in order of priority). If Router C comes online with a higher priority than Router B, VRRP selects Router C as the new master, even though Router B has not failed.

If you disable preemption, VRRP switches only if the original master recovers or the new master fails.

21.2.5 vPC and VRRP

VRRP interoperates with virtual port channels (vPCs). vPCs allow links that are physically connected to two different Inspur CN12700 series devices to appear as a single port channel by a third device. See the Inspur CN12700 Series INOS Layer 2 Switching Configuration Guide, for more information on vPCs.

vPC forwards traffic through both the master VRRP router as well as the backup VRRP router.

21.2.6 VRRP Advertisements

The VRRP master sends VRRP advertisements to other VRRP routers in the same group. The advertisements communicate the priority and state of the master. Inspur INOS encapsulates the VRRP advertisements in IP packets and sends them to the IP multicast address assigned to the VRRP group. Inspur INOS sends the advertisements once every second by default, but you can configure a different advertisement interval.

21.2.7 VRRP Authentication

VRRP supports the following authentication functions:

- No authentication
- Plain text authentication

VRRP rejects packets in any of the following cases:

- The authentication schemes differ on the router and in the incoming packet.
- Text authentication strings differ on the router and in the incoming packet.

21.2.8 VRRP Tracking

VRRP supports the following two options for tracking:

- Native interface tracking—Tracks the state of an interface and uses that state to determine the priority of the VRRP router in a VRRP group. The tracked state is down if the interface is down or if the interface does not have a primary IP address.
- Object tracking—Tracks the state of a configured object and uses that state to determine the priority of the VRRP router in a VRRP group.

If the tracked state (interface or object) goes down, VRRP updates the priority based on what you configure the new priority to be for the tracked state. When the tracked state comes up, VRRP restores the original priority for the virtual router group.

For example, you may want to lower the priority of a VRRP group member if its uplink to the network goes down so another group member can take over as master for the VRRP group.

21.2.9 VRRPv3 and VRRS

VRRP version 3 (VRRPv3) enables a group of switches to form a single virtual switch in order to provide redundancy and reduce the possibility of a single point of failure in a network. The LAN clients can then be configured with the virtual switch as their default gateway. The virtual switch, representing a group of switches, is also known as a VRRPv3 group.

Virtual router redundancy service (VRRS) improves the scalability of VRRPv3 by providing a stateless redundancy service to VRRS pathways and VRRS clients by monitoring VRRPv3. VRRPv3 acts as a VRRS server that pushes VRRPv3 status information (such as current and previous redundancy states, active and inactive Layer 2 and Layer 3 addresses, and so on) to VRRS pathways and all registered VRRS clients.

VRRS clients are other Inspur processes or applications that use VRRPv3 to provide or withhold a service or resource dependent upon the state of the group. VRRS pathways are special VRRS clients that use the VRRS database information to provide scaled first-hop gateway redundancy across scaled interface environments.

VRRS by itself is limited to maintaining its own state. Linking a VRRS client to a VRRPv3 group provides a mechanism that allows VRRS to provide a service to client applications so that they can implement stateless or stateful failovers. A stateful failover requires communication with a nominated backup before the failure so that operational data is not lost when the failover occurs.

VRRS pathways operate in a similar way to clients but are integrated with the VRRS architecture. They provide a means to scale first-hop gateway redundancy by allowing you to configure a virtual address across hundreds of interfaces. The virtual gateway state of a VRRS pathway follows the state of a First-Hop Redundancy Protocol (FHRP) VRRS server.

VRRPv3 notifies VRRS of its current state (master, backup, or nonoperational initial state [INIT]) and passes that infromation to pathways or clients. The VRRPv3 group name activates VRRS and associates the VRRPv3 group with any clients or pathways that are configured as part of VRRS with the same name.

Pathways and clients act on the VRRPv3 server state. When a VRRPv3 group changes states, VRRS pathways and clients alter their behavior (performing tasks such as shutting down interfaces or appending accounting logs) depending on the state received from VRRS.

21.2.10 BFD for VRRP

This feature supports bidirectional forwarding detection (BFD). BFD is a detection protocol that provides fast-forwarding and path-failure detection times. BFD provides subsecond failure detection between two adjacent devices and can be less CPU-intensive than protocol hello messages because some of the BFD load can be distributed onto the data plane on supported modules. See the *Inspur CN12700 Series INOS Interfaces Configuration Guide*, for more information.

21.2.11 High Availability

VRRP supports high availability through stateful restarts and stateful switchovers. A stateful restart occurs when the VRRP process fails and is restarted. Stateful switchover occurs when the active supervisor switches to the standby supervisor. Inspur INOS applies the run-time configuration after the switchover.

VRRPv3 does not support stateful switchovers.

21.2.12 Virtualization Support

VRRP supports virtual routing and forwarding (VRF) instances. VRF exists within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. If you change the VRF membership of an interface, Inspur INOS removes all Layer 3 configurations, including VRRP.

21.3 Licensing Requirements for VRRP

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the Inspur INOS Licensing Guide.

21.4 Guidelines and Limitations for VRRP

- You cannot configure VRRP on the management interface.
- When VRRP is enabled, you should replicate the VRRP configuration across devices in your network.
- We recommend that you do not configure more than one first-hop redundancy protocol on the same interface.
- You must configure an IP address for the interface where you configure VRRP and enable that interface before VRRP becomes active.
- Inspur INOS removes all Layer 3 configurations on an interface when you change the interface VRF membership, port channel membership, or when you change the port mode to Layer 2.
- When you configure VRRP to track a Layer 2 interface, you must shut down the Layer 2 interface and reenable the interface to update the VRRP priority to reflect the state of the Layer 2 interface
- BFD for VRRP can only be configured between two routers.
- The VRRP IP address must be different than any physical IP address of the devices participating in the VRRP, otherwise the ARP or MAC entries will be corrupted and may cause forwarding problems.
- VRRPv3 has the following configuration guidelines and limitations:
    - VRRPv3 is not intended as a replacement for existing dynamic protocols. VRRPv3 is designed for use over multi-access, multicast, or broadcast-capable Ethernet LANs.
    - VRRPv3 is supported only on Ethernet and Fast Ethernet interfaces, bridge group virtual interfaces (BVIs), and Gigabit Ethernet interfaces as well as on Multiprotocol Label Switching (MPLS) virtual private networks (VPNs), VRF-aware MPLS VPNs, and VLANs.

- When VRRPv3 is in use, VRRPv2 is unavailable. To configure VRRPv3, you must disable any VRRPv2 configuration.
- VRRS is currently available only for use with VRRPv3.
- Use VRRPv3 millisecond timers only where absolutely necessary and with careful consideration and testing. Millisecond values work only under favorable circumstances. The millisecond timer values are compatible with third-party vendors, as long as they also support VRRPv3.
- Full network redundancy can be achieved only if VRRPv3 operates over the same network path as the VRRS pathway redundant interfaces. For full redundancy, the following restrictions apply:
  - VRRS pathways should use the same physical interface as the parent VRRPv3 group or be configured on a subinterface with the same physical interface as the parent VRRPv3 group.
  - VRRS pathways can be configured on switch virtual interfaces (SVIs) only if the associated VLAN shares the same trunk as the VLAN on which the parent VRRPv3 group is configured.

21.5 Default Settings for VRRP Parameters

Default RIP Parameters

Parameters	**Default**
Advertisement interval	1 second
Authentication	No authentication
Preemption	Enabled
Priority	100
VRRP feature	Disabled
VRRPv3	Disabled
VRRS	Disabled
VRRPv3 secondary address matching	Enables
Priority of a VRRPv3 group	100
VRRPv3 advertisement timer	1000 milliseconds

21.6 Configuring VRRP

21.6.1 Enabling VRRP

You must globally enable the VRRP feature before you configure and enable any VRRP groups.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **[no] feature vrrp**
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **[no] feature vrrp**	Enables the VRRP feature. Use the **no** form of this command to disable this feature. Using the no form of this command will disable the feature in a VDC and remove all associated configurations.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example enables VRRP:

```
switch # configure terminal

switch(config)# feature vrrp

switch(config)#  copy  running-config  startup-config
```

21.6.2 Configuring VRRP Groups

You can create a VRRP group, assign the virtual IP address, and enable the group.

You can configure one virtual IPv4 address for a VRRP group. By default, the master VRRP router drops the packets addressed directly to the virtual IP address because the VRRP master is only intended as a next-hop router to forward packets. Some applications require that Inspur INOS accept packets addressed to the virtual router IP. Use the secondary option to the virtual IP address to accept these packets when the local router is the VRRP master.

Once you have configured the VRRP group, you must explicitly enable the group before it becomes active.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command. Ensure that you have configured an IP address on the interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **vrrp** *number*
4. switch(config-if-vrrp)# **address** *ip-address* [**secondary**]
5. switch(config-if-vrrp)# **no shutdown**
6. switch(config-if-vrrp)# **copy running-config startup-config**
7. (Optional) switch(config-if-vrrp)# **show vrrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrp** *number*	Creates a virtual router group. the range is from 1 to 255.

Step 4	switch(config-if-vrrp)# **address** *ip-address* [**secondary**]	Configures the virtual IPv4 address for the specified VRRP group. This address should be in the same subnet as the IPv4 address of the interface. Use the secondary option only if applications require that VRRP routers accept the packets sent to the virtual router's IP address and deliver to applications.
Step 5	switch(config-if-vrrp)# **no shutdown**	Enables the VRRP group. Disabled by default.
Step 6	switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 7	(Optional) switch(config-if-vrrp)# **show vrrp**	(Optional) Displays VRRP information.

Example
The following example enables VRRP:

```
switch # configure terminal

switch(config)# interface ethernet 2/1



switch(config-if)# vrrp 250

switch(config-if-vrrp)#  address 192.0.2.8

switch(config-if-vrrp)# no shutdown

switch(config-if-vrrp)# copy running-config startup-config

switch(config-if-vrrp)# show vrrp
```

21.6.3 Configuring VRRP Priority

The valid priority range for a virtual router is from 1 to 254 (1 is the lowest priority and 254 is the highest). The default priority value for backups is 100. For devices whose interface IP address is the same as the primary virtual IP address (the master), the default value is 255.

If you configure VRRP on a vPC-enabled interface, you can optionally configure the upper and lower threshold values to control when to fail over to the vPC trunk. If the backup router priority falls below the lower threshold,

VRRP sends all backup router traffic across the vPC trunk to forward through the master VRRP router. VRRP maintains this scenario until the backup VRRP router priority increases above the upper threshold.

Before you begin

- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have configured an IP address on the interface.
- You must enable VRRP.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **vrrp** *number*
4. switch(config-if-vrrp)# **shutdown**
5. switch(config-if-vrrp)# **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]

6. switch(config-if-vrrp)# **no shutdown**
7. switch(config-if-vrrp)# **copy running-config startup-config**
8. (Optional) switch(config-if-vrrp)# **show vrrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrp** *number*	Creates a virtual router group. the range is from 1 to 255.
Step 4	switch(config-if-vrrp)# **shutdown**	Disables the VRRP group. Disabled by default.
Step 5	switch(config-if-vrrp)# **priority** *level* [**forwarding-threshold lower** *lower-value* **upper** *upper-value*]	Sets the priority level used to select the active router in a VRRP group. The level range is from 1 to 254. The default is 100 for backups and 255 for a master that has an interface IP address equal to the virtual IP address. Optionally, sets the upper and lower threshold values used by vPC to determine when to fail over to the vPC trunk. The lower-value range is from 1 to 255. The default is 1. The upper-value range is from 1 to 255. The default is 255.
Step 6	switch(config-if-vrrp)# **no shutdown**	Enables the VRRP group. Disabled by default.
Step 7	switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config-if-vrrp)# **show vrrp**	Displays VRRP information.

Example

The following example enables VRRP:

```
switch # configure terminal

switch(config)# interface ethernet 2/1



switch(config-if)# vrrp 250

switch(config-if)# shutdown

switch(config-if-vrrp)#   priority  60  forwarding-threshold  lower  40
upper 50

switch(config-if-vrrp)# no shutdown
switch(config)#  copy  running-config  startup-config
switch(config-if-vrrp)# show vrrp
```

21.6.4 Configuring VRRP Authentication

You can configure simple text authentication for a VRRP group.

Before you begin

- Ensure that the authentication configuration is identical for all VRRP devices in the network.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have configured an IP address on the interface. See Configuring IPv4, on page 19.
- You must enable VRRP.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **vrrp** *number*
4. switch(config-if-vrrp)# **shutdown**
5. switch(config-if-vrrp)# **authentication text** *password*
6. switch(config-if-vrrp)# **no shutdown**
7. switch(config-if-vrrp)# **copy running-config startup-config**
8. (Optional) switch(config-if-vrrp)# **show vrrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrp** *number*	Creates a virtual router group. The range is from 1 to 255.
Step 4	switch(config-if-vrrp)# **shutdown**	Disables the VRRP group. Disabled by default.
Step 5	switch(config-if-vrrp)# **authentication text** *password*	Assigns the simple text authentication option and specifies the keyname password. The keyname range is from 1 to 255 characters. We recommend that you use at least 16 characters. The text password is up to eight alphanumeric characters.
Step 6	switch(config-if-vrrp)# **no shutdown**	Enables the VRRP group. Disabled by default.
Step 7	switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config-if-vrrp)# **show vrrp**	Displays VRRP information.

Example

The following example enables VRRP:

```
switch # configure terminal

switch(config)# interface ethernet 2/1


switch(config-if)# vrrp 250

switch(config-if)# shutdown

switch(config-if-vrrp)#   authentication   text aPassword

switch(config-if-vrrp)#   no   shutdown
switch(config)#   copy   running-config
```

```
startup-config  switch(config-if-vrrp)# show vrrp
```

21.6.5 Configuring Time Intervals for Advertisement Packets

You can configure the time intervals for advertisement packets.

Before you begin

- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that you have configured an IP address on the interface.
- You must enable VRRP.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **vrrp** *number*
4. switch(config-if-vrrp)# **shutdown**
5. switch(config-if-vrrp)# **advertisement interval** *seconds*
6. switch(config-if-vrrp)# **no shutdown**
7. switch(config-if-vrrp)# **copy running-config startup-config**
8. (Optional) switch(config-if-vrrp)# **show vrrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrp** *number*	Creates a virtual router group. The range is from 1 to 255.
Step 4	switch(config-if-vrrp)# **shutdown**	Disables the VRRP group. Disabled by default.
Step 5	switch(config-if-vrrp)# **advertisement interval** *seconds*	Sets the interval time in seconds between sending advertisement frames. The range is from 1 to 255. The default is 1 second.
Step 6	switch(config-if-vrrp)# **no shutdown**	Enables the VRRP group. Disabled by default.
Step 7	switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config-if-vrrp)# **show vrrp**	Displays VRRP information.

Example

The following example enables VRRP:

```
switch # configure terminal

switch(config)# interface ethernet 2/1

switch(config-if)# vrrp 250
switch(config-if)# shutdown

switch(config-if-vrrp)# advertisement-interval 15
switch(config-if-vrrp)# no shutdown
switch(config)# copy running-config startup-config
switch(config-if-vrrp)# show vrrp
```

21.6.6 Disabling Preemption

You can disable preemption for a VRRP group member. If you disable preemption, a higher-priority backup router does not take over for a lower-priority master router. Preemption is enabled by default.

Before you begin

- You must enable VRRP.
- Ensure that you have configured an IP address on the interface.
- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **vrrp** *number*
4. switch(config-if-vrrp)# **shutdown**
5. switch(config-if-vrrp)# **no preempt**
6. switch(config-if-vrrp)# **no shutdown**
7. switch(config-if-vrrp)# **copy running-config startup-config**
8. (Optional) switch(config-if-vrrp)# **show vrrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrp** *number*	Creates a virtual router group. The range is from 1 to 255.
Step 4	switch(config-if-vrrp)# **shutdown**	Disables the VRRP group. Disabled by default.
Step 5	switch(config-if-vrrp)# **no preempt**	Disables the preempt option and allows the master to remain when a higher-priority backup appears.
Step 6	switch(config-if-vrrp)# **no shutdown**	Enables the VRRP group. Disabled by default.
Step 7	switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config-if-vrrp)# **show vrrp**	Displays VRRP information.

Example

The following example enables VRRP:

```
switch # configure terminal

switch(config)# interface ethernet 2/1
switch(config-if)#         vrrp        250
switch(config-if)# shutdown
switch(config-if-vrrp)#      no      preempt
switch(config-if-vrrp)# no shutdown
switch(config)#  copy  running-config  startup-config

switch(config-if-vrrp)# show vrrp
```

21.6.7 Configuring VRRP Interface State Tracking

Interface state tracking changes the priority of the virtual router based on the state of another interface in the device. When the tracked interface goes down or the IP address is removed, Inspur INOS assigns the tracking priority value to the virtual router. When the tracked interface comes up and an IP address is configured on this interface, Inspur INOS restores the configured priority to the virtual router.

Before you begin

- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- You must enable VRRP.
- Ensure that you have enabled the virtual router.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *interface-type slot/port*
3. switch(config-if)# **vrrp** *number*
4. switch(config-if-vrrp)# **shutdown**
5. switch(config-if-vrrp)# **track interface** *type number* **priority** *value*
6. switch(config-if-vrrp)# **no shutdown**
7. switch(config-if-vrrp)# **copy running-config startup-config**
8. (Optional) switch(config-if-vrrp)# **show vrrp**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *interface-type slot/port*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrp** *number*	Creates a virtual router group. The range is from 1 to 255.
Step 4	switch(config-if-vrrp)# **shutdown**	Disables the VRRP group. Disabled by default.
Step 5	switch(config-if-vrrp)# **track interface** *type number* **priority** *value*	Enables interface priority tracking for a VRRP group. The priority range is from 1 to 254.
Step 6	switch(config-if-vrrp)# **no shutdown**	Enables the VRRP group. Disabled by default.
Step 7	switch(config-if-vrrp)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 8	(Optional) switch(config-if-vrrp)# **show vrrp**	Displays VRRP information.

Example

The following example enables VRRP:

```
switch # configure terminal

switch(config)# interface ethernet 2/1

switch(config-if)# vrrp 250

switch(config-if)# shutdown

switch(config-if-vrrp)# track interface ethernet 2/10 priority 254

switch(config-if-vrrp)#  no  shutdown
switch(config)#  copy  running-config startup-config
switch(config-if-vrrp)# show vrrp
```

21.6.8 Enabling the VRRPv3 Feature

You must globally enable the VRRPv3 feature before you can configure and enable any VRRPv3 groups.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **feature vrrpv3**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **feature vrrpv3**	Enables VRRP version 3 and Virtual Router Redundancy Service (VRRS). The no form of this command disables VRRPv3 and VRRS in a VDC. If VRRPv2 is currently configured, use the **no feature vrrp** command in global configuration mode to remove the VRRPv2 configuration and then use the **feature vrrpv3** command to enable VRRPv3.

Example

The following example shows how to enable VRRPv3:

```
switch# configure terminal

switch(config)# enable vrrpv3
```

21.6.9 Creating VRRPv3 Groups

You can create a VRRPv3 group, assign the virtual IP address, and enable the group.

Before you begin

- Ensure that the VRRPv3 feature is enabled.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).
- Ensure that you configure an IP address on the interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type/number*
3. switch(config-if)# **vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
4. switch(config-if-vrrpv3-group)# **address** *ip-address* [**primary** | **secondary**]
5. switch(config-if-vrrpv3-group)# **description** *description*
6. switch(config-if-vrrpv3-group)# **match-address**
7. switch(config-if-vrrpv3-group)# **preempt** [**delay minimum** *seconds*]
8. switch(config-if-vrrpv3-group)# **priority** *level*
9. switch(config-if-vrrpv3-group)# **timers advertise** *interval*
10. switch(config-if-vrrpv3-group)# **vrrp2**
11. switch(config-if-vrrpv3-group)# **vrrs leader** *vrrs-leader-name*
12. switch(config-if-vrrpv3-group)# **shutdown**
13. switch(config-if-vrrpv3-group)# **show fhrp** [*interface-type interface-number*] [**verbose**]
14. switch(config-if-vrrpv3-group)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type/number*	Enters interface configuration mode.
Step 3	switch(config-if)# **vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]	Creates a VRRPv3 group and enters VRRPv3 group configuration mode. The range is from 1 to 255.
Step 4	switch(config-if-vrrpv3-group)# **address** *ip-address* [**primary** | **secondary**]	(Optional) Specifies a primary or secondary IPv4 or IPv6 address for the VRRPv3 group. To utilize secondary IP addresses in a VRRPv3 group, you must first configure a primary IP address on the same group.
Step 5	switch(config-if-vrrpv3-group)# **description** *description*	(Optional) Specifies a description for the VRRPv3 group. You can enter up to 80 alphanumeric characters.
Step 6	switch(config-if-vrrpv3-group)# **match-address**	(Optional) Matches the secondary address in the advertisement packet against the configured address.
Step 7	switch(config-if-vrrpv3-group)# **preempt** [**delay minimum** *seconds*]	(Optional) Enables preemption of a lower priority master switch with an optional delay. The range is from 0 to 3600.
Step 8	switch(config-if-vrrpv3-group)# **priority** *level*	(Optional) Specifies the priority of the VRRPv3 group. The range is from 1 to 254.

Step 9	switch(config-if-vrrpv3-group)# **timers advertise***interval*	(Optional) Sets the advertisement timer in milliseconds. The range is from 100 to 40950. Inspur recommends that you set this timer to a value greater than or equal to 1 second.
Step 10	switch(config-if-vrrpv3-group)# **vrrp2**	(Optional) Enables support for VRRPv2 simultaneously, to ensure interoperability with devices that support only VRRPv2. VRRPv2 compatibility mode is provided to allow an upgrade from VRRPv2 to VRRPv3. This is not a full VRRPv2 implementation and should be used only to perform an upgrade.
Step 11	switch(config-if-vrrpv3-group)# **vrrs leader** *vrrs-leader-name*	(Optional) Specifies a leader's name to be registered with VRRS.
Step 12	switch(config-if-vrrpv3-group)# **shutdown**	(Optional) Disables VRRP configuration for the VRRPv3 group.
Step 13	switch(config-if-vrrpv3-group)# **show fhrp** [*interface-type interface-number*] [**verbose**]	(Optional) Displays First Hop Redundancy Protocol (FHRP) information. Use the **verbose** keyword to view detailed information.
Step 14	switch(config-if-vrrpv3-group)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

The following example shows how to create a VRRPv3 group:

```
switch# configure terminal

switch(config)#  interface  ethernet 1/2

switch(config-if)# vrrpv3 5 address-family ipv4

switch(config-if)# hsrp version 2

switch(config-if-vrrpv3-group)# address 100.0.1.10 primary

switch(config-if-vrrpv3-group)# description group3

switch(config-if-vrrpv3-group)# match-adress

switch(config-if-vrrpv3-group)# preempt delay minimum 30

switch(config-if-vrrpv3-group)# priority 3

switch(config-if-vrrpv3-group)# timers advertise 1000

switch(config-if-vrrpv3-group)# vrrp2

switch(config-if-vrrpv3-group)# vrrs leader leader1

switch(config-if-vrrpv3-group)# shutdown
switch(config-if-vrrpv3-group)#  show  fhrp  ethernet  1/2 verbose
switch(config-if-vrrpv3-group)# show running-config startup-config
```

21.6.10 Configuring the Delay Period for FHRP Client Initialization

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **fhrp delay** {[**minimum**] [**reload**] *seconds*}
3. switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **fhrp delay** {[**minimum**] [**reload**] *seconds*}	Specifies the delay period for the initialization of FHRP clients. The range is from 0 to 3600 seconds. The **minimum** keyword configures the delay period after an interface beomes available. The **reload** command configures the delay period after the device reloads.
Step 3	switch(config)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

The following example shows how to configure the delay period for initializing FHRP clients:

```
switch# configure terminal

switch(config)# fhrp delay minimum 14
```

21.6.11 Configuring VRRPv3 Control Groups

Before you begin

- Ensure that the VRRPv3 feature is enabled.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command). Ensure that you configure an IP address on the interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type/number*
3. switch(config-if)# **ip address** *ip address mask* [**secondary**]
4. switch(config-if)# **vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]
5. switch(config-if-vrrpv3-group)# **address** *ip-address* [**primary** | **secondary**]
6. switch(config-if-vrrpv3-group)# **vrrs leader** *vrrs-leader-name*
7. switch(config-if-vrrpv3-group)# **shutdown**
8. switch(config-if-vrrpv3-group)# **show fhrp** [*interfice-type interface-number*] [**verbose**]
9. switch(config-if-vrrpv3-group)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type/number*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip address** *ip address mask* [**secondary**]	Configures the IP address on the interface. You can use the **secondary** keyword to configure additional IP addresses on the interface.
Step 4	switch(config-if)# **vrrpv3** *number* **address-family** [**ipv4** | **ipv6**]	Creates a VRRPv3 group and enters VRRPv3 group configuration mode. The range is from 1 to 255.
Step 5	switch(config-if-vrrpv3-group)# **address** *ip-address* [**primary** | **secondary**]	(Optional) Specifies a primary or secondary IPv4 or IPv6 address for the VRRPv3 group.
Step 6	switch(config-if-vrrpv3-group)# **vrrs leader** *vrrs-leader-name*	(Optional) Specifies a leader's name to be registered with VRRS.
Step 7	switch(config-if-vrrpv3-group)# **shutdown**	(Optional) Disables VRRP configuration for the VRRPv3 group.
Step 8	switch(config-if-vrrpv3-group)# **show fhrp** [*interfice-type interface-number*] [**verbose**]	(Optional) Displays First Hop Redundancy Protocol (FHRP) information. Use the **verbose**keyword to view detailed information
Step 9	switch(config-if-vrrpv3-group)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

The following example shows how to configure a VRRPv3 control group:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)# ip address 209.165.200.230 255.255.255.224

switch(config-if)# vrrpv3 5 address-family ipv4

switch(cofig-if-vrrpv3-group)#          address  209.165.200.227  primary
switch(cofig-if-vrrpv3-group)#  vrrs  leader  leader1
switch(cofig-if-vrrpv3-group)# shutdown

switch(cofig-if-vrrpv3-group)#  show  fhrp  ethernet  1/2 verbose

 switch(cofig-if-vrrpv3-group)# show running-config startup-config
```

21.6.12 Configuring VRRS Pathways

You can configure a Virtual Router Redundancy Service (VRRS) pathway. In scaled environments, VRRS pathways should be used in combination with VRRPv3 control groups.

Before you begin

- Ensure that the VRRPv3 feature is enabled.
- Ensure that you are in the correct VDC (or use the **switchto vdc** command).
- Ensure that you configure an IP address on the interface.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **interface** *type/number*
3. switch(config-if)# **ip address** *ip-address mask* [**secondary**]
4. switch(config-if)# **vrrs pathway** *vrrs-tag*
5. switch(configif-vrrs-pw)# **mac address** {*mac-address* | **inherit**}
6. switch(configif-vrrs-pw)# **address** *ip-address*
7. switch(configif-vrrs-pw)# **show vrrs pathway** *interface-type interface-number*
8. switch(config-if-vrrs-pw)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **interface** *type/number*	Enters interface configuration mode.
Step 3	switch(config-if)# **ip address** *ip-address mask* [**secondary**]	Configures the IP address on the interface. You can use the **secondary** keyword to configure additional IP addresses on the interface.
Step 4	switch(config-if)# **vrrs pathway** *vrrs-tag*	Defines the VRRS pathway for a VRRS group and enters VRRS pathway configuration mode. The *vrrs-tag* argument specifies the name of the VRRS tag that is being associated with the pathway.
Step 5	switch(configif-vrrs-pw)# **mac address** {*mac-address* | **inherit**}	Specifies a MAC address for the pathway. The **inherit** keyword causes the pathway to inherit the virtual MAC address of the VRRPv3 group with which the pathway is associated.
Step 6	switch(configif-vrrs-pw)# **address** *ip-address*	Defines the virtual IPv4 or IPv6 address for a pathway. A VRRPv3 group is capable of controlling more than one pathway.
Step 7	switch(configif-vrrs-pw)# **show vrrs pathway** *interface-type interface-number*	(Optional) Displays the VRRS pathway information for different pathway states, such as active, inactive, and not ready.
Step 8	switch(config-if-vrrs-pw)# **copy running-config startup-config**	(Optional) Saves this configuration change.

Example

The following example shows how to configure VRRS pathways:

```
switch# configure terminal

switch(config)# interface ethernet 1/2

switch(config-if)# ip address 209.165.200.230 255.255.255.224

switch(config-if)# vrrs pathway path1

switch(config-if-vrrs-pw)#       mac       address  fe24.fe24.fe24
switch(config-if-vrrs-pw)# address 209.165.201.10
switch(config-if-vrrs-pw)#  show vrrs  pathway  ethernet  1/2
switch(config-if-vrrs-pw)# show running-config startup-config
```

21.7 Verifying the VRRP Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show vrrp	Displays the VRRP status for all groups.
show vrrp vr *group-number*	Displays the VRRP status for a VRRP group.
show vrrs client [*client-name*]	Displays the VRRS client information.
show vrrs pathway [*interface type/number*]	Displays the VRRS pathway information for different pathway states, such as active, inactive, and not ready.
show vrrs server	Displays the VRRS server information.
show vrrs tag [*tag-name*]	Displays the VRRS tag information.
show fhrp [*interface-type interface-name*] [**verbose**]	Displays First Hop Redundancy Protocol (FHRP) information.
show interface interface-type	Displays the virtual router configuration for an interface.

21.8 Monitoring VRRP Statistics

Use one of the following commands to display statistics about the feature:

Command	Purpose
show vrrp statistics	Displays the VRRP statistics.

- Use the **clear vrrp vr** command to clear the IPv4 VRRP statistics for the specified interface.
- Use the **clear vrrp statistics** command to clear all the VRRP statistics for all interfaces in the device.
- Use the **clear vrrp ipv4** command to clear all the statistics for the specified IPv4 virtual router.

21.8.1 Configuration Example for VRRP

In this example, Router A and Router B each belong to three VRRP groups. In the configuration, each group has the following properties:

- Group 1:

  - Virtual IP address is 10.1.0.10.
  - Router A will become the master for this group with priority 120.
  - Advertising interval is 3 seconds.
  - Preemption is enabled.
- Group 5:
  - Router B will become the master for this group with priority 200.
  - Advertising interval is 30 seconds.
  - Preemption is enabled.

Group 100:

  - Router A will become the master for this group first because it has a higher IP address (10.1.0.2).
  - Advertising interval is the default 1 second.
  - Preemption is disabled.

Router A

```
switch (config)# interface ethernet 1/0
switch (config-if)# ip address 10.1.0.2/16
switch (config-if)# no shutdown
switch (config-if)# vrrp 1
switch (config-if-vrrp)# priority 120
switch                (config-if-vrrp)#  authentication    text    Inspur
switch (config-if-vrrp)#        advertisement-interval  3
switch  (config-if-vrrp)# address 10.1.0.10
 switch     (config-if-vrrp)#    no        shutdown
 switch (config-if-vrrp)#    exit
 switch  (config-if)#  vrrp 5
 switch (config-if-vrrp)# priority 100
 switch (config-if-vrrp)# advertisement-interval 30
 switch       (config-if-vrrp)#address    10.1.0.50
 switch(config-if-vrrp)#          no          shutdown
 switch (config-if-vrrp)# exit
 switch (config-if)# vrrp 100
 switch (config-if-vrrp)# no preempt
 switch (config-if-vrrp)# address 10.1.0.100
 switch (config-if-vrrp)# no shutdown
```

Router B

```
switch (config)# interface ethernet 1/0
switch (config-if)# ip address 10.2.0.1/2
witch (config-if)# no shutdown
switch (config-if)# vrrp 1
switch (config-if-vrrp)# priority 100

switch              (config-if-vrrp)# authentication   text   Inspur
switch (config-if-vrrp)#        advertisement-interval  3
switch  (config-if-vrrp)# address 10.2.0.10

switch   (config-if-vrrp)#  no      shutdown
switch (config-if-vrrp)#    exit
switch  (config-if)#  vrrp 5

switch (config-if-vrrp)# priority 200

switch (config-if-vrrp)# advertisement-interval 10

switch        (config-if-vrrp)#address    10.2.0.50
switch
(config-if-vrrp)#             no             shutdown
switch (config-if-vrrp)# exit

switch (config-if)# vrrp 100

switch (config-if-vrrp)# no preempt

switch (config-if-vrrp)# address 10.2.0.100

switch (config-if-vrrp)# no shutdown
```

This example shows how to enable VRRPv3 and create and customize a VRRPv3 group:

```
switch#  configure  terminal
switch(config)#               feature            vrrp
switch(config)# interface ethernet 4/6
switch  (config-if)#  vrrpv3  5  address-family ipv4
switch (config-if-vrrp3-group)# address 209.165.200.225 primary
switch   (config-if-vrrp3-group)#   description group3
switch (config-if-vrrp3-group)# match-address
switch  (config-if-vrrp3-group)#  preempt  delay minimum 30
```

21.9 Related Documents for VRRP

Related Topic	**Document Title**
Configuring the gateway load balancing protocol	Configuring GLBP
Configuring the hot standby routing protocol	Configuring HSRP
VRRP CLI commands	*Inspur CN12700 Series Unicast Routing Command Reference*
Configuring high availability	*Inspur CN12700 Series INOS High Availability and Redundancy Guide*

21.10 Feature History for VRRP

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	**Releases**	**Feature Information**
VRRPv3 and VRRS	8.4(1)	These features were introduced.
BFD for VRRP	8.4(1)	Added support for BFD.
VRRP priority thresholds	8.4(1)	Added support for priority thresholds and vPC.
VRRP object tracking	8.4(1)	Added support for tracking multiple object types in VRRP.
VRRP	8.4(1)	This feature was introduced.

CHAPTER 22 Configuring Object Tracking

This chapter contains the following sections:

- Finding Feature Information
- Information About Object Tracking
- Licensing Requirements for Object Tracking
- Prerequisites for Object Tracking
- Guidelines and Limitations for Object Tracking
- Default Settings for Object Tracking Parameters
- Configuring Object Tracking
- Verifying the Object Tracking Configuration
- Configuration Example for Object Tracking
- Related Documents for Object Tracking
- Standards for Object Tracking
- Feature History for Object Tracking

22.1 Finding Feature Information

Your software release might not support all the features documented in this module. To find information about the features documented in this module, and to see a list of the releases in which each feature is supported, see the "New and Changed Information"chapter or the Feature History table in this chapter.

22.2 Information About Object Tracking

Object tracking allows you to track specific objects on the device, such as the interface line protocol state, IP routing, and route reachability, and to take action when the state of the tracked object changes. This feature allows you to increase the availability of the network and shorten recovery time if an object state goes down.

The object tracking feature allows you to create a tracked object that multiple clients can use to modify the client behavior when a tracked object changes. Several clients register their interest with the tracking process, track the same object, and take different actions when the object state changes.

Clients include the following features:

- Embedded Event Manager (EEM)
- Gateway Load Balancing Protocol (GLBP)
- Hot Standby Redundancy Protocol (HSRP)
- Virtual port channel (vPC)
- Virtual Router Redundancy Protocol (VRRP)

The object tracking monitors the status of the tracked objects and communicates any changes made to interested clients. Each tracked object is identified by a unique number that clients can use to configure the action to take when a tracked object changes state.

Inspur INOS tracks the following object types:

- Interface line protocol state—Tracks whether the line protocol state is up or down.
- Interface IP routing state—Tracks whether the interface has an IPv4 or IPv6 address and if IPv4 or IPv6 routing is enabled and active.
- IP route reachability—Tracks whether an IPv4 or IPv6 route exists and is reachable from the local device.

For example, you can configure HSRP to track the line protocol of the interface that connects one of the redundant routers to the rest of the network. If that link protocol goes down, you can modify the priority of the affected HSRP router and cause a switchover to a backup router that has better network connectivity.

22.2.1 Object Track List

An object track list allows you to track the combined states of multiple objects. Object track lists support the following capabilities:

- Boolean "and" function—Each object defined within the track list must be in an up state so that the track list object can become up.
- Boolean "or" function—At least one object defined within the track list must be in an up state so that the tracked object can become up.
- Threshold percentage—The percentage of up objects in the tracked list must be greater than the configured up threshold for the tracked list to be in the up state. If the percentage of down objects in the tracked list is above the configured track list down threshold, the tracked list is marked as down.
- Threshold weight—Assign a weight value to each object in the tracked list, and a weight threshold for the track list. If the combined weights of all up objects exceeds the track list weight up threshold, the track list is in an up state. If the combined weights of all the down objects exceeds the track list weight down threshold, the track list is in the down state.

Other entities, such as virtual Port Channels (vPCs) can use an object track list to modify the state of a vPC based on the state of the multiple peer links that create the vPC.

See the *Inspur CN12700 Series INOS Interfaces Configuration Guide*, for more information on vPCs.

22.2.2 High Availability

Object tracking supports high availability through stateful restarts. A stateful restart occurs when the object tracking process crashes. Object tracking also supports a stateful switchover on a dual supervisor system. Inspur INOS applies the runtime configuration after the switchover.

You can also use object tracking to modify the behavior of a client to improve overall network availability.

22.2.3 Virtualization Support

Object tracking supports Virtual Routing and Forwarding (VRF) instances. VRFs exist within virtual device contexts (VDCs). By default, Inspur INOS places you in the default VDC and default VRF unless you specifically configure another VDC and VRF. By default, Inspur INOS tracks the route reachability state of objects in the default VRF. If you want to track objects in another VRF, you must configure the object to be a member of that VRF.

22.3 Licensing Requirements for Object Tracking

This feature does not require a license. Any feature not included in a license package is bundled with the Inspur INOS system images and is provided at no extra charge to you. For a complete explanation of the Inspur INOS licensing scheme, see the Inspur INOS Licensing Guide.

22.4 Prerequisites for Object Tracking

If you configure VDCs, install the Advanced Series license and enter the desired VDC. See the *Inspur CN12700 Series INOS Virtual Device Context Configuration Guide*.

22.5 Guidelines and Limitations for Object Tracking

Object Tracking has the following configuration guidelines and limitations:

- Supports up to 500 tracked objects per VDC.
- Supports Ethernet, subinterfaces, tunnels, port channels, loopback interfaces, and VLAN interfaces.
- Supports one tracked object per HSRP group or GLBP group.
- If you are familiar with the Inspur IOS CLI, be aware that the Inspur INOS commands for this feature might differ from the Inspur IOS commands that you would use.

22.6 Default Settings for Object Tracking Parameters

Default Object Tracking Parameters

Parameters	**Default**
Tracked Object VRF	Member of default VRF

22.7 Configuring Object Tracking

22.7.1 Configuring Object Tracking for an Interface

You can configure Inspur INOS to track the line protocol or IPv4 or IPv6 routing state of an interface.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *object-id* **interface** *interface-type number* {{**ip** | **ipv6**} **routing** |**line-protocol**}
3. (Optional) switch(config-track)# **show track** [*object-id*]
4. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **track** *object-id* **interface** *interface-type number* {{**ip** | **ipv6**} **routing** |**line-protocol**}	Creates a tracked object for an interface and enters tracking configuration mode. The object-id range is from 1 to 500.
Step 3	(Optional) switch(config-track)# **show track** [*object-id*]	Displays object tracking information.
Step 4	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to configure object tracking for the line protocol state on Ethernet 1/2:

```
switch # configure terminal

switch(config)# track 1 interface ethernet 1/2 line-protocol

switch(config)#  copy  running-config  startup-config
```

The following example shows how to configure object tracking for the IPv4 routing state on Ethernet 1/2:

```
switch # configure terminal

switch(config)# track 2 interface ethernet 1/2 ip routing

switch(config)#  copy  running-config  startup-config
```

The following example shows how to configure object tracking for the IPv6 routing state on Ethernet 1/2:

```
switch # configure terminal

switch(config)# track 3 interface ethernet 1/2 ipv6 routing

switch(config)#  copy  running-config  startup-config
```

22.7.2 Deleting a Tracking Object

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **no track** *1*
3. (Optional) switch(config)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **no track** *1*	Deletes a tracked object for an interface. The object-id range is from 1 to 500.
Step 3	(Optional) switch(config)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.

Example

The following example shows how to delete an object tracking:

```
switch # configure terminal

switch(config)# no track 1

switch(config)#  copy  running-config  startup-config
```

22.7.3 Configuring Object Tracking for Route Reachability

You can configure Inspur INOS to track the existence and reachability of an IP route or IPv6 route.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**
3. (Optional) switch(config-track)# **show track** [*object-id*]
4. switch(config-track)# **copy running-config startup-config**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**	Creates a tracked object for a route and enters tracking configuration mode. The object-id range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32. The prefix format for IPv6 is A:B::C:D/length, where the length range is from 1 to 128. The prefix format for IPv6 is A:B::C:D/length, where the length range is from 1 to 128.
Step 3	(Optional) switch(config-track)# **show track** [*object-id*]	Displays object tracking information.
Step 4	switch(config-track)# **copy running-configstartup-config**	Saves the change persistently through reboots and restarts by copying and running configuration to the startup configuration.

Example

The following example shows how to configure object tracking for an IPv4 route in the default VRF:

```
switch # configure terminal

switch(config)# track 4 ip route 192.0.2.0/8 reachability

switch(config-track)# copy running-config startup-config
```

The following example shows how to configure object tracking for an IPv6 route in the default VRF:

```
switch # configure terminal

switch(config)# track 5 ipv6 route 10::10/128 reachability

switch(config-track)# copy running-config startup-config
```

22.7.4 Configuring an Object Track List with a Boolean Expression

You can configure an object track list that contains multiple tracked objects. A tracked list contains one or more objects. The Boolean expression enables two types of calculation by using either "and" or "or" operators. For example, when tracking two interfaces using the "and" operator, up means that both interfaces are up, and down means that either interface is down.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *track-number* **list boolean** {**and** | **or**}
3. switch(config-track)# **object** *object-number* [**not**]
4. (Optional) switch(config-track)# **copy running-config startup-config**
5. (Optional) switch(config-track)# **show track**

DETAILED STEPS

	Command or Action	**Purpose**
Step 1	switch# **configure terminal**	Enters global configuration mode.

Step 2	switch(config)# **track** *track-number* **list boolean** {**and** | **or**}	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a Boolean calculation. The keywords are as follows: • and—Specifies that the list is up if all objects are up or down if one or more objects are down. For example, when tracking two interfaces, up means that both interfaces are up, and down means that either interface is down. • or—Specifies that the list is up if at least one object is up. For example, when tracking two interfaces, up means that either interface is up, and down means that both interfaces are down. The track-number range is from 1 to 500.
Step 3	switch(config-track)# **object** *object-number* [**not**]	Adds a tracked object to the track list. The object-id range is from 1 to 500. The not keyword optionally negates the tracked object state. **Note** The example means that when object 10 is up, the tracked list detects object 10 as down.
Step 4	(Optional) switch(config-track)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	(Optional) switch(config-track)# **show track**	Displays object tracking information.

Example

The following example shows how to configure a track list with multiple objects as a Boolean ‟and″:

```
switch     #     configure terminal

switch(config)# track 1 list boolean and
switch(config-track)# object 10

switch(config-track)# object 20 not

switch(config)# copy running-config startup-config
```

22.7.5 Configuring an Object Track List with a Percentage Threshold

You can configure an object track list that contains a percentage threshold. A tracked list contains one or more objects. The percentage of up objects must exceed the configured track list up percent threshold before the track list is in an up state. For example, if the tracked list has three objects, and you configure an up threshold of 60 percent, two of the objects must be in the up state (66 percent of all objects) for the track list to be in the up state.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *track-number* **list threshold percentage**

3. switch(config-track)# **threshold percentage up** *up-value* **down** *down-value*
4. switch(config-track)# **object** *object-number*
5. (Optional) switch(config-track)# **copy running-config startup-config**
6. (Optional) switch(config-track)# **show track**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **track** *track-number* **list threshold percentage**	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold percent. The track-number range is from 1 to 500.
Step 3	switch(config-track)# **threshold percentage up** *up-value* **down** *down-value*	Configures the threshold percent for the tracked list. The range from 0 to 100 percent.
Step 4	switch(config-track)# **object** *object-number*	Adds a tracked object to the track list. The object-id range is from 1 to 500. **Note** The example means that when object 10 is up, the tracked list detects object 10 as down.
Step 5	(Optional) switch(config-track)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 6	(Optional) switch(config-track)# **show track**	Displays object tracking information.

Example

The following example shows how to configure a track list with multiple objects as a Boolean "and":

```
switch # configure terminal

switch(config)#    track    1    list    threshold   percentage
switch(config-track)#  threshold percentage  up  70  down  30
switch(config-track)# object 10

switch(config-track)# object 20

switch(config-track)# object 30

switch(config-track)#   copy   running-config startup-config
```

22.7.6 Configuring an Object Track List with a Weight Threshold

You can configure an object track list that contains a weight threshold. A tracked list contains one or more objects. The combined weight of up objects must exceed the configured track list up weight threshold before the track list is in an up state. For example, if the tracked list has three objects with the default weight of 10 each, and you configure an up threshold of 15, two of the objects must be in the up state (combined weight of 20) for the track list to be in the up state.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**

2. switch(config)# **track** *track-number* **list threshold weight**
3. switch(config-track)# **threshold weight up** *up-value* **down** *down-value*
4. switch(config-track)# **object** *object-id* **weight** *value*
5. switch(config-track)# **copy running-config startup-config**
6. (Optional) switch(config-track)# **show track**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **track** *track-number* **list threshold weight**	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold weight. The *track-number* range is from 1 to 500.
Step 3	switch(config-track)# **threshold weight up** *up-value* **down** *down-value*	Configures the threshold weight for the tracked list. The range is from 1 to 255.
Step 4	switch(config-track)# **object** *object-id* **weight** *value*	Adds a tracked object to the track list. The *object-id* range is from 1 to 500. The *value* range is from 1 to 255. The default weight value is 10
Step 5	switch(config-track)# **copy running-configstartup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 6	(Optional) switch(config-track)# **show track**	Displays object tracking information.

Example

The following example shows how to configure a track list with a up weight threshold of 30 and a down threshold of 10:

```
switch # configure terminal

switch(config)#track1 list threshold    weight

switch(config-track)#threshold  weight  up  30  down  10

switch(config-track)#object10weight 15

switch(config-track)# object 20 weight 15

switch(config-track)# object 30

switch(config-track)# copy running-config startup-config
```

22.7.7 Configuring an Object Tracking Delay

You can configure a delay for a tracked object or an object track list that delays when the object or list triggers a stage change. The tracked object or track list starts the delay timer when a state change occurs but does not recognize a state change until the delay timer expires. At that point, Inspur INOS checks the object state again and records a state change only if the object or list currently has a changed state. Object tracking ignores any intermediate state changes before the delay timer expires.

For example, for an interface line-protocol tracked object that is in the up state with a 20 second down delay, the delay timer starts when the line protocol goes down. The object is not in the down state unless the line protocol is down 20 seconds later.

You can configure independent up delay and down delay for a tracked object or track list. When you delete the delay, object tracking deletes both the up and down delay.

You can change the delay at any point. If the object or list is already counting down the delay timer from a triggered event, the new delay is computed as the following:

- If the new configuration value is less than the old configuration value, the timer starts with the new value.
- If the new configuration value is more than the old configuration value, the timer is calculated as the new configuration value minus the current timer countdown minus the old configuration value.

Before you begin

Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *object-id* {*parameters*}
3. switch(config-track)# **track** *track-number* **list** {*parameters*}
4. switch(config-track)# **delay** {**up** *up-time* [**down** *down-time*] | **down** *down-time* [**up** *up-time*]}
5. (Optional) switch(config-track)# **copy running-config startup-config**
6. (Optional) switch(config-track)# **show track**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **track** *object-id* {*parameters*}	Creates a tracked object for a route and enters tracking configuration mode. The object-id range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32. The prefix format for IPv6 is A:B::C:D/length, where the length range is from 1 to 128.
Step 3	switch(config-track)# **track** *track-number* **list** {*parameters*}	Configures a tracked list object and enters tracking configuration mode. Specifies that the state of the tracked list is based on a configured threshold weight. The *track-number* range is from 1 to 500.
Step 4	switch(config-track)# **delay** {**up** *up-time* [**down***down-time*] | **down** *down-time* [**up** *up-time*]}	Configures the object delay timers. The range is from 0 to 180 seconds. The *track-number* range is from 1 to 500.
Step 5	(Optional) switch(config-track)# **copy running-config startup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 6	(Optional) switch(config-track)# **show track**	Displays object tracking information.

Example

The following example shows how to configure object tracking for a route and use delay timers:

```
switch # configure terminal

switch(config)#   track   2   ip   route   209.165.201.0/8 reachability
switch(config-track)# delay up 20 down 30

switch(config-track)# copy running-config startup-config
```

The following example shows how to configure a track list with an up weight threshold of 30 and a down threshold of 10 with delay timers:

```
switch# configure terminal

switch(config)#track1 list threshold    weight

switch(config-track)#threshold  weight  up  30  down 10

switch(config-track)#object10weight 15

switch(config-track)# object 20 weight 15

switch(config-track)# object 30

switch(config-track)# delay up 20 down 30

switch(config-track)# copy running-config startup-config
```

The following example shows the delay timer in the show track command output before and after an interface is shut down:

```
switch(config-track)# show track

Track 1

  Interface
  loopback1      Line
  Protocol       Line
  Protocol is UP

  1  changes,  last
  change   00:00:13
  Delay   down   10
  secs

switch(config-track)# interface loopback 1

switch(config-
if)#   shutdown
switch(config-
if)#       show
track Track 1

  Interface loopback1 Line Protocol

  Line Protocol is delayed DOWN (8 secs remaining)<------- delay timer
  counting down

  1  changes,  last
  change   00:00:22
  Delay   down   10
  secs
```

22.7.8 Configuring Object Tracking for a Nondefault VRF

You can configure Inspur INOS to track an object in a specific VRF.

Before you begin

- Confirm that you are in the correct VDC. To change the VDC, use the **switchto vdc** command.
- Ensure that nondefault VRFs are created first.

SUMMARY STEPS

1. switch# **configure terminal**
2. switch(config)# **track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**
3. switch(config-track)# **vrf member** *vrf-name*
4. switch(config-track)# **copy running-config startup-config**
5. (Optional) switch(config-track)# **show track**

DETAILED STEPS

	Command or Action	Purpose
Step 1	switch# **configure terminal**	Enters global configuration mode.
Step 2	switch(config)# **track** *object-id* {**ip** | **ipv6**} **route** *prefix/length* **reachability**	Creates a tracked object for a route and enters tracking configuration mode. The *object-id* range is from 1 to 500. The prefix format for IP is A.B.C.D/length, where the length range is from 1 to 32. The prefix format for IPv6 is A:B::C:D/length, where the length range is from 1 to 128.
Step 3	switch(config-track)# **vrf member** *vrf-name*	Configures the VRF to use for tracking the configured object.
Step 4	switch(config-track)# **copy running-configstartup-config**	Saves the change persistently through reboots and restarts by copying the running configuration to the startup configuration.
Step 5	(Optional) switch(config-track)# **show track**	Displays object tracking information.

Example

The following example shows how to configure object tracking for a route and use VRF Red to look up reachability information for this object:

```
switch # configure terminal

switch(config)# track 2 ip route 209.165.201.0/8 reachability

switch(config-track)# vrf member Red

switch(config-track)#    copy    running-config startup-config
```

The following example shows how to configure object tracking for an IPv6 route and use VRF Red to look up reachability information for this object:

```
switch# configure terminal

switch(config)#  track  3  ipv6  route  1::2/64 reachability

switch(config-track)# vrf member Red

switch(config-track)#    copy    running-config startup-config
```

The following example how to modify tracked object 2 to use VRF Blue instead of VRF Red to look up reachability information for this object:

```
switch(config-track)# show track
switch(config)#  track  2
switch(config-track)#  vrf member Blue

switch(config-track)#    copy    running-config startup-config
```

22.8 Verifying the Object Tracking Configuration

Use one of the following commands to verify the configuration:

Command	Purpose
show track [*object-id*] [**brief**]	Displays the object tracking information for one or more objects.
show track [*object-id*] **interface** [**brief**]	Displays the interface-based object tracking information.
show track [*object-id*] {**ip** | **ipv6**} **route** [**brief**]	Displays the IPv4 or IPv6 route-based object tracking information.
show trun track	Displays the IP route IPv6 object tracking configuration information.

22.9 Configuration Example for Object Tracking

This example shows how to configure object tracking for route reachability and use VRF Red to look up reachability information for this route:

```
switch# configure terminal

switch(config)#   track   2   ip   route   209.165.201.0/8 reachability

switch(config-track)# vrf member Red
```

22.10 Related Documents for Object Tracking

Related Topic	Document Title
Object Tracking CLI commands	*Inspur CN12700 Series INOS Unicast Routing Command Reference*
Configuring the Embedded Event Manager	*Inspur CN12700 Series INOS System Management Configuration Guide*

22.11 Standards for Object Tracking

No new or modified standards are supported by this feature, and support for existing standards has not been modified by this feature.

22.12 Feature History for Object Tracking

This table includes only the updates for those releases that have resulted in additions or changes to the feature.

Feature Name	Releases	Feature Information
IPv6	8.4(1)	Added support for IPv6.
Tracking delay	8.4(1)	Added support for delaying a tracked object update.
Object track list	8.4(1)	Added support for object track lists and Boolean expressions.
Object tracking	8.4(1)	This feature was introduced.