



Inspur

CN12900 Series

INOS-CN Unicast Routing Configuration Guide



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: icnt_service@inspur.com

Technical Document Support Email: icnt_service@inspur.com

Address: 1036 Langchao Road, Lixia District, Jinan City, Shandong Province

Postal code: 250101

Notice

Copyright © 2020

Inspur Group.

All rights reserved.

No part of this publication may be excerpted, reproduced, translated or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm, without permission in Writing from **Inspur-Cisco Networking Technology Co.,Ltd.**

inspur 浪潮

is the trademark of **Inspur-Cisco Networking Technology Co.,Ltd.**

All other trademarks and trade names mentioned in this document are the property of their respective holders.

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 CN12900 Series. To have a quick grasp of the CN12900 Series, please read this manual carefully.

Versions





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

Product name	Version
CN12900 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 .

Convention	Description
	For example, log in as user root .
Italic	Book titles are in <i>italics</i> .
Lucida Console	Terminal display is in <code>Lucida Console</code> .

Command conventions

Convention	Description
Boldface	The keywords of a command line are in boldface .
Italic	Command arguments are in <i>italics</i> .
[]	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 .
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

Format	Description
	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

CHAPTER 1 Overview.....	1
1.1 About Layer 3 Unicast Routing.....	1
1.2 Routing Algorithms.....	6
1.3 Layer 3 Virtualization.....	7
1.4 Inspur INOS-CN Forwarding Architecture.....	7
1.5 Summary of Layer 3 Unicast Routing Features.....	9
1.6 Related Topics.....	11
CHAPTER 2 Configuring IPv4.....	12
2.1 About IPv4.....	12
2.2 Licensing Requirements for IPv4.....	16
2.3 Prerequisites for IPv4.....	16
2.4 Guidelines and Limitations for IPv4.....	16
2.5 Default Settings.....	16
2.6 Configuring IPv4.....	17
2.7 Verifying the IPv4 Configuration.....	27
2.8 Additional References.....	27
CHAPTER 3 Configuring IPv6.....	29
3.1 About IPv6.....	29
3.2 Licensing Requirements for IPv6.....	39
3.3 Prerequisites for IPv6.....	39
3.4 Guidelines and Limitations for IPv6.....	39
3.5 Configuring IPv6.....	39
3.6 Verifying the IPv6 Configuration.....	44
3.7 Configuration Examples for IPv6.....	44
CHAPTER 4 Configuring DNS.....	45
4.2 Licensing Requirements for DNS Clients.....	46
4.3 Prerequisites for DNS Clients.....	46
4.4 Guidelines and Limitations for DNS.....	46
4.5 Default Settings.....	46

4.6 Configuring DNS Clients.....	46
4.7 Verifying the DNS Client Configuration.....	50
4.8 Configuration Examples for the DNS Client.....	50
CHAPTER 5 Configuring OSPFv2.....	51
5.1 About OSPFv2.....	51
5.2 Licensing Requirements for OSPFv2.....	60
5.3 Prerequisites for OSPFv2.....	60
5.4 Guidelines and Limitations for OSPFv2.....	60
5.5 Default Settings.....	61
5.6 Configuring Basic OSPFv2.....	61
5.7 Configuring Advanced OSPFv2.....	70
5.8 Verifying the OSPFv2 Configuration.....	91
5.9 Monitoring OSPFv2.....	92
5.10 Configuration Examples for OSPFv2.....	92
CHAPTER 6 Configuring OSPFv3.....	94
6.1 About OSPFv3.....	94
6.2 Licensing Requirements for OSPFv3.....	103
6.3 Prerequisites for OSPFv3.....	103
6.4 Guidelines and Limitations for OSPFv3.....	103
6.5 Default Settings.....	104
6.6 Configuring Basic OSPFv3.....	104
6.7 Configuring Advanced OSPFv3.....	111
6.8 Verifying the OSPFv3 Configuration.....	131
6.9 Monitoring OSPFv3.....	132
6.10 Configuration Examples for OSPFv3.....	133
CHAPTER 7 Configuring EIGRP.....	134
7.1 About EIGRP.....	134
7.2 Licensing Requirements for EIGRP.....	139
7.3 Prerequisites for EIGRP.....	140
7.4 Guidelines and Limitations for EIGRP.....	140
7.5 Default Settings.....	140
7.6 Configuring Basic EIGRP.....	141

7.7 Configuring Advanced EIGRP.....	145
7.8 Configuring Virtualization for EIGRP.....	157
7.9 Verifying the EIGRP Configuration.....	159
7.10 Monitoring EIGRP.....	159
7.11 Configuration Examples for EIGRP.....	160
CHAPTER 8 Configuring IS-IS.....	161
8.1 About IS-IS.....	161
8.2 Licensing Requirements for IS-IS.....	164
8.3 Prerequisites for IS-IS.....	164
8.4 Guidelines and Limitations for IS-IS.....	164
8.5 Default Settings.....	164
8.6 Configuring IS-IS.....	165
8.7 Verifying the IS-IS Configuration.....	185
8.8 Monitoring IS-IS.....	186
8.9 Configuration Examples for IS-IS.....	187
CHAPTER 9 Configuring Basic BGP.....	188
9.1 About Basic BGP.....	188
9.2 Licensing Requirements for Basic BGP.....	194
9.3 Prerequisites for BGP.....	194
9.4 Guidelines and Limitations for BGP.....	195
9.5 Default Settings.....	195
9.6 CLI Configuration Modes.....	196
9.7 Configuring Basic BGP.....	198
9.8 Verifying the Basic BGP Configuration.....	209
9.9 Monitoring BGP Statistics.....	210
9.10 Configuration Examples for Basic BGP.....	211
CHAPTER 10 Configuring Advanced BGP.....	212
10.1 About Advanced BGP.....	212
10.2 Licensing Requirements for Advanced BGP.....	220
10.3 Prerequisites for Advanced BGP.....	220
10.4 Guidelines and Limitations for Advanced BGP.....	220
10.5 Default Settings for Advanced BGP.....	221

10.6 Configuring Advanced BGP.....	221
10.7 Configuring BMP.....	252
10.8 Configuring a Graceful Restart.....	254
10.9 Verifying the Advanced BGP Configuration.....	257
10.10 Monitoring BGP Statistics.....	259
10.11 Configuration Examples.....	259
CHAPTER 11 Configuring RIP.....	261
11.1 About RIP.....	261
11.2 Licensing Requirements for RIP.....	263
11.3 Prerequisites for RIP.....	263
11.4 Guidelines and Limitations.....	263
11.5 Default Settings.....	263
11.6 Configuring RIP.....	263
11.7 Verifying the RIP Configuration.....	275
11.8 Displaying RIP Statistics.....	275
11.9 Configuration Examples for RIP.....	275
CHAPTER 12 Configuring Static Routing.....	277
12.1 About Static Routing.....	277
12.2 Licensing Requirements for Static Routing.....	278
12.3 Prerequisites for Static Routing.....	278
12.4 Default Settings.....	278
12.5 Configuring Static Routing.....	279
12.6 Verifying the Static Routing Configuration.....	282
12.7 Configuration Example for Static Routing.....	283
CHAPTER 13 Configuring Layer 3 Virtualization.....	284
13.1 About Layer 3 Virtualization.....	284
13.2 Licensing Requirements for VRFs.....	287
13.3 Guidelines and Limitations for VRFs.....	287
13.4 Guidelines and Limitations for VRF Route Leaking.....	287
13.5 Default Settings.....	287
13.6 Configuring VRFs.....	287
13.7 Verifying the VRF Configuration.....	294

13.8 Configuration Examples for VRF.....	294
CHAPTER 14 Managing the Unicast RIB and FIB.....	300
14.1 About the Unicast RIB and FIB.....	300
14.2 Licensing Requirements for the Unicast RIB and FIB.....	301
14.3 Managing the Unicast RIB and FIB.....	301
14.4 Verifying the Unicast RIB and FIB.....	307
CHAPTER 15 Configuring Route Policy Manager.....	309
15.1 About Route Policy Manager.....	309
15.2 Licensing Requirements for Route Policy Manager.....	311
15.3 Guidelines and Limitations.....	312
15.4 Default Settings.....	312
15.5 Configuring Route Policy Manager.....	312
15.6 Verifying the Route Policy Manager Configuration.....	324
15.7 Configuration Examples for Route Policy Manager.....	324
CHAPTER 16 Configuring Policy-Based Routing.....	326
16.1 About Policy-Based Routing.....	326
16.2 Licensing Requirements for Policy-Based Routing.....	327
16.3 Prerequisites for Policy-Based Routing.....	327
16.4 Guidelines and Limitations.....	328
16.5 Default Settings.....	328
16.6 Configuring Policy-Based Routing.....	328
16.7 Verifying the Policy-Based Routing Configuration.....	331
16.8 Configuration Examples for Policy-Based Routing.....	331
CHAPTER 17 Configuring HSRP.....	332
17.1 Information About HSRP.....	332
17.2 Licensing Requirements for HSRP.....	337
17.3 Prerequisites for HSRP.....	337
17.4 Guidelines and Limitations for HSRP.....	337
17.5 Default Settings.....	338
17.6 Configuring HSRP.....	339
17.7 Verifying the HSRP Configuration.....	350
17.8 Configuration Examples for HSRP.....	350

CHAPTER 18 Configuring VRRP	353
18.1 Information About VRRP.....	353
18.2 Information About VRRPv3 and VRRS.....	357
18.3 High Availability.....	357
18.4 Virtualization Support.....	358
18.5 Licensing Requirements for VRRP.....	358
18.6 Guidelines and Limitations for VRRP.....	358
18.7 Guidelines and Limitations for VRRPv3.....	358
18.8 Default Settings for VRRP Parameters.....	359
18.9 Default Settings for VRRPv3 Parameters.....	359
18.10 Configuring VRRP.....	359
18.11 Configuring VRRPv3.....	368
18.12 Verifying the VRRP Configuration.....	375
18.13 Verifying the VRRPv3 Configuration.....	375
18.14 Monitoring and Clearing VRRP Statistics.....	375
18.15 Monitoring and Clearing VRRPv3 Statistics.....	376
18.16 Configuration Examples for VRRP.....	376
18.17 Configuration Examples for VRRPv3.....	377
18.18 Additional References.....	378
CHAPTER 19 Configuring Object Tracking	379
19.1 Information About Object Tracking.....	379
19.2 Licensing Requirements for Object Tracking.....	380
19.3 Guidelines and Limitations.....	380
19.4 Default Settings.....	380
19.5 Configuring Object Tracking.....	380
19.6 Verifying the Object Tracking Configuration.....	390
19.7 Configuration Examples for Object Tracking.....	390
19.8 Related Topics.....	390
APPENDIX A IETF RFCs Supported by Inspur INOS-CN Unicast Features	391

Figure

Figure 1-1 Packet Header Updates Through a Network.....	2
Figure 1-2 Simple Hub-and-Spoke Network.....	5
Figure 1-3 Inspur INOS-CN Forwarding Architecture.....	8
Figure 2-1 ARP Process.....	13
Figure 2-2 Reverse ARP.....	14
Figure 3-1 ggregatable Global Address Format.....	30
Figure 3-2 Link-Local Address Format.....	32
Figure 3-3 Pv4-Compatible IPv6 Address Format.....	32
Figure 3-4 Unique Local Address Structure.....	33
Figure 3-5 Subnet Router Anycast Address Format.....	33
Figure 3-6 IPv6 Multicast Address Format.....	34
Figure 3-7 IPv6 Solicited-Node Multicast Address Format.....	34
Figure 3-8 IPv4 Packet Header Format.....	35
Figure 3-9 Pv6 Packet Header Format.....	36
Figure 3-10 IPv6 Extension Header Format.....	37
Figure 5-1 DR in Multi-Access Network.....	53
Figure 5-2 OSPFv2 Areas.....	54
Figure 5-3 Stub Area.....	57
Figure 5-4 Virtual Links.....	58
Figure 6-1 DR in Multi-Access Network.....	96
Figure 6-2 OSPFv3 Areas.....	97
Figure 6-3 Stub Area.....	100
Figure 6-4 Virtual Links.....	101
Figure 8-1 IS-IS Network Divided into Areas.....	162
Figure 9-1 BGP PIC Edge Unipath.....	192
Figure 9-2 BGP PIC Edge Multipaths.....	193
Figure 10-1 iBGP Network.....	213
Figure 10-2 AS Confederation.....	214
Figure 10-3 Route Reflector.....	215

<i>Figure 10-4 BGP Route Advertisement with the Additional Paths Capability</i>	216
<i>Figure 11-1 RIP with Split Horizon Poison Reverse</i>	261
<i>Figure 13-1 VRFs in a Netwo</i>	284
<i>Figure 13-2 Service VRF Reachability</i>	286
<i>Figure 13-3 Service VRF Filtering</i>	286
<i>Figure 13-4 Service VRF Reachability Filtering</i>	286
<i>Figure 14-1 Inspur INOS-CN Forwarding Architecture</i>	300
<i>Figure 17-1 HSRP Topology with Two Enabled Routers</i>	333
<i>Figure 17-2 HSRP Load Sharing</i>	336
<i>Figure 18-1 Basic VRRP Topology</i>	354
<i>Figure 18-2 Load Sharing and Redundancy VRRP Topology</i>	355

Table

<i>Table 2-1 LPM Routing Modes for Inspur CN12900 Series Switches.....</i>	<i>12</i>
<i>Table 2-2 Default IP Parameters.....</i>	<i>16</i>
<i>Table 3-1 Compressed IPv6 Address Formats.....</i>	<i>29</i>
<i>Table 3-2 Base IPv6 Packet Header Fields.....</i>	<i>35</i>
<i>Table 3-3 IPv6 Extension Header Types.....</i>	<i>37</i>
<i>Table 3-4 IPv6 DNS Record Types.....</i>	<i>38</i>
<i>Table 3-5 LPM Routing Modes for inspur CN12900 Series Switches.....</i>	<i>38</i>
<i>Table 4-1 Default DNS Client Parameters.....</i>	<i>46</i>
<i>Table 5-1 LSA Types.....</i>	<i>54</i>
<i>Table 5-2 Default OSPFv2 Parameters.....</i>	<i>61</i>
<i>Table 6-1 LSA Types.....</i>	<i>97</i>
<i>Table 6-2 Default OSPFv3 Parameters.....</i>	<i>104</i>
<i>Table 7-1 Default EIGRP Parameters.....</i>	<i>140</i>
<i>Table 8-1 Default IS-IS Parameters.....</i>	<i>165</i>
<i>Table 9-1 BGP Default Administrative Distances.....</i>	<i>188</i>
<i>Table 9-2 BGP PIC Feature Support Matrix.....</i>	<i>194</i>
<i>Table 9-3 Default BGP Parameters.....</i>	<i>195</i>
<i>Table 10-1 Default BGP Parameters.....</i>	<i>221</i>
<i>Table 11-1 Default RIP Parameters.....</i>	<i>263</i>
<i>Table 12-1 Default Static Routing Parameters.....</i>	<i>278</i>
<i>Table 13-1 Default VRF Parameters.....</i>	<i>287</i>
<i>Table 15-1 Default Route Policy Manager Parameters.....</i>	<i>312</i>
<i>Table 16-1 Default Policy-Based Routing Parameters.....</i>	<i>328</i>
<i>Table 17-1 HSRP and IPv6 ND Addresses.....</i>	<i>334</i>
<i>Table 17-2 Default HSRP Parameters.....</i>	<i>338</i>
<i>Table 18-1 Default VRRP Parameters.....</i>	<i>359</i>
<i>Table 18-2 Default VRRPv3 Parameters.....</i>	<i>359</i>
<i>Table 19-1 Default Object Tracking Parameters.....</i>	<i>380</i>

CHAPTER 1 Overview

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

This chapter includes the following sections:

- About Layer 3 Unicast Routing
- Routing Algorithms
- Layer 3 Virtualization
- Inspur INOS-CN Forwarding Architecture
- Summary of Layer 3 Unicast Routing Features
- Guidelines and Limitations for Unicast

1.1 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.

1.1.1 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.

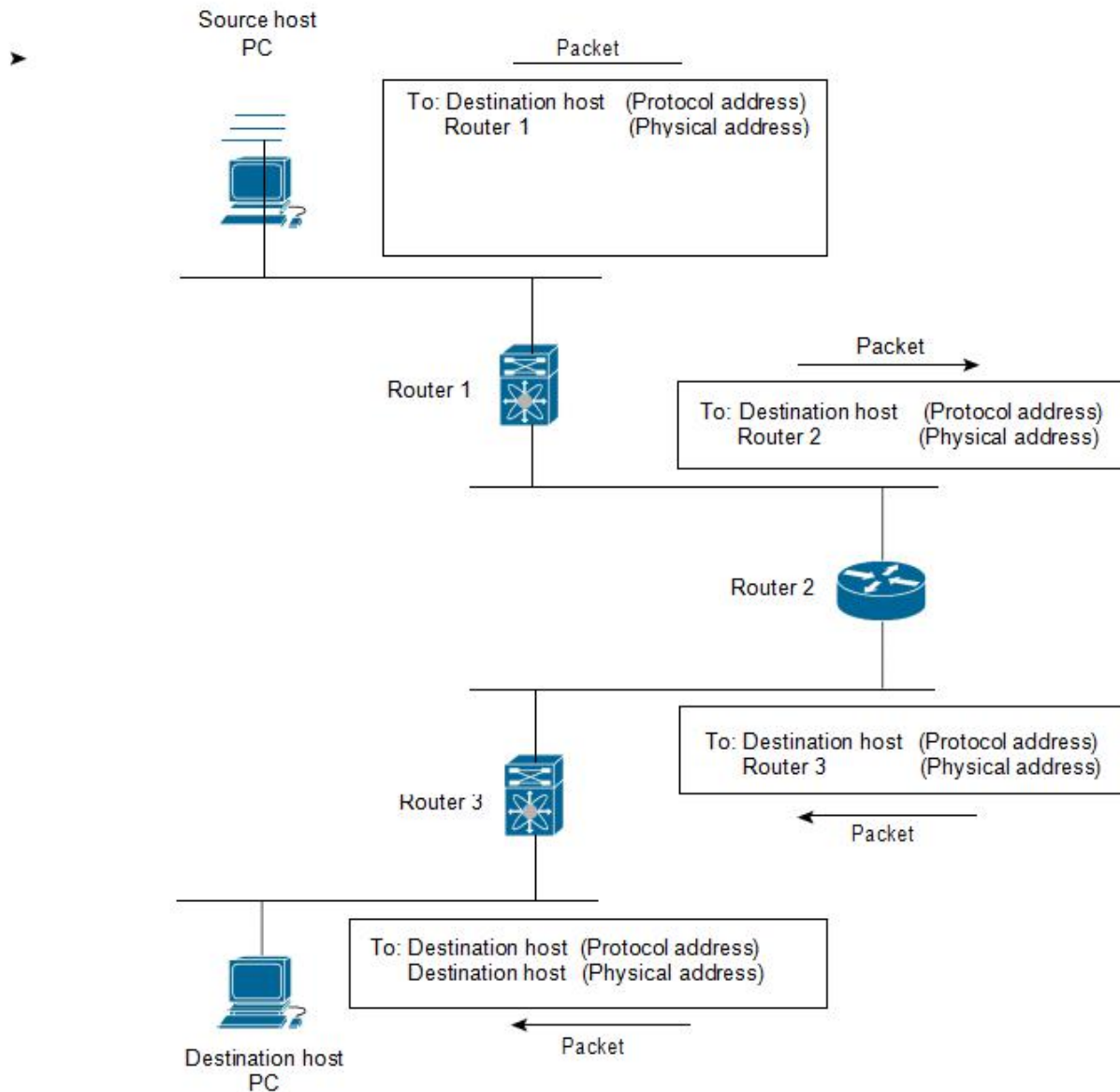
Routers communicate with one another and maintain their routing tables by transmitting a variety of messages. The routing update message is one such message that consists of all or a portion of a routing table. By analyzing routing updates from all other routers, a router can build a detailed picture 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.

1.1.2 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 Figure 1-1).

Figure 1-1 Packet Header Updates Through a Network

1.1.3 Routing Metrics

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

This section includes the following metrics:

- Path Length
- Reliability
- Routing Delay
- Bandwidth
- Load
- Communication Cost

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 a packet 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.

1.1.4 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-CN selects the router ID based on the following criteria:

- Inspur INOS-CN prefers loopback0 over any other interface. If loopback0 does not exist, then Inspur INOS-CN prefers the first loopback interface over any other interface type.
- If you have not configured a loopback interface, Inspur INOS-CN uses the first interface in the configuration file as the router ID. If you configure any loopback interface after Inspur INOS-CN 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.

1.1.5 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 (RIR) 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.

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-CN does not remove private autonomous system numbers from routing updates.

1.1.6 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.

1.1.7 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. For the number of ECMP paths supported by each routing protocol, see the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

1.1.8 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.

1.1.9 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.

1.1.10 Stub Routing

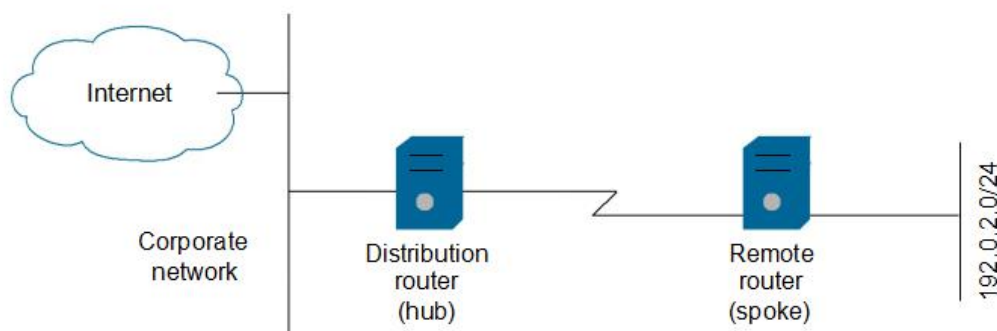
You can use stub routing in a hub-and-spoke network topology, where one or more end (stub) networks are connected to a remote router (the spoke) that is connected to one or more distribution routers (the hub). The remote router is adjacent only to one or more distribution routers. The only route for 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.

Figure 1-2 shows a simple hub-and-spoke configuration.

Figure 1-2 Simple Hub-and-Spoke Network



Stub routing does not prevent routes from being advertised to the remote router. Figure 1-2 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 reduces only the amount of memory required 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 nonlocal 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 the Enhanced Interior Gateway Routing Protocol (EIGRP) supports stub routers.

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.

This section includes the following topics:

- Static Routes and Dynamic Routing Protocols
- Interior and Exterior Gateway Protocols
- Distance Vector Protocols
- Link-State Protocols

1.2.1 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 subnet with a static route to the IP default gateway or router of last resort (a router to which all unroutable packets are sent).

1.2.2 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.

1.2.3 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.

1.2.4 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.

1.3 Layer 3 Virtualization

Inspur INOS-CN supports multiple virtual routing and forwarding (VRF) 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). A VRF represents a Layer 3 addressing domain. Each Layer 3 interface (logical or physical) belongs to one VRF. For more information, see Chapter 13, "Configuring Layer 3 Virtualization."

Inspur INOS-CN can segment operating system and hardware resources into virtual device contexts (VDCs) that emulate virtual devices. The Inspur CN12900 Series switch currently does not support multiple VDCs. All switch resources are managed in the default VDC.

1.4 Inspur INOS-CN Forwarding Architecture

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

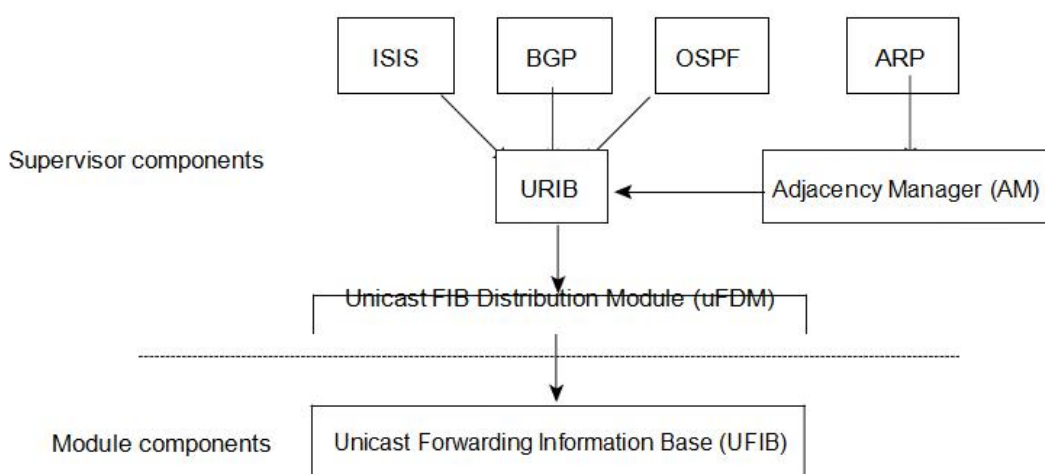
This section includes the following topics:

- Unicast RIB
- Adjacency Manager
- Unicast Forwarding Distribution Module
- FIB
- Hardware Forwarding
- Software Forwarding

1.4.1 Unicast RIB

The Inspur INOS-CN forwarding architecture consists of multiple components, as shown in Figure 1-3.

Figure 1-3 Inspur INOS-CN 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 FIB 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).

1.4.2 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. For more information, see Chapter 3, “Configuring IPv6.”

1.4.3 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 consist 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.

1.4.4 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.

1.4.5 Hardware Forwarding

Inspur INOS-CN 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.

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

1.4.6 Software Forwarding

The software forwarding path in Inspur INOS-CN 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. For more information, see the *Inspur CN12900 Series INOS-CN Security Configuration Guide*.

1.5 Summary of Layer 3 Unicast Routing Features

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

IPv4 and IPv6

Layer 3 uses either the IPv4 or IPv6 protocol. 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. EIGRP relies on its neighbors to provide the routes. It constructs the network topology from the routes advertised by its neighbors, similar to a link-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.

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-CN 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-CN. 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 the Hot Standby Router Protocol (HSRP) and the 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 because 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.

1.6 Related Topics

The following Inspur documents are related to the Layer 3 features:

- *Inspur CN12900 Series INOS-CN Multicast Routing Configuration Guide*
- *Inspur CN12900 Series INOS-CN High Availability and Redundancy Guide*

CHAPTER 2 Configuring IPv4

This chapter describes how to configure Internet Protocol version 4 (IPv4), which includes addressing, Address Resolution Protocol (ARP), and Internet Control Message Protocol (ICMP), on the Inspur INOS-CN device.

2.1 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 devices 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 and software access control list (ACL) 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.

2.1.1 Multiple IPv4 Addresses

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

2.1.2 LPM Routing Modes

By default, Inspur INOS-CN programs routes in a hierarchical fashion to allow for the longest prefix match (LPM) on the device. However, you can configure the device for different routing modes to support significantly more LPM route entries.

The following tables list the LPM routing modes that are supported on Inspur CN12900 Series switches.

Table 2-1 LPM Routing Modes for Inspur CN12900 Series Switches

LPM Routing Mode	Broadcom T2 Mode	CLI Command
Default system routing mode	3 (for line cards); 4 (for fabric modules)	

LPM Routing Mode	Broadcom T2 Mode	CLI Command
Max-host routing mode	2 (for line cards); 3 (for fabric modules)	system routing max-mode host
Nonhierarchical routing mode	3 (for line cards); 4 with max-l3-mode option (for line cards)	system routing non-hierarchical-routing [max-l3-mode]
64-bit ALPM routing mode	Submode of mode 4 (for fabric modules)	system routing mode hierarchical 64b-alpm
LPM heavy routing mode ¹		system routing template-lpm-heavy
LPM Internet-peering mode ²		system routing template-internet-peering

1. This mode is not supported for Inspur CN12900 Series switches.
2. This mode is not supported for Inspur CN12900 Series switches.

Host to LPM Spillover

Host routes can be stored in the LPM table in order to achieve a larger host scale. In ALPM mode, the switch allows fewer host routes. If you add more host routes than the supported scale, the routes that are spilled over from the host table take the space of the LPM routes in the LPM table. The total number of LPM routes allowed in that mode is reduced by the number of host routes stored. This feature is supported on Inspur CN12900 Series switches.

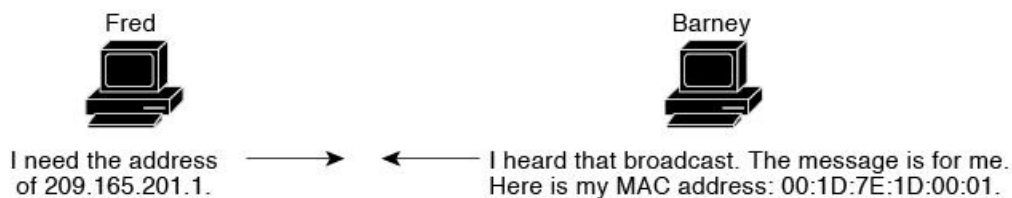
For Inspur CN12900 Series switches, only the default system routing and nonhierarchical routing modes support this feature on line cards. Fabric modules do not support this feature.

2.1.3 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 2-1 shows the ARP broadcast and response process.

Figure 2-1 ARP Process



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.

2.1.4 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.

2.1.5 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.

2.1.6 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.

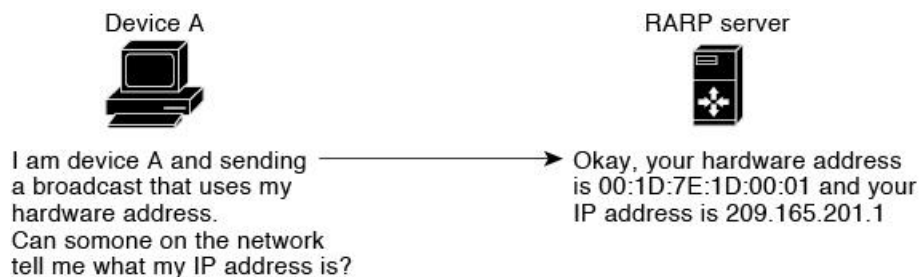
Layer 2 switches determine which port of a device receives a message that is sent only to that port. However, Layer 3 switches are devices that build an ARP cache (table).

2.1.7 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 2-2 shows how RARP works.

Figure 2-2 Reverse ARP



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:

- Because 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.

2.1.8 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.

2.1.9 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.

2.1.10 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-CN supports enabling or disabling gratuitous ARP requests or ARP cache updates.

2.1.11 Glean Throttling

If the Address Resolution Protocol (ARP) request for the next hop is not resolved when incoming IP packets are forwarded in a line card, 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.

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.

2.1.12 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.

2.1.13 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

2.1.14 Virtualization Support

IPv4 supports virtual routing and forwarding (VRF) instances.

2.2 Licensing Requirements for IPv4

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	IP requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

2.3 Prerequisites for IPv4

IPv4 has the following prerequisites:

- IPv4 can only be configured on Layer 3 interfaces.

2.4 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.
- Local proxy ARP is not supported for an interface with more than one HSRP group that belongs to multiple subnets.

2.5 Default Settings

Table 2-5 lists the default settings for IP parameters.

Table 2-2 Default IP Parameters

Parameters	Default
------------	---------

ARP timeout	1500 seconds
Proxy ARP	Disabled

2.6 Configuring IPv4

2.6.1 Configuring IPv4 Addressing

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

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip address** *ip-address/length*
4. (Optional) **show ip interface**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.168.1.1 255.0.0.0	Specifies a primary or secondary IPv4 address for an interface. <ul style="list-style-type: none"> • 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, which is the 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 4	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 5	copy running-config startup-config Example:	(Optional) Copies the running configuration to the startup configuration.

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

2.6.2 Configuring Multiple IP Addresses

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

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip address** *ip-address/length*
4. (Optional) **show ip interface**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip address <i>ip-address/length</i> [secondary] Example: switch(config-if)# ip address 192.168.1.1 255.0.0.0 secondary	Specifies the configured address as a secondary IPv4 address.
Step 4	show ip interface Example: switch(config-if)# show ip interface	(Optional) Displays interfaces configured for IPv4.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

2.6.3 Configuring Max-Host Routing Mode

By default, Inspur INOS-CN programs routes in a hierarchical fashion (with fabric modules configured to be in mode 4 and line card modules configured to be in mode 3), which allows for longest prefix match (LPM) and host scale on the device.

You can modify the default LPM and host scale to program more hosts in the system, as might be required when the node is positioned as a Layer-2 to Layer-3 boundary node.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing max-mode host**
3. (Optional)
4. **show forwarding route summary**
5. **copy running-config startup-config**
6. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing max-mode host Example: switch(config)# system routing max-mode host	Puts the line cards in Broadcom T2 mode 2 and the fabric modules in Broadcom T2 mode 3 to increase the number of supported hosts.
Step 3	show forwarding route summary Example: switch(config)# show forwarding route summary	(Optional) Displays the LPM routing mode.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload Example: switch(config)# reload	Reboots the entire device.

2.6.4 Configuring Nonhierarchical Routing Mode

If the host scale is small (as in a pure Layer 3 deployment), we recommend programming the longest prefix match (LPM) routes in the line cards to improve convergence performance. Doing so programs routes and hosts in the line cards and does not program any routes in the fabric modules.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing non-hierarchical-routing [max-l3-mode]**
3. (Optional) **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

Command	Purpose
----------------	----------------

Step 1	<pre>configure terminal Example: switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<pre>[no] system routing non-hierarchical-routing [max-l3-mode] Example: switch(config)# system routing non-hierarchical-routing max-l3-mode</pre>	Puts the line cards in Broadcom T2 mode 3 (or Broadcom T2 mode 4 if you use the max-l3-mode option) to support a larger LPM scale. As a result, all of the IPv4 and IPv6 routes will be programmed on the line cards rather than on the fabric modules.
Step 3	<pre>show forwarding route summary Example: switch(config)# show forwarding route summary Mode 3: 120K IPv4 Host table 16k LPM table (> 65 < 127 1k entry reserved) Mode 4: 16k V4 host/4k V6 host 128k v4 LPM/20K V6 LPM</pre>	(Optional) Displays the LPM routing mode.
Step 4	<pre>copy running-config startup-config Example: switch(config)# copy running-config startup-config</pre>	Saves this configuration change.
Step 5	<pre>reload Example: switch(config)# reload</pre>	Reboots the entire device.

2.6.5 Configuring 64-Bit ALPM Routing Mode

You can use the 64-bit algorithmic longest prefix match (ALPM) feature to manage IPv4 and IPv6 route table entries. In 64-bit ALPM routing mode, the device can store significantly more route entries. In this mode, you can program one of the following:

- 80,000 IPv6 entries and no IPv4 entries
- No IPv6 entries and 128,000 IPv4 entries
- x IPv6 entries and y IPv4 entries, where $2x + y \leq 128,000$

2.6.6 Configuring a Static ARP Entry

You can configure a static ARP entry on the device to map IP addresses to MAC hardware addresses.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **ip arp ipaddr mac_addr**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip arp <i>ipaddr mac_addr</i> Example: switch(config-if)# ip arp 192.168.1.1 0019.076c.1a78	Associates an IP address with a MAC address as a static entry.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

2.6.7 Configuring Proxy ARP

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

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **ip proxy-arp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ip proxy-arp Example: switch(config-if)# ip proxy-arp	Enables proxy ARP on the interface.
Step 4	copy running-config startup-config	(Optional) Copies the running configuration to the

Command	Purpose
Example: switch(config-if)# copy running-config startup-config	startup configuration.

2.6.8 Configuring Local Proxy ARP on Ethernet Interfaces

You can configure local proxy ARP on Ethernet interfaces.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *number*
3. **[no] ip local-proxy-arp**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	[no] ip local-proxy-arp Example: switch(config-if)# ip local-proxy-arp	Enables local proxy ARP on the interface.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

2.6.9 Configuring Local Proxy ARP on SVIs

You can configure local proxy ARP on SVIs and suppress ARP broadcasts on corresponding VLANs.

BEFORE YOU BEGIN

If you are planning to suppress ARP broadcasts, configure the double-wide ACL TCAM region size for ARP/Layer 2 Ethertype using the **hardware access-list tcam region arp-ether 256 double-wide** command, save the configuration, and reload the switch. (For more information, see the "Configuring ACL TCAM Region Sizes" section in the *Inspur CN12900 Series INOS-CN Security Configuration Guide*.)

SUMMARY STEPS

1. **configure terminal**
2. **interface vlan** *vlan-id*
3. **[no] ip local-proxy-arp [no-hw-flooding]**

4. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface vlan <i>vlan-id</i> Example: <pre>switch(config)# interface vlan 5 switch(config-if)#</pre>	Creates a VLAN interface and enters the configuration mode for the SVI.
Step 3	[no] ip local-proxy-arp [no-hw-flooding] Example: <pre>switch(config-if)# ip local-proxy-arp no-hw-flooding</pre>	Enables local proxy ARP on SVIs. The no-hw-flooding option suppresses ARP broadcasts on corresponding VLANs. Note If you configure the no-hw-flooding option and then want to change the configuration to allow ARP broadcasts on SVIs, you must first disable this feature using the no ip local-proxy-arp no-hw-flooding command and then enter the ip local-proxy-arp command.
Step 4	copy running-config startup-config Example: <pre>switch(config-if)# copy running-config startup-config</pre>	(Optional) Copies the running configuration to the startup configuration.

2.6.10 Configuring Gratuitous ARP

You can configure gratuitous ARP on an interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **ip arp gratuitous {request | update}**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface ethernet <i>number</i> Example:	Enters interface configuration mode.

	Command	Purpose
	switch(config)# interface ethernet 2/3 switch(config-if)#	
Step 3	ip arp gratuitous {request update} Example: switch(config-if)# ip arp gratuitous request	Enables gratuitous ARP on the interface. The default is enabled.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

2.6.11 Configuring Path MTU Discovery

You can configure path MTU discovery.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip tcp path-mtu-discovery Example: switch(config)# ip tcp path-mtu-discovery	Enables path MTU discovery.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

2.6.12 Configuring IP Directed Broadcasts

An IP directed broadcast is an IP packet whose destination address is a valid broadcast address for some IP subnet but which originates from a node that is not itself 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 it forwards unicast IP packets destined to 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 them as directed

broadcasts intended for the subnet to which that interface is attached are broadcasted on that subnet. You can optionally filter those broadcasts through an IP access list such that only those packets that pass through the access list are broadcasted on the subnet.

To enable IP directed broadcasts, use the following command in interface configuration mode:

Command	Purpose
<code>ip directed-broadcast [acl]</code>	Enables the translation of a directed broadcast to physical broadcasts. You can optionally filter those broadcasts through an IP access list.

2.6.13 Configuring IP Glean Throttling

We recommend that you configure IP glean throttling to filter the unnecessary glean packets that are sent to the supervisor for ARP resolution for the next hops that are not reachable or do not exist. IP glean throttling boosts software performance and helps to manage traffic more efficiently.

SUMMARY STEPS

1. **configure terminal**
2. **[no] hardware ip glean throttle**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>[no] hardware ip glean throttle</code> Example: switch(config)# <code>hardware ip glean throttle</code>	Enables IP glean throttling.
Step 3	<code>copy running-config startup-config</code> Example: switch(config)# <code>copy running-config startup-config</code>	(Optional) Copies the running configuration to the startup configuration.

2.6.14 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).

SUMMARY STEPS

1. **configure terminal**
2. **[no] hardware ip glean throttle maximum *count***
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>[no] hardware ip glean throttle maximum count</code> Example: switch(config)# <code>hardware ip glean throttle maximum 2134</code>	Configures the number of drop adjacencies that are installed in the FIB.
Step 3	<code>copy running-config startup-config</code> Example: switch(config)# <code>copy running-config startup-config</code>	(Optional) Copies the running configuration to the startup configuration.

2.6.15 Configuring a Hardware IP Glean Throttle Timeout

You can configure a timeout for the installed drop adjacencies to remain in the FIB.

SUMMARY STEPS

1. `configure terminal`
2. `[no] hardware ip glean throttle maximum timeout timeout`
3. (Optional) `copy running-config startup-config`

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>[no] hardware ip glean throttle maximum timeout timeout</code> Example: switch(config)# <code>hardware ip glean throttle maximum timeout 300</code>	Configures the timeout for the installed drop adjacencies to remain in the FIB. 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 3	<code>copy running-config startup-config</code> Example: switch(config)# <code>copy running-config startup-config</code>	(Optional) Copies the running configuration to the startup configuration.

2.6.16 Configuring the Interface IP Address for the ICMP Source IP Field

You can configure an interface IP address for the ICMP source IP field to handle ICMP error messages.

SUMMARY STEPS

1. **configure terminal**
2. **[no] ip source {ethernet *slot/port* | loopback *number* | port-channel *number*} icmp-errors**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] ip source {ethernet <i>slot/port</i> loopback <i>number</i> port-channel <i>number</i>} icmp-errors Example: switch(config)# ip source loopback 0 icmp-errors	Configures an interface IP address for the ICMP source IP field to route ICMP error messages.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

2.7 Verifying the IPv4 Configuration

To display the IPv4 configuration information, perform one of the following tasks:

Command	Purpose
show ip adjacency	Displays the adjacency table.
show ip adjacency summary	Displays a summary of throttle adjacencies.
show ip arp	Displays the ARP table.
show ip arp summary	Displays a summary of the number of throttle adjacencies.
show ip interface	Displays IP-related interface information.
show ip arp statistics [vrf <i>vrf-name</i>]	Displays the ARP statistics.
show system routing mode	Displays the LPM routing mode.

2.8 Additional References

2.8.1 Related Documents

Related Topic	Document Title

TCAM regions

See the "Configuring ACL TCAM Region Sizes" section in the *Inspur CN12900 Series INOS-CN Security Configuration Guide*.

CHAPTER 3 Configuring IPv6

This chapter describes how to configure Internet Protocol version 6 (IPv6), which includes addressing, on the Inspur INOS-CN device.

3.1 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 Integrated Intermediate System-to-Intermediate System (IS-IS), Open Shortest Path First (OSPF) for IPv6, and multiprotocol Border Gateway Protocol (BGP).

This section includes the following topics:

- IPv6 Address Formats
- IPv6 Unicast Addresses
- IPv6 Anycast Addresses
- IPv6 Multicast Addresses
- IPv4 Packet Header
- Simplified IPv6 Packet Header
- DNS for IPv6
- Path MTU Discovery for IPv6
- LPM Routing Modes
- Virtualization Support

3.1.1 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. 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. Table 3-1 shows a list of compressed IPv6 address formats.

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 3-1 Compressed IPv6 Address Formats

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	2001:0:0:0:0DB8:800:200C: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 Table 3-1 to send an IPv6 packet to itself. The loopback address in IPv6 is the same as the loopback address in IPv4. For more information, see Chapter 1, “Overview.”

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.

3.1.2 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. This section includes the following topics:

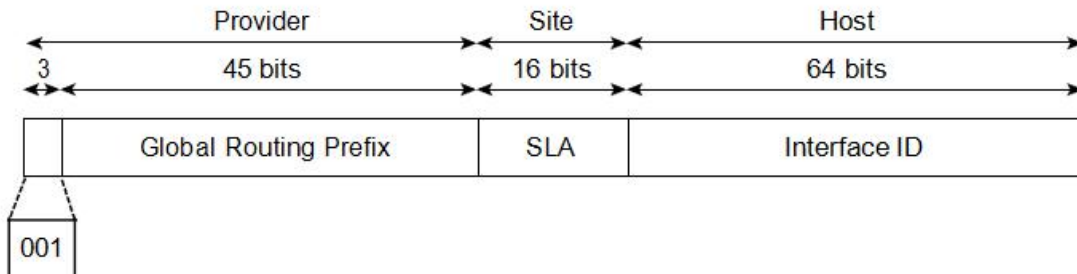
- Aggregatable Global Addresses
- Link-Local Addresses
- IPv4-Compatible IPv6 Addresses
- Unique Local Addresses
- Site-Local Address

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). Figure 3-1 shows the structure of an aggregatable global address.

Figure 3-1 aggregatable Global Address Format



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 TLA 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, and Frame Relay types), 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).

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:

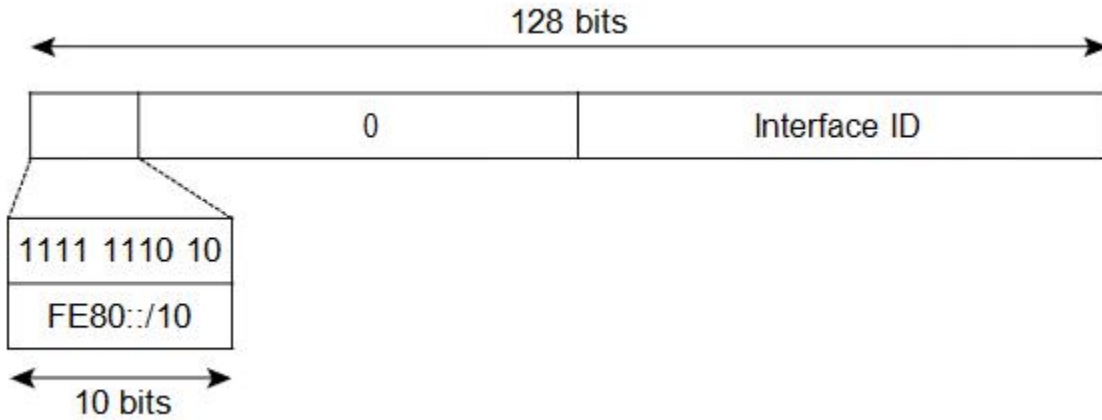
1. The router is queried for MAC addresses (from the pool of MAC addresses in the router).
2. If no MAC addresses are available in the router, the serial number of the router is used to form the link-local addresses.
3. If the serial number of the router cannot be used to form the link-local addresses, 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. Figure 3-2 shows the structure of a link-local address.

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

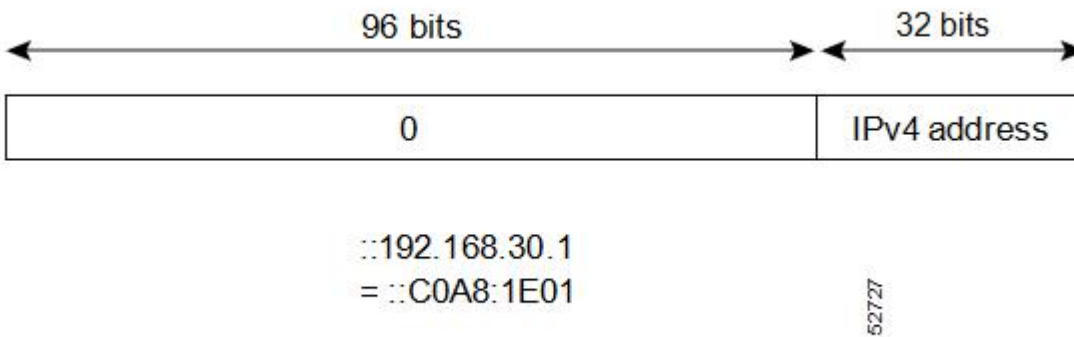
Figure 3-2 Link-Local Address Format



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: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 3-3 shows the structure of an IPv4-compatible IPv6 address and a few acceptable formats for the address.

Figure 3-3 IPv4-Compatible IPv6 Address Format



Unique Local Addresses

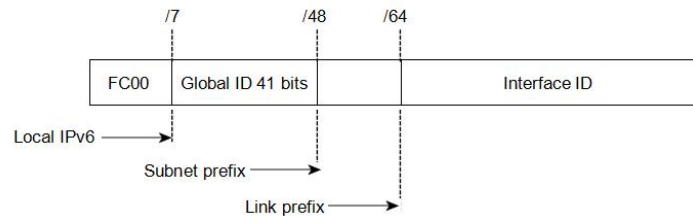
A unique local address is an IPv6 unicast address that is globally unique and is intended for local communications. It is not expected to be routable on the global Internet and is routable inside of a limited area, such as a site, and it may be routed between a limited set of sites. Applications may 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 3-4 shows the structure of a unique local address.

Figure 3-4 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 Address

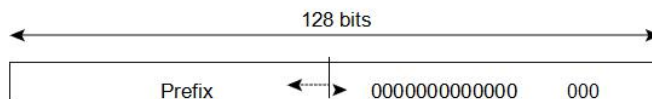
Because RFC 3879 deprecates the use of site-local addresses, you should follow the recommendations of unique local addressing (ULA) in RFC 4193 when you configure private IPv6 addresses.

3.1.3 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.

Figure 3-5 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 3-5 Subnet Router Anycast Address Format

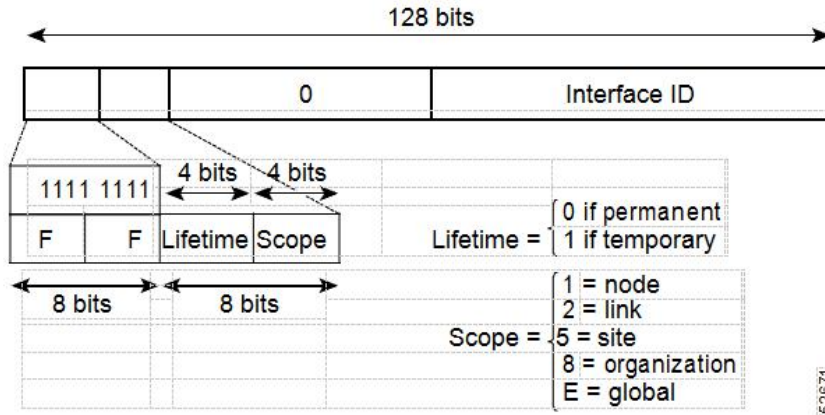


3.1.4 IPv6 Multicast Addresses

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 3-6 shows the format of the IPv6 multicast address.

Figure 3-6 IPv6 Multicast Address Format



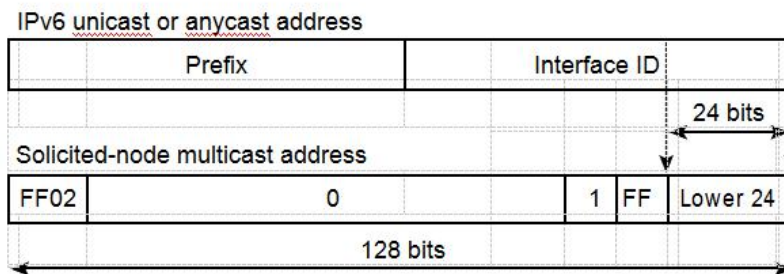
IPv6 nodes (hosts and routers) are required to join (where received packets are destined for) the following multicast groups:

- 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 (see Figure 3-7). 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 3-7 IPv6 Solicited-Node Multicast Address Format

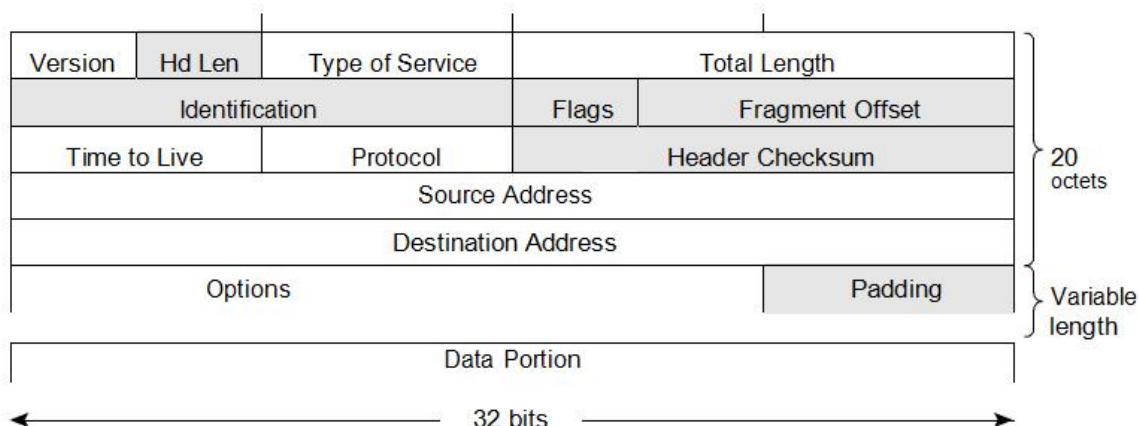


3.1.5 IPv4 Packet Header

The base IPv4 packet header has 12 fields with a total size of 20 octets (160 bits) (see Figure 3-8). 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 3-8 IPv4 Packet Header Format



3.1.6 Simplified IPv6 Packet Header

The base IPv6 packet header has 8 fields with a total size of 40 octets (320 bits) (see Figure 3-9). 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.

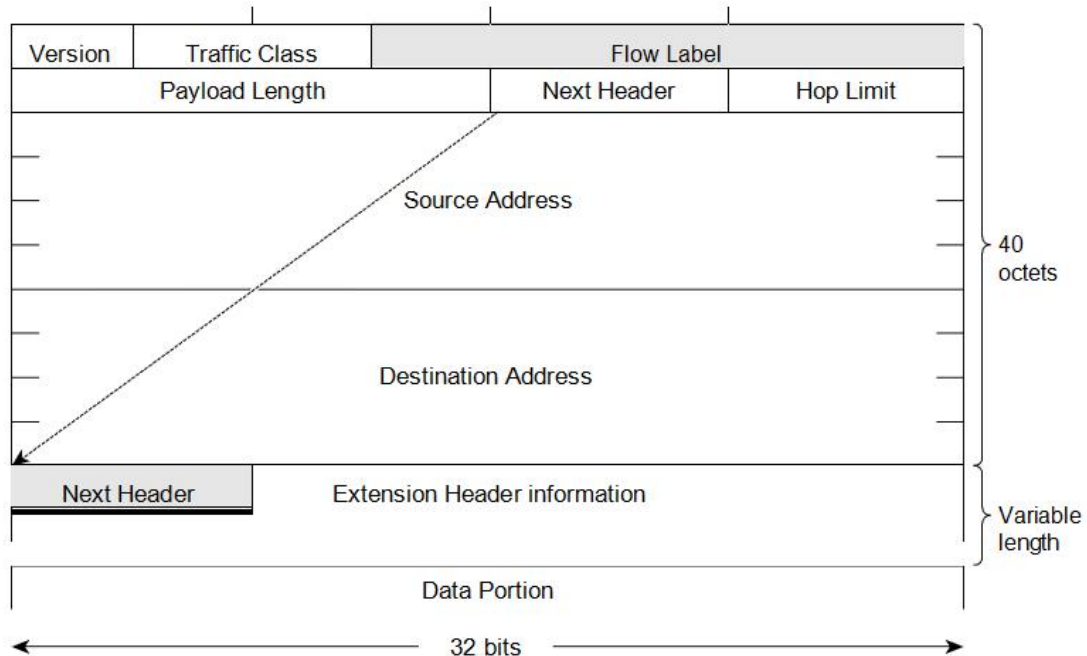
Table 3-2 lists the fields in the base IPv6 packet header.

Table 3-2 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, as shown in Figure 3-9.
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

Field	Description
	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 3-9 Pv6 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 3-10 shows the IPv6 extension header format.

Figure 3-10 IPv6 Extension Header Format

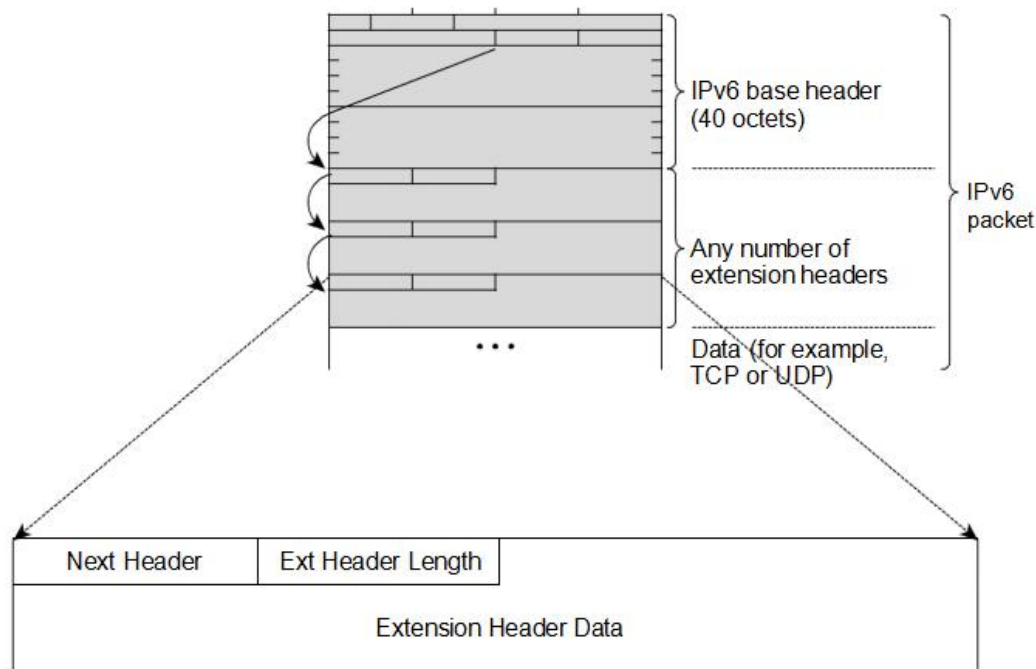


Table 3-3 lists the extension header types and their Next Header field values.

Table 3-3 IPv6 Extension Header Types

Header Type	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 options header	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
Routing header	43	Header that is used for source 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 inside a packet to transport the data. The two main transport protocols are TCP and UDP.

3.1.7 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 (see Table 3-4).

Table 3-4 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.)	2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.8.1.c.0 .y.y.y.y.e.f.f.3.ip6.int PTR www.abc.test

3.1.8 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-CN retains the lower value. The connection does not increase the segment size to gauge the throughput.

3.1.9 LPM Routing Modes

By default, Inspur INOS-CN programs routes in a hierarchical fashion to allow for the longest prefix match (LPM) on the device. However, you can configure the device for different routing modes to support significantly more LPM route entries.

The following tables list the LPM routing modes that are supported on the Inspur CN12900 Series switches.

Table 3-5 LPM Routing Modes for Inspur CN12900 Series Switches

LPM Routing Mode	Broadcom T2 Mode	CLI Command
Default system routing mode	3 (for line cards); 4 (for fabric modules)	
Max-host routing mode	2 (for line cards); 3 (for fabric modules)	system routing max-mode host
Nonhierarchical routing mode	3 (for line cards); 4 with max-l3-mode option (for line cards)	system routing non-hierarchical-routing [max-l3-mode]
64-bit ALPM routing mode	Submode of mode 4 (for fabric modules)	system routing mode hierarchical 64b-alpm
LPM heavy routing mode ¹		system routing template-lpm-heavy
LPM Internet-peering mode ²		system routing template-internet-peering

1. This mode is not supported for Inspur CN12900 Series switches.

2. This mode is not supported for Inspur CN12900 Series switches.

For detailed configuration information, see the “Configuring IPv6” section.

Host to LPM Spillover

Host routes can be stored in the LPM table in order to achieve a larger host scale. In ALPM mode, the switch allows fewer host routes. If you add more host routes than the supported scale, the routes that are spilled over from the

host table take the space of the LPM routes in the LPM table. The total number of LPM routes allowed in that mode is reduced by the number of host routes stored. This feature is supported on Inspur CN12900 Series switches.

For Inspur CN12900 Series switches, only the default system routing and nonhierarchical routing modes support this feature on line cards. Fabric modules do not support this feature.

3.1.10 Virtualization Support

IPv6 supports virtual routing and forwarding (VRF) instances.

3.2 Licensing Requirements for IPv6

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	IPv6 requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

3.3 Prerequisites for IPv6

IPv6 has the following prerequisites:

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

3.4 Guidelines and Limitations for 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.
- IPv6 static route next-hop link-local addresses cannot be configured at any local interface.
- You must define the BGP update source when using a link-local IPv6 address.
- 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.

3.5 Configuring IPv6

3.5.1 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet *number***
3. **ipv6 address {*addr* [*eui64*] [*route-preference preference*] [*secondary*] tag *tag-id*]} or **ipv6****

- address ipv6-address use-link-local-only**
4. (Optional) **show ipv6 interface**
 5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet number Example: switch(config)# interface ethernet 2/3 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 address {addr [eui64] [route-preference preference] [secondary] tag tag-id or ipv6 address ipv6-address use-link-local-only Example: switch(config-if)# ipv6 address 2001:0DB8::1/10 or switch(config-if)# 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	show ipv6 interface Example: switch(config-if)# show ipv6 interface	(Optional) Displays interfaces configured for IPv6.
Step 5	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure an IPv6 address:

```
switch# configure terminal
switch(config)# interface ethernet
3/1
switch(config-if)# ipv6 address ?
A:::C:D/LEN IPv6 prefix format: xxxx:xxxx/ml,
xxxx:xxxx::/ml, xxxx::xx/128
use-link-local-only Enable IPv6 on interface using only a single link-
```

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

This example shows how to display an IPv6 interface:

```
switch(config-if)# show ipv6 interface ethernet 3/1
Ethernet3/1, Interface status: protocol-down/link-down/admin-down, iod:
 36 IPv6 address: 2001:db8:0000:0000:0218:baff:fed8:239d
 IPv6 subnet: 2001:db8::/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
```

3.5.2 Configuring Max-Host Routing Mode

By default, the device programs routes in a hierarchical fashion (with fabric modules configured to be in mode 4 and line card modules configured to be in mode 3), which allows for longest prefix match (LPM) and host scale on the device.

You can modify the default LPM and host scale to program more hosts in the system, as might be required when the node is positioned as a Layer-2 to Layer-3 boundary node.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing max-mode host**
3. (Optional) **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing max-mode host Example: switch(config)# system routing max-mode host	Puts the line cards in Broadcom T2 mode 2 and the fabric modules in Broadcom T2 mode 3 to increase the number of supported hosts.
Step 3	show forwarding route summary	(Optional) Displays the LPM routing mode.

	Command	Purpose
	Example: <pre>switch(config)# show forwarding route summary</pre>	
Step 4	copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	Saves this configuration change.
Step 5	reload Example: <pre>switch(config)# reload</pre>	Reboots the entire device.

3.5.3 Configuring Nonhierarchical Routing Mode

If the host scale is small (as in a pure Layer 3 deployment), we recommend programming the longest prefix match (LPM) routes in the line cards to improve convergence performance. Doing so programs routes and hosts in the line cards and does not program any routes in the fabric modules.

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing non-hierarchical-routing [max-l3-mode]**
3. (Optional) **show forwarding route summary**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	[no] system routing non-hierarchical-routing [max-l3-mode] Example: <pre>switch(config)# system routing non-hierarchical-routing max-l3-mode</pre>	Puts the line cards in Broadcom T2 mode 3 (or Broadcom T2 mode 4 if you use the max-l3-mode option) to support a larger LPM scale. As a result, all of the IPv4 and IPv6 routes will be programmed on the line cards rather than on the fabric modules.
Step 3	show forwarding route summary Example: <pre>switch(config)# show forwarding route summary Mode 3: 120K IPv4 Host table 16k LPM table (> 65 < 127 1k entry reserved) Mode 4: 16k V4 host/4k V6 host</pre>	(Optional) Displays the LPM routing mode.

	Command	Purpose
	128k v4 LPM/20K V6 LPM	
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload Example: switch(config)# reload	Reboots the entire device.

3.5.4 Configuring 64-Bit ALPM Routing Mode

You can use the 64-bit algorithmic longest prefix match (ALPM) feature to manage IPv4 and IPv6 route table entries. In 64-bit ALPM routing mode, the device can store significantly more route entries. Using this mode, you can program one of the following:

- 80,000 IPv6 entries and no IPv4 entries
- No IPv6 entries and 128,000 IPv4 entries
- x IPv6 entries and y IPv4 entries, where $2x + y \leq 128,000$

SUMMARY STEPS

1. **configure terminal**
2. **[no] system routing mode hierarchical 64b-alpm**
3. (Optional) **show system routing mode**
4. **copy running-config startup-config**
5. **reload**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] system routing mode hierarchical 64b-alpm Example: switch(config)# system routing mode hierarchical 64b-alpm	Causes all IPv4 and IPv6 LPM routes with a mask length that is less than or equal to 64 to be programmed in the fabric module. All host routes for IPv4 and IPv6 and all LPM routes with a mask length of 65 through 127 are programmed in the line card.
Step 3	show system routing mode Example: switch(config)# show system routing mode	(Optional) Displays the LPM routing mode.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	Saves this configuration change.
Step 5	reload	Reboots the entire device.

Command	Purpose
Example: switch(config)# reload	

3.6 Verifying the IPv6 Configuration

To display the IPv6 configuration, perform one of the following tasks:

Command	Purpose
show ipv6 interface	Displays IPv6-related interface information.
show ipv6 adjacency	Displays the adjacency table.
show system routing mode	Displays the LPM routing mode.

3.7 Configuration Examples for IPv6

This example shows how to configure IPv6:

```
configure terminal
interface ethernet 3/1
  ipv6 address 2001:db8::/64 eui64
  ipv6 nd reachable-time 10
```

CHAPTER 4 Configuring DNS

This chapter describes how to configure the Domain Name Server (DNS) client on the Inspur INOS-CN device.

4.1.1 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 *icntnetworks.com*. A specific hostname in this domain, the File Transfer Protocol (FTP) system, for example, is identified as *ftp.icntnetworks.com*.

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-CN, you must identify the hostnames, specify a name server, and enable the DNS service.

Inspur INOS-CN allows you to statically map IP addresses to domain names. You can also configure Inspur INOS-CN 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.

4.1.2 High Availability

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

4.1.3 Virtualization Support

Inspur INOS-CN supports multiple instances of the DNS clients that run on the same system. You can configure a DNS client. You can optionally have a different DNS client configuration in each virtual routing and forwarding (VRF) instance.

4.2 Licensing Requirements for DNS Clients

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	DNS requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

4.3 Prerequisites for DNS Clients

The DNS client has the following prerequisites:

- You must have a DNS name server on your network.

4.4 Guidelines and Limitations for DNS

The DNS client has the following configuration guidelines and limitations:

- You configure the DNS client in a specific VRF. If you do not specify a VRF, Inspur INOS-CN uses the default VRF.
- DNS supports IPv6 addresses.

4.5 Default Settings

Table 4-1 lists the default settings for DNS client parameters.

Table 4-1 Default DNS Client Parameters

Parameters	Default
DNS client	Enabled

4.6 Configuring DNS Clients

4.6.1 Configuring the DNS Client

You can configure the DNS client to use a DNS server on your network.

BEFORE YOU BEGIN

Ensure that you have a domain name server on your network.

SUMMARY STEPS

- configure terminal**
- {ip | ipv6} host name ip/ipv6-address1 [ip/ipv6-address2... ip/ipv6-address6]**
- (Optional) **ip domain-name name [use-vrf vrf-name]**
- (Optional) **ip domain-list name [use-vrf vrf-name]**
- (Optional) **ip name-server ip/ipv6-address1 [ip/ipv6-address2... ip/ipv6-address6] [use-vrf vrf-name]**
- (Optional) **ip domain lookup**
- (Optional) **show hosts**

8. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	{ip ipv6} host name ip/ipv6-address1 [ip/ipv6-address2... ip/ipv6-address6] Example: <pre>switch(config)# ip host inspur-rtp 192.0.2.1</pre> Example: <pre>switch(config)# ipv6 host inspur-rtp 2001:DB8:1::1</pre>	Defines up to six static hostname-to-address mappings in the hostname cache. The address can be either an IPv4 or IPv6 address.
Step 3	ip domain-name name [use-vrf vrf-name] Example: <pre>switch(config)# ip domain-name myserver.com</pre>	(Optional) Defines the default domain name that Inspur INOS-CN uses to complete unqualified host names. You can optionally define a VRF that Inspur INOS-CN uses to resolve this domain name if it cannot be resolved in the VRF that you configured this domain name under. Inspur INOS-CN appends the default domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup.
Step 4	ip domain-list name [use-vrf vrf-name] Example: <pre>switch(config)# ip domain-list mycompany.com</pre>	(Optional) Defines additional domain names that Inspur INOS-CN can use to complete unqualified hostnames. You can optionally define a VRF that Inspur INOS-CN uses to resolve these domain names if they cannot be resolved in the VRF that you configured this domain name under. Inspur INOS-CN 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-CN continues this process for each entry in the domain list until it finds a match.
Step 5	ip name-server ip/ipv6-address1 [ip/ipv6-address2... ip/ipv6-address6] [use-vrf vrf-name] Example: <pre>switch(config)# ip name-server 192.0.2.22</pre> Example: <pre>switch(config)# ip name-server 2001:DB8:1::1</pre>	(Optional) Defines up to six name servers. The address can be either an IPv4 or IPv6 address. You can optionally define a VRF that Inspur INOS-CN uses to reach this name server if it cannot be reached in the VRF that you configured this name server under.

	Command	Purpose
Step 6	ip domain-lookup Example: switch(config)# ip domain-lookup	(Optional) Enables DNS-based address translation. This feature is enabled by default.
Step 7	show hosts Example: switch(config)# show hosts	(Optional) Displays information about DNS.
Step 8	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure a default domain name and enable DNS lookup:

```
switch# configure terminal
switch(config)# ip domain-name inspur.com
switch(config)# ip name-server 192.0.2.1 use-vrf
management
switch(config)# ip domain-lookup
switch(config)# copy running-config startup-config
```

4.6.2 Configuring Virtualization

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.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. (Optional) **ip domain-name** *name* [**use-vrf** *vrf-name*]
4. (Optional) **ip domain-list** *name* [**use-vrf** *vrf-name*]
5. (Optional) **ip name-server** *ip/ipv6-address1* [*ip/ipv6-address2...* *ip/ipv6-address6*] [**use-vrf** *vrf-name*]
6. (Optional) **show hosts**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.

	Command	Purpose
Step 2	vrf context <i>vrf-name</i> Example: <pre>switch(config)# vrf context Red switch(config-vrf)#</pre>	Creates a VRF and enters VRF configuration mode.
Step 3	ip domain-name <i>name</i> [use-vrf <i>vrf-name</i>] Example: <pre>switch(config-vrf)# ip domain-name myserver.com</pre>	(Optional) Defines the default domain name server that Inspur INOS-CN uses to complete unqualified hostnames. You can optionally define a VRF that Inspur INOS-CN uses to resolve this domain name server if it cannot be resolved in the VRF under which you configured this domain name. Inspur INOS-CN appends the default domain name to any hostname that does not contain a complete domain name before starting a domain-name lookup.
Step 4	ip domain-list <i>name</i> [use-vrf <i>vrf-name</i>] Example: <pre>switch(config-vrf)# ip domain-list mycompany.com</pre>	(Optional) Defines additional domain name servers that Inspur INOS-CN can use to complete unqualified hostnames. You can optionally define a VRF that Inspur INOS-CN uses to resolve this domain name server if it cannot be resolved in the VRF under which you configured this domain name. Inspur INOS-CN 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-CN continues this process for each entry in the domain list until it finds a match.
Step 5	ip name-server <i>ip/ipv6-address1</i> <i>[ip/ipv6-address2... ip/ipv6-address6]</i> [use-vrf vrf-name] Example: <pre>switch(config-vrf)# ip name-server 192.0.2.22</pre> Example: <pre>switch(config)# ip name-server 2001:DB8:1::1</pre>	(Optional) Defines up to six name servers. The address can be either an IPv4 or IPv6 address. You can optionally define a VRF that Inspur INOS-CN uses to reach this name server if it cannot be reached in the VRF that you configured this name server under.
Step 6	show hosts Example: <pre>switch(config-vrf)# show hosts</pre>	(Optional) Displays information about DNS.
Step 7	copy running-config startup-config Example: <pre>switch(config-vrf)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This example shows how to configure a default domain name and enable DNS lookup within a VRF:

```
switch# configure terminal
switch(config)# vrf context Red
switch(config-vrf)# ip domain-name inspur.com
switch(config-vrf)# ip name-server 192.0.2.1 use-vrf
management
switch(config-vrf)# copy running-config startup-config
```

4.7 Verifying the DNS Client Configuration

To display the DNS client configuration, perform one of the following tasks:

Command	Purpose
<code>show hosts</code>	Displays information about DNS.

4.8 Configuration Examples for the DNS Client

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
```


CHAPTER 5 Configuring OSPFv2

This chapter describes how to configure Open Shortest Path First version 2 (OSPFv2) for IPv4 networks on the Inspur INOS-CN device.

5.1 About OSPFv2

OSPFv2 is an IETF link-state protocol (see the “Link-State Protocols” section) 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 Hello packet to determine if the routers have compatible configurations. The neighbor routers try to establish adjacency, which means that the routers synchronize their link-state databases to ensure that they have identical OSPFv2 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 OSPFv2 routers eventually have identical link-state databases. When all OSPFv2 routers have identical link-state databases, the network is converged (see the “Convergence” section). 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 Chapter 6, “Configuring OSPFv3.”

5.1.1 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 (see the “Designated Routers” section)

The Hello packet contains information about the originating OSPFv2 interface and router, including the assigned OSPFv2 cost of the link, the hello interval, 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, 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 dead interval (usually a multiple of the hello interval), then the neighbor is removed from the local neighbor table.

5.1.2 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 (see the “Areas” section)
- 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 (see the “Designated Routers” section).
- 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 (see the “Designated Routers” section).
- Local interface—The local interface that received the Hello packet for this neighbor.

5.1.3 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. For more information, see the “Designated Routers” section.

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 (see the “Link-State Database” section). 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.

5.1.4 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 designated router (*DR*), to control the LSA floods and represent the network to the rest of the OSPFv2 area (see the “Areas” section). If the DR fails, OSPFv2 selects a backup designated router (*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.

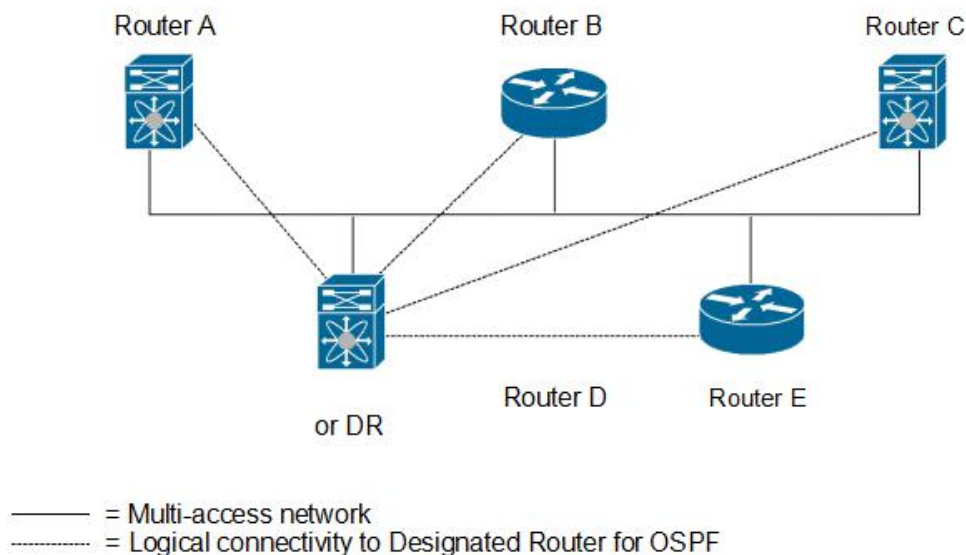
The DR and BDR are selected based on the information in the Hello packet. When an interface sends a Hello packet, it sets the priority field and the DR and BDR field if it knows who the DR and BDR are. The routers follow an election procedure based on which routers declare themselves in the DR and BDR 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 5-1 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 5-1 DR in Multi-Access Network

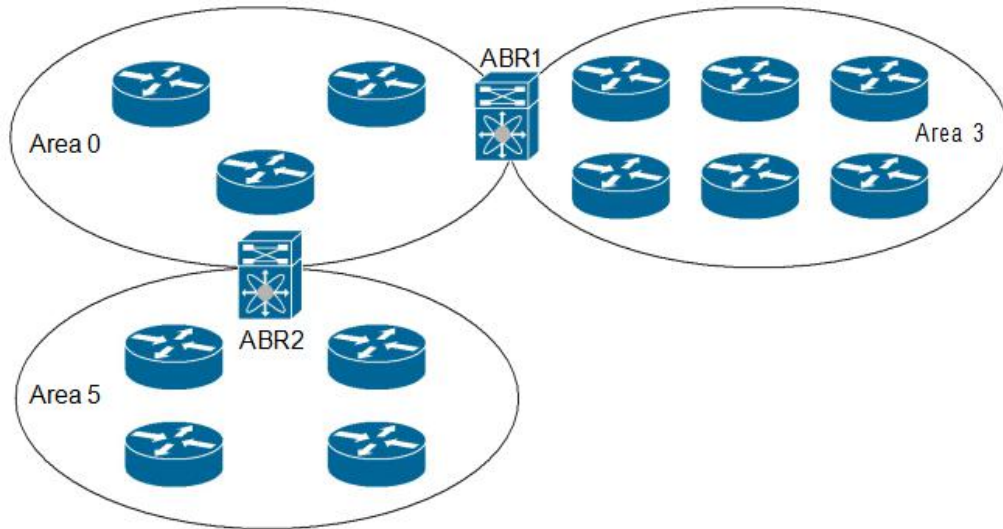


5.1.5 Areas

You can limit the CPU and memory requirements that OSPFv2 puts on the routers by dividing an OSPFv2 network into areas. 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-CN 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 area border routers (ABRs). An ABR connects to both the backbone area and at least one other defined area (see Figure 5-2).

Figure 5-2 OSPFv2 Areas

The ABR has a separate link-state database for each area to which it connects. The ABR sends Network Summary (type 3) LSAs (see the “Route Summarization” section) from one connected area to the backbone area. The backbone area sends summarized information about one area to another area. In Figure 5-2, 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.

5.1.6 Link-State Advertisements

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

LSA Types

Table 5-1 shows the LSA types supported by Inspur INOS-CN.

Table 5-1 LSA Types

Type	Name	Description
1	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.
2	Network LSA	LSA sent by the DR. This LSA lists all routers in the multi-access network. Network LSAs trigger an SPF recalculation. See the “Designated Routers” section.
3	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. See the “Areas” section.
4	ASBR Summary LSA	LSA sent by the area border router to an external area. This LSA advertises the link cost to the ASBR only. See the “Areas” section.
5	AS External	LSA generated by the ASBR. This LSA includes the link cost to an external

Type	Name	Description
	LSA	autonomous system destination. AS External LSAs are flooded throughout the autonomous system. See the “Areas” section.
7	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. See the “Areas” section
9–11	Opaque LSAs	LSA used to extend OSPF. See the “Opaque LSAs” section.

Link Cost

Each OSPFv2 interface is assigned a link cost. The cost is an arbitrary number. By default, Inspur INOS-CN 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 (see the “Areas” section). 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 usage. 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-CN 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 (see the “High Availability and Graceful Restart” section). 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.

5.1.7 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 Routing Information Base (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 (see the “OSPFv2 Stub Router Advertisements” section) OSPFv2 also runs a modified Dijkstra algorithm for fast recalculation for summary and external (type 3, 4, 5, and 7) LSA changes.

5.1.8 Authentication

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

- Simple password authentication
- Cryptographic authentication

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

Simple Password Authentication

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

Cryptographic Authentication

Cryptographic authentication uses an encrypted password for OSPFv2 authentication. The transmitter computes a code using the packet to be transmitted and the key string, inserts the code and the key ID in the packet, and transmits the packet. The receiver validates the code in the packet by computing the code locally using the received packet and the key string (corresponding to the key ID in the packet) configured locally.

Both message digest 5 (MD5) and hash-based message authentication code secure hash algorithm (HMAC-SHA) cryptographic authentication are supported.

MD5 Authentication

You can use MD5 authentication to authenticate OSPFv2 messages by configuring a password that is shared at the local router and all remote OSPFv2 neighbors. For each OSPFv2 message, Inspur INOS-CN 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.

HMAC-SHA Authentication

OSPFv2 supports RFC 5709 to allow the use of HMAC-SHA algorithms, which offer more security than MD5. The HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 algorithms are supported for OSPFv2 authentication.

5.1.9 Advanced Features

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

Stub Area

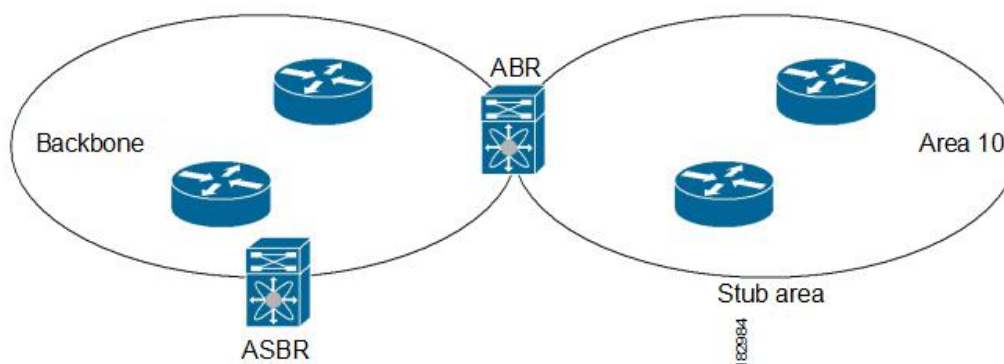
You can limit the amount of external routing information that floods an area by making it a stub area. A stub area

is an area that does not allow AS External (type 5) LSAs (see the “Link-State Advertisements” section). 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. See the “Stub Routing” section.
- No ASBR routers exist in the stub area.
- You cannot configure virtual links in the stub area.

Figure 5-3 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 5-3 Stub Area



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

Not-So-Stubby Area

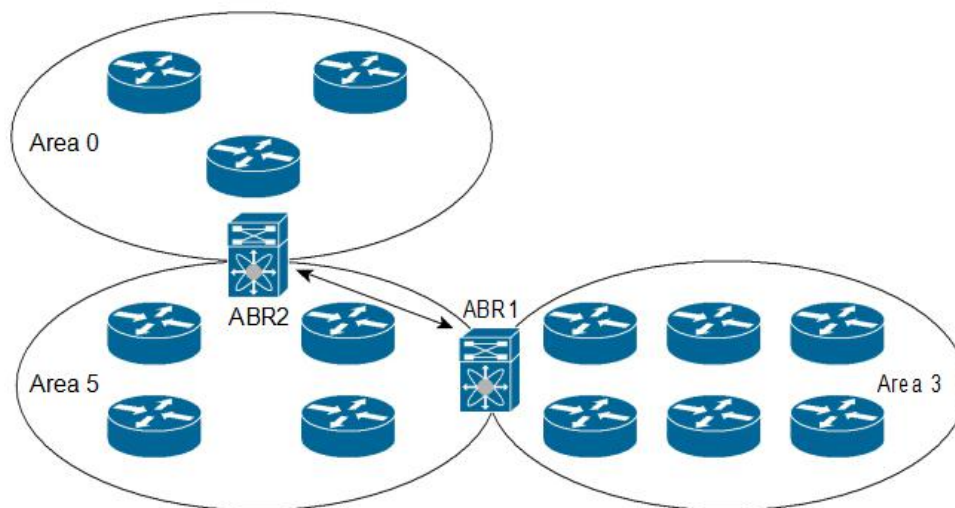
A Not-so-Stubby Area (NSSA) is similar to a stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates NSSA External (type 7) LSAs that it floods throughout the NSSA. You can optionally 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. See the “Link-State Advertisements” section for information about NSSA External LSAs.

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 (see the “Configuring NSSA” section).

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. Figure 5-4 shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 5-4 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. See the “Route Redistribution” section. 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.

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-CN automatically configures a discard route for the summary address to prevent routing black holes and route loops.

High Availability and Graceful Restart

Inspur INOS-CN 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 (see the “Opaque LSAs” section). 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
 - 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 ospf** command
 - Active supervisor removal
 - Active supervisor reload using the **reload module active-sup** command

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-CN 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. For the number of supported OSPFv2 instances, see the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

SPF Optimization

Inspur INOS-CN 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-CN 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) for IPv4. 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information.

Virtualization Support

Inspur INOS-CN supports multiple process instances for OSPFv2. Each OSPFv2 instance can support multiple virtual routing and forwarding (VRF) instances, up to the system limit. For the number of supported OSPFv2 instances, see the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

5.2 Licensing Requirements for OSPFv2

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	OSPFv2 requires an Essentials license package.

5.3 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 (see the “Enabling OSPFv2” section).

5.4 Guidelines and Limitations for OSPFv2

OSPFv2 has the following configuration guidelines and limitations:

- Inspur INOS-CN 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-CN 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 you observe a timeout on the querying SNMP agent while polling OSPF MIB objects, increase the timeout value on the polling SNMP agent.

- The following guidelines and limitations apply to the administrative distance feature:
 - When an OSPF route has two or more equal cost paths, configuring the administrative distance is nondeterministic 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.
 - The discard route is always assigned an administrative distance of 220. No configuration in the table map applies to OSPF discard routes.
- If you configure the **delay restore seconds** command in vPC configuration mode and if the VLANs on the multi-chassis EtherChannel trunk (MCT) are announced by OSPFv2 or OSPFv3 using switch virtual interfaces (SVIs), those SVIs are announced with MAX_LINK_COST on the vPC secondary node for the duration of the configured time. As a result, all route or host programming completes after the vPC synchronization operation (on a peer reload of the secondary vPC node) before attracting traffic. This behavior allows for minimal packet loss for any north-to-south traffic.
- The output of the **show run ospf** command might show the default values for some OSPF commands.

5.5 Default Settings

Table 5-2 lists the default settings for OSPFv2 parameters.

Table 5-2 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 calculation minimum hold time	1000 milliseconds
SPF calculation maximum wait time	5000 milliseconds

5.6 Configuring Basic OSPFv2

Configure OSPFv2 after you have designed your OSPFv2 network.

5.6.1 Enabling OSPFv2

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

SUMMARY STEPS

1. **configure terminal**

2. **feature ospf**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature ospf Example: switch(config)# feature ospf	Enables the OSPFv2 feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To disable the OSPFv2 feature and remove all associated configuration, use the **no feature ospf** command in global configuration mode:

Command	Purpose
no feature ospf Example: switch(config)# no feature ospf	Disables the OSPFv2 feature and removes all associated configuration.

5.6.2 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf instance-tag**
3. **router-id ip-address**
4. (Optional) **show ip ospf instance-tag**

5. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	router-id ip-address Example: switch(config-router)# router-id 192.0.2.1	(Optional) Configures the OSPFv2 router ID. This IP address identifies this OSPFv2 instance and must exist on a configured interface in the system.
Step 4	show ip ospf instance-tag Example: switch(config-router)# show ip ospf 201	(Optional) Displays OSPF information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To remove the OSPFv2 instance and all associated configuration, use the **no router ospf** command in global configuration mode.

Command	Purpose
no router ospf instance-tag Example: switch(config)# no router ospf 201	Deletes the OSPF instance and the associated configuration.

5.6.3 Configuring Optional Parameters on an OSPFv2 Instance

You can configure optional parameters for OSPF.

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.

DETAILED STEPS

You can configure the following optional parameters for OSPFv2 in router configuration mode:

Command	Purpose
---------	---------

distance <i>number</i> Example: <pre>switch(config-router)# distance 25</pre>	Configures the administrative distance for this OSPFv2 instance. The range is from 1 to 255. The default is 110.
log-adjacency-changes [<i>detail</i>] Example: <pre>switch(config-router)# log-adjacency-changes</pre>	Generates a system message whenever a neighbor changes state.
maximum-paths <i>path-number</i> Example: <pre>switch(config-router)# maximum-paths 4</pre>	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 64. The default is 8.
passive-interface default Example: <pre>switch(config-router)# passive-interface default</pre>	Suppresses routing updates on all interfaces. This command is overridden by the VRF or interface command mode configuration.

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
```

5.6.4 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip address** *ip-prefix/length*
4. **ip router ospf** *instance-tag area area-id* [**secondaries none**]
5. (Optional) **show ip ospf** *instance-tag interface interface-type slot/port*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i>	Enters interface configuration mode.

	Command	Purpose
	Example: <pre>switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	
Step 3	ip address <i>ip-prefix/length</i> Example: <pre>switch(config-if)# ip address 192.0.2.1/16</pre>	Assigns an IP address and subnet mask to this interface.
Step 4	ip router ospf <i>instance-tag area area-id</i> [secondaries none] Example: <pre>switch(config-if)# ip router ospf 201 area 0.0.0.15</pre>	Adds the interface to the OSPFv2 instance and area.
Step 5	show ip ospf <i>instance-tag interface</i> <i>interface-type slot/port</i> Example: <pre>switch(config-if)# show ip ospf 201 interface ethernet 1/2</pre>	(Optional) Displays OSPF information.
Step 6	copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

You can configure the following optional parameters for OSPFv2 in interface configuration mode:

Command	Purpose
ip ospf cost <i>number</i> Example: <pre>switch(config-if)# ip ospf cost 25</pre>	Configures the OSPFv2 cost metric for this interface. The default is to calculate cost metric, based on the reference bandwidth and interface bandwidth. The range is from 1 to 65535.
ip ospf dead-interval <i>seconds</i> Example: <pre>switch(config-if)# ip ospf dead-interval 50</pre>	Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
ip ospf hello-interval <i>seconds</i> Example: <pre>switch(config-if)# ip ospf hello-interval 25</pre>	Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
ip ospf mtu-ignore Example: <pre>switch(config-if)# ip ospf mtu-ignore</pre>	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.
[default no] ip ospf passive-interface Example:	Suppresses routing updates on the interface. This command overrides the router or VRF command

Command	Purpose
switch(config-if)# ip ospf passive-interface	mode configuration. The default option removes this interface mode command and reverts to the router or VRF configuration, if present.
ip ospf priority <i>number</i> Example: switch(config-if)# ip ospf priority 25	Configures the OSPFv2 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1.
ip ospf shutdown Example: switch(config-if)# ip ospf shutdown	Shuts down the OSPFv2 instance on this interface.

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.

5.6.5 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 keychain for this authentication configuration. See the *Inspur CN12900 Series INOS-CN Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **area** *area-id* **authentication** [**message-digest**]
4. **interface** *interface-type slot/port*
5. (Optional) **ip ospf authentication-key** [**0** | **3**] *key* or
ip ospf message-digest-key *key-id md5* [**0** | **3**] *key*
6. (Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.

	Command	Purpose
	Example: <pre>switch# configure terminal switch(config)#</pre>	
Step 2	<pre>router ospf instance-tag</pre> Example: <pre>switch(config)# router ospf 201 switch(config-router)#</pre>	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	<pre>area area-id authentication [message-digest]</pre> Example: <pre>switch(config-router)# area 0.0.0.10 authentication</pre>	Configures the authentication mode for an area.
Step 4	<pre>interface interface-type slot/port</pre> Example: <pre>switch(config-router)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 5	<pre>ip ospf authentication-key [0 3] key</pre> Example: <pre>switch(config-if)# ip ospf authentication-key 0 mypass</pre>	(Optional) Configures simple password authentication for this interface. Use this command if the authentication is not set to keychain or message-digest. 0 configures the password in cleartext. 3 configures the password as 3DES encrypted.
	<pre>ip ospf message-digest-key key-id md5 [0 3] key</pre> Example: <pre>switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass</pre>	(Optional) Configures message digest authentication for this interface. Use this command if the authentication is set to message-digest. The <i>key-id</i> range is from 1 to 255. The MD5 option 0 configures the password in cleartext and 3 configures the pass key as 3DES encrypted.
Step 6	<pre>show ip ospf instance-tag interface interface-type slot/port</pre> Example: <pre>switch(config-if)# show ip ospf 201 interface ethernet 1/2</pre>	(Optional) Displays OSPF information.
Step 7	<pre>copy running-config startup-config</pre> Example: <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

5.6.6 Configuring Authentication for an Interface

You can configure authentication 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 keychain for this authentication configuration. To configure OSPFv2 HMAC-SHA authentication, you must specify the HMAC-SHA algorithm to be used for the key. OSPFv2 will use the MD5 cryptographic algorithm if cryptographic authentication using keychain is configured without selecting a cryptographic-algorithm. See the *Inspur CN12900 Series INOS-CN Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip ospf authentication** [**message-digest**]
4. (Optional) **ip ospf authentication key-chain** *key-id*
5. (Optional) **ip ospf authentication-key** [**0 | 3 | 7**] *key*
6. (Optional) **ip ospf message-digest-key** *key-id md5* [**0 | 3 | 7**] *key*
7. (Optional) **show ip ospf instance-tag interface** *interface-type slot/port*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ip ospf authentication [message-digest] Example: switch(config-if)# ip ospf authentication	Enables interface authentication mode for OSPFv2 for either cleartext or message-digest type. Use this command to override area-based authentication for this interface. All neighbors must share this authentication type.
Step 4	ip ospf authentication key-chain <i>key-id</i> Example: switch(config-if)# ip ospf authentication key-chain Test1	(Optional) Configures interface authentication to use keychains for OSPFv2. See the <i>Inspur CN12900 Series INOS-CN Security Configuration Guide</i> for details on keychains.
Step 5	ip ospf authentication-key [0 3 7] <i>key</i> Example: switch(config-if)# ip ospf authentication-key 0 mypass	(Optional) Configures simple password authentication for this interface. Use this command if the authentication is not set to keychain or message-digest. The options are as follows: <ul style="list-style-type: none"> • 0—Configures the password in cleartext. • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Inspur type 7 encrypted.
Step 6	ip ospf message-digest-key <i>key-id md5</i> [0	(Optional) Configures message digest authentication

Command	Purpose
<pre> 3 7] key</pre> <p>Example:</p> <pre>switch(config-if)# ip ospf message-digest-key 21 md5 0 mypass</pre>	<p>for this interface. Use this command if the authentication is set to message-digest. The <i>key-id</i> range is from 1 to 255. The MD5 options are as follows:</p> <ul style="list-style-type: none"> • 0—Configures the password in cleartext. • 3—Configures the pass key as 3DES encrypted. • 7—Configures the key as Inspur type 7 encrypted.
<p>Step 7</p> <pre>show ip ospf instance-tag interface interface-type slot/port</pre> <p>Example:</p> <pre>switch(config-if)# show ip ospf 201 interface ethernet 1/2</pre>	<p>(Optional) Displays OSPF information.</p>
<p>Step 8</p> <pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config)# copy running-config startup-config</pre>	<p>(Optional) Saves this configuration change.</p>

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
```

This example shows how to configure OSPFv2 HMAC-SHA-1 and MD5 cryptographic authentication:

```
switch# configure terminal
switch(config)# key chain chain1
switch(config-keychain)# key 1
switch(config-keychain-key)# key-string 7 070724404206
switch(config-keychain-key)# accept-lifetime 01:01:01 Jan 01 2015 infinite
switch(config-keychain-key)# send-lifetime 01:01:01 Jan 01 2015 infinite
switch(config-keychain-key)# cryptographic-algorithm HMAC-SHA-1
switch(config-keychain-key)# exit
switch(config-keychain)# key 2
switch(config-keychain-key)# key-string 7
070e234f1f5b4a
switch(config-keychain-key)# accept-lifetime 10:51:01 Jul 24 2015 infinite
switch(config-keychain-key)# send-lifetime 10:51:01 Jul 24 2015 infinite
switch(config-keychain-key)# cryptographic-algorithm MD5
switch(config-keychain-key)# exit
switch(config-keychain)# exit

switch(config)# interface ethernet 1/1
switch(config-if)# ip router ospf 1 area 0.0.0.0
switch(config-if)# ip ospf authentication message-
```

```

digest
switch(config-if)# ip ospf authentication key-chain chain1

switch(config-if)# show key chain
chain1 Key-Chain chain1
  Key 1 -- text 7 "070724404206"
    cryptographic-algorithm HMAC-SHA-1
    accept lifetime UTC (01:01:01 Oct 01 2018)-(always valid) [active]
    send lifetime UTC (01:01:01 Oct 01 2018)-(always valid) [active]
  Key 2 -- text 7 "070e234f1f5b4a"
    cryptographic-algorithm MD5
    accept lifetime UTC (10:51:00 Oct 01 2018)-(always valid) [active]
    send lifetime UTC (10:51:00 Oct 01 2018)-(always valid) [active]

switch(config-if)# show ip ospf interface ethernet 1/1
Ethernet1/1 is up, line protocol is
  up IP address 11.11.11.1/24
  Process ID 1 VRF default, area 0.0.0.3
  Enabled by interface configuration
  State BDR, Network type BROADCAST, cost 40
  Index 6, Transmit delay 1 sec, Router Priority 1
  Designated Router ID: 33.33.33.33, address: 11.11.11.3
  Backup Designated Router ID: 1.1.1.1, address:
  11.11.11.1 2 Neighbors, flooding to 2, adjacent with 2
  Timer intervals: Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello timer due in 00:00:08
  Message-digest authentication, using keychain key1 (ready)
    Sending SA: Key id 2, Algorithm MD5
  Number of opaque link LSAs: 0, checksum sum 0

```

5.7 Configuring Advanced OSPFv2

Configure OSPFv2 after you have designed your OSPFv2 network.

5.7.1 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 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area area-id filter-list route-map map-name {in out} Example: switch(config-router)# area 0.0.0.10 filter-list route-map FilterLSAs in	Filters incoming or outgoing Network Summary (type 3) LSAs on an ABR.
Step 4	show ip ospf policy statistics area id filter-list {in out} Example: switch(config-if)# show ip ospf policy statistics area 0.0.0.10 filter-list in	(Optional) Displays OSPF policy information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

5.7.2 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf instance-tag**
3. **area area-id stub**

4. (Optional) **area** *area-id* **default-cost** *cost*
5. (Optional) **show ip ospf** *instance-tag*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area <i>area-id</i> stub Example: switch(config-router)# area 0.0.0.10 stub	Creates this area as a stub area.
Step 4	area <i>area-id</i> default-cost <i>cost</i> Example: switch(config-router)# area 0.0.0.10 default-cost 25	(Optional) 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	show ip ospf <i>instance-tag</i> Example: switch(config-if)# show ip ospf 201	(Optional) Displays OSPF information.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

5.7.3 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
area <i>area-id</i> stub no-summary	Creates this area as a totally stubby area.

Example:

```
switch(config-router)# area 20 stub
no-summary
```

5.7.4 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: <pre>switch(config)# router ospf 201 switch(config-router)#</pre>	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area <i>area-id</i> nssa [no-redistribution] [default-information-originate	Creates this area as an NSSA.

	Command	Purpose
	<pre>[route-map map-name]] [no-summary] [translate type7 {always never} [suppress-fa]] Example: switch(config-router)# area 0.0.0.10 nssa</pre>	
Step 4	<pre>area area-id default-cost cost Example: switch(config-router)# area 0.0.0.10 default-cost 25</pre>	(Optional) Sets the cost metric for the default summary route sent into this NSSA.
Step 5	<pre>show ip ospf instance-tag Example: switch(config-if)# show ip ospf 201</pre>	(Optional) Displays OSPF information.
Step 6	<pre>copy running-config startup-config Example: switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```


5.7.5 Configuring Multi-Area Adjacency

You can add more than one area to an existing OSPFv2 interface. The additional logical interfaces support multi-area adjacency.

BEFORE YOU BEGIN

You must enable OSPFv2.

Ensure that you have configured a primary area for the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip router ospf** [*instance-tag*] **multi-area** *area-id*
4. (Optional) **show ip ospf** *instance-tag* **interface** *interface-type slot/port*

(Optional) copy running-config startup-config DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# <code>interface ethernet 1/2</code> switch(config-if)#	Enters interface configuration mode.
Step 3	ip router ospf [<i>instance-tag</i>] multi-area <i>area-id</i> Example: switch(config-if)# <code>ip router ospf 201</code> <code>multi-area 3</code>	Adds the interface to another area. Note The <i>instance-tag</i> argument is optional. If you do not specify an instance, the multi-area configuration is applied to the same instance that is configured for the primary area on that interface.
Step 4	show ip ospf <i>instance-tag</i> interface <i>interface-type slot/port</i> Example: switch(config-if)# <code>show ip ospf 201</code> <code>interface ethernet 1/2</code>	(Optional) Displays OSPFv2 information.
Step 5	copy running-config startup-config Example: switch(config)# <code>copy running-config</code> <code>startup-config</code>	(Optional) Saves this configuration change.

This example shows how to add a second area to an OSPFv2 interface:

```
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) # ip router ospf 201 multi-area 20
switch(config-if) # copy running-config startup-config
```

5.7.6 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:

- 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 OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf *instance-tag***
3. **area *area-id* virtual-link *router-id***
4. (Optional) **show ip ospf virtual-link [brief]**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area <i>area-id</i> virtual-link <i>router-id</i> Example: switch(config-router)# area 0.0.0.10 virtual-link 10.1.2.3 switch(config-router-vlink)#	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	show ip ospf virtual-link [brief] Example: switch(config-router-vlink)# show ip ospf virtual-link	(Optional) Displays OSPF virtual link information.
Step 5	copy running-config startup-config Example: switch(config-router-vlink)# copy	(Optional) Saves this configuration change.

Command	Purpose
running-config startup-config	

You can configure the following optional commands in virtual link configuration mode:

Command	Purpose
authentication [key-chain <i>key-id</i> message-digest null] Example: <pre>switch(config-router-vlink)# authentication message-digest</pre>	(Optional) Overrides area-based authentication for this virtual link.
authentication-key [0 3] <i>key</i> Example: <pre>switch(config-router-vlink)# authentication-key 0 mypass</pre>	(Optional) Configures a simple password for this virtual link. Use this command if the authentication is not set to keychain or message-digest. 0 configures the password in cleartext. 3 configures the password as 3DES encrypted.
dead-interval <i>seconds</i> Example: <pre>switch(config-router-vlink)# dead-interval 50</pre>	(Optional) Configures the OSPFv2 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
hello-interval <i>seconds</i> Example: <pre>switch(config-router-vlink)# hello-interval 25</pre>	(Optional) Configures the OSPFv2 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
message-digest-key <i>key-id</i> md5 [0 3] <i>key</i> Example: <pre>switch(config-router-vlink)# message-digest-key 21 md5 0 mypass</pre>	(Optional) 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.
retransmit-interval <i>seconds</i> Example: <pre>switch(config-router-vlink)# retransmit-interval 50</pre>	(Optional) Configures the OSPFv2 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
transmit-delay <i>seconds</i> Example: <pre>switch(config-router-vlink)# transmit-delay 2</pre>	(Optional) Configures the OSPFv2 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.

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
```

5.7.7 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 OSPF.

Create the necessary route maps used for redistribution.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	redistribute {<i>bgp id</i> <i>direct</i> <i>eigrp id</i> <i>isis id</i> <i>ospf id</i> <i>rip id</i> <i>static</i>} route-map <i>map-name</i> Example: switch(config-router)# redistribute bgp route-map FilterExternalBGP	Redistributes the selected protocol into OSPF through the configured route map. Note If you redistribute static routes, Inspur INOS-CN also redistributes the default static route.
Step 4	default-information originate [<i>always</i>] [<i>route-map map-name</i>] Example: switch(config-router)# default-information-originate route-map DefaultRouteFilter	Creates a default route into this OSPF domain if the default route exists in the RIB. Use the following optional keywords: <ul style="list-style-type: none">• always —Always generates the default route of 0.0.0. even if the route does not exist in the RIB.

Command	Purpose
	<ul style="list-style-type: none"> • route-map—Generates the default route if the route map returns true. <p>Note This command ignores match statements in the route map.</p>
<p>Step 5 <code>default-metric cost</code></p> <p>Example:</p> <pre>switch(config-router)# default-metric 25</pre>	<p>Sets the cost metric for the redistributed routes. This command does not apply to directly connected routes. Use a route map to set the default metric for directly connected routes.</p> <p>Note If you do not specify a metric, OSPF uses a default value of 20 when redistributing routes from all protocols except Border Gateway Protocol (BGP) routes, which use a metric of 1.</p>
<p>Step 6 <code>copy running-config startup-config</code></p> <p>Example:</p> <pre>switch(config-router)# copy running-config startup-config</pre>	<p>(Optional) Saves this configuration change.</p>

This example shows how to redistribute the Border Gateway Protocol (BGP) into OSPF:

```
switch# configure terminal
switch(config)# router ospf 201
switch(config-router)# redistribute bgp route-map FilterExternalBGP
switch(config-router)# copy running-config startup-config
```

5.7.8 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 OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **redistribute** {**bgp** *id* | **direct** | **eigrp** *id* | **isis** *id* | **ospf** *id* | **rip** *id* | **static**} **route-map** *map-name*

4. **redistribute maximum-prefix** *max* [*threshold*] [**warning-only** | **withdraw** [*num-retries* *timeout*]]
5. (Optional) **show running-config ospf**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	redistribute { <i>bgp id</i> <i>direct</i> <i>eigrp id</i> <i>isis id</i> <i>ospf id</i> <i>rip id</i> <i>static</i> } route-map <i>map-name</i> Example: switch(config-router)# redistribute bgp route-map FilterExternalBGP	Redistributes the selected protocol into OSPF through the configured route map.
Step 4	redistribute maximum-prefix <i>max</i> [<i>threshold</i>] [warning-only withdraw [<i>num-retries</i> <i>timeout</i>]] Example: switch(config-router)# redistribute maximum-prefix 1000 75 warning-only	Specifies a maximum number of prefixes that OSPFv2 distributes. The range is from 1 to 65535. Optionally specifies the following: <ul style="list-style-type: none"> • threshold—Percentage of maximum prefixes that trigger 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 <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> is 60 to 600 seconds. The default is 300 seconds. Use the clear ip ospf redistribution command if all routes are withdrawn.
Step 5	show running-config ospf Example: switch(config-router)# show running-config ospf	(Optional) Displays the OSPFv2 configuration.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

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

5.7.9 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

Ensure that you have enabled OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf *instance-tag***
3. **area *area-id* range *ip-prefix/length* [no-advertise] [cost *cost*] or**
4. **summary-address *ip-prefix/length* [no-advertise | tag *tag-id*]**
5. (Optional) **show ip ospf summary-address**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	area <i>area-id</i> range <i>ip-prefix/length</i> [no-advertise] [cost <i>cost</i>] Example: switch(config-router)# area 0.0.0.10 range 10.3.0.0/16	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 <i>cost</i> range is from 0 to 16777215.
Step 4	summary-address <i>ip-prefix/length</i> [no-advertise tag <i>tag</i>] Example: switch(config-router)# summary-address 10.5.0.0/16 tag 2	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 5	show ip ospf summary-address Example: switch(config-router)# show ip ospf summary-address	(Optional) Displays information about OSPF summary addresses.
Step 6	copy running-config startup-config	(Optional) Saves this configuration change.

Command	Purpose
Example: switch(config-router)# copy running-config startup-config	

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)# no discard-route internal
switch(config-router)# copy running-config startup-config
```

5.7.10 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 OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf instance-tag**
3. **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) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	max-metric router-lsa [external-lsa [max-metric-value]] [include-stub]	Configures OSPFv2 stub route advertisements.

Command	Purpose
<pre>[on-startup {seconds wait-for bgp tag}] [summary-lsa [max-metric-value]]</pre> <p>Example:</p> <pre>switch(config-router)# max-metric router-lsa</pre>	
<p>Step 4</p> <pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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)# 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)# no discard-route internal
switch(config-router)# copy running-config startup-config
```

5.7.11 xConfiguring the Administrative Distance of Routes

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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf** *instance-tag*
3. **[no] table-map** *map-name*
4. **exit**
5. **route-map** *map-name* [**permit** | **deny**] [*seq*]
6. **match route-type** *route-type*
7. **match ip route-source prefix-list** *name*
8. **match ip address prefix-list** *name*
9. **set distance** *value*
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	[no] table-map map-name Example: switch(config-router)# table-map foo	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.
Step 4	exit Example: switch(config-router)# exit switch(config)#	Exits router configuration mode.
Step 5	route-map map-name [permit deny] [seq] Example: switch(config)# route-map foo permit 10 switch(config-route-map)#	Creates a route map or enters route-map configuration mode for an existing route map. Use <i>seq</i> 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	match route-type route-type Example: switch(config-route-map)# match route-type external	Matches against one of the following route types: <ul style="list-style-type: none"> 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	match ip route-source prefix-list name Example: switch(config-route-map)# match ip route-source prefix-list p1	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	match ip address prefix-list name Example: switch(config-route-map)# match ip address prefix-list p1	Matches against one or more IPv4 prefix lists. Use the ip prefix-list command to create the prefix list.

	Command	Purpose
Step 9	set distance value Example: switch(config-route-map)# set distance 150	Sets the administrative distance of routes for OSPFv2. The range is from 1 to 255.
Step 10	copy running-config startup-config Example: switch(config-route-map)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

5.7.12 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 OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf *instance-tag***
3. **timers lsa-arrival *msec***
4. **timers lsa-group-pacing *seconds***
5. **timers throttle lsa *start-time hold-interval max-time***
6. **timers throttle spf *delay-time hold-time max-time***
7. **interface *type slot/port***
8. **ip ospf hello-interval *seconds***
9. **ip ospf dead-interval *seconds***
10. **ip ospf retransmit-interval *seconds***
11. **ip ospf transmit-delay *seconds***
12. (Optional) **show ip ospf**
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	timers lsa-arrival <i>msec</i> Example: switch(config-router)# timers lsa-arrival 2000	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	timers lsa-group-pacing <i>seconds</i> Example: switch(config-router)# timers lsa-group-pacing 200	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 10 seconds.
Step 5	timers throttle lsa <i>start-time hold-interval max-time</i> Example: switch(config-router)# timers throttle lsa 3000	Sets the rate limit in milliseconds for generating LSAs with the following timers: <i>start-time</i> —The range is from 0 to 5000 milliseconds. The default value is 0 milliseconds. <i>hold-interval</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. <i>max-time</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	timers throttle spf <i>delay-time hold-time max-wait</i> Example: switch(config-router)# timers throttle	Sets the SPF best-path schedule initial delay time, minimum hold time, and maximum wait time in milliseconds between SPF best-path calculations. The range is from 1 to 600000 milliseconds. The default

	Command	Purpose
	<code>spf 3000 2000 5000</code>	and 5000-ms wait time.
Step 7	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 8	ip ospf hello-interval <i>seconds</i> Example: switch(config-if)# ip ospf hello-interval 30	Sets the hello interval for this interface. The range is from 1 to 65535. The default is 10.
Step 9	ip ospf dead-interval <i>seconds</i> Example: switch(config-if)# ip ospf dead-interval 30	Sets the dead interval for this interface. The range is from 1 to 65535.
Step 10	ip ospf retransmit-interval <i>seconds</i> Example: switch(config-if)# ip ospf retransmit-interval 30	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	ip ospf transmit-delay <i>seconds</i> Example: switch(config-if)# ip ospf transmit-delay 600 switch(config-if)#	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	show ip ospf Example: switch(config-if)# show ip ospf	(Optional) Displays information about OSPF.
Step 13	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

5.7.13 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf *instance-tag***
3. **graceful-restart**
4. (Optional) **graceful-restart grace-period *seconds***
5. (Optional) **graceful-restart helper-disable**
6. (Optional) **graceful-restart planned-only**
7. (Optional) **show ip ospf *instance-tag***
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	graceful-restart Example: switch(config-router)# graceful-restart	Enables a graceful restart. A graceful restart is enabled by default.
Step 4	graceful-restart grace-period <i>seconds</i> Example: switch(config-router)# graceful-restart grace-period 120	(Optional) Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	graceful-restart helper-disable Example: switch(config-router)# graceful-restart helper-disable	(Optional) Disables helper mode. This feature is enabled by default.
Step 6	graceful-restart planned-only Example: switch(config-router)# graceful-restart planned-only	(Optional) Configures a graceful restart for planned restarts only.

Command	Purpose
<pre>show ip ospf instance-tag</pre> <p>Example:</p> <pre>switch(config-if)# show ip ospf 201</pre>	(Optional) Displays OSPF information.
<p>Step 8</p> <pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

5.7.14 Restarting an OSPFv2 Instance

You can restart an OSPFv2 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
<pre>restart ospf instance-tag</pre> <p>Example:</p> <pre>switch(config)# restart ospf 201</pre>	Restarts the OSPFv2 instance and removes all neighbors.

5.7.15 Configuring OSPFv2 with Virtualization

You can configure multiple OSPFv2 instances. You can also create multiple VRFs and use the same or multiple OSPFv2 instances in each VRF. You assign an OSPFv2 interface to a VRF.

BEFORE YOU BEGIN

Ensure that you have enabled OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf_name*
3. **router ospf** *instance-tag*
4. **vrf** *vrf-name*
5. (Optional) **maximum-paths** *paths*
6. **interface** *interface-type slot/port*
7. **vrf member** *vrf-name*
8. **ip-address** *ip-prefix/length*
9. **router ospf** *instance-tag area area-id*
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context vrf-name Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	router ospf instance-tag Example: switch(config-vrf)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 4	vrf vrf-name Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters VRF configuration mode.
Step 5	maximum-paths paths Example: switch(config-router-vrf)# maximum-paths 4	(Optional) Configures the maximum number of equal OSPFv2 paths to a destination in the route table for this VRF. This feature is used for load balancing.
Step 6	interface interface-type slot/port Example: switch(config-router-vrf)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 7	vrf member vrf-name Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 8	ip address ip-prefix/length Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 9	ip router ospf instance-tag area area-id Example: switch(config-if)# ip router ospf 201 area 0	Assigns this interface to the OSPFv2 instance and area configured.
Step 10	copy running-config startup-config	(Optional) Saves this configuration change.

Command	Purpose
Example: <pre>switch(config)# copy running-config startup-config</pre>	

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)# 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
```

5.8 Verifying the OSPFv2 Configuration

To display the OSPFv2 configuration, perform one of the following tasks:

Command	Purpose
show ip ospf [<i>instance-tag</i>] [<i>vrf vrf-name</i>]	Displays information about one or more OSPF routing instances. The output includes the following area-level counts: <ul style="list-style-type: none"> • 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 [<i>vrf {vrf-name all default management}</i>]	Displays the OSPFv2 border router configuration.
show ip ospf database [<i>vrf {vrf-name all default management}</i>]	Displays the OSPFv2 link-state database summary.
show ip ospf interface <i>number</i> [<i>vrf {vrf-name all default management}</i>]	Displays OSPFv2-related interface information.
show ip ospf lsa-content-changed-list <i>neighbor-id interface-type number</i> [<i>vrf {vrf-name all default management}</i>]	Displays the OSPFv2 LSAs that have changed.
show ip ospf neighbors [<i>neighbor-id</i>] [<i>detail</i>] [<i>interface-type number</i>] [<i>vrf {vrf-name all default management}</i>]	Displays the list of OSPFv2 neighbors.

Command	Purpose
[summary]	
show ip ospf request-list <i>neighbor-id</i> <i>interface-type number</i> [vrf {vrf-name all default management}]	Displays the list of OSPFv2 link-state requests.
show ip ospf retransmission-list <i>neighbor-id interface-type number</i> [vrf {vrf-name all default management}]	Displays the list of OSPFv2 link-state retransmissions.
show ip ospf route [<i>ospf-route</i>] [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 the VRF-based OSPFv2 configuration.
show running-configuration ospf	Displays the current running OSPFv2 configuration.

5.9 Monitoring OSPFv2

To display OSPFv2 statistics, use the following commands:

Command	Purpose
show ip ospf policy statistics area <i>area-id</i> 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 {vrf-name all default management}]	Displays the OSPFv2 event counters.
show ip ospf traffic [<i>interface-type number</i>] [vrf {vrf-name all default management}]	Displays the OSPFv2 packet counters.

5.10 Configuration Examples for OSPFv2

The following example shows how to configure 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
```

5.10.1 OSPF RFC Compatibility Mode Example

The following example shows how to configure OSPF to be compatible with routers that comply with RFC 1583:

```
switch# configure terminal
switch(config)# feature ospf
switch(config)# router ospf Test1
switch(config-router)# rfc1583compatibility
switch(config-router)# vrf A
switch(config-router-vrf)# rfc1583compatibility
```

5.10.2 Related Documents

Related Topic	Document Title
Keychains	<i>Inspur CN12900 Series INOS-CN Security Configuration Guide</i>

CHAPTER 6 Configuring OSPFv3

This chapter describes how to configure Open Shortest Path First version 3 (OSPFv3) for IPv6 networks on the Inspur INOS-CN device.

6.1 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.

6.1.1 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, Inspur INOS-CN does not support RFC 6506 and provides only partial support for RFC 4552.
- OSPFv3 redefines LSA types.

6.1.2 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, 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.

6.1.3 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:

- Neighbor ID—The router ID of the neighbor router.
- 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 (see the “Designated Routers” section).
- 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, the neighbor is moved to the down state and is no longer considered adjacent.

6.1.4 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. For more information, see the “Designated Routers” section.

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 (see the “Link-State Database” section). 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.

6.1.5 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 selects a backup designated router (*BDR*). If the DR fails, OSPFv3 uses

the BDR.

Network types are as follows:

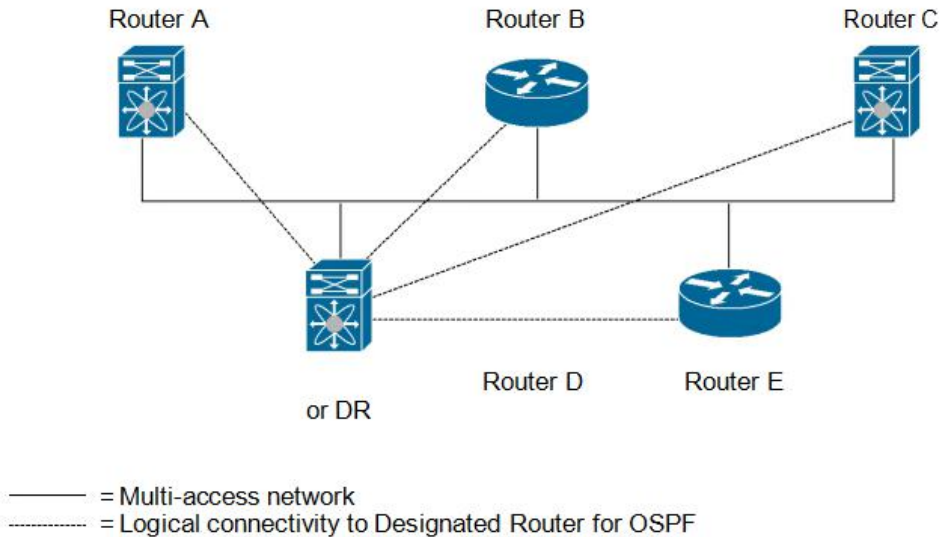
- Point-to-point—A network that exists only between two routers. All neighbors on a point-to-point network establish adjacency and there is no DR.
- Broadcast—A network with multiple routers that can communicate over a shared medium that allows broadcast traffic, such as Ethernet. OSPFv3 routers establish 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. Figure 6-1 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 6-1 DR in Multi-Access Network

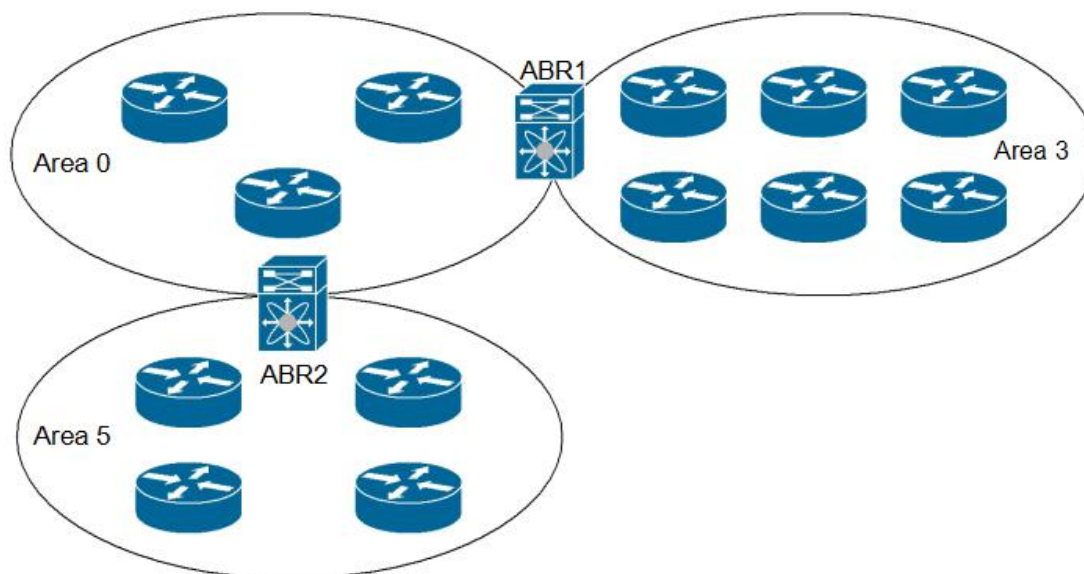


6.1.6 Areas

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-CN 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 (see Figure 6-2).

Figure 6-2 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 Figure 6-2, 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.

6.1.7 Link-State Advertisement

OSPFv3 uses link-state advertisements (LSAs) to build its routing table.

LSA Types

Table 6-1 shows the LSA types supported by Inspur INOS-CN.

Table 6-1 LSA Types

Type	Name	Description
1	Router LSA	LSA sent by every router. 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.

Type	Name	Description
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.
11	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-CN 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 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-CN supports the LSA grouping feature to prevent all LSAs from refreshing at the same time.

6.1.8 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.

6.1.9 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 Routing Information Base (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.

6.1.10 Address Family Support

Inspur INOS-CN 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.

6.1.11 Authentication

You can configure authentication on OSPFv3 messages to prevent unauthorized or invalid routing updates in your network.

RFC 4552 provides authentication to OSPFv3 using an IPv6 authentication header (AH) or encapsulating security payload (ESP) extension header. Inspur INOS-CN partially supports RFC 4552 by using the IPv6 AH header to authenticate OSPFv3 packets.

Inspur INOS-CN supports the IP security (IPSec) authentication method and the message digest 5 (MD5) or secure hash algorithm 1 (SHA1) algorithm to authenticate OSPFv3 packets. OSPFv3 IPSec authentication supports only static keys.

You can configure IPSec authentication for an OSPFv3 process, area, or interface.

6.1.12 Advanced Features

Inspur INOS-CN supports advanced OSPFv3 features that enhance the usability and scalability of OSPFv3 in the network.

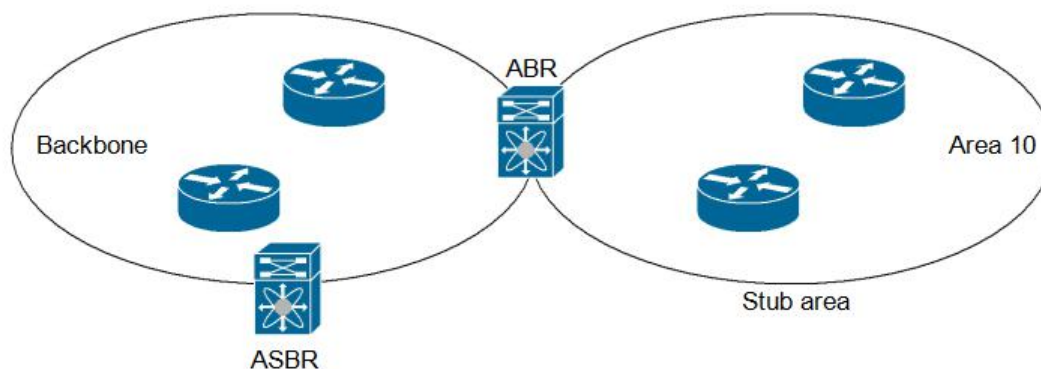
Stub Area

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.

Figure 6-3 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 6-3 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.

Not-So-Stubby Area

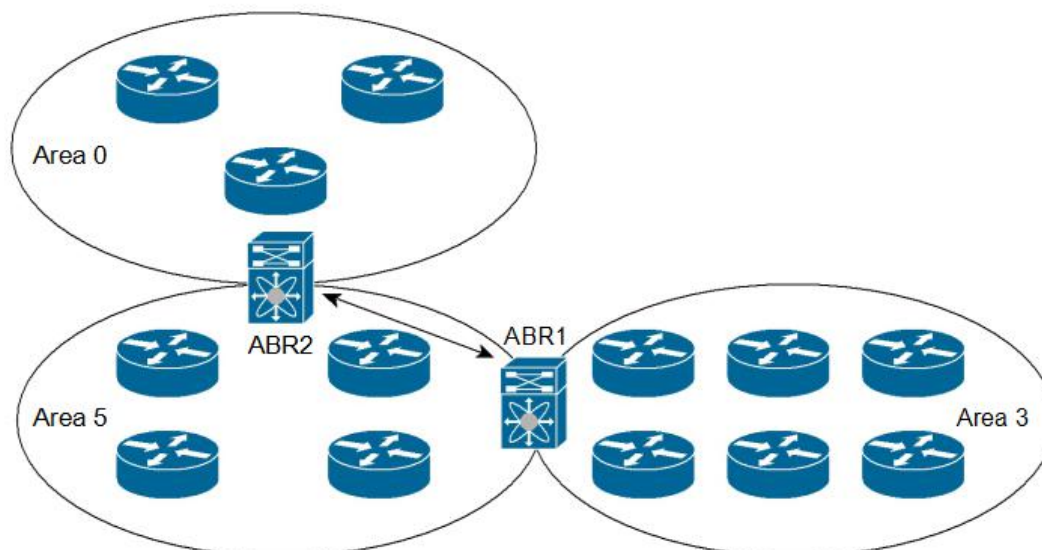
A Not-So-Stubby Area (NSSA) is similar to the stub area, except that an NSSA allows you to import autonomous system external routes within an NSSA using redistribution. The NSSA ASBR redistributes these routes and generates type-7 LSAs that it floods throughout the NSSA. You can optionally configure 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.

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. Figure 6-4 shows a virtual link that connects Area 3 to the backbone area through Area 5.

Figure 6-4 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

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 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 OSPFv3 autonomous system.

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-CN automatically configures a discard route for the summary address to prevent routing black holes and route loops.

High Availability and Graceful Restart

Inspur INOS-CN 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
 - 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-sup** command

Multiple OSPFv3 Instances

Inspur INOS-CN 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. For the number of supported OSPFv3 instances, see the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

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-CN allows only one OSPFv3 instance on an interface.

SPF Optimization

Inspur INOS-CN 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-CN 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) for IPv6. 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information.

Virtualization Support

Inspur INOS-CN supports multiple process instances for OSPFv3. Each OSPFv3 instance can support multiple virtual routing and forwarding (VRF) instances, up to the system limit. For the number of supported OSPFv3 instances, see the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

6.2 Licensing Requirements for OSPFv3

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	OSPFv3 requires an Essentials license package

6.3 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 (see the “Enabling OSPFv3” section).
- You are familiar with IPv6 addressing and basic configuration. See Chapter 3, “Configuring IPv6” for information on IPv6 routing and addressing.

6.4 Guidelines and Limitations for OSPFv3

OSPFv3 has the following configuration guidelines and limitations:

- Inspur INOS-CN displays areas in dotted decimal notation regardless of whether you enter the area in decimal or dotted decimal notation.
- 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
```

- 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 you observe a timeout on the querying SNMP agent while polling OSPF MIB objects, increase the timeout

- value on the polling SNMP agent.
- The following guidelines and limitations apply to the administrative distance feature:
 - 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-CN OSPF is different from that in Inspur INOS OSPF.
- If you configure the **delay restore seconds** command in vPC configuration mode and if the VLANs on the multichassis EtherChannel trunk (MCT) are announced by OSPFv2 or OSPFv3 using switch virtual interfaces (SVIs), those SVIs are announced with MAX_LINK_COST on the vPC secondary node for the duration of the configured time. As a result, all route or host programming completes after the vPC synchronization operation (on a peer reload of the secondary vPC node) before attracting traffic. This behavior allows for minimal packet loss for any north-to-south traffic.

6.5 Default Settings

Table 6-2 lists the default settings for OSPFv3 parameters.

Table 6-2 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	200 milliseconds
SPF calculation minimum hold time	1000 milliseconds
SPF calculation maximum wait time	5000 milliseconds

6.6 Configuring Basic OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

6.6.1 Enabling OSPFv3

You must enable OSPFv3 before you can configure OSPFv3.

SUMMARY STEPS

1. **configure terminal**
2. **feature ospfv3**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature ospfv3 Example: switch(config)# feature ospfv3	Enables OSPFv3.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To disable the OSPFv3 feature and remove all associated configuration, use the following command in global configuration mode:

Command	Purpose
no feature ospfv3 Example: switch(config)# no feature ospfv3	Disables the OSPFv3 feature and removes all associated configuration.

6.6.2 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. For more information, see the “Administrative Distance” section.
- Log adjacency changes—Creates a system message whenever an OSPFv3 neighbor changes its state.
- Name lookup—Translates OSPF router IDs to hostnames, 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 (see the “Enabling OSPFv3” section).

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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. (Optional) **router-id ip-address**
4. (Optional) **show ipv6 ospfv3 instance-tag**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	router-id ip-address Example: switch(config-router)# router-id 192.0.2.1	(Optional) 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.
Step 4	show ipv6 ospfv3 instance-tag Example: switch(config-router)# show ipv6 ospfv3 201	(Optional) Displays OSPFv3 information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To remove the OSPFv3 instance and all associated configuration, use the following command in configuration mode:

Command	Purpose
no router ospfv3 <i>instance-tag</i> Example: switch(config)# no router ospfv3 201	Deletes the OSPFv3 instance and all associated configuration.

You can configure the following optional parameters for OSPFv3 in router configuration mode:

Command	Purpose
log-adjacency-changes [detail] Example: switch(config-router)# log-adjacency-changes	Generates a system message whenever a neighbor changes state.
passive-interface default Example: switch(config-router)# passive-interface default	Suppresses routing updates on all interfaces. This command is overridden by the VRF or interface command mode configuration.

You can configure the following optional parameters for OSPFv3 in address family configuration mode:

Command	Purpose
distance <i>number</i> Example: switch(config-router-af)# distance 25	Configures the administrative distance for this OSPFv3 instance. The range is from 1 to 255. The default is 110.
maximum-paths <i>paths</i> Example: switch(config-router-af)# maximum-paths 4	Configures the maximum number of equal OSPFv3 paths to a destination in the route table. This command is used for load balancing. The range is from 1 to 64. The default is 8.

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
```

6.6.3 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 (see the “Neighbors” section). 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.

SUMMARY STEPS

1. **configure terminal**

2. **interface** *interface-type slot/port*
3. **ipv6 address** *ipv6-prefix/length*
4. **ipv6 router ospfv3** *instance-tag area area-id* [**secondaries none**]
5. (Optional) **show ipv6 ospfv3** *instance-tag interface interface-type slot/port*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ipv6 address <i>ipv6-prefix/length</i> Example: switch(config-if)# ipv6 address 2001:0DB8::1/48	Assigns an IPv6 address to this interface.
Step 4	ipv6 router ospfv3 <i>instance-tag area area-id</i> [secondaries none] Example: switch(config-if)# ipv6 router ospfv3 201 area 0	Adds the interface to the OSPFv3 instance and area.
Step 5	show ipv6 ospfv3 <i>instance-tag interface interface-type slot/port</i> Example: switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2	(Optional) Displays OSPFv3 information.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional parameters for OSPFv3 in interface configuration mode:

Command	Purpose
ospfv3 cost <i>number</i> Example: switch(config-if)# ospfv3 cost 25	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.
ospfv3 dead-interval <i>seconds</i> Example:	Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.

Command	Purpose
switch(config-if)# ospfv3 dead-interval 50	
ospfv3 hello-interval <i>seconds</i> Example: switch(config-if)# ospfv3 hello-interval 25	Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.
ospfv3 instance <i>instance</i> Example: switch(config-if)# ospfv3 instance 25	Configures the OSPFv3 instance ID. The range is from 0 to 255. The default is 0. The instance ID is link-local in scope.
ospfv3 mtu-ignore Example: 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.
ospfv3 network (broadcast point-point) Example: switch(config-if)# ospfv3 network broadcast	Sets the OSPFv3 network type.
[default no] ospfv3 passive-interface Example: switch(config-if)# ospfv3 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.
ospfv3 priority <i>number</i> Example: switch(config-if)# ospfv3 priority 25	Configures the OSPFv3 priority, used to determine the DR for an area. The range is from 0 to 255. The default is 1.
ospfv3 shutdown Example: switch(config-if)# ospfv3 shutdown	Shuts down the OSPFv3 instance on this interface.

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
```

6.6.4 Configuring OSPFv3 IPsec Authentication

You can configure OSPFv3 IP security (IPsec) authentication for a process, an area, and/or an interface.

The authentication configuration is inherited from process to area to interface level. If authentication is configured at all three levels, the interface configuration takes precedence over the process and area configurations.

BEFORE YOU BEGIN

Ensure that you have enabled OSPFv3.

Ensure that you have enabled the Internet messaging program (IMP) using the **feature imp** command.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. **exit**
4. **authentication ipsec spi spi auth [0 | 3 | 7] key** or **area area authentication ipsec spi spi auth [0 | 3 | 7] key** or **interface interface-type slot/port ospfv3 authentication ipsec spi spi auth [0 | 3 | 7] key**
5. (Optional) **show ospfv3 process**
6. (Optional) **show ospfv3 interface interface-type slot/port**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: <pre>switch(config)# router ospfv3 100 switch(config-router)#</pre>	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	exit Example: <pre>switch(config-router)# exit switch(config)#</pre>	Exits OSPFv3 router configuration mode.
Step 4	authentication ipsec spi spi auth [0 3 7] key Example: <pre>switch(config)# authentication ipsec spi 475 md5 111111111111111111112222222222222222</pre>	Configures OSPFv3 IPsec authentication at the process (or VRF) level. The <i>spi</i> argument specifies the security parameter index (SPI). The range is from 256 to 4294967295. The <i>auth</i> argument specifies the type of authentication. The supported values are md5 or sha1. 0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Inspur type 7 encrypted. If the cleartext option (0) is used, the <i>key</i> argument must be 32 characters long for md5 or 40 characters long for sha1.
	area area authentication ipsec spi spi auth [0 3 7] key Example: <pre>switch(config)# area 0 authentication ipsec spi 475 md5 111111111111111111112222222222222222</pre>	Configures OSPFv3 IPsec authentication at the area level. The <i>spi</i> argument specifies the security parameter index (SPI). The range is from 256 to 4294967295. The <i>auth</i> argument specifies the type of authentication. The supported values are md5 or sha1.

Command	Purpose
	<p>0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Inspur type 7 encrypted.</p> <p>If the cleartext option (0) is used, the <i>key</i> argument must be 32 characters long for md5 or 40 characters long for sha1.</p> <p>Note Use the area area authentication disable command to disable OSPFv3 IPsec authentication at the area level.</p>
<pre>interface interface-type slot/port ospfv3 authentication ipsec spi spi auth [0 3 7] key Example: switch(config)# interface ethernet 1/1 switch(config-if)# ospfv3 authentication ipsec spi 475 md5 1111111111111111112222222222222222</pre>	<p>Configures OSPFv3 IPsec authentication for the specified interface.</p> <p>The <i>spi</i> argument specifies the security parameter index (SPI). The range is from 256 to 4294967295.</p> <p>The <i>auth</i> argument specifies the type of authentication.</p> <p>The supported values are md5 or sha1.</p> <p>0 configures the password in cleartext. 3 configures the pass key as 3DES encrypted. 7 configures the key as Inspur type 7 encrypted.</p> <p>If the cleartext option (0) is used, the <i>key</i> argument must be 32 characters long for md5 or 40 characters long for sha1.</p> <p>Note Use the ospfv3 authentication disable command to disable OSPFv3 IPsec authentication for the specified interface.</p>
<p>Step 5 <code>show ospfv3 process</code></p> <p>Example:</p> <pre>switch(config)# show ospfv3 100</pre>	<p>(Optional) Displays the OSPFv3 authentication configuration at the process level.</p>
<p>Step 6 <code>show ospfv3 interface interface-type slot/port</code></p> <p>Example:</p> <pre>switch(config)# show ospfv3 interface ethernet 1/1</pre>	<p>(Optional) Displays the OSPFv3 authentication configuration at the interface level.</p>
<p>Step 7 <code>copy running-config startup-config</code></p> <p>Example:</p> <pre>switch(config)# copy running-config startup-config</pre>	<p>(Optional) Saves this configuration change.</p>

6.7 Configuring Advanced OSPFv3

Configure OSPFv3 after you have designed your OSPFv3 network.

6.7.1 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. **address-family ipv6 unicast**
4. **area area-id filter-list route-map map-name {in | out}**
5. (Optional) **show ipv6 ospfv3 policy statistics area id filter-list {in | out}**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	area area-id filter-list route-map map-name {in out} Example: switch(config-router-af)# area 0.0.0.10 filter-list route-map FilterLSAs in	Filters incoming or outgoing Inter-Area Prefix (type 3) LSAs on an ABR.
Step 5	show ipv6 ospfv3 policy statistics area id filter-list {in out} Example: switch(config-if)# show ipv6 ospfv3 policy statistics area 0.0.0.10	(Optional) Displays OSPFv3 policy information.

	Command	Purpose
	<code>filter-list in</code>	
Step 6	copy running-config startup-config Example: <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

6.7.2 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. **area area-id stub**
4. (Optional) **address-family ipv6 unicast**
5. (Optional) **area area-id default-cost cost**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: <pre>switch(config)# router ospfv3 201 switch(config-router)#</pre>	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	area area-id stub Example: <pre>switch(config-router)# area 0.0.0.10</pre>	Creates this area as a stub area.

	Command	Purpose
	stub	
Step 4	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	(Optional) Enters IPv6 unicast address family mode.
Step 5	area area-id default-cost cost Example: switch(config-router-af)# area 0.0.0.10 default-cost 25	(Optional) Sets the cost metric for the default summary route sent into this stub area. The range is from 0 to 16777215.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example 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
```

6.7.3 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
area area-id stub no-summary Example: switch(config-router)# area 20 stub no-summary	Creates this area as a totally stubby area.

6.7.4 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. **area area-id nssa [no-redistribution] [default-information-originate] [route-map map-name] [no-summary] [translate type7 {always | never}] [suppress-fa]**
4. (Optional) **address-family ipv6 unicast**
5. (Optional) **area area-id default-cost cost**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	area area-id nssa [no-redistribution] [default-information-originate] [route-map map-name] [no-summary] [translate type7 {always never}] [suppress-fa] Example: switch(config-router)# area 0.0.0.10 nssa	Creates this area as an NSSA.
Step 4	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	(Optional) Enters IPv6 unicast address family mode.
Step 5	area area-id default-cost cost Example: switch(config-router-af)# area 0.0.0.10 default-cost 25	(Optional) Sets the cost metric for the default summary route sent into this NSSA. The range is from 0 to 16777215.
Step 6	copy running-config startup-config	(Optional) Saves this configuration change.

Command	Purpose
Example: <pre>switch(config-router)# copy running-config startup-config</pre>	

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
```

6.7.5 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 OSPFv3.

Ensure that you have configured a primary area for the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ipv6 router ospfv3** [*instance-tag*] **multi-area** *area-id*
4. (Optional) **show ipv6 ospfv3** *instance-tag* **interface** *interface-type slot/port*

5. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: <pre>switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 3	ipv6 router ospfv3 [instance-tag] multi-area area-id Example: <pre>switch(config-if)# ipv6 router ospfv3 201 multi-area 3</pre>	Adds the interface to another area. Note The <i>instance-tag</i> argument is optional. If you do not specify an instance, the multi-area configuration is applied to the same instance that is configured for the primary area on that interface.
Step 4	show ipv6 ospfv3 instance-tag interface interface-type slot/port Example: <pre>switch(config-if)# show ipv6 ospfv3 201 interface ethernet 1/2</pre>	(Optional) Displays OSPFv3 information.
Step 5	copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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 router ospfv3 201 area 0.0.0.10
switch(config-if)# ipv6 router ospfv3 201 multi-area 20
switch(config-if)# copy running-config startup-config
```

6.7.6 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **area *area-id* virtual-link *router-id***
4. (Optional) **show ipv6 ospfv3 virtual-link [brief]**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	area <i>area-id</i> virtual-link <i>router-id</i> Example: switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1 switch(config-router-vlink)#	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	show ipv6 ospfv3 virtual-link [brief] Example: switch(config-if)# show ipv6 ospfv3 virtual-link	(Optional) Displays OSPFv3 virtual link information.
Step 5	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

You can configure the following optional commands in virtual link configuration mode:

Command	Purpose
dead-interval <i>seconds</i> Example: switch(config-router-vlink)# dead-interval 50	(Optional) Configures the OSPFv3 dead interval, in seconds. The range is from 1 to 65535. The default is four times the hello interval, in seconds.
hello-interval <i>seconds</i> Example: switch(config-router-vlink)# hello-interval 25	(Optional) Configures the OSPFv3 hello interval, in seconds. The range is from 1 to 65535. The default is 10 seconds.

Command	Purpose
retransmit-interval <i>seconds</i> Example: <pre>switch(config-router-vlink)# retransmit-interval 50</pre>	(Optional) Configures the OSPFv3 retransmit interval, in seconds. The range is from 1 to 65535. The default is 5.
transmit-delay <i>seconds</i> Example: <pre>switch(config-router-vlink)# transmit-delay 2</pre>	(Optional) Configures the OSPFv3 transmit-delay, in seconds. The range is from 1 to 450. The default is 1.

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 ospf 101
switch(config-router)# area 0.0.0.10 virtual-link 2001:0DB8::1
switch(config-router)# copy running-config startup-config
```

6.7.7 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 .

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3** *instance-tag*
3. **address-family ipv6 unicast**
4. **redistribute** {*bgp id* | *direct* | *isis id* | *rip id* | *static*} **route-map** *map-name*
5. **default-information originate** [*always*] [*route-map map-name*]
6. **default-metric** *cost*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.

	Command	Purpose
	Example: <pre>switch# configure terminal switch(config)#</pre>	
Step 2	router ospfv3 instance-tag Example: <pre>switch(config)# router ospfv3 201 switch(config-router)#</pre>	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: <pre>switch(config-router)# address-family ipv6 unicast switch(config-router-af)#</pre>	Enters IPv6 unicast address family mode.
Step 4	redistribute {bgp id direct isis id rip id static} route-map map-name Example: <pre>switch(config-router-af)# redistribute bgp route-map FilterExternalBGP</pre>	Redistributes the selected protocol into OSPFv3 through the configured route map. Note If you redistribute static routes, Inspur INOS-CN also redistributes the default static route.
Step 5	default-information originate [always] [route-map map-name] Example: <pre>switch(config-router-af)# default-information-originate route-map DefaultRouteFilter</pre>	Creates a default route into this OSPFv3 domain if the default route exists in the RIB. Use the following optional keywords: <ul style="list-style-type: none"> • 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	default-metric cost Example: <pre>switch(config-router-af)# default-metric 25</pre>	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	copy running-config startup-config Example: <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

6.7.8 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

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **address-family ipv6 unicast**
4. **redistribute {*bgp id* | *direct* | *isis id* | *rip id* | *static*} route-map *map-name***
5. **redistribute maximum-prefix *max* [*threshold*] [*warning-only* | *withdraw* [*num-retries* *timeout*]]**
6. (Optional) **show running-config ospfv3**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 <i>instance-tag</i> Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	redistribute {<i>bgp id</i> <i>direct</i> <i>isis id</i> <i>rip id</i> <i>static</i>} route-map <i>map-name</i> Example: switch(config-router-af)# redistribute bgp route-map FilterExternalBGP	Redistributes the selected protocol into OSPFv3 through the configured route map.

	Command	Purpose
Step 5	<pre>redistribute maximum-prefix max [threshold] [warning-only withdraw [num-retries timeout]]</pre> <p>Example:</p> <pre>switch(config-router)# redistribute maximum-prefix 1000 75 warning-only</pre>	<p>Specifies a maximum number of prefixes that OSPFv2 distributes. The range is from 1 to 65535. Optionally, specifies the following:</p> <ul style="list-style-type: none"> threshold—Percentage 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 <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> range is from 60 to 600 seconds. The default is 300 seconds.
Step 6	<pre>show running-config ospfv3</pre> <p>Example:</p> <pre>switch(config-router)# show running-config ospf</pre>	(Optional) Displays the OSPFv3 configuration.
Step 7	<pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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)# redistribute maximum-prefix 1000 75
```

6.7.9 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 instance-tag**
3. **address-family ipv6 unicast**
4. **area area-id range ipv6-prefix/length [no-advertise] [cost cost] or**
5. **summary-address ipv6-prefix/length [no-advertise] [tag tag]**
6. (Optional) **show ipv6 ospfv3 summary-address**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	area area-id range ipv6-prefix/length [no-advertise] [cost cost] Example: switch(config-router-af)# area 0.0.0.10 range 2001:0DB8::/48 advertise	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 <i>cost</i> range is from 0 to 16777215.
Step 5	summary-address ipv6-prefix/length [no-advertise] [tag tag] Example: switch(config-router-af)# summary-address 2001:0DB8::/48 tag 2	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	show ipv6 ospfv3 summary-address Example: switch(config-router)# show ipv6 ospfv3 summary-address	(Optional) Displays information about OSPFv3 summary addresses.
Step 7	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

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

```
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 ospf 201
switch(config-router)# address-family ipv6 unicast
switch(config-router)# summary-address 2001:0DB8::/48
```

```
switch(config-router)# no discard route internal
switch(config-router)# copy running-config startup-config
```

6.7.10 Configuring the Administrative Distance of Routes

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 OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf *instance-tag***
3. **address-family ipv6 unicast**
4. **[no] table-map *map-name***
5. **exit**
6. **exit**
7. **route-map *map-name* [permit | deny] [*seq*]**
8. **match route-type *route-type***
9. **match ip route-source prefix-list *name***
10. **match ipv6 address prefix-list *name***
11. **set distance *value***
12. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf <i>instance-tag</i> Example: switch(config)# router ospf 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 4	[no] table-map <i>map-name</i> Example: switch(config-router-af)# table-map foo	Configures the policy for filtering or modifying OSPFv3 routes before sending them to the RIB. You can enter up to 63 alphanumeric characters for the map name.

	Command	Purpose
	exit Example: <pre>switch(config-router-af)# exit switch(config-router)#</pre>	Exits router address-family configuration mode.
Step 6	exit Example: <pre>switch(config-router)# exit switch(config)#</pre>	Exits router configuration mode.
Step 7	route-map <i>map-name</i> [permit deny] [<i>seq</i>] Example: <pre>switch(config)# route-map foo permit 10 switch(config-route-map)#</pre>	Creates a route map or enters route-map configuration mode for an existing route map. Use <i>seq</i> 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	match route-type <i>route-type</i> Example: <pre>switch(config-route-map)# match route-type external</pre>	Matches against one of the following route types: <ul style="list-style-type: none"> • 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	match ip route-source prefix-list <i>name</i> Example: <pre>switch(config-route-map)# match ip route-source prefix-list p1</pre>	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	match ipv6 address prefix-list <i>name</i> Example: <pre>switch(config-route-map)# match ipv6 address prefix-list p1</pre>	Matches against one or more IPv6 prefix lists. Use the ip prefix-list command to create the prefix list.
Step 11	set distance <i>value</i> Example: <pre>switch(config-route-map)# set distance 150</pre>	Sets the administrative distance of routes for OSPFv3. The range is from 1 to 255.
Step 12	copy running-config startup-config Example: <pre>switch(config-route-map)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

Command	Purpose
---------	---------

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 ospf 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
```

6.7.11 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.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **timers lsa-arrival *msec***
4. **timers lsa-group-pacing *seconds***
5. **timers throttle lsa *start-time hold-interval max-time***
6. **address-family ipv6 unicast**
7. **timers throttle spf *delay-time hold-time max-time***
8. **interface *type slot/port***
9. **ospfv3 retransmit-interval *seconds***
10. **ospfv3 transmit-delay *seconds***
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	timers lsa-arrival msec Example: switch(config-router)# timers lsa-arrival 2000	Sets the LSA arrival time in milliseconds. The range is from 10 to 600000. The default is 1000 milliseconds.
Step 4	timers lsa-group-pacing seconds Example: switch(config-router)# timers lsa-group-pacing 200	Sets the interval in seconds for grouping LSAs. The range is from 1 to 1800. The default is 10 seconds.
Step 5	timers throttle lsa start-time hold-interval max-time Example: switch(config-router)# timers throttle lsa network 350 5000 6000	Sets the rate limit in milliseconds for generating LSAs. You can configure the following timers: <i>start-time</i> —The range is from 50 to 5000 milliseconds. The default value is 50 milliseconds. <i>hold-interval</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds. <i>max-time</i> —The range is from 50 to 30,000 milliseconds. The default value is 5000 milliseconds.
Step 6	address-family ipv6 unicast Example: switch(config-router)# address-family ipv6 unicast switch(config-router-af)#	Enters IPv6 unicast address family mode.
Step 7	timers throttle spf delay-time hold-time max-time Example: switch(config-router)# timers throttle spf 3000 2000 5000	Sets the SPF best-path schedule initial delay time, minimum hold time, and maximum wait time in milliseconds between SPF best-path calculations. The range is from 1 to 600000 milliseconds. The default values are a 200-ms delay time, 1000-ms hold time, and 5000-ms wait time.
Step 8	interface type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 9	ospfv3 retransmit-interval seconds	Sets the estimated time in seconds between LSAs

	Command	Purpose
	Example: <pre>switch(config-if)# ospfv3 retransmit-interval 30</pre>	transmitted from this interface. The range is from 1 to 65535. The default is 5.
Step 10	ospfv3 transmit-delay <i>seconds</i> Example: <pre>switch(config-if)# ospfv3 transmit-delay 600 switch(config-if)#</pre>	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	copy running-config startup-config Example: <pre>switch(config-if)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

6.7.12 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 OSPFv3.

Ensure that all neighbors are configured for graceful restart with matching optional parameters set.

SUMMARY STEPS

1. **configure terminal**
2. **router ospfv3 *instance-tag***
3. **graceful-restart**
4. **graceful-restart grace-period *seconds***
5. **graceful-restart helper-disable**
6. **graceful-restart planned-only**
7. (Optional) **show ipv6 ospfv3 *instance-tag***
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

Command	Purpose
---------	---------

	Command	Purpose
	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 3	graceful-restart Example: switch(config-router)# graceful-restart	Enables graceful restart. A graceful restart is enabled by default.
Step 4	graceful-restart grace-period seconds Example: switch(config-router)# graceful-restart grace-period 120	Sets the grace period, in seconds. The range is from 5 to 1800. The default is 60 seconds.
Step 5	graceful-restart helper-disable Example: switch(config-router)# graceful-restart helper-disable	Disables helper mode. Enabled by default.
Step 6	graceful-restart planned-only Example: switch(config-router)# graceful-restart planned-only	Configures graceful restart for planned restarts only.
Step 7	Show ipv6 ospfv3 instance-tag Example: switch(config-if)# show ipv6 ospfv3 201	(Optional) Displays OSPFv3 information.
Step 8	Copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to enable graceful restart if it has been disabled and set the grace period to 120 seconds:

```
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
```

6.7.13 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:

Command	Purpose
restart ospfv3 instance-tag Example: switch(config)# restart ospfv3 201	Restarts the OSPFv3 instance and removes all neighbors.

6.7.14 Configuring OSPFv3 with Virtualization

You can configure multiple OSPFv3 instances. You can also create multiple VRFs and use the same or multiple OSPFv3 instances in each VRF. You assign an OSPFv3 interface to a VRF.

BEFORE YOU BEGIN

You must enable OSPF.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** vrf_name
3. **router ospfv3** instance-tag
4. **vrf** vrf-name
5. (Optional) **maximum-paths** paths
6. **interface** type slot/port
7. **vrf member** vrf-name
8. **ipv6 address** ipv6-prefix/length
9. **ipv6 ospfv3** instance-tag area area-id
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context vrf-name Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	router ospfv3 instance-tag Example: switch(config)# router ospfv3 201 switch(config-router)#	Creates a new OSPFv3 instance with the configured instance tag.
Step 4	vrf vrf-name Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters VRF configuration mode.
Step 5	maximum-paths paths	(Optional) Configures the maximum number of equal

	Command	Purpose
	Example: <pre>switch(config-router-vrf)# maximum-paths 4</pre>	OSPFv3 paths to a destination in the route table for this VRF. Use this command for load balancing.
Step 6	interface <i>type slot/port</i> Example: <pre>switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 7	vrf member <i>vrf-name</i> Example: <pre>switch(config-if)# vrf member RemoteOfficeVRF</pre>	Adds this interface to a VRF.
Step 8	ipv6 address <i>ipv6-prefix/length</i> Example: <pre>switch(config-if)# ipv6 address 2001:0DB8::1/48</pre>	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 9	ipv6 ospfv3 <i>instance-tag area area-id</i> Example: <pre>switch(config-if)# ipv6 ospfv3 201 area 0</pre>	Assigns this interface to the OSPFv3 instance and area configured.
Step 10	copy running-config startup-config Example: <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

6.8 Verifying the OSPFv3 Configuration

To display the OSPFv3 configuration, perform one of the following tasks:

Command	Purpose
show ipv6 ospfv3 [<i>instance-tag</i>] [<i>vrf vrf-name</i>]	Displays information about one or more OSPFv3 routing instances. The output includes the

Command	Purpose
	following area-level counts: <ul style="list-style-type: none"> • 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 <i>type number</i> [vrf { <i>vrf-name</i> all default management }]	Displays OSPFv3-related interface information.
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 ospfv3 process	Displays the OSPFv3 authentication configuration at the process level.
show ospfv3 interface <i>interface-type slot/port</i>	Displays the OSPFv3 authentication configuration at the interface level.
show running-configuration ospfv3	Displays the current running OSPFv3 configuration.

6.9 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	Displays the OSPFv3 route policy statistics for an area.

Command	Purpose
<i>area-id</i> filter-list {in out} [vrf { <i>vrf-name</i> all default management}]	
show ipv6 ospfv3 policy statistics redistribute { bgp id direct isis id rip id static } vrf { <i>vrf-name</i> all default management}]	Displays the OSPFv3 route policy statistics.
show ipv6 ospfv3 statistics [vrf { <i>vrf-name</i> all default management}]	Displays the OSPFv3 event counters.
show ipv6 ospfv3 traffic [<i>interface-type number</i>] [vrf { <i>vrf-name</i> all default management}]	Displays the OSPFv3 packet counters.

6.10 Configuration Examples for 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
```

CHAPTER 7 Configuring EIGRP

This chapter describes how to configure the Enhanced Interior Gateway Routing Protocol (EIGRP) on the Inspur INOS-CN device.

7.1 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.

This section includes the following topics:

- EIGRP Components
- EIGRP Route Updates
- Advanced EIGRP

7.1.1 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. See the “EIGRP Route Updates” section.

EIGRP also uses the Hello messages as a keepalive to its neighbors. As long as Hello messages are received, Inspur INOS-CN 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.

7.1.2 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.

This section includes the following topics:

- Internal Route Metrics
- Wide Metrics
- External Route Metrics
- EIGRP and the Unicast RIB

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:

$$\text{metric} = [k1 \times \text{bandwidth} + (k2 \times \text{bandwidth}) / (256 - \text{load}) + k3 \times \text{delay} + k6 \times \text{extended attributes}] \times [k5 / (\text{reliability} + k4)]$$

Since 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:

$$\text{RIB Metric} = (\text{Wide Metric} / \text{RIB scale value}).$$

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.

7.1.3 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 keychain management for the authentication keys. Keychain management allows you to control changes to the authentication keys used by MD5 authentication digest. See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for more details about creating keychains.

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-CN 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

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-CN 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.

Split Horizon

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-CN 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) for IPv4 and IPv6. 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information.

Virtualization Support

EIGRP supports Virtual Routing and Forwarding instances (VRFs).

Graceful Restart and High Availability

Inspur INOS-CN 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-CN 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-CN 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-CN 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 rediscovers 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-CN applies the running configuration, and EIGRP informs the neighbors that it is operational again.

Multiple EIGRP Instances

Inspur INOS-CN 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 CN12900 Series INOS-CN Verified Scalability Guide*.

7.2 Licensing Requirements for EIGRP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	EIGRP requires an Essentials license package.

7.3 Prerequisites for EIGRP

EIGRP has the following prerequisites:

- You must enable EIGRP (see the “Enabling the EIGRP Feature” section).

7.4 Guidelines and Limitations for EIGRP

EIGRP has the following configuration guidelines and limitations:

- 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 (see Chapter 15, “Configuring Route Policy Manager”).
- 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.
- 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-CN does not support IGRP or connecting IGRP and EIGRP clouds.
- Autosummarization is disabled by default and cannot be enabled.
- Inspur INOS-CN supports only IP.
- High availability is not supported with EIGRP aggressive timers.
- When EIGRP is using narrow metrics and the summary route is relearned as internal, the internal route is preferred over the summary route because it has better administrative distance, so the route will keep toggling between summary and internal in the RIB.

7.5 Default Settings

Table 7-1 lists the default settings for EIGRP parameters.

Table 7-1 Default EIGRP Parameters

Parameters	Default
Administrative distance	<ul style="list-style-type: none"> • Internal routes—90 • External routes—170
Bandwidth percent	50 percent
Default metric for redistributed routes	<ul style="list-style-type: none"> • Bandwidth—100000 Kb/s • Delay—100 (10 microsecond units) • Reliability—255 • Loading—1 • MTU—1500

Parameters	Default
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

7.6 Configuring Basic EIGRP

7.6.1 Enabling the EIGRP Feature

You must enable EIGRP before you can configure EIGRP.

SUMMARY STEPS

1. **configure terminal**
2. **feature eigrp**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature eigrp Example: switch(config)# feature eigrp	Enables the EIGRP feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays information about enabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To disable the EIGRP feature and remove all associated configuration, use the following command in configuration mode:

Command	Purpose
no feature eigrp Example: <pre>switch(config)# no feature eigrp</pre>	Disables the EIGRP feature and removes all associated configuration.

7.6.2 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 (see the “Autonomous Systems” section). Routes are not advertised or accepted from other autonomous systems unless you enable route redistribution.

BEFORE YOU BEGIN

You must enable EIGRP (see the “Enabling the EIGRP Feature” section).

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.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp** *instance-tag*
3. (Optional) **autonomous-system** *as-number*
4. (Optional) **log-adjacency-changes**
5. (Optional) **log-neighbor-warnings** [*seconds*]
6. **interface** *interface-type slot/port*
7. **{ip | ipv6} router eigrp** *instance-tag*
8. (Optional) **show {ip | ipv6} eigrp interfaces**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router eigrp <i>instance-tag</i> Example: <pre>switch(config)# router eigrp Test1 switch(config-router)#</pre>	<p>Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>If you configure an <i>instance-tag</i> 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.</p>

	Command	Purpose
Step 3	autonomous-system <i>as-number</i> Example: switch(config-router)# autonomous-system 33	(Optional) Configures a unique AS number for this EIGRP instance. The range is from 1 to 65535.
Step 4	log-adjacency-changes Example: switch(config-router)# log-adjacency-changes	(Optional) Generates a system message whenever an adjacency changes state. This command is enabled by default.
Step 5	log-neighbor-warnings [<i>seconds</i>] Example: switch(config-router)# log-neighbor-warnings	(Optional) 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	interface <i>interface-type slot/port</i> Example: switch(config-router)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use ? to determine the slot and port ranges.
Step 7	{ip ipv6} router eigrp <i>instance-tag</i> Example: switch(config-if)# ip router eigrp Test1	Associates this interface with the configured EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 8	show {ip ipv6} eigrp interfaces Example: switch(config-if)# show ip eigrp interfaces	(Optional) Displays information about EIGRP interfaces.
Step 9	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To remove the EIGRP process and the associated configuration, use the following command in the configuration mode:

Command	Purpose
no router eigrp <i>instance-tag</i> Example: switch(config)# no router eigrp Test1	Deletes the EIGRP process and all associated configuration.

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
```

For more information about other EIGRP parameters, see the “Configuring Advanced EIGRP” section.

7.6.3 Restarting an EIGRP Instance

You can restart an EIGRP instance. This action clears all neighbors for the instance.

To restart an EIGRP instance and remove all associated neighbors, use the following commands:

Command	Purpose
flush-routes Example: <pre>switch(config)# flush-routes</pre>	(Optional) Flushes all EIGRP routes in the unicast RIB when this EIGRP instance restarts.
restart eigrp instance-tag Example: <pre>switch(config)# restart eigrp Test1</pre>	Restarts the EIGRP instance and removes all neighbors. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

7.6.4 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.

To disable an EIGRP instance, use the following command in router configuration mode:

Command	Purpose
<pre>switch(config-router)# shutdown</pre> Example: <pre>switch(config-router)# shutdown</pre>	Disables this instance of EIGRP. The EIGRP router configuration remains.

7.6.5 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.

To configure a passive interface for EIGRP, use the following command in interface configuration mode:

Command	Purpose
<pre>{ip ipv6} passive-interface eigrp instance-tag</pre> Example: <pre>switch(config-if)# ip passive-interface eigrp tag10</pre>	Suppresses EIGRP hellos, which prevents neighbors from forming and sending routing updates on an EIGRP interface. The <i>instance-tag</i> argument can be any case-sensitive, alphanumeric string up to 20 characters.

7.6.6 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.

To disable EIGRP on an interface, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# {ip ipv6} eigrp instance-tag shutdown</pre> <p>Example:</p> <pre>switch(config-router)# ip eigrp Test1 shutdown</pre>	Disables EIGRP on this interface. The EIGRP interface configuration remains. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.

7.7 Configuring Advanced EIGRP

7.7.1 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

You must enable EIGRP.

Ensure that all neighbors for an EIGRP process share the same authentication configuration, including the shared authentication key.

Create the keychain for this authentication configuration. For more information, see the *Inspur CN12900 Series INOS-CN Security Configuration Guide*.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp instance-tag**
3. **address-family {ipv4 | ipv6} unicast**
4. **authentication key-chain key-chain**
5. **authentication mode md5**
6. **interface interface-type slot/port**
7. **{ip | ipv6} router eigrp instance-tag**
8. **{ip | ipv6} authentication key-chain eigrp instance-tag key-chain**
9. **{ip | ipv6} authentication mode eigrp instance-tag md5**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<pre>configure terminal</pre> <p>Example:</p> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<pre>router eigrp instance-tag</pre> <p>Example:</p> <pre>switch(config)# router eigrp Test1 switch(config-router)#</pre>	<p>Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>If you configure an <i>instance-tag</i> that does not qualify as an AS number, you must use the</p>

	Command	Purpose
		autonomous-system command to configure the AS number explicitly or this EIGRP instance will remain in the shutdown state.
Step 3	address-family { <i>ipv4</i> <i>ipv6</i> } unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	authentication key-chain <i>key-chain</i> Example: switch(config-router-af)# authentication key-chain routeKeys	Associates a keychain with this EIGRP process for this VRF. The keychain can be any case-sensitive, alphanumeric string up to 20 characters.
Step 5	authentication mode md5 Example: switch(config-router-af)# authentication mode md5	Configures MD5 message digest authentication mode for this VRF.
Step 6	interface <i>interface-type slot/port</i> Example: switch(config-router-af) interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use ? to find the supported interfaces.
Step 7	{ <i>ip</i> <i>ipv6</i> } router eigrp <i>instance-tag</i> Example: switch(config-if)# ip router eigrp Test1	Associates this interface with the configured EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 8	{ <i>ip</i> <i>ipv6</i> } authentication key-chain eigrp <i>instance-tag key-chain</i> Example: switch(config-if)# ip authentication key-chain eigrp Test1 routeKeys	Associates a keychain 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	{ <i>ip</i> <i>ipv6</i> } authentication mode eigrp <i>instance-tag md5</i> Example: switch(config-if)# ip authentication mode eigrp Test1 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	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This 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

```

7.7.2 Configuring EIGRP Stub Routing

To configure a router for EIGRP stub routing, use the following command in address-family configuration mode:

Command	Purpose
<pre> switch(config-router-af)# stub [direct receive-only redistributed [direct] leak-map map-name] Example: switch(config-router-af)# eigrp stub redistributed </pre>	<p>Configures a remote router as an EIGRP stub router. The map name can be any case-sensitive, alphanumeric string up to 20 characters.</p>

This example shows how to configure a stub router to advertise directly connected and redistributed routes:

```

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

```

Use the **show ip eigrp neighbor detail** command to verify that a router has been configured as a stub router. The last line of the output shows the stub status of the remote or spoke router.

This example shows output from the **show ip eigrp neighbor detail** command:

```

Router# 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

```

7.7.3 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.

To configure a summary aggregate address, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# {ip ipv6} summary-address eigrp instance-tag ip-prefix/length [distance leak-map map-name] Example: switch(config-if)# ip summary-address eigrp Test1 192.0.2.0/8</pre>	<p>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.</p> <p>You can optionally configure the administrative distance for this aggregate address. The default administrative distance is 5 for aggregate addresses.</p>

This 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
```

7.7.4 Redistributing Routes into EIGRP

You can redistribute routes in EIGRP from other routing protocols.

BEFORE YOU BEGIN

You must enable EIGRP.

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.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp instance-tag**
3. **address-family {ipv4 | ipv6} unicast**
4. **redistribute {bgp as | {eigrp | isis | ospf | ospfv3 | rip} instance-tag | direct | static} route-map name**
5. **default-metric bandwidth delay reliability loading mtu**
6. (Optional) **show {ip | ipv6} eigrp route-map statistics redistribute**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<pre>configure terminal Example: switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<pre>router eigrp instance-tag Example: switch(config)# router eigrp Test1 switch(config-router)#</pre>	<p>Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.</p> <p>If you configure an <i>instance-tag</i> that does not qualify as an AS number, you must use the autonomous-system command to configure</p>

	Command	Purpose
		the AS number explicitly or this EIGRP instance will remain in the shutdown state.
Step 3	address-family { <i>ipv4</i> <i>ipv6</i> } unicast Example: <pre>switch(config-router)# address-family ipv4 unicast switch(config-router-af)#</pre>	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	redistribute { <i>bgp as</i> (<i>eigrp</i> <i>isis</i> <i>ospf</i> <i>ospfv3</i> <i>rip</i>) <i>instance-tag</i> <i>direct</i> <i>static</i> } route-map <i>name</i> Example: <pre>switch(config-router-af)# redistribute bgp 100 route-map BGPFilter</pre>	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	default-metric <i>bandwidth delay reliability loading mtu</i> Example: <pre>switch(config-router-af)# default-metric 500000 30 200 1 1500</pre>	Sets the metrics assigned to routes learned through route redistribution. The default values are as follows: <ul style="list-style-type: none"> bandwidth—100000 Kb/s delay—100 (10 microsecond units) reliability—255 loading—1 MTU—1492
Step 6	show { <i>ip</i> <i>ipv6</i> } eigrp route-map statistics redistribute Example: <pre>switch(config-router-af)# show ip eigrp route-map statistics redistribute bgp</pre>	(Optional) Displays information about EIGRP route map statistics.
Step 7	copy running-config startup-config Example: <pre>switch(config)# copy running- config startup-config</pre>	(Optional) Saves this configuration change.

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
```

7.7.5 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

You must enable EIGRP.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router eigrp <i>instance-tag</i> Example: <pre>switch(config)# router eigrp Test1 switch(config-router)#</pre>	Creates a new EIGRP instance with the configured instance tag.
Step 3	redistribute { <i>bgp id</i> direct <i>eigrp id</i> <i>isis id</i> <i>ospf id</i> <i>rip id</i> static } route-map <i>map-name</i> Example: <pre>switch(config-router)# redistribute bgp route-map FilterExternalBGP</pre>	Redistributes the selected protocol into EIGRP through the configured route map.
Step 4	redistribute maximum-prefix <i>max</i> [<i>threshold</i>] [warning-only withdraw [<i>num-retries</i> <i>timeout</i>]] Example: <pre>switch(config-router)# redistribute maximum-prefix 1000 75 warning-only</pre>	Specifies a maximum number of prefixes that EIGRP distributes. The range is from 1 to 65535. Optionally specifies the following: <ul style="list-style-type: none"> • threshold—Percentage 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.

	Command	Purpose
		Optionally tries to retrieve the redistributed routes. The <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> 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	show running-config eigrp Example: switch(config-router)# show running-config eigrp	(Optional) Displays the EIGRP configuration.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This 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
```

7.7.6 Configuring Load Balancing in EIGRP

You can configure load balancing in EIGRP. You can configure the number of Equal Cost Multiple Path (ECMP) routes using the maximum paths option.

BEFORE YOU BEGIN

You must enable EIGRP).

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp instance-tag**
3. **address-family {ipv4 | ipv6} unicast**
4. **maximum-paths num-paths**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp instance-tag Example: switch(config)# router eigrp Test1	Creates a new EIGRP process with the configured instance tag. The instance tag can be any case-sensitive, alphanumeric string up

	Command	Purpose
	switch(config-router)#	to 20 characters. If you configure an <i>instance-tag</i> 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	address-family {ipv4 ipv6} unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	maximum-paths num-paths Example: switch(config-router-af)# maximum-paths 5	Sets the number of equal cost paths that EIGRP accepts in the route table. The range is from 1 to 32. The default is 8.
Step 5	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This 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)# maximum-paths 6
switch(config-router)# copy running-config startup-config
```

7.7.7 Configuring Graceful Restart for EIGRP

You can configure graceful restart or nonstop forwarding for EIGRP.

BEFORE YOU BEGIN

You must enable EIGRP.

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.

SUMMARY STEPS

1. **configure terminal**
2. **router eigrp instance-tag**
3. **address-family {ipv4 | ipv6} unicast**
4. **graceful-restart**
5. **timers nsf converge seconds**
6. **timers nsf route-hold seconds**
7. **timers nsf signal seconds**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router eigrp instance-tag Example: switch(config)# router eigrp Test1 switch(config-router)#	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 <i>instance-tag</i> 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	address-family {ipv4 ipv6} unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters the address-family configuration mode. This command is optional for IPv4.
Step 4	graceful-restart Example: switch(config-router-af)# graceful-restart	Enables graceful restart. This feature is enabled by default.
Step 5	timers nsf converge seconds Example: switch(config-router-af)# timers nsf converge 100	Sets the time limit for the convergence after a switchover. The range is from 60 to 180 seconds. The default is 120.
Step 6	timers nsf route-hold seconds Example: switch(config-router-af)# timers nsf route-hold 200	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	timers nsf signal seconds Example: switch(config-router-af)# timers nsf signal 15	Sets the time limit for signaling a graceful restart. The range is from 10 to 30 seconds. The default is 20.
Step 8	copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

7.7.8 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.

To change the interval between hello packets, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# {ip ipv6} hello-interval eigrp instance-tag seconds Example: switch(config-if)# ip hello-interval eigrp Test1 30</pre>	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.

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.

To change the hold time, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# {ip ipv6} hold-time eigrp instance-tag seconds Example: switch(config-if)# ipv6 hold-time eigrp Test1 30</pre>	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.

Use the **show ip eigrp interface detail** command to verify the timer configuration.

7.7.9 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.

To disable split horizon, use the following command in interface configuration mode:

Command	Purpose
<pre>switch(config-if)# no {ip ipv6} split-horizon eigrp instance-tag Example: switch(config-if)# no ip split-horizon eigrp</pre>	Disables split horizon.

Test1

7.7.10 Enabling Wide Metrics

To enable wide metrics, use the following command in router or address family configuration mode:

Command	Purpose
<pre>switch(config-router)# metrics version 64bit</pre> <p>Example:</p> <pre>switch(config-router)# metrics version 64bit</pre>	Enables 64-bit metric values.

To optionally configure a scaling factor for the RIB, use the following commands in router or address family configuration mode:

Command	Purpose
<pre>switch(config-router)# metrics rib-scale value</pre> <p>Example:</p> <pre>switch(config-router)# metrics rib-scale 128</pre>	(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 is 128.

7.7.11 Tuning EIGRP

You can configure optional parameters to tune EIGRP for your network.

You can configure the following optional parameters in address-family configuration mode:

Command	Purpose
<pre>default-information originate [always route-map map-name]</pre> <p>Example:</p> <pre>switch(config-router-af)# default-information originate always</pre>	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 map name can be any case-sensitive, alphanumeric string up to 20 characters.
<pre>distance internal external</pre> <p>Example:</p> <pre>switch(config-router-af)# distance 25 100</pre>	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).
<pre>metric max-hops hop-count</pre> <p>Example:</p> <pre>switch(config-router-af)# metric max-hops 70</pre>	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.
<pre>metric weights tos k1 k2 k3 k4 k5 k6</pre> <p>Example:</p> <pre>switch(config-router-af)# metric weights 0</pre>	Adjusts the EIGRP metric or K value. EIGRP uses the following formula to determine the total metric to the network:

Command	Purpose
1 3 2 1 0	$\text{metric} = [k1 \times \text{bandwidth} + (k2 \times \text{bandwidth}) / (256 - \text{load}) + k3 \times \text{delay} + k6 \times \text{extended attributes}] * [k5 / (\text{reliability} + k4)]$ <p>Default values and ranges are as follows:</p> <ul style="list-style-type: none"> • 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.
nsf await-redirect-protocol-convergence Example: <pre>switch(config-router-af)# nsf await-redirect-protocol-convergence</pre>	<p>Causes EIGRP to wait for the convergence of redistributed protocols before installing its own routes in the Routing Information Base (RIB) during nonstop forwarding (NSF).</p> <p>This command is useful in switchover scenarios when NSF is in progress and you want EIGRP to wait for BGP to converge and install its routes. It prevents EIGRP from installing transient routes and modifying the Forwarding Information Base (FIB) entries before BGP converges and EIGRP finds an alternate path to a destination.</p> <p>Note If you use this command when mutual redistribution is configured between EIGRP and BGP (for example, in a PE-CE environment), some traffic loss might occur because the provider-edge (PE) router will not install EIGRP routes into the RIB until BGP routes are available. This behavior delays the routes that the customer-edge (CE) router learns from EIGRP and advertises to the peer PE router.</p>
timers active-time {time-limit disabled} Example: <pre>switch(config-router-af)# timers active-time 200</pre>	<p>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.</p>

You can configure the following optional parameters in interface configuration mode:

Command	Purpose
<pre>{ip ipv6} bandwidth eigrp instance-tag bandwidth</pre> Example:	<p>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</p>

Command	Purpose
<pre>switch(config-if)# ip bandwidth eigrp Test1 30000</pre>	2,560,000,000 Kb/s.
<pre>{ip ipv6} bandwidth-percent eigrp instance-tag percent</pre> <p>Example:</p> <pre>switch(config-if)# ip bandwidth-percent eigrp Test1 30</pre>	<p>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.</p> <p>The percent range is from 0 to 100. The default is 50.</p>
<pre>no {ip ipv6} delay eigrp instance-tag delay</pre> <p>Example:</p> <pre>switch(config-if)# ip delay eigrp Test1 100</pre>	<p>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).</p>
<pre>{ip ipv6} distribute-list eigrp instance-tag {prefix-list name route-map name} {in out}</pre> <p>Example:</p> <pre>switch(config-if)# ip distribute-list eigrp Test1 route-map EigrpTest in</pre>	<p>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.</p>
<pre>no {ip ipv6} next-hop-self eigrp instance-tag</pre> <p>Example:</p> <pre>switch(config-if)# ipv6 next-hop-self eigrp Test1</pre>	<p>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.</p>
<pre>{ip ipv6} offset-list eigrp instance-tag {prefix-list name route-map name} {in out} offset</pre> <p>Example:</p> <pre>switch(config-if)# ip offset-list eigrp Test1 prefix-list EigrpList in</pre>	<p>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.</p>
<pre>{ip ipv6} passive-interface eigrp instance-tag</pre> <p>Example:</p> <pre>switch(config-if)# ip passive-interface eigrp Test1</pre>	<p>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.</p>

7.8 Configuring Virtualization for EIGRP

You can configure multiple EIGRP processes, create multiple VRFs, and use the same or multiple EIGRP processes in each VRF. You assign an interface to a VRF.

BEFORE YOU BEGIN

- You must enable EIGRP.
- Create the VRFs.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **router eigrp** *instance-tag*
4. **interface ethernet** *slot/port*
5. **vrf member** *vrf-name*
6. **{ip | ipv6} router eigrp** *instance-tag*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	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	router eigrp <i>instance-tag</i> Example: switch(config)# router eigrp Test1 switch(config-router)#	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 <i>instance-tag</i> 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	interface ethernet <i>slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode. Use ? to find the slot and port ranges.
Step 5	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF. The VRF name can be any case-sensitive, alphanumeric string up to 20 characters.
Step 6	{ip ipv6} router eigrp <i>instance-tag</i> Example: switch(config-if)# ip router eigrp Test1	Adds this interface to the EIGRP process. The instance tag can be any case-sensitive, alphanumeric string up to 20 characters.
Step 7	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

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)# 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
```

7.9 Verifying the EIGRP Configuration

To display the EIGRP configuration, perform one of the following tasks:

Command	Purpose
show {ip ipv6} eigrp [<i>instance-tag</i>]	Displays a summary of the configured EIGRP processes.
show {ip ipv6} eigrp [<i>instance-tag</i>] interfaces [<i>type number</i>] [brief] [detail]	Displays information about all configured EIGRP interfaces.
show {ip ipv6} eigrp <i>instance-tag</i> neighbors [<i>type number</i>] [detail]	Displays information about all the EIGRP neighbors. Use this command to verify the EIGRP neighbor configuration.
show {ip ipv6} eigrp [<i>instance-tag</i>] route [<i>ip-prefix/length</i>] [active] [all-links] [detail-links] [pending] [summary] [zero-successors] [<i>vrf vrf-name</i>]	Displays information about all the EIGRP routes.
show {ip ipv6} eigrp [<i>instance-tag</i>] topology [<i>ip-prefix/length</i>] [active] [all-links] [detail-links] [pending] [summary] [zero-successors] [<i>vrf vrf-name</i>]	Displays information about the EIGRP topology table.
show running-configuration eigrp	Displays the current running EIGRP configuration.

7.10 Monitoring EIGRP

To display EIGRP statistics, use the following commands:

Command	Purpose
show {ip ipv6} eigrp [<i>instance-tag</i>] accounting [<i>vrf vrf-name</i>]	Displays accounting statistics for EIGRP.
show {ip ipv6} eigrp [<i>instance-tag</i>] route-map statistics redistribute	Displays redistribution statistics for EIGRP.
show {ip ipv6} eigrp [<i>instance-tag</i>] traffic [<i>vrf vrf-name</i>]	Displays traffic statistics for EIGRP.

7.11 Configuration Examples for EIGRP

This example shows how to configure EIGRP:

```
feature eigrp interface
ethernet 1/2
 ip address 192.0.2.55/24
 ip router eigrp Test1
 no shutdown
router eigrp Test1
 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 map to match an EIGRP external protocol metric route with an allowable deviation of 100, a source protocol of BGP, and an autonomous system number of 45000. When the two match clauses are 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 external 500 +-
100
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
```

CHAPTER 8 Configuring IS-IS

This chapter describes how to configure Integrated Intermediate System-to-Intermediate System (IS-IS) on the Inspur INOS-CN device.

8.1 About IS-IS

IS-IS is an Interior Gateway Protocol (IGP) based on Standardization (ISO)/International Engineering Consortium (IEC) 10589. Inspur INOS-CN supports Internet Protocol version 4 (IPv4) and 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.

8.1.1 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.

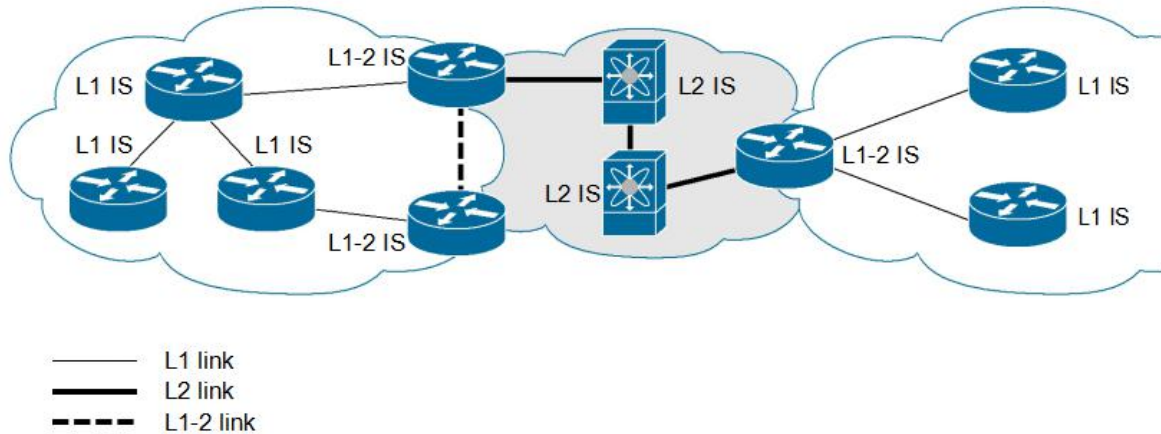
IS-IS Areas

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 (see Figure 8-1).

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-CN 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 8-1 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.

8.1.2 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 CN12900 Series INOS-CN Security Configuration Guide* for information on keychain management.

8.1.3 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.

8.1.4 Overload Bit

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.

8.1.5 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.

8.1.6 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-CN 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.

8.1.7 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-CN 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.

8.1.8 BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4 and IPv6. 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information.

8.1.9 Virtualization Support

Inspur INOS-CN supports multiple process instances for IS-IS. Each IS-IS instance can support multiple virtual

routing and forwarding (VRF) instances, up to the system limit. For the number of supported IS-IS instances, see the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

8.1.10 High Availability and Graceful Restart

Inspur INOS-CN 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
 - 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

8.1.11 Multiple IS-IS Instances

Inspur INOS-CN supports multiple 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.

8.2 Licensing Requirements for IS-IS

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	IS-IS requires an Essentials license package.

8.3 Prerequisites for IS-IS

IS-IS has the following prerequisites:

- You must enable IS-IS (see the “Enabling the IS-IS Feature” section).

8.4 Guidelines and Limitations for IS-IS

IS-IS has the following configuration guidelines and limitations:

- You can configure IS-IS over segment routing for all Inspur CN12900 Series switches.

8.5 Default Settings

Table 8-1 lists the default settings for IS-IS parameters.

Table 8-1 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	8
Metric	40
Reference bandwidth	40 Gbps

8.6 Configuring IS-IS

To configure IS-IS, follow these steps:

-
- Step 1** Enable the IS-IS feature.
 - Step 2** Create an IS-IS instance.
 - Step 3** Add an interface to the IS-IS instance.
 - Step 4** Configure optional features, such as authentication, mesh groups, and dynamic host exchange.
-

8.6.1 IS-IS Configuration Modes

The following sections show how to enter each of the configuration modes. From a mode, you can enter the ? command to display the commands available in that mode.

Router Configuration Mode

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

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)#
```

8.6.2 Enabling the IS-IS Feature

You must enable the IS-IS feature before you can configure IS-IS.

SUMMARY STEPS

1. **configure terminal**
2. **feature isis**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature isis Example: switch(config)# feature isis	Enables the IS-IS feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To disable the IS-IS feature and remove all associated configuration, use the following command in configuration mode:

Command	Purpose
no feature isis Example: switch(config)# no feature isis	Disables the IS-IS feature and removes all associated configurations.

8.6.3 Creating an IS-IS Instance

You can create an IS-IS instance and configure the area level for that instance.

BEFORE YOU BEGIN

You must enable IS-IS.

SUMMARY STEPS

1. **configure terminal**
2. **router isis** *instance-tag*
3. **net** *network-entity-title*
4. (Optional) **is-type** {*level-1* | *level-2* | *level-1-2*}
5. (Optional) **show isis** [*vrf vrf-name*] **process**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i> Example: switch(config)# router isis Enterprise switch(config-router)#	Creates a new IS-IS instance with the configured <i>instance tag</i> .
Step 3	net <i>network-entity-title</i> Example: switch(config-router)# net 47.0004.004d.0001.0001.0c11.1111.00	Configures the NET for this IS-IS instance.
Step 4	is-type { <i>level-1</i> <i>level-2</i> <i>level-1-2</i> } Example: switch(config-router)# is-type level-2	(Optional) Configures the area level for this IS-IS instance. The default is level-1-2.
Step 5	show isis [<i>vrf vrf-name</i>] process Example: switch(config)# show isis process	(Optional) Displays a summary of IS-IS information for all IS-IS instances.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To remove the IS-IS instance and the associated configuration, use the following command in configuration mode:

Command	Purpose
no router isis <i>instance-tag</i> Example: switch(config)# no router isis Enterprise	Deletes the IS-IS instance and all associated configurations.

You can configure the following optional parameters for IS-IS:

Command	Purpose
---------	---------

distance <i>value</i> Example: <pre>switch(config-router)# distance 30</pre>	Sets the administrative distance for IS-IS. The range is from 1 to 255. The default is 115.
log-adjacency-changes Example: <pre>switch(config-router)# log-adjacency-changes</pre>	Sends a system message whenever an IS-IS neighbor changes the state.
lsp-mtu <i>size</i> Example: <pre>switch(config-router)# lsp-mtu 600</pre>	Sets the MTU for LSPs in this IS-IS instance. The range is from 128 to 4352 bytes. The default is 1492.
maximum-paths <i>number</i> Example: <pre>switch(config-router)# maximum-paths 6</pre>	Configures the maximum number of equal-cost paths that IS-IS maintains in the route table. The range is from 1 to 64. The default is 8.
reference-bandwidth <i>bandwidth-value</i> { Mbps Gbps } Example: <pre>switch(config-router)# reference-bandwidth 100 Gbps</pre>	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.

This 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
```

To clear neighbor statistics and remove adjacencies, use the following command in router configuration mode:

Command	Purpose
<pre>clear isis [<i>instance-tag</i>] adjacency [* <i>system-id</i> <i>interface</i>]</pre> Example: <pre>switch(config-if)# clear isis adjacency *</pre>	Clears neighbor statistics and removed adjacencies for this IS-IS instance.

8.6.4 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:

Command	Purpose
<pre>restart isis <i>instance-tag</i></pre> Example: <pre>switch(config)# restart isis Enterprise</pre>	Restarts the IS-IS instance and removes all neighbors.

8.6.5 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:

Command	Purpose
shutdown Example: <pre>switch(config-router)# shutdown</pre>	Disables the IS-IS instance.

8.6.6 Configuring IS-IS on an Interface

You can add an interface to an IS-IS instance.

BEFORE YOU BEGIN

You must enable IS-IS.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. (Optional) **medium** {**broadcast** | **p2p**}
4. {**ip** | **ipv6**} **router isis** *instance-tag*
5. (Optional) **show isis** [**vrf vrf-name**] [*instance-tag*] **interface** [*interface-type slot/port*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: <pre>switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 3	medium { broadcast p2p } Example: <pre>switch(config-if)# medium p2p</pre>	(Optional) Configures the broadcast or point-to-point mode for the interface. IS-IS inherits this mode.
Step 4	{ ip ipv6 } router isis <i>instance-tag</i> Example: <pre>switch(config-if)# ip router isis Enterprise</pre>	Associates this IPv4 or IPv6 interface with an IS-IS instance.
Step 5	show isis [vrf vrf-name] [<i>instance-tag</i>] interface [<i>interface-type slot/port</i>] Example: <pre>switch(config)# show isis Enterprise ethernet 1/2</pre>	(Optional) Displays IS-IS information for an interface.

	Command	Purpose
Step 6	<pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

You can configure the following optional parameters for IS-IS in interface mode:

Command	Purpose
<pre>isis circuit-type {level-1 level-2 level-1-2}</pre> <p>Example:</p> <pre>switch(config-if)# isis circuit-type level-2</pre>	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.
<pre>isis metric value {level-1 level-2}</pre> <p>Example:</p> <pre>switch(config-if)# isis metric 30</pre>	Sets the IS-IS metric for this interface. The range is from 1 to 16777214. The default is 10.
<pre>isis passive {level-1 level-2 level-1-2}</pre> <p>Example:</p> <pre>switch(config-if)# isis passive level-2</pre>	Prevents the interface from forming adjacencies but still advertises the prefix associated with the interface.

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
```

8.6.7 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:

Command	Purpose
<pre>switch(config-if)# isis shutdown</pre> <p>Example:</p> <pre>switch(config-router)# isis shutdown</pre>	Disables IS-IS on this interface. The IS-IS interface configuration remains.

8.6.8 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.

SUMMARY STEPS

1. **configure terminal**
2. **router isis *instance-tag***
3. **authentication-type {cleartext | md5} {level-1 | level-2}**
4. **authentication key-chain *key* {level-1 | level-2}**
5. (Optional) **authentication-check {level-1 | level-2}**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i> Example: switch(config)# router isis Enterprise switch(config-router)#	Creates a new IS-IS instance with the configured <i>instance tag</i> .
Step 3	authentication-type {cleartext md5} {level-1 level-2} Example: switch(config-router)# authentication-type cleartext 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	authentication key-chain <i>key</i> {level-1 level-2} Example: switch(config-router)# authentication key-chain ISISKey level-2	Configures the authentication key used for an IS-IS area-level authentication.
Step 5	authentication-check {level-1 level-2} Example: switch(config-router)# authentication-check level-2	(Optional) Enables checking the authentication parameters in a received packet.
Step 6	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

8.6.9 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **isis authentication-type** {cleartext | md5} {level-1 | level-2}
4. **isis authentication key-chain** *key* {level-1 | level-2}
5. (Optional) **isis authentication-check** {level-1 | level-2}
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	isis authentication-type {cleartext md5} {level-1 level-2} Example: switch(config-if)# isis authentication-type cleartext level-2	Sets the authentication type for IS-IS on this interface as cleartext or as an MD5 authentication digest.
Step 4	isis authentication key-chain <i>key</i> {level-1 level-2} Example: switch(config-if)# isis authentication-key ISISKey level-2	Configures the authentication key used for IS-IS on this interface.
Step 5	isis authentication-check {level-1 level-2} Example: switch(config-if)# isis authentication-check	(Optional) Enables checking the authentication parameters in a received packet.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

8.6.10 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:

Command	Purpose
isis mesh-group {blocked mesh-id} Example: switch(config-if)# isis mesh-group 1	Adds this interface to a mesh group. The range is from 1 to 4294967295.

8.6.11 Configuring 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:

Command	Purpose
isis priority number {level-1 level-2} Example: switch(config-if)# isis priority 100 level-1	Sets the priority for DIS selection. The range is from 0 to 127. The default is 64.

8.6.12 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:

Command	Purpose
hostname dynamic Example: switch(config-router)# hostname dynamic	Enables dynamic host exchange.

8.6.13 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:

Command	Purpose
<pre>set-overload-bit {always on-startup {seconds wait-for bgp as-number}} [suppress [interlevel external]] Example: switch(config-router)# set-overload-bit on-startup 30</pre>	Sets the overload bit for IS-IS. The <i>seconds</i> range is from 5 to 86400.

8.6.14 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:

Command	Purpose
<pre>[no] attached-bit Example: switch(config-router)# no attached-bit</pre>	Configures the Level 1/Level 2 router to set the attached bit. This feature is enabled by default.

8.6.15 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:

Command	Purpose
<pre>[no] isis hello-padding Example: switch(config-if)# no isis hello-padding</pre>	Pads the hello packet to the full maximum transmission unit (MTU). The default is enabled. Use the no form of this command to configure the transient mode of hello padding.

8.6.16 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-CN advertises the smallest metric of all the more-specific routes.

BEFORE YOU BEGIN

You must enable IS-IS.

SUMMARY STEPS

1. **configure terminal**
2. **router isis *instance-tag***
3. **address-family {*ipv4* | *ipv6*} unicast**
4. **summary-address *ip-prefix/mask-len* {*level-1* | *level-2* | *level-1-2*}**
5. (Optional) **show isis [*vrf vrf-name*] {*ip* | *ipv6*} summary-address *ip-prefix* [*longer-prefixes*]**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i> Example: switch(config)# router isis Enterprise switch(config-router)#	Creates a new IS-IS instance with the configured <i>instance tag</i> .
Step 3	address-family {<i>ipv4</i> <i>ipv6</i>} unicast Example: switch(config-router)# address-family <i>ipv4</i> unicast switch(config-router-af)#	Enters address family configuration mode.
Step 4	summary-address <i>ip-prefix/mask-len</i> {<i>level-1</i> <i>level-2</i> <i>level-1-2</i>} Example: switch(config-router-af)# summary-address 192.0.2.0/24 <i>level-2</i>	Configures a summary address for an IS-IS area for IPv4 or IPv6 addresses.
Step 5	show isis [<i>vrf vrf-name</i>] {<i>ip</i> <i>ipv6</i>} summary-address <i>ip-prefix</i> [<i>longer-prefixes</i>] Example: switch(config-if)# show isis <i>ip</i> summary-address	(Optional) Displays IS-IS IPv4 or IPv6 summary address information.
Step 6	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure an IPv4 unicast summary address for IS-IS:

```
switch# configure terminal
```

```
switch(config)# router isis Enterprise
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
```

8.6.17 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.

SUMMARY STEPS

1. **configure terminal**
2. **router isis** *instance-tag*
3. **address-family** {*ipv4* | *ipv6*} **unicast**
4. **redistribute** {*bgp as* | *direct* | {*eigrp* | *isis* | *ospf* | *ospfv3* | *rip*} *instance-tag* | **static**} **route-map** *map-name*
5. (Optional) **default-information originate** [*always*] [**route-map** *map-name*]
6. (Optional) **distribute** {*level-1* | *level-2*} **into** {*level-1* | *level-2*} {**route-map** *route-map* | **all**}
7. (Optional) **show isis** [*vrf vrf-name*] {*ip* | *ipv6*} **route** *ip-prefix* [*detail* | **longer-prefixes**] [**summary** | **detail**]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i> Example: switch(config)# router isis Enterprise switch(config-router)#	Creates a new IS-IS instance with the configured <i>instance tag</i> .
Step 3	address-family { <i>ipv4</i> <i>ipv6</i> } unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address family configuration mode.
Step 4	redistribute { <i>bgp as</i> { <i>eigrp</i> <i>isis</i> <i>ospf</i> <i>ospfv3</i> <i>rip</i> } <i>instance-tag</i> static direct } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute eigrp 201 route-map ISISmap	Redistributes routes from other protocols into IS-IS.
Step 5	default-information originate [<i>always</i>] [route-map <i>map-name</i>]	(Optional) Generates a default route into IS-IS.

	Command	Purpose
	Example: <pre>switch(config-router-af)# default-information originate always</pre>	
Step 6	<pre>distribute {level-1 level-2} into {level-1 level-2} {route-map route-map all} Example: switch(config-router-af)# distribute level-1 into level-2 all</pre>	(Optional) Redistributes routes from one IS-IS level to the other IS-IS level.
Step 7	<pre>show isis [vrf vrf-name] {ip ipv6} route ip-prefix [detail longer-prefixes [summary detail]] Example: switch(config-router-af)# show isis ip route</pre>	(Optional) Shows the IS-IS routes.
Step 8	<pre>copy running-config startup-config Example: switch(config-router-af)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This example shows how to redistribute EIGRP into IS-IS:

```
switch# configure terminal
switch(config)# router isis Enterprise
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
```

8.6.18 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.

SUMMARY STEPS

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

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i> Example: <pre>switch(config)# router isis Enterprise switch(config-router)#</pre>	Creates a new IS-IS instance with the configured instance tag.
Step 3	redistribute { <i>bgp id</i> direct <i>eigrp id</i> <i>isis id</i> <i>ospf id</i> <i>rip id</i> static } route-map <i>map-name</i> Example: <pre>switch(config-router)# redistribute bgp route-map FilterExternalBGP</pre>	Redistributes the selected protocol into IS-IS through the configured route map.
Step 4	redistribute maximum-prefix <i>max</i> [<i>threshold</i>] [warning-only withdraw [<i>num-retries</i> <i>timeout</i>]] Example: <pre>switch(config-router)# redistribute maximum-prefix 1000 75 warning-only</pre>	Specifies a maximum number of prefixes that IS-IS distributes. The range is from 1 to 65535. You can optionally specify the following: <ul style="list-style-type: none"> • threshold—Percentage 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. You can optionally try to retrieve the redistributed routes. The <i>num-retries</i> range is from 1 to 12. The <i>timeout</i> is 60 to 600 seconds. The default is 300 seconds. Use the clear isis redistribution command if all routes are withdrawn.
Step 5	show running-config isis Example: <pre>switch(config-router)# show running-config isis</pre>	(Optional) Displays the IS-IS configuration.
Step 6	copy running-config startup-config Example: <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

8.6.19 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.

SUMMARY STEPS

1. **configure terminal**
2. **router isis *instance-tag***
3. **address-family ipv4 unicast**
4. **no adjacency-check**
5. **exit**
6. **address-family ipv6 unicast**
7. **no adjacency-check**
8. (Optional) **show running-config isis**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i> Example: switch(config)# router isis Enterprise switch(config-router)#	Creates a new IS-IS instance with the configured instance tag.
Step 3	address-family ipv4 unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address family configuration mode.
Step 4	no adjacency-check Example: switch(config-router-af)# no adjacency-check	Disables strict adjacency mode for the IPv4 address family.
Step 5	exit	Exits address family configuration mode.

	Command	Purpose
	Example: <pre>switch(config-router-af)# exit switch(config-router)#</pre>	
Step 6	address-family ipv6 unicast Example: <pre>switch(config-router)# address-family ipv6 unicast switch(config-router-af)#</pre>	Enters address family configuration mode.
Step 7	no adjacency-check Example: <pre>switch(config-router-af)# no adjacency-check</pre>	Disables strict adjacency mode for the IPv6 address family.
Step 8	show running-config isis Example: <pre>switch(config-router-af)# show running-config isis</pre>	(Optional) Displays the IS-IS configuration.
Step 9	copy running-config startup-config Example: <pre>switch(config-router-af)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

8.6.20 Configuring a Graceful Restart

You can configure a graceful restart for IS-IS.

BEFORE YOU BEGIN

You must enable IS-IS.
Create the VRFs.

SUMMARY STEPS

1. **configure terminal**
2. **router isis *instance-tag***
3. **graceful-restart**
4. **graceful-restart t3 *manual time***
5. (Optional) **show running-config isis**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router isis <i>instance-tag</i>	Creates a new IS-IS process with the configured name.

	Command	Purpose
	Example: <pre>switch(config)# router isis Enterprise switch(config-router)#</pre>	
Step 3	graceful-restart Example: <pre>switch(config-router)# graceful-restart</pre>	Enables a graceful restart and the graceful restart helper functionality. Enabled by default.
Step 4	graceful-restart t3 manual time Example: <pre>switch(config-router)# graceful-restart t3 manual 300</pre>	Configures the graceful restart T3 timer. The range is from 30 to 65535 seconds. The default is 60.
Step 5	show running-config isis Example: <pre>switch(config-router)# show running-config isis</pre>	(Optional) Displays the IS-IS configuration.
Step 6	copy running-config startup-config Example: <pre>switch(config-router)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

8.6.21 Configuring Virtualization

You can configure multiple IS-IS instances and multiple VRFs 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.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf_name*
3. **exit**
4. **router isis** *instance-tag*
5. (Optional) **vrf** *vrf_name*
6. **net** *network-entity-title*
7. **exit**
8. **interface** *type slot/port*
9. **vrf member** *vrf-name*
10. **{ip | ipv6} address** *ip-prefix/length*

11. `{ip | ipv6} router isis instance-tag`
12. (Optional) `show isis [vrf vrf-name] [instance-tag] interface [interface-type slot/port]`
13. (Optional) `copy running-config startup-config`

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code> Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	<code>vrf context vrf-name</code> Example: switch(config)# <code>vrf context</code> RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	<code>exit</code> Example: switch(config-vrf)# <code>exit</code> switch(config)#	Exits VRF configuration mode.
Step 4	<code>router isis instance-tag</code> Example: switch(config)# <code>router isis Enterprise</code> switch(config-router)#	Creates a new IS-IS instance with the configured instance tag.
Step 5	<code>vrf vrf-name</code> Example: switch(config-router)# <code>vrf</code> RemoteOfficeVRF switch(config-router-vrf)#	(Optional) Enters VRF configuration mode.
Step 6	<code>net network-entity-title</code> Example: switch(config-router-vrf)# <code>net</code> 47.0004.004d.0001.0001.0c11.1111.00	Configures the NET for this IS-IS instance.
Step 7	<code>exit</code> Example: switch(config-router-vrf)# <code>exit</code> switch(config-router)#	Exits router VRF configuration mode.
Step 8	<code>interface ethernet slot/port</code> Example: switch(config)# <code>interface ethernet 1/2</code> switch(config-if)#	Enters interface configuration mode.
Step 9	<code>vrf member vrf-name</code> Example:	Adds this interface to a VRF.

	Command	Purpose
	switch(config-if)# vrf member RemoteOfficeVRF	
Step 10	{ip ipv6} address ip-prefix/length Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 11	{ip ipv6} router isis instance-tag Example: switch(config-if)# ip router isis Enterprise	Associates this IPv4 or IPv6 interface with an IS-IS instance.
Step 12	show isis [vrf vrf-name] [instance-tag] interface [interface-type slot/port] Example: switch(config-if)# show isis Enterprise ethernet 1/2	(Optional) Displays IS-IS information for an interface in a VRF.
Step 13	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

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 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
```

8.6.22 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:

Command	Purpose
lsp-gen-interval [level-1 level-2] <i>lsp-max-wait [lsp-initial-wait lsp-second-wait]</i> Example: switch(config-router)# lsp-gen-interval level-1 500 500 500	Configures the IS-IS throttle for LSP generation. The optional parameters are as follows: <ul style="list-style-type: none"> 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

Command	Purpose
	<p>trigger and LSP generation. The range is from 50 to 65535 milliseconds.</p> <ul style="list-style-type: none"> lsp-second-wait—The second wait used for LSP throttle during backoff. The range is from 50 to 65535 milliseconds.
<p>max-lsp-lifetime <i>lifetime</i></p> <p>Example:</p> <pre>switch(config-router)# max-lsp-lifetime 500</pre>	<p>Sets the maximum LSP lifetime in seconds. The range is from 1 to 65535. The default is 1200.</p>
<p>metric-style transition</p> <p>Example:</p> <pre>switch(config-router)# metric-style transition</pre>	<p>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.</p>
<p>spf-interval [<i>level-1</i> <i>level-2</i>]</p> <p><i>spf-max-wait</i> [<i>spf-initial-wait</i> <i>spf-second-wait</i>]</p> <p>Example:</p> <pre>switch(config-router)# spf-interval level-2 500 500 500</pre>	<p>Configures the interval between LSA arrivals. The optional parameters are as follows:</p> <ul style="list-style-type: none"> 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.

You can use the following optional command in router address configuration mode:

Command	Purpose
<p>adjacency-check</p> <p>Example:</p> <pre>switch(config-router-af)# adjacency-check</pre>	<p>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.</p>

You can use the following optional commands in interface configuration mode to tune IS-IS:

Command	Purpose
<p>isis csnp-interval <i>seconds</i> [<i>level-1</i> <i>level-2</i>]</p> <p>Example:</p> <pre>switch(config-if)# isis csnp-interval 20</pre>	<p>Sets the complete sequence number PDU (CNSP) interval in seconds for IS-IS. The range is from 1 to 65535. The default is 10.</p>
<p>isis hello-interval <i>seconds</i> [<i>level-1</i> <i>level-2</i>]</p> <p>Example:</p>	<p>Sets the hello interval in seconds for IS-IS. The range is from 1 to 65535. The default is 10.</p>

Command	Purpose
switch(config-if)# isis hello-interval 20	
isis hello-multiplier <i>num</i> [level-1 level-2] Example: switch(config-if)# isis hello-multiplier 20	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.
isis lsp-interval <i>milliseconds</i> Example: switch(config-if)# isis lsp-interval 20	Sets the interval in milliseconds between LSPs sent on this interface during flooding. The range is from 10 to 65535. The default is 33.

8.7 Verifying the IS-IS Configuration

To display the IS-IS configuration, perform one of the following tasks:

Command	Purpose
show isis [<i>instance-tag</i>] adjacency [<i>interface</i>] [detail summary] [vrf <i>vrf-name</i>]	Displays the IS-IS adjacencies. Use the clear isis adjacency command to clear these statistics.
show isis [<i>instance-tag</i>] database [level-1 level-2] [detail summary] [<i>LSP ID</i>] [{ ip ipv6 } prefix <i>ip-prefix</i>] [router-id <i>router-id</i>] [adjacency <i>node-id</i>] [zero-sequence]} [vrf <i>vrf-name</i>]	Displays the IS-IS LSP database.
show isis [<i>instance-tag</i>] hostname [vrf <i>vrf-name</i>]	Displays the dynamic host exchange information.
show isis [<i>instance-tag</i>] interface [brief <i>interface</i>] [level-1 level-2] [vrf <i>vrf-name</i>]	Displays the IS-IS interface information.
show isis [<i>instance-tag</i>] mesh-group [<i>mesh-id</i>] [vrf <i>vrf-name</i>]	Displays the mesh group information.
show isis [<i>instance-tag</i>] protocol [vrf <i>vrf-name</i>]	Displays information about the IS-IS protocol.
show isis [<i>instance-tag</i>] { ip ipv6 } redistribute route [<i>ip-address</i> summary] [[<i>ip-prefix</i>] [longer-prefixes [summary]]] [vrf <i>vrf-name</i>]	Displays the IS-IS route redistribution information.
show isis [<i>instance-tag</i>] { ip ipv6 } route [<i>ip-address</i> summary] [<i>ip-prefix</i>] [longer-prefixes [summary]]] [detail] [vrf <i>vrf-name</i>]	Displays the IS-IS route table.
show isis [<i>instance-tag</i>] rrm [<i>interface</i>] [vrf <i>vrf-name</i>]	Displays the IS-IS interface retransmission information.
show isis [<i>instance-tag</i>] srm [<i>interface</i>] [vrf <i>vrf-name</i>]	Displays the IS-IS interface flooding information.
show isis [<i>instance-tag</i>] ssn [<i>interface</i>] [vrf <i>vrf-name</i>]	Displays the IS-IS interface PSNP information.
show isis [<i>instance-tag</i>] { ip ipv6 } summary-address [<i>ip-address</i>] [<i>ip-prefix</i>] [vrf <i>vrf-name</i>]	Displays the IS-IS summary address information.

Command	Purpose
<i>vrf-name</i>]	
show running-configuration isis	Displays the current running IS-IS configuration.
show tech-support isis [<i>detail</i>]	Displays the technical support details for IS-IS.

8.8 Monitoring IS-IS

To display IS-IS statistics, use the following commands:

Command	Purpose
show isis [<i>instance-tag</i>] adjacency [<i>interface</i>] [<i>system-ID</i>] [<i>detail</i>] [<i>summary</i>] [<i>vrf vrf-name</i>]	Displays the IS-IS adjacency statistics.
show isis [<i>instance-tag</i>] database [<i>level-1</i> <i>level-2</i>] [<i>detail</i> <i>summary</i>] [<i>lsip</i>] {{ <i>adjacency id</i> } { <i>ip</i> <i>ipv6</i> } <i>prefix prefix</i> } [<i>router-id id</i>] [<i>zero-sequence</i>]} [<i>vrf vrf-name</i>]	Displays the IS-IS database statistics.
show isis [<i>instance-tag</i>] statistics [<i>interface</i>] [<i>vrf vrf-name</i>]	Displays the IS-IS interface statistics.
show isis { <i>ip</i> <i>ipv6</i> } route-map statistics redistribute { <i>bgp id</i> <i>eigrp id</i> <i>isis id</i> <i>ospf id</i> <i>rip id</i> <i>static</i> } [<i>vrf vrf-name</i>]	Displays the IS-IS redistribution statistics.
show isis route-map statistics distribute { <i>level-1</i> <i>level-2</i> } into { <i>level-1</i> <i>level-2</i> }} [<i>vrf vrf-name</i>]	Displays IS-IS distribution statistics for routes distributed between levels.
show isis [<i>instance-tag</i>] spf-log [<i>detail</i>] [<i>vrf vrf-name</i>]	Displays the IS-IS SPF calculation statistics.
show isis [<i>instance-tag</i>] traffic [<i>interface</i>] [<i>vrf vrf-name</i>]	Displays the IS-IS traffic statistics.

To clear IS-IS configuration statistics, perform one of the following tasks:

Command	Purpose
clear isis [<i>instance-tag</i>] adjacency [* [<i>interface</i>] [<i>system-id id</i>]] [<i>vrf vrf-name</i>]	Clears the IS-IS adjacency statistics.
clear isis { <i>ip</i> <i>ipv6</i> } route-map statistics redistribute { <i>bgp id</i> <i>direct</i> <i>eigrp id</i> <i>isis id</i> <i>ospf id</i> <i>rip id</i> <i>static</i> } [<i>vrf vrf-name</i>]	Clears the IS-IS redistribution statistics.
clear isis route-map statistics distribute { <i>level-1</i> <i>level-2</i> } into { <i>level-1</i> <i>level-2</i> } [<i>vrf vrf-name</i>]	Clears IS-IS distribution statistics for routes distributed between levels.
clear isis [<i>instance-tag</i>] statistics [* <i>interface</i>] [<i>vrf vrf-name</i>]	Clears the IS-IS interface statistics.
clear isis [<i>instance-tag</i>] traffic [* <i>interface</i>] [<i>vrf vrf-name</i>]	Clears the IS-IS traffic statistics.

8.9 Configuration Examples for IS-IS

This 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 192.0.2.1/24
 isis circuit-type level-1
 ip router isis Enterprise
```

CHAPTER 9 Configuring Basic BGP

This chapter describes how to configure Border Gateway Protocol (BGP) on the Inspur INOS-CN device.

9.1 About Basic BGP

Inspur INOS-CN 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). See the “Load Sharing and Multipath” section for more information.

9.1.1 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 autonomous system (AS) numbers in plain-text notation or as.dot notation and 4-byte AS numbers in plain-text notation.

When BGP is configured with a 4-byte AS number, the **route-target auto VXLAN** command cannot be used because the AS number along with the VNI (which is already a 3-byte value) is used to generate the route target. For more information, see the *Inspur CN12900 Series INOS-CN VXLAN Configuration Guide*.

9.1.2 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 Table 9-1.

Table 9-1 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	220	Applied to routes originated by the router.

9.1.3 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-CN supports the following peer configuration options:

- Individual IPv4 or IPv6 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-CN 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-CN 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.

9.1.4 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-CN 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.

9.1.5 BGP Path Selection

BGP supports sending and receiving multiple paths per prefix and advertising such paths. For information on configuring additional BGP paths, see Chapter 10, “Configuring Advanced BGP.”

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.

Inspur INOS-CN implements the BGP best-path algorithm in the following steps:

-
- | | |
|---------------|--|
| Step 1 | Compares two paths to determine which is better. |
| Step 2 | Explores all paths and determines in which order to compare the paths to select the overall best path. |
| Step 3 | 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.

Step 1—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-CN uses to compare two paths to determine the better path:

1. Inspur INOS-CN chooses a valid path for comparison. (For example, a path that has an unreachable next hop is not valid.)
2. Inspur INOS-CN chooses the path with the highest weight.
3. Inspur INOS-CN chooses the path with the highest local preference.
4. If one of the paths is locally originated, Inspur INOS-CN chooses that path.
5. Inspur INOS-CN chooses the path with the shorter AS-path.
6. Inspur INOS-CN chooses the path with the lower origin. The Interior Gateway Protocol (IGP) is considered lower than EGP.
7. Inspur INOS-CN chooses the path with the lower multiexit discriminator (MED).

You can configure Inspur INOS-CN to always perform the best-path algorithm MED comparison, regardless of the peer autonomous system in the paths. Otherwise, Inspur INOS-CN performs a MED comparison that depends on the AS-path attributes of the two paths being compared:

- a. If a path has no AS-path or the AS-path starts with an AS_SET, the path is internal and Inspur INOS-CN compares the MED to other internal paths.
 - b. If the AS-path starts with an AS_SEQUENCE, the peer autonomous system is the first AS number in the sequence and Inspur INOS-CN compares the MED to other paths that have the same peer autonomous system.
 - c. 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-CN compares the MED to other internal paths.
 - d. 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-CN compares the MED to other paths that have the same peer autonomous system.
8. If one path is from an internal peer and the other path is from an external peer, Inspur INOS-CN chooses the path from the external peer.
 9. If the paths have different IGP metrics to their next-hop addresses, Inspur INOS-CN chooses the path with the lower IGP metric.
 10. Inspur INOS-CN uses the path that was selected by the best-path algorithm the last time that it was run.

If all path parameters in Step 1 through Step 9 are the same, you can configure the best-path algorithm to compare the router IDs. If the path includes an originator attribute, Inspur INOS-CN uses that attribute as the router ID to compare to; otherwise, Inspur INOS-CN uses the router ID of the peer that sent the path. If the paths have different router IDs, Inspur INOS-CN chooses the path with the lower router ID.

11. Inspur INOS-CN selects the path with the shorter cluster length. If a path was not received with a cluster list attribute, the cluster length is 0.
12. Inspur INOS-CN 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.

Step 2—Determining the Order of Comparisons

The second step of the BGP best-path algorithm implementation is to determine the order in which Inspur INOS-CN compares the paths:

1. Inspur INOS-CN partitions the paths into groups. Within each group, Inspur INOS-CN compares the MED among all 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-CN chooses just one group that contains all the paths.
2. Inspur INOS-CN 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-CN 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-CN compares it with the next path in the group.
3. Inspur INOS-CN forms a set of paths that contain the best path selected from each group in Step 2. Inspur INOS-CN selects the overall best path from this set of paths by going through them as in Step 2.

Step 3—Determining the Best-Path Change Suppression

The next part of the implementation is to determine whether Inspur INOS-CN 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-CN 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. 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.

9.1.6 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.

9.1.7 BGP Prefix Independent Convergence

The BGP prefix independent convergence (PIC) edge feature achieves faster convergence in the forwarding plane for BGP IP routes to a BGP backup path when there is a link failure.

The BGP PIC edge feature improves BGP convergence after a network failure. This convergence applies to edge failures in an IP network. This feature creates and stores a backup path in the routing information base (RIB) and forwarding information base (FIB) so that when the primary path fails, the backup path can immediately take over, enabling fast failover in the forwarding plane. BGP PIC edge supports only IPv4 address families.

When BGP PIC edge is configured, BGP calculates a 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 remote next hop through 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.

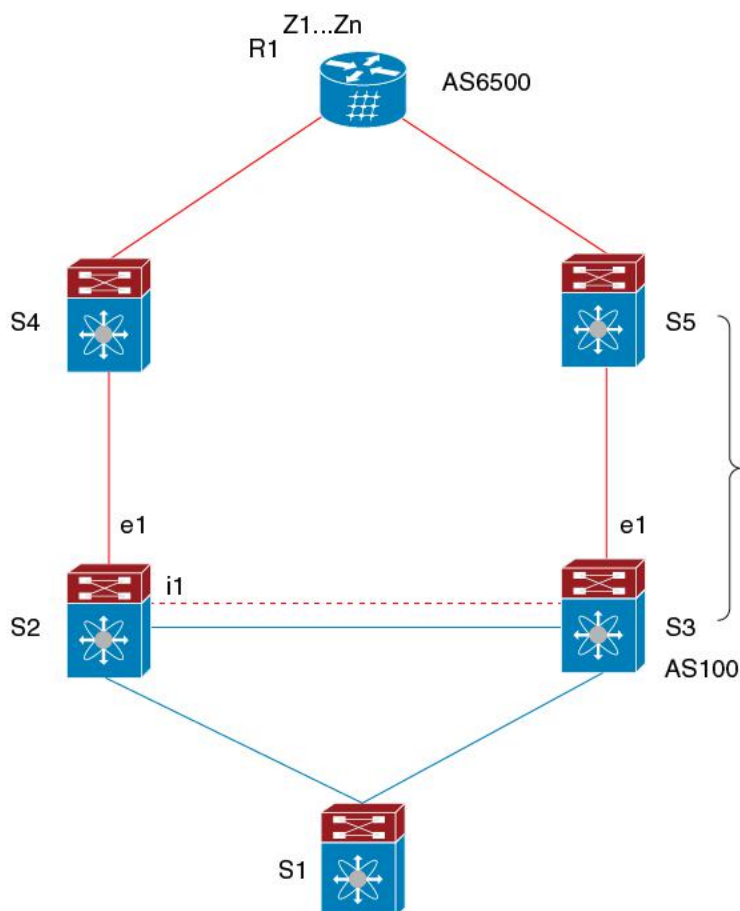
This feature detects both local interface failures and remote interface or link failures and triggers the use of the backup path.

BGP PIC edge supports both unipath and multipath.

BGP PIC Edge Unipath

The following figure shows a BGP PIC edge unipath topology.

Figure 9-1 BGP PIC Edge Unipath



In this figure:

- eBGP sessions are between S2-S4 and S3-S5.
- The 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:
 - A primary path through S4
 - A backup path through S5

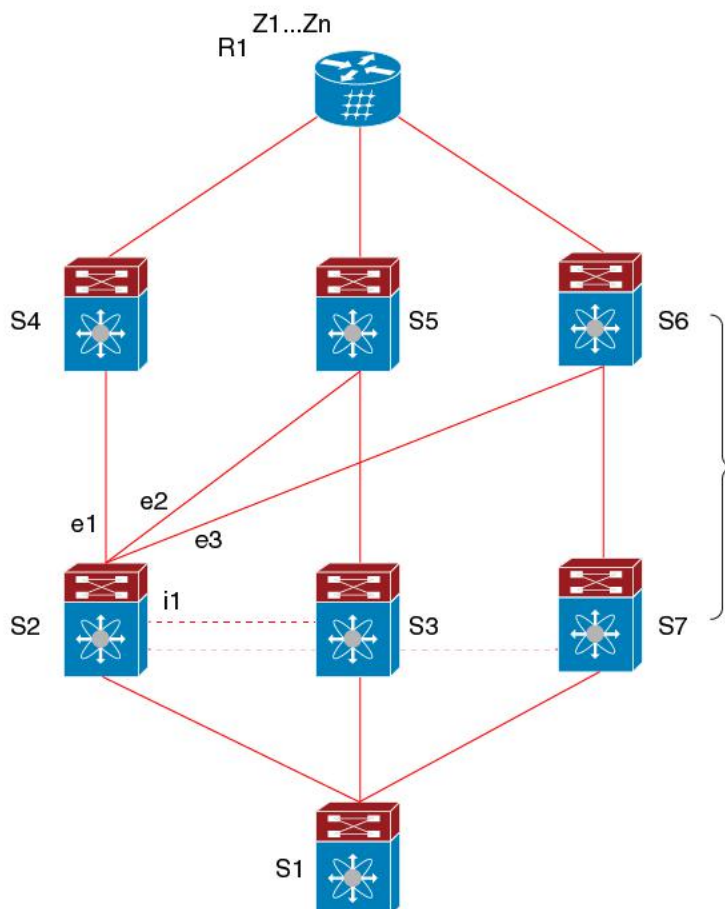
In this example, S3 advertises to S2 the prefixes Z1...Zn to reach (with itself as the next hop). With BGP PIC edge enabled, BGP on S2 installs both the best path (through S4) and the backup path (through S3 or S5) toward the AS6500 into the RIB. Then the RIB downloads both routes to the FIB.

If the S2-S4 link goes down, the FIB on S2 detects the link failure. It automatically switches from the primary path to the backup path and points to the new next hop S3. Traffic is quickly rerouted due to the local fast re-convergence in the FIB. After learning of the link failure event, BGP on S2 recomputes the best path (which is the previous backup path), removes next hop S4 from the RIB, and reinstalls S3 as the primary next hop into the RIB. BGP also computes a new backup path, if any, and notifies the RIB. With the support of the BGP PIC edge feature, the FIB can switch to the available backup route instantly upon detection of a link failure on the primary route without waiting for BGP to select the new best path and converge to achieve a fast reroute.

BGP PIC Edge with Multipaths

The following figure shows a BGP PIC edge multipath topology.

Figure 9-2 BGP PIC Edge Multipaths



In this topology, there are six paths for a given prefix:

- 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 eBGP 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.

When the Equal Cost Multipath Protocol (ECMP) is enabled, none of the multipaths can be selected as the backup path.

For multipaths with the backup path scenario, faster convergence is not expected with simultaneous failure of all active multipaths.

BGP PIC Feature Support Matrix

Table 9-2 BGP PIC Feature Support Matrix

BGP PIC	IPv4 Unicast	IPv6 Unicast
Edge unipath	Yes	No
Edge with multipath (multiple active ECMP, only one backup)	Yes	No

9.1.8 BGP Virtualization

BGP supports virtual routing and forwarding (VRF) instances.

9.2 Licensing Requirements for Basic BGP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	BGP requires an Essentials license package.

9.3 Prerequisites for BGP

BGP has the following prerequisites:

- You must enable BGP (see the “Enabling BGP” section).
- 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.

9.4 Guidelines and Limitations for BGP

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 BGP/eBGP multihop sessions.
- You must define the BGP update source when using a link-local IPv6 address.
- Specify a BGP policy if you configure redistribution.
- Define the BGP router ID within a VRF.
- For IPv6 neighbors, Inspur recommends that you configure a router ID per VRF. If a VRF does not have any IPv4 interfaces, the IPv6 BGP neighbor will not come up because its router ID must be an IPv4 address. The numerically lowest loopback IPv4 address is elected to be the router ID. If a loopback address does not exist, the lowest IP address from the VRF interfaces is elected. If that does not exist, the BGP neighbor relationship is not established.
- 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 VRFs, install the Advanced Services license and enter the desired VRF.
- Although the **show ip bgp** commands are available for verifying the BGP configuration, Inspur recommends using the **show bgp** commands instead.

The following guidelines and limitations apply to the BGP PIC edge feature:

- BGP PIC edge is supported for the Inspur CN12900 Series switches beginning with Inspur INOS-CN Release 9.2(2).
- BGP PIC edge supports only the IPv4 address family.
- BGP PIC edge is not supported for VXLAN EVPN and MPLS segment routing.
- BGP PIC edge supports only one repair (backup) path.
- When BGP PIC edge is configured, BGP supports a maximum of 63 active paths. A backup path is not added if there are 64 active paths.

9.5 Default Settings

Table 9-2 lists the default settings for BGP parameters.

Table 9-3 Default BGP Parameters

Parameters	Default
BGP feature	Disabled
Keep alive interval	60 seconds

Hold timer	180 seconds
BGP PIC edge	Disabled
Auto-summary	Always disabled
Synchronization	Always disabled

9.6 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.

9.6.1 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 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)#
```

9.6.2 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.

This 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)#
```

This 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)#
```

9.6.3 Neighbor Configuration Mode

Inspur INOS-CN provides the neighbor configuration mode to configure BGP peers. You can use neighbor configuration mode to configure all parameters for a peer.

This 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)#
```

This 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)#
```

9.6.4 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.

With the introduction of RFC 5549, you can configure an IPv4 address family for a neighbor with an IPv6 address.

This example shows how to enter the IPv4 neighbor address family configuration mode for a neighbor with an IPv4 address:

```
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 the IPv4 neighbor address family configuration mode for a neighbor with an IPv6 address:

```
switch(config)# router bgp 64496
switch(config-router)# neighbor 2001:db8::/64 eui64
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)#
```

This example shows how to enter the VRF IPv4 neighbor address family configuration mode for a neighbor with an IPv4 address:

```
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)#
```

This example shows how to enter the VRF IPv4 neighbor address family configuration mode for a neighbor with an IPv6 address:

```
switch(config)# router bgp 64497
switch(config-router)# vrf vrf_A
switch(config-router-vrf)# neighbor 2001:db8::/64 eui64
switch(config-router-vrf-neighbor)# address-family ipv4 unicast
switch(config-router-vrf-neighbor-af)#
```

9.7 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.

9.7.1 Enabling BGP

You must enable BGP before you can configure BGP.

SUMMARY STEPS

1. **configure terminal**
2. **feature bgp**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature bgp Example: switch(config)# feature bgp	Enables BGP.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

Use the **no feature bgp** command to disable BGP and remove all associated configuration.

Command	Purpose
no feature bgp Example: switch(config)# no feature bgp	Disables BGP and removes all associated configuration.

9.7.2 Creating a BGP Instance

You can create a BGP instance and assign a router ID to the BGP instance.

BEFORE YOU BEGIN

You must enable BGP.

BGP must be able to obtain a router ID (for example, a configured loopback address).

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. (Optional) **router-id** *ip-address*
4. (Optional) **address-family** {*ipv4* | *ipv6*} {*unicast* | *multicast*}
5. (Optional) **network** {*ip-address/length* | *ip-address mask mask*} [*route-map map-name*]
6. (Optional) **show bgp all**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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	router-id <i>ip-address</i> Example: switch(config-router)# router-id 192.0.2.255	(Optional) Configures the BGP router ID. This IP address identifies this BGP speaker.
Step 4	address-family { <i>ipv4</i> <i>ipv6</i> } { <i>unicast</i> <i>multicast</i> } Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	(Optional) Enters global address family configuration mode for the IPv4 or IPv6 address family.
Step 5	network { <i>ip-address/length</i> <i>ip-address mask mask</i> } [<i>route-map map-name</i>] Example: switch(config-router-af)# network 10.10.10.0/24 Example: switch(config-router-af)# network 10.10.10.0 mask 255.255.255.0	(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	show bgp all Example: switch(config-router-af)# show bgp all	(Optional) Displays information about all BGP address families.
Step 7	copy running-config startup-config Example: switch(config-router-af)# copy	(Optional) Saves this configuration change.

Command	Purpose
<code>running-config startup-config</code>	

Use the **no router bgp** command to remove the BGP process and the associated configuration.

Command	Purpose
no router bgp <i>autonomous-system-number</i> Example: <code>switch(config)# no router bgp 201</code>	Deletes the BGP process and the associated configuration.

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
```

9.7.3 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:

Command	Purpose
restart bgp <i>instance-tag</i> Example: <code>switch(config)# restart bgp 201</code>	Restarts the BGP instance and resets or reestablishes all peering sessions.

9.7.4 Shutting Down BGP

You can shut down the BGP and gracefully disable BGP while retaining the configuration.

To shut down BGP, use the following command in router configuration mode:

Command	Purpose
shutdown Example: <code>switch(config-router)# shutdown</code>	Gracefully shuts down BGP.

9.7.5 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.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** {*ip-address* | *ipv6-address*} **remote-as** *as-number*
4. (Optional) **description** *text*
5. (Optional) **timers** *keepalive-time hold-time*
6. (Optional) **shutdown**
7. **address-family** {*ipv4* | *ipv6*} {*unicast* | *multicast*}
8. (Optional) **weight** *value*
9. (Optional) **show bgp** {*ipv4* | *ipv6*} {*unicast* | *multicast*} **neighbors**
10. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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	neighbor { <i>ip-address</i> <i>ipv6-address</i> } remote-as <i>as-number</i> Example: switch(config-router)# neighbor 209.165.201.1 remote-as 64497 switch(config-router-neighbor)#	Configures the IPv4 or IPv6 address and AS number for a remote BGP peer. The <i>ip-address</i> format is x.x.x.x. The <i>ipv6-address</i> format is A:B::C:D.
Step 4	description <i>text</i> Example: switch(config-router-neighbor)# description Peer Router B switch(config-router-neighbor)#	(Optional) Adds a description for the neighbor. The description is an alphanumeric string up to 80 characters.
Step 5	timers <i>keepalive-time hold-time</i> Example: switch(config-router-neighbor)# timers 30 90	(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	shutdown Example: 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	address-family { <i>ipv4</i> <i>ipv6</i> } { <i>unicast</i> <i>multicast</i> }	Enters neighbor address family configuration mode for the IPv4 or IPv6 address family.

	Command	Purpose
	Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	
Step 8	<pre>weight value</pre> Example: <pre>switch(config-router-neighbor-af)# weight 100</pre>	(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	<pre>show bgp {ipv4 ipv6} {unicast multicast} neighbors</pre> Example: <pre>switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors</pre>	(Optional) Displays information about BGP peers.
Step 10	<pre>copy running-config startup-config</pre> Example: <pre>switch(config-router-neighbor-af) copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This 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-af)# address-family ipv4 unicast
switch(config-router-neighbor-af)# weight 100
switch(config-router-neighbor-af)# copy running-config startup-
config
```

9.7.6 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.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *prefix* **remote-as** **route-map** *map-name*
4. (Optional) **show bgp** {**ipv4** | **ipv6**} {**unicast** | **multicast**} **neighbors**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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	neighbor <i>prefix</i> remote-as route-map <i>map-name</i> Example: switch(config-router)# neighbor 192.0.2.0/8 remote-as routemap BGPPeers switch(config-router-neighbor)#	Configures the IPv4 or IPv6 prefix and a route map for the list of accepted AS numbers for the remote BGP peers. The <i>prefix</i> format for IPv4 is x.x.x.x/length. The length range is from 1 to 32. The <i>prefix</i> format for IPv6 is A:B::C:D/length. The length range is from 1 to 128. The <i>map-name</i> can be any case-sensitive, alphanumeric string up to 63 characters.
Step 4	show bgp { ipv4 ipv6 } { unicast multicast } neighbors Example: switch(config-router-neighbor-af)# show bgp ipv4 unicast neighbors	(Optional) Displays information about BGP peers.
Step 5	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

9.7.7 Configuring BGP PIC Edge

Follow these steps to configure BGP PIC edge.

BEFORE YOU BEGIN

You must enable BGP.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **neighbor** *ip-address*
4. **remote-as** *as-number*
5. **address-family ipv4 unicast**
6. **additional-paths install backup**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 64496 switch(config-router)#	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	neighbor <i>ip-address</i> Example: switch(config-router)# neighbor 209.165.201.1 switch(config-router-neighbor)#	Configures the IPv4 address for a remote BGP peer. The <i>ip-address</i> format is x.x.x.x.
Step 4	remote-as <i>as-number</i> Example: switch(config-router-neighbor)# remote-as 64497	Configures the AS number for a remote BGP peer.
Step 5	address-family ipv4 unicast Example: switch(config-router-neighbor)# address- family ipv4 unicast switch(config-router-neighbor-af)#	Enters neighbor address family configuration mode for the IPv4 address family.
Step 6	additional-paths install backup	Enables BGP to install the backup path to the routing

Command	Purpose
Example: switch(config-router-neighbor-af)# additional-paths install backup	table.
Step 7 copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to configure the device to support BGP PIC edge in an 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
 neighbor 1.1.1.6
  remote-as 200
  address-family ipv4 unicast
  additional-paths install backup
  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 are 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:

```
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>
99.0.0.0/24, 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: Wed Jul 11 19:00:12 2018
Vobj id: 141 orig_as: 65002 peer_as: 65100 rnh: 10.3.0.2
10.4.0.2 Ethernet8/4 DMAC: 0431.10d8.4dfd
  packets: 2 bytes: 3484 Repair path 10.3.0.2 Ethernet8/3 DMAC: 0431.10d8.4dfd
  packets: 0 bytes: 1
```

9.7.8 Clearing BGP Information

To clear BGP information, use the following commands:

Command	Purpose
clear bgp all { <i>neighbor</i> * <i>as-number</i> <i>peer-template name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	Clears one or more neighbors from all address families. * clears all neighbors in all address families. The arguments are as follows: <ul style="list-style-type: none"> <i>neighbor</i>—IPv4 or IPv6 address of a neighbor. <i>as-number</i>—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. <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. <i>prefix</i>—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. <i>vrf-name</i>—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 <i>vrf-name</i>]	Clears route flap dampening networks in all address families. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp all flap-statistics [vrf <i>vrf-name</i>]	Clears route flap statistics in all address families. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp { ipv4 ipv6 } { unicast multicast } dampening [vrf <i>vrf-name</i>]	Clears route flap dampening networks in the selected address family. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp { ipv4 ipv6 } { unicast multicast } flap-statistics [vrf <i>vrf-name</i>]	Clears route flap statistics in the selected address family. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.
clear bgp { ipv4 ipv6 } { unicast multicast } { <i>neighbor</i> * <i>as-number</i> <i>peer-template name</i>	Clears one or more neighbors from the selected address family. * clears all neighbors in the

Command	Purpose
<i>prefix</i> { vrf <i>vrf-name</i> }	<p>address family. The arguments are as follows:</p> <ul style="list-style-type: none"> • <i>neighbor</i>—The IPv4 or IPv6 address of a neighbor. • <i>as-number</i>— 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. • <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • <i>prefix</i>—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp { ip { unicast multicast } } { <i>neighbor</i> * <i>as-number</i> peer-template <i>name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	<p>Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows:</p> <ul style="list-style-type: none"> • <i>neighbor</i>—IPv4 or IPv6 address of a neighbor. • <i>as-number</i>— 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. • <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • <i>prefix</i>—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp dampening [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	<p>Clears route flap dampening in one or more networks. The arguments are as follows:</p> <ul style="list-style-type: none"> • <i>ip-neighbor</i>—IPv4 address of a neighbor. • <i>ip-prefix</i>—IPv4. All neighbors within that prefix are cleared. • <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any

Command	Purpose
	case-sensitive, alphanumeric string up to 64 characters.
clear ip bgp flap-statistics [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	Clears route flap statistics in one or more networks. The arguments are as follows: <ul style="list-style-type: none"> • <i>ip-neighbor</i>—IPv4 address of a neighbor. • <i>ip-prefix</i>—IPv4. All neighbors within that prefix are cleared. • <i>vrf-name</i>—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}} { <i>neighbor</i> * <i>as-number</i> peer-template <i>name</i> <i>prefix</i> } [vrf <i>vrf-name</i>]	Clears one or more neighbors. * clears all neighbors in the address family. The arguments are as follows: <ul style="list-style-type: none"> • <i>neighbor</i>—IPv4 or IPv6 address of a neighbor. • <i>as-number</i>— 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. • <i>name</i>—Peer template name. The name can be any case-sensitive, alphanumeric string up to 64 characters. • <i>prefix</i>—IPv4 or IPv6 prefix. All neighbors within that prefix are cleared. • <i>vrf-name</i>—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 [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	Clears route flap dampening in one or more networks. The arguments are as follows: <ul style="list-style-type: none"> • <i>ip-neighbor</i>—IPv4 address of a neighbor. • <i>ip-prefix</i>—IPv4. All neighbors within that prefix are cleared. • <i>vrf-name</i>—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 [<i>ip-neighbor</i> <i>ip-prefix</i>] [vrf <i>vrf-name</i>]	Clears route flap statistics one or more networks. The arguments are as follows: <ul style="list-style-type: none"> • <i>ip-neighbor</i>—IPv4 address of a neighbor. • <i>ip-prefix</i>—IPv4. All neighbors within that

Command	Purpose
	<ul style="list-style-type: none"> prefix are cleared. <i>vrf-name</i>—VRF name. All neighbors in that VRF are cleared. The name can be any case-sensitive, alphanumeric string up to 64 characters.

9.8 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 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] community { regex <i>expression</i> [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 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] community-list list-name [vrf vrf-name]	Displays the BGP routes that match a BGP community list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity { regex <i>expression</i> generic [non-transitive transitive] <i>aa4:nn</i> [exact-match] } [vrf vrf-name]	Displays the BGP routes that match a BGP extended community.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity-list list-name [exact-match] } [vrf vrf-name]	Displays the BGP routes that match a BGP extended community list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] { dampening dampened-paths [regex <i>expression</i>] } [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 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] history-paths [regex <i>expression</i>] [vrf vrf-name]	Displays the BGP route history paths.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] filter-list list-name [vrf vrf-name]	Displays the information for the BGP filter list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] neighbors [<i>ip-address</i> <i>ipv6-prefix</i>] [vrf vrf-name]	Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors.

Command	Purpose
show bgp {ipv4 ipv6} {unicast multicast} [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} {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} {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} {unicast multicast} [ip-address ipv6-prefix] received-paths [vrf vrf-name]	Displays the BGP paths stored for soft reconfiguration.
show bgp {ipv4 ipv6} {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} {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 {ip ipv6} bgp options	Displays the BGP status and configuration information.
show {ip ipv6} mbgp options	Displays the BGP status and configuration information.
show running-configuration bgp	Displays the current running BGP configuration.

9.9 Monitoring BGP Statistics

To display BGP statistics, use the following commands:

Command	Purpose
show bgp {ipv4 ipv6} {unicast multicast} [ip-address ipv6-prefix] flap-statistics [vrf vrf-name]	Displays the BGP route flap statistics.
show bgp sessions [vrf vrf-name]	Displays the BGP sessions for all peers.
show bgp statistics	Displays the BGP statistics.

9.10 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:DB8:0:1::55 remote-as 64496
switch(config-router-neighbor)# address-family ipv6
unicast
switch(config-router-af)# next-hop-self
```

CHAPTER 10 Configuring Advanced BGP

This chapter describes how to configure advanced features of the Border Gateway Protocol (BGP) on the Inspur INOS-CN device.

10.1 About Advanced BGP

BGP is an interdomain routing protocol that provides loop-free routing between organizations or autonomous systems. Inspur INOS-CN 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.

10.1.1 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-CN 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-CN 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.

10.1.2 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.

10.1.3 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-CN supports the following 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-CN 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-CN 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.

10.1.4 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.

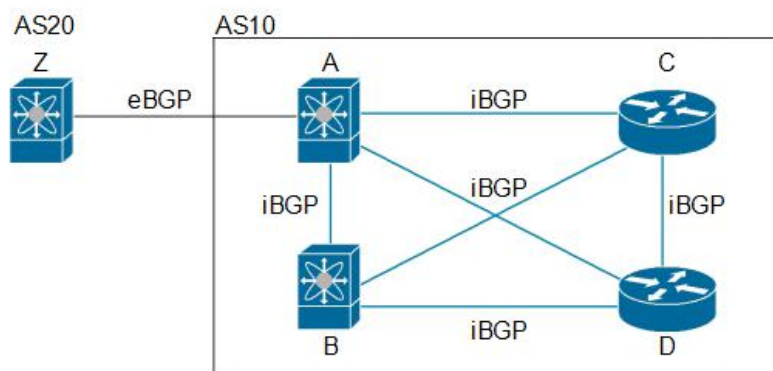
Typically eBGP peerings need to be over directly connected interfaces so that convergence will be faster when the interface goes down.

10.1.5 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).

Figure 10-1 shows an iBGP network within a larger BGP network.

Figure 10-1 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 failover.

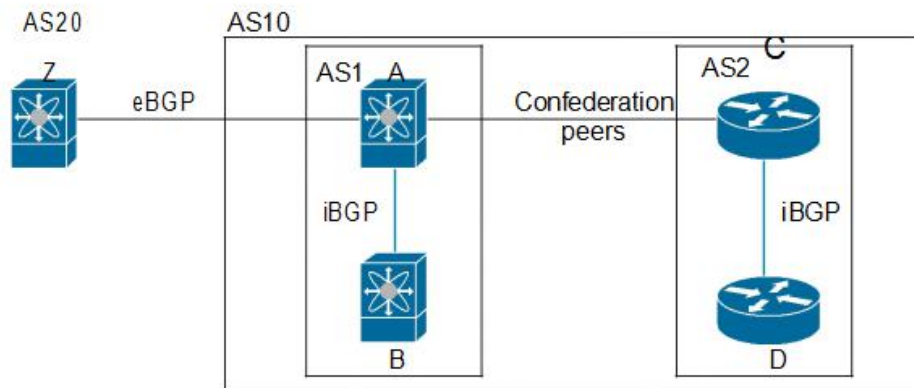
You should use loopback interfaces for establishing eBGP peering sessions because loopback interfaces are less susceptible to interface flapping. An interface flap occurs when the interface is administratively brought up or down because of a failure or maintenance issue.

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.

Figure 10-2 shows the BGP network from Figure 10-1, split into two subautonomous systems and one confederation.

Figure 10-2 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 in Figure 10-1.

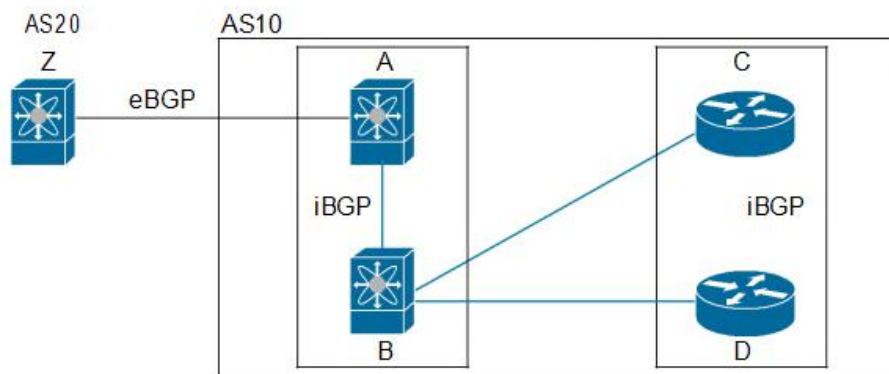
Route Reflector

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.

Figure 10-1 shows a simple iBGP configuration with four meshed iBGP speakers (routers A, B, C, and D). Without route reflectors, when router A receives a route from an external neighbor, it advertises 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 Figure 10-3, 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 10-3 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.

10.1.6 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-CN 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.

10.1.7 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.

10.1.8 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

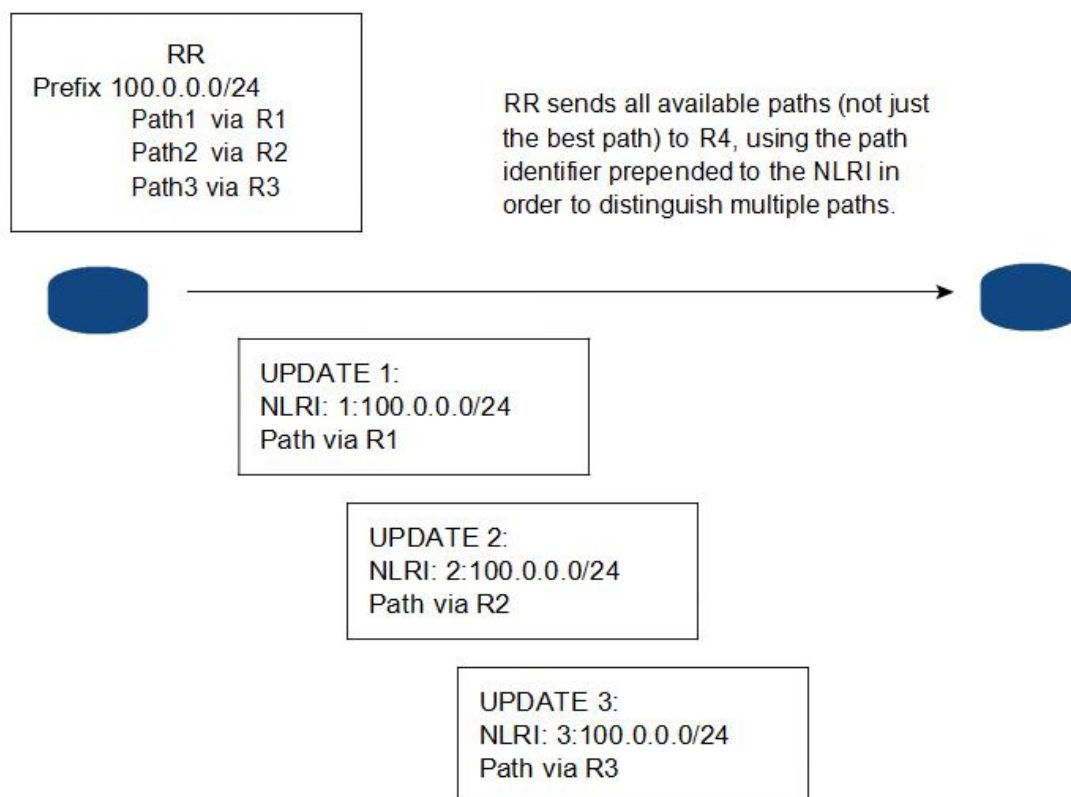
BGP supports sending and receiving multiple paths per prefix and advertising such paths.

10.1.9 BGP Additional Paths

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.

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. Figure 10-4 illustrates the BGP additional paths capability.

Figure 10-4 BGP Route Advertisement with the Additional Paths Capability



10.1.10 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.

10.1.11 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.

10.1.12 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:

- The next hop becomes unreachable.
- The next hop becomes reachable.
- The fully recursed Interior Gateway Protocol (IGP) metric to the next hop changes.
- The first hop IP address or first hop interface changes.
- The next hop becomes connected.
- The next hop becomes unconnected.
- The next hop becomes a local address.
- The 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.

10.1.13 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. iBGP is not 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
```

10.1.14 BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4 and IPv6. 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information.

10.1.15 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.

10.1.16 Multiprotocol BGP

BGP on Inspur INOS-CN 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.

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.

10.1.17 BGP Monitoring Protocol

The BGP Monitoring Protocol (BMP) monitors BGP updates and peer statistics and is supported for all Inspur CN12900 Series switches.

Using this protocol, the BGP speaker connects to external BMP servers and sends them information regarding BGP events. A maximum of two BMP servers can be configured in a BGP speaker, and each BGP peer can be configured for monitoring by all or a subset of the BMP servers. The BGP speaker does not accept any information from the BMP server.

10.1.18 Weighted ECMP over BGP

Basic equal-cost multipath (ECMP) allows multiple routes to the same destination with different next hops and load balances the routed traffic over those multiple next hops.

To ensure optimal and efficient load balancing and to administer more control over routing behavior, weighted ECMP distributes traffic to BGP peers that are only one hop away. Rather than equally distributing traffic over multiple paths, weighted ECMP distributes traffic based on the bandwidth available for each next hop.

Weighted ECMP is not configurable. It is enabled automatically when a BGP peer sends attributes with the BGP extended community link bandwidth. The **show ip route ip-address detail vrf all | i bw** command shows which BGP peers are sending BGP extended attributes with the bandwidth.

10.1.19 Graceful Restart and High Availability

Inspur INOS-CN 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-CN 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-CN applies the startup configuration, BGP reestablishes peering sessions and relearns the routes.

A Inspur INOS-CN 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.

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.

10.1.20 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 reestablished.
- 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.

10.1.21 Virtualization Support

You can configure one BGP instance. BGP supports virtual routing and forwarding (VRF) instances.

10.2 Licensing Requirements for Advanced BGP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	BGP requires an Essentials license package.

10.3 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.

10.4 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.
- You must define the BGP update source when using a link-local IPv6 address.
- 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.
- 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.
- BFD is supported for single-hop eBGP peers and single-hop iBGP peers. To enable BFD for single-hop iBGP peers, you must configure the update-source option on the physical interface.
- BFD is supported for multi-hop eBGP peers and multi-hop iBGP peers.

- BGP supports prefix-based peers, but BFD is not supported for prefix-based peers.
- 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.
- If you use the `aggregate-address` command to configure aggregate addresses and the `suppress-fib-pending` command to suppress BGP routes, lossless traffic for aggregates cannot be ensured on BGP or system triggers.
- When you enable FIB suppression on the switch and route programming fails in the hardware, BGP advertises routes that are not programmed locally in the hardware.
- If you disable a command in the neighbor, template peer, template peer-session, or template peer-policy configuration mode (and the `inherit peer` or `inherit peer-session` command is present), you must use the `default` keyword to return the command to its default state. For example, to disable the `update-source loopback 0` command from the running configuration, you must enter the `default update-source loopback 0` command.
- When next-hop-self is configured for route-reflector clients, the route reflector advertises routes to its clients with itself as the next hop.

10.5 Default Settings for Advanced BGP

Table 10-1 lists the default settings for advanced BGP parameters.

Table 10-1 Default BGP Parameters

Parameters	Default
BGP feature	Disabled
BGP additional paths	Disabled
Hold timer	180 seconds
Keep alive interval	60 seconds
Dynamic capability	Enabled

10.6 Configuring Advanced BGP

10.6.1 Enabling IP Forward on an Interface

To use RFC 5549, you must configure at least one IPv4 address. If you do not want to configure an IPv4 address, you must enable the IP forward feature to use RFC 5549.

1. `configure terminal`
2. `interface type slot/port`
3. `ip forward`
4. (Optional) `copy running-config startup-config`

DETAILED STEPS

	Command	Purpose
Step 1	<code>configure terminal</code>	Enters global configuration mode.

	Command	Purpose
	Example: switch# configure terminal switch(config)#	
Step 2	interface <i>type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ip forward Example: switch(config-if)# ip forward	Allows IPv4 traffic on the interface even when there is no IP address configuration on that interface.
Step 4	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Saves this configuration change.

10.6.2 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.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-session** *template-name*
4. (Optional) **password number** *password*
5. (Optional) **timers** *keepalive hold*
6. **exit**
7. **neighbor** *ip-address remote-as as-number*
8. **inherit peer-session** *template-name*
9. (Optional) **description** *text*
10. (Optional) **show bgp peer-session** *template-name*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-session <i>template-name</i> Example: switch(config-router)# template peer-session BaseSession switch(config-router-stmp)#	Enters peer-session template configuration mode.
Step 4	password <i>number password</i> Example: switch(config-router-stmp)# password 0 test	(Optional) Adds the cleartext password <i>test</i> to the neighbor. The password is stored and displayed in type 3 encrypted form (3DES).
Step 5	timers <i>keepalive hold</i> Example: switch(config-router-stmp)# timers 30 90	(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	exit Example: switch(config-router-stmp)# exit switch(config-router)#	Exits peer-session template configuration mode.
Step 7	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#	Places the router in the neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	inherit peer-session <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer-session BaseSession switch(config-router-neighbor)	Applies a peer-session template to the peer.
Step 9	description <i>text</i> Example: switch(config-router-neighbor)# description Peer Router A switch(config-router-neighbor)	(Optional) Adds a description for the neighbor.
Step 10	show bgp peer-session <i>template-name</i>	(Optional) Displays the peer-policy template.

Command	Purpose
Example: <pre>switch(config-router-neighbor)# show bgp peer-session BaseSession</pre>	
Step 11 copy running-config startup-config Example: <pre>switch(config-router-neighbor)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

Use the **show bgp neighbor** command to see the template applied.

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
```

10.6.3 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-CN 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 .

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer-policy** *template-name*
4. (Optional) **advertise-active-only**
5. (Optional) **maximum-prefix** *number*
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {*ipv4* | *ipv6*} {**multicast** | **unicast**}
9. **inherit peer-policy** *template-name* *preference*
10. (Optional) **show bgp peer-policy** *template-name*
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Enables BGP and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer-policy <i>template-name</i> Example: switch(config-router)# template peer-policy BasePolicy switch(config-router-ptmp)#	Creates a peer-policy template.
Step 4	advertise-active-only Example: switch(config-router-ptmp)# advertise-active-only	(Optional) Advertises only active routes to the peer.
Step 5	maximum-prefix <i>number</i> Example: switch(config-router-ptmp)# maximum-prefix 20	(Optional) Sets the maximum number of prefixes allowed from this peer.
Step 6	exit Example: switch(config-router-ptmp)# exit switch(config-router)#	Exits peer-policy template configuration mode.
Step 7	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 8	address-family { <i>ipv4</i> <i>ipv6</i> } { multicast unicast } Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters global address-family configuration mode for the specified address family.
Step 9	inherit peer-policy <i>template-name</i> <i>preference</i> Example: switch(config-router-neighbor-af)# inherit peer-policy BasePolicy 1	Applies a peer-policy template to the peer address-family configuration and assigns the preference value for this peer policy.
Step 10	show bgp peer-policy <i>template-name</i>	(Optional) Displays the peer-policy template.

Command	Purpose
Example: <pre>switch(config-router-neighbor-af) # show bgp peer-policy BasePolicy</pre>	
Step 11 copy running-config startup-config Example: <pre>switch(config-router-neighbor-af) # copy running-config startup-config</pre>	(Optional) Saves this configuration change.

Use the **show bgp neighbor** command to see the template applied.

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
```

10.6.4 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.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *autonomous-system-number*
3. **template peer** *template-name*
4. **inherit peer-session** *template-name*
5. **address-family** {*ipv4* | *ipv6*} {*multicast* | *unicast*}
6. **inherit peer** *template-name*
7. **exit**
8. **timers** *keepalive hold*
9. **exit**
10. **neighbor** *ip-address* **remote-as** *as-number*
11. **inherit peer** *template-name*
12. **timers** *keepalive hold*
13. (Optional) **show bgp peer-template** *template-name*
14. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>autonomous-system-number</i> Example: switch(config)# router bgp 65535	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	template peer <i>template-name</i> Example: switch(config-router)# template peer BasePeer switch(config-router-neighbor)#	Enters peer template configuration mode.
Step 4	inherit peer-session <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer-session BaseSession	(Optional) Inherits a peer-session template in the peer template.
Step 5	address-family { <i>ipv4</i> <i>ipv6</i> } { multicast unicast } Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	(Optional) Configures the global address-family configuration mode for the specified address family.
Step 6	inherit peer <i>template-name</i> Example: switch(config-router-neighbor-af)# inherit peer BasePolicy	(Optional) Applies a peer template to the neighbor address-family configuration.
Step 7	exit Example: switch(config-router-neighbor-af)# exit switch(config-router-neighbor)#	Exits BGP neighbor address-family configuration mode.
Step 8	timers <i>keepalive hold</i> Example: switch(config-router-neighbor)# timers 45 100	(Optional) Adds the BGP timer values to the peer. These values override the timer values in the peer-session template, BaseSession.
Step 9	exit Example: switch(config-router-neighbor)# exit switch(config-router)#	Exits BGP peer template configuration mode.
Step 10	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example:	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.

	Command	Purpose
	<pre>switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#</pre>	
Step 11	<pre>inherit peer <i>template-name</i> Example: switch(config-router-neighbor)# inherit peer BasePeer</pre>	Inherits the peer template.
Step 12	<pre>timers <i>keepalive hold</i> Example: switch(config-router-neighbor)# timers 60 120</pre>	(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	<pre>show bgp peer-template <i>template-name</i> Example: switch(config-router-neighbor-af)# show bgp peer-template BasePeer</pre>	(Optional) Displays the peer template.
Step 14	<pre>copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

Use the **show bgp neighbor** command to see the template applied.

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
```

10.6.5 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.

When a BGP peer that is part of a prefix peering disconnects, Inspur INOS-CN holds its peer structures for a defined prefix peer timeout value. An established peer can reset and reconnect without danger of being blocked because other peers have consumed all slots for that prefix peering.

To configure the BGP prefix peering timeout value, use the following command in neighbor configuration mode:

Command	Purpose
timers prefix-peer-timeout <i>value</i> Example: <pre>switch(config-router-neighbor)# timers prefix-peer-timeout 120</pre>	Configures the timeout value for prefix peering. The range is from 0 to 1200 seconds. The default value is 30.

To configure the maximum number of peers, use the following command in neighbor configuration mode:

Command	Purpose
maximum-peers <i>value</i> Example: <pre>switch(config-router-neighbor)# maximum-peers 120</pre>	Configures the maximum number of peers for this prefix peering. The range is from 1 to 1000.

This example shows how to configure a prefix peering that accepts up to 10 peers:

```
switch# configure terminal
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)#
```

Use the **show bgp ipv4 unicast neighbors** command to show the details of the configuration for that prefix peering with a list of the currently accepted instances and the counts of active, maximum concurrent, and total accepted peers.

10.6.6 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:

Command	Purpose
password [0 3 7] <i>string</i> Example: <pre>switch(config-router-neighbor)# password BGPpassword</pre>	Configures an MD5 password for BGP neighbor sessions.

10.6.7 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-CN automatically attempts a soft reset for the session.

To configure soft reconfiguration inbound, use the following command in neighbor address-family configuration mode:

Command	Purpose
soft-reconfiguration inbound Example: <pre>switch(config-router-neighbor-af)# soft-reconfiguration inbound</pre>	Enables soft reconfiguration to store the inbound BGP route updates. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

To reset a BGP neighbor session, use the following command in any mode:

Command	Purpose
clear bgp {ipv4 ipv6} {unicast multicast} ip-address soft {in out} Example: <pre>switch# clear bgp ip unicast 192.0.2.1 soft in</pre>	Resets the BGP session without tearing down the TCP session.

10.6.8 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:

Command	Purpose
next-hop-self Example: <pre>switch(config-router-neighbor-af)# next-hop-self</pre>	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.
next-hop-third-party Example: <pre>switch(config-router-neighbor-af)# next-hop-third-party</pre>	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.

10.6.9 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:

Command	Purpose
nexthop trigger-delay {critical non-critical} milliseconds Example:	Specifies the next-hop address tracking delay timer for critical next-hop reachability routes and for noncritical routes. The range is from 1 to

<pre>switch(config-router-af)# nexthop trigger-delay critical 5000</pre>	4294967295 milliseconds. The critical timer default is 3000. The noncritical timer default is 10000.
--	--

10.6.10 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:

Command	Purpose
<pre>nexthop route-map name</pre> <p>Example:</p> <pre>switch(config-router-af)# nexthop route-map nextHopLimits</pre>	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.

10.6.11 Shrinking Next-Hop Groups When A Session Goes Down

You can configure BGP to shrink ECMP groups in an accelerated way when a session goes down.

This feature applies to the following BGP path failure events:

- Any single or multiple Layer 3 link failures
- Line card failures
- BFD failure detections for BGP neighbors
- Administrative shutdown of BGP neighbors (using the **shutdown** command)

The accelerated handling of the first two events (Layer 3 link failures and line card failures) is enabled by default and does not require a configuration command to be enabled.

To configure the accelerated handling of the last two events, use the following command in router configuration mode:

Command	Purpose
<pre>neighbor-down fib-accelerate</pre> <p>Example:</p> <pre>switch(config-router)# neighbor-down fib-accelerate</pre>	Withdraws the corresponding next hop from all next-hop groups (ECMP groups and single next-hop routes) whenever a BGP session goes down. <p>Note This command applies to both IPv4 and IPv6 address-family routes.</p> <p>Note This command is supported only in a BGP-only environment where all non-direct routes are installed by BGP.</p>

10.6.12 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:

Command	Purpose
dont-capability-negotiate Example: <pre>switch(config-router-neighbor) # dont-capability-negotiate</pre>	Disables capabilities negotiation. You must manually reset the BGP sessions after configuring this command.

10.6.13 Disabling Policy Batching

In BGP deployments where prefixes have unique attributes, BGP tries to identify routes with similar attributes to bundle in the same BGP update message. To avoid the overhead of this additional BGP processing, you can disable batching.

Inspur recommends that you disable policy batching for BGP deployments that have a large number of routes with unique next hops.

To disable policy batching, use the following command in router configuration mode:

Command	Purpose
disable-policy-batching Example: <pre>switch(config-router) # disable-policy-batching</pre>	Disables the batching evaluation of prefix advertisements to all peers.

10.6.14 Configuring BGP Additional Paths

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. To do so, use the following commands in neighbor address-family configuration mode:

Command	Purpose
<pre>[no] capability additional-paths send [disable]</pre> Example: <pre>switch(config-router-neighbor-af) # capability additional-paths send</pre>	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.
<pre>[no] capability additional-paths receive [disable]</pre> Example: <pre>switch(config-router-neighbor-af) # capability additional-paths receive</pre>	Advertises the capability to receive additional paths from the BGP peer. The disable option disables the advertising capability of receiving additional paths. The no form of this command disables the

Command	Purpose
	capability of receiving additional paths.
show bgp neighbor Example: <pre>switch(config-router-neighbor-af)# show bgp neighbor</pre>	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.

This example shows how to configure BGP to advertise the capability to send and receive additional paths to the BGP peer:

```
switch# configure terminal
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
```

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. To do so, use the following commands in address-family configuration mode:

Command	Purpose
[no] additional-paths send Example: <pre>switch(config-router-af)# additional-paths send</pre>	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.
[no] additional-paths receive Example: <pre>switch(config-router-af)# additional-paths receive</pre>	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 receive capability.
show bgp neighbor Example: <pre>switch(config-router-af)# show bgp neighbor</pre>	Displays whether the local peer has advertised the additional paths send or receive capability to the remote peer.

This example shows how to enable the additional paths send and receive capability for all neighbors under the specified address family for which this capability has not been disabled:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths send
switch(config-router-af)# additional-paths receive
```

Configuring Advertised Paths

You can specify the paths that are advertised for BGP. To do so, use the following commands in route-map configuration mode:

Command	Purpose
<pre>[no] set ip next-hop unchanged</pre> <p>Example:</p> <pre>switch(config-route-map)# set ip next-hop unchanged</pre>	Specifies an unchanged next-hop IP address.
<pre>[no] set path-selection {all backup best2 multipaths} advertise</pre> <p>Example:</p> <pre>switch(config-route-map)# set path-selection all advertise</pre>	<p>Specifies the paths to be advertised for a given prefix. You can use one of the following options:</p> <ul style="list-style-type: none"> • all—Advertises all available valid paths. • backup—Advertises paths marked as backup paths. This option requires that backup paths be enabled using the additional-path install backup command. • best2—Advertises the second best path, which is the best path of the remaining available paths, except the already calculated best path. • multipaths—Advertises all multipaths. This option requires that multipaths be enabled using the maximum-paths command. <p>Note: If there are no multipaths, the backup and best2 options are the same. If there are multipaths, best2 is the first path on the list of multipaths while backup is the best path of all available paths, except the calculated best path and multipaths.</p> <p>The no form of this command specifies that only the best path be advertised.</p>
<pre>show bgp {ipv4 ipv6} unicast [ip-address ipv6-prefix] [vrf vrf-name]</pre> <p>Example:</p> <pre>switch(config-route-map)# show bgp ipv4 unicast</pre>	Displays the path ID for the additional paths of a prefix and advertisement information for these paths.

This example shows how to specify that all paths be advertised for the specified prefix:

```
switch# configure terminal
switch(config)# route-map PATH_SELECTION_RMAP
switch(config-route-map)# match ip address prefix-list p1
switch(config-route-map)# set path-selection all advertise
```

Configuring Additional Path Selection

You can configure the capability of selecting additional paths for a prefix. To do so, use the following commands in address-family configuration mode:

Command	Purpose
<pre>[no] additional-paths selection route-map map-name</pre>	Configures the capability of selecting additional paths for a prefix.

<p>Example:</p> <pre>switch(config-router-af)# additional-paths selection route-map map1</pre>	<p>The no form of this command disables the additional paths selection capability.</p>
<p>show bgp {ipv4 ipv6} unicast [<i>ip-address</i> <i>ipv6-prefix</i>] [vrf <i>vrf-name</i>]</p> <p>Example:</p> <pre>switch(config-route-af)# show bgp ipv4 unicast</pre>	<p>Displays the path ID for the additional paths of a prefix and advertisement information for these paths.</p>

This example shows how to configure the additional paths selection under the specified address family:

```
switch# configure terminal
switch(config)# router bgp 100
switch(config-router)# address-family ipv4 unicast
switch(config-router-af)# additional-paths selection route-map PATH_SELECTION_RMAP
```

10.6.15 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:

Command	Purpose
<p>disable-connected-check</p> <p>Example:</p> <pre>switch(config-router-neighbor)# disable-connected-check</pre>	<p>Disables checking whether or not a single-hop eBGP peer is directly connected. You must manually reset the BGP sessions after using this command.</p>

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:

Command	Purpose
<p>ebgp-multihop <i>ttl-value</i></p> <p>Example:</p> <pre>switch(config-router-neighbor)# ebgp-multihop 5</pre>	<p>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.</p>

Disabling a Fast External Fallover

By default, the Inspur INOS-CN 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:

Command	Purpose
no fast-external-fallover Example: <pre>switch(config-router)# no fast-external-fallover</pre>	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 excessive AS numbers in the AS-path attribute.

To discard routes that have excessive AS numbers in the AS-path attribute, use the following command in router configuration mode:

Command	Purpose
maxas-limit <i>number</i> Example: <pre>switch(config-router)# maxas-limit 50</pre>	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 subautonomous systems.

To configure eBGP local AS support, use the following command in neighbor configuration mode:

Command	Purpose
local-as <i>number</i> [no-prepend [replace-as [dual-as]]] Example: <pre>switch(config-router-neighbor)# local-as 1.1</pre>	Configures eBGP to prepend the local AS <i>number</i> to the AS_PATH attribute. The AS <i>number</i> 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.

This example shows how to configure local AS support on a VRF:

```
switch# configure terminal
switch(config)# router bgp 1
switch(config-router)# vrf test
switch(config-router-vrf)# local-as 1
switch(config-router-vrf)# show running-config
bgp
```

10.6.16 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.

To configure a BGP confederation identifier, use the following command in router configuration mode:

Command	Purpose
confederation identifier <i>as-number</i> Example: <pre>switch(config-router)# confederation identifier 4000</pre>	Configures a confederation identifier for an AS confederation. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

To configure the autonomous systems that belong to the AS confederation, use the following command in router configuration mode:

Command	Purpose
bgp confederation peers <i>as-number</i> [<i>as-number2...</i>] Example: <pre>switch(config-router)# bgp confederation peers 5 33 44</pre>	Specifies a list of autonomous systems that belong to the confederation. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

10.6.17 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 .

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **cluster-id** *cluster-id*
4. **address-family** {*ipv4* | *ipv6*} {*unicast* | *multicast*}
5. (Optional) **client-to-client reflection**
6. **exit**
7. **neighbor** *ip-address* **remote-as** *as-number*
8. **address-family** {*ipv4* | *ipv6*} {*unicast* | *multicast*}
9. **route-reflector-client**
10. (Optional) **show bgp** {*ipv4* | *ipv6*} {*unicast* | *multicast*} **neighbors**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp as-number Example: switch(config)# router bgp 65535 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	cluster-id cluster-id Example: switch(config-router)# cluster-id 192.0.2.1	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	address-family {ipv4 ipv6} {unicast multicast} Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters router address-family configuration mode for the specified address family.
Step 5	client-to-client reflection Example: 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	exit Example: switch(config-router-neighbor)# exit switch(config-router)#	Exits router address configuration mode.
Step 7	neighbor ip-address remote-as as-number Example: switch(config-router)# neighbor 192.0.2.10 remote-as 65535 switch(config-router-neighbor)#	Configures the IP address and AS number for a remote BGP peer.
Step 8	address-family {ipv4 ipv6} {unicast multicast} Example: switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#	Enters neighbor address-family configuration mode for the specified address family.
Step 9	route-reflector-client Example: 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	show bgp {ipv4 ipv6} {unicast 	(Optional) Displays the BGP peers.

Command	Purpose
multicast} neighbors Example: <pre>switch(config-router-neighbor-af) # show bgp ipv4 unicast neighbors</pre>	
Step 11 copy running-config startup-config Example: <pre>switch(config-router-neighbor-af) # copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This example shows how to configure the router as a route reflector and add one neighbor as a client:

```
switch# configure terminal
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
```

10.6.18 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.

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. **configure terminal**
2. **router bgp as-number**
3. **neighbor ip-address remote-as as-number**
4. (Optional) **update-source interface number**
5. **address-family {ipv4 | ipv6} {unicast | multicast}**
6. **route-reflector-client**
7. **route-map map-name out**
8. (Optional) **show bgp {ipv4 | ipv6} {unicast | multicast} neighbors**

9. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router bgp as-number Example: <pre>switch(config)# router bgp 200 switch(config-router)#</pre>	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor ip-address remote-as as-number Example: <pre>switch(config-router)# neighbor 192.0.2.12 remote-as 200 switch(config-router-neighbor)#</pre>	Configures the IP address and AS number for a remote BGP peer.
Step 4	update-source interface number Example: <pre>switch(config-router-neighbor)# update-source loopback 300</pre>	(Optional) Specifies and updates the source of the BGP session.
Step 5	address-family {ipv4 ipv6} {unicast multicast} Example: <pre>switch(config-router-neighbor)# address-family ipv4 unicast switch(config-router-neighbor-af)#</pre>	Enters router address-family configuration mode for the specified address family.
Step 6	route-reflector-client Example: <pre>switch(config-router-neighbor-af)# route-reflector-client</pre>	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	route-map map-name out Example: <pre>switch(config-router-neighbor-af)# route-map setrrnh out</pre>	Applies the configured BGP policy to outgoing routes.
Step 8	show bgp {ipv4 ipv6} {unicast multicast} [ip-address ipv6-prefix] route-map map-name [vrf vrf-name] Example: <pre>switch(config-router-neighbor-af)# show bgp ipv4 unicast route-map setrrnh</pre>	(Optional) Displays the BGP routes that match the route map.
Step 9	copy running-config startup-config	(Optional) Saves this configuration change.

Command	Purpose
Example: <pre>switch(config-router-neighbor-af) # copy running-config startup-config</pre>	

This example shows how to configure the next hop on reflected routes on a BGP route reflector using an outbound route map:

```
switch# configure terminal
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 ospf 1 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
```

10.6.19 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:

Command	Purpose
<pre>dampening [{half-life reuse-limit suppress-limit max-suppress-time route-map map-name}]</pre> Example: <pre>switch(config-router-af)# dampening route-map bgpDamp</pre>	Disables capabilities negotiation. The parameter values are as follows: <ul style="list-style-type: none"> • 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.

10.6.20 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
maximum-paths [<i>ibgp</i>] <i>maxpaths</i> Example: switch(config-router-af)# maximum-paths 8	Configures the maximum number of equal-cost paths for load sharing. The default is 1.

10.6.21 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:

Command	Purpose
maximum-prefix <i>maximum</i> [<i>threshold</i>] [<i>restart time</i> <i>warning-only</i>] Example: switch(config-router-neighbor-af)# maximum-prefix 12	Configures the maximum number of prefixes from a peer. The parameter ranges are as follows: <ul style="list-style-type: none"> <i>maximum</i>—The range is from 1 to 300000. <i>Threshold</i>—The range is from 1 to 100 percent. The default is 75 percent. <i>time</i>—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.

10.6.22 Configuring Dynamic Capability

You can configure dynamic capability for a BGP peer.

To configure dynamic capability, use the following command in neighbor configuration mode:

Command	Purpose
dynamic-capability Example: switch(config-router-neighbor)# dynamic-capability	Enables dynamic capability. This command triggers an automatic notification and session reset for the BGP neighbor sessions.

10.6.23 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
<pre>aggregate-address ip-prefix/length [as-set] [summary-only] [advertise-map map-name] [attribute-map map-name] [suppress-map map-name]</pre> <p>Example:</p> <pre>switch(config-router-af)# aggregate-address 192.0.2.0/8 as-set</pre>	<p>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:</p> <ul style="list-style-type: none"> • 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 keyword and argument specify the route map used to select attribute information from selected routes. • The attribute-map keyword and argument specify the route map used to select attribute information from the aggregate. • The suppress-map keyword and argument conditionally filter more specific routes.

10.6.24 Suppressing BGP Routes

You can configure Inspur INOS-CN to advertise newly learned BGP routes only after these routes are confirmed by the Forwarding Information Base (FIB) and programmed in the hardware. After the routes are programmed, subsequent changes to these routes do not require this hardware-programming check.

To suppress BGP routes, use the following command in router configuration mode:

Command	Purpose
<pre>suppress-fib-pending</pre> <p>Example:</p> <pre>switch(config-router)# suppress-fib-pending</pre>	<p>Suppresses newly learned BGP routes (IPv4 or IPv6) from being advertised to downstream BGP neighbors until the routes have been programmed in the hardware.</p>

10.6.25 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 nonexistent map**—Defines the prefix that must exist in the BGP table before BGP propagates a route that matches the advertise map. The nonexistent 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.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **neighbor** *ip-address* **remote-as** *as-number*
4. **address-family** {*ipv4* | *ipv6*} {**unicast** | **multicast**}
5. **advertise-map** *adv-map* {**exist-map** *exist-rmap* | **non-exist-map** *nonexist-rmap*}
6. (Optional) **show bgp** {*ipv4* | *ipv6*} {**unicast** | **multicast**} **neighbors**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65534 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family { <i>ipv4</i> <i>ipv6</i> } { unicast multicast }	Enters address-family configuration mode.
Step 5	advertise-map <i>adv-map</i> { exist-map <i>exist-rmap</i> non-exist-map <i>nonexist-rmap</i> }	Configures BGP to conditionally advertise routes based on the two configured route maps: <ul style="list-style-type: none"> • <i>adv-map</i>—Specifies a route map with match statements that the route must pass before BGP passes the route to the next route map. The <i>adv-map</i> is a case-sensitive, alphanumeric string up to 63 characters. • <i>exist-rmap</i>—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 <i>exist-rmap</i> is a case-sensitive, alphanumeric string up to 63 characters. • <i>nonexist-rmap</i>—Specifies a route map with match

Command	Purpose
	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 <i>nonexist-rmap</i> is a case-sensitive, alphanumeric string up to 63 characters.
Step 6 <code>show bgp {ipv4 ipv6} {unicast multicast} neighbors</code> Example: <pre>switch(config-router-neighbor-af) # show bgp ipv4 unicast neighbors</pre>	(Optional) Displays information about BGP and the configured conditional advertisement route maps.
Step 7 <code>copy running-config startup-config</code> Example: <pre>switch(config-router-neighbor-af) # copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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 65534
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 172.16.201.0/27
```

10.6.26 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 route for redistributed routes.

BEFORE YOU BEGIN

You must enable BGP.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **address-family** {ipv4 | ipv6} {unicast | multicast}
4. **redistribute** {direct | {eigrp | isis | ospf | ospfv3 | rip} *instance-tag* | static} **route-map** *map-name*
5. (Optional) **default-metric** *value*

6. (Optional) copy running-config startup-config

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	address-family { <i>ipv4</i> <i>ipv6</i> } { <i>unicast</i> <i>multicast</i> } Example: switch(config-router)# address-family <i>ipv4 unicast</i> switch(config-router-af)#	Enters address-family configuration mode.
Step 4	redistribute { <i>direct</i> { <i>eigrp</i> <i>isis</i> <i>ospf</i> <i>ospfv3</i> <i>rip</i> } <i>instance-tag</i> <i>static</i> } route-map <i>map-name</i> Example: switch(config-router-af)# redistribute <i>eigrp 201 route-map Eigrpmap</i>	Redistributes routes from other protocols into BGP.
Step 5	default-metric <i>value</i> Example: switch(config-router-af)# default-metric <i>33</i>	(Optional) Generates a default route into BGP.
Step 6	copy running-config startup-config Example: switch(config-router-af)# copy <i>running-config startup-config</i>	(Optional) Saves this configuration change.

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
```

10.6.27 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.

SUMMARY STEPS

1. **configure terminal**
2. **route-map allow permit**
3. **exit**
4. **ip route *ip-address network-mask null null-interface-number***
5. **router bgp *as-number***
6. **address-family {*ipv4 | ipv6*} unicast**
7. **default-information originate**
8. **redistribute static route-map allow**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	route-map allow permit Example: switch(config)# route-map allow permit switch(config-route-map)#	Enters router map configuration mode and defines the conditions for redistributing routes.
Step 3	exit Example: switch(config-route-map)# exit switch(config)#	Exits router map configuration mode.
Step 4	ip route <i>ip-address network-mask null null-interface-number</i> Example: switch(config)# ip route 0.0.0.0/0 null 0	Configures the IP address.
Step 5	router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Enters BGP mode and assigns the AS number to the local BGP speaker.
Step 6	address-family {<i>ipv4 ipv6</i>} unicast Example: switch(config-router)# address-family ipv4 unicast switch(config-router-af)#	Enters address family configuration mode.
Step 7	default-information originate Example: switch(config-router-af)# default-information originate	Advertises the default route.
Step 8	redistribute static route-map allow	Redistributes the default route.

Command	Purpose
Example: <pre>switch(config-router-af)# redistribute static route-map allow</pre>	
Step 9 <code>copy running-config startup-config</code> Example: <pre>switch(config-router-af)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

10.6.28 Tuning BGP

You can tune BGP characteristics through a series of optional parameters.

To tune BGP, use the following optional commands in router configuration mode:

Command	Purpose
<pre>bestpath [always-compare-med as-path multipath-relax compare-routerid cost-community ignore igp-metric ignore med {confed missing-as-worst non-deterministic}]</pre> Example: <pre>switch(config-router)# bestpath always-compare-med</pre>	Modifies the best-path algorithm. The optional parameters are as follows: <ul style="list-style-type: none"> • 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. • igp-metric ignore—Ignores the Interior Gateway Protocol (IGP) metric for next hop during best-path selection. This option is supported beginning with INOS-CN Release 9.2(2). • 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.
<pre>enforce-first-as</pre> Example:	Enforces the neighbor autonomous system to be the first AS number listed in the AS_path attribute for eBGP.

Command	Purpose
<pre>switch(config-router)# enforce-first-as</pre> <p>log-neighbor-changes Example:</p> <pre>switch(config-router)# log-neighbor-changes</pre>	<p>Generates a system message when any neighbor changes state.</p> <p>Note To suppress neighbor status change messages for a specific neighbor, you can use the log-neighbor-changes disable command in router address-family configuration mode.</p>
<p>router-id <i>id</i></p> <p>Example:</p> <pre>switch(config-router)# router-id 10.165.20.1</pre>	<p>Manually configures the router ID for this BGP speaker.</p>
<p>timers [bestpath-delay <i>delay</i> bgp <i>keepalive holdtime</i> prefix-peer-timeout <i>timeout</i>]</p> <p>Example:</p> <pre>switch(config-router)# timers bgp 90 270</pre>	<p>Sets the BGP timer values. The optional parameters are as follows:</p> <ul style="list-style-type: none"> <i>delay</i>—Initial best-path timeout value after a restart. The range is from 0 to 3600 seconds. The default value is 300. <i>keepalive</i>—BGP session keepalive time. The range is from 0 to 3600 seconds. The default value is 60. <i>holdtime</i>—BGP session hold time. The range is from 0 to 3600 seconds. The default value is 180. <i>timeout</i>—Prefix peer timeout value. The range is from 0 to 1200 seconds. The default value is 30. <p>You must manually reset the BGP sessions after configuring this command.</p>

To tune BGP, use the following optional commands in router address-family configuration mode:

Command	Purpose
<p>distance <i>ebgp-distance ibgp-distance local-distance</i></p> <p>Example:</p> <pre>switch(config-router-af)# distance 20 100 200</pre>	<p>Sets the administrative distance for BGP. The range is from 1 to 255. The defaults are as follows:</p> <ul style="list-style-type: none"> <i>ebgp-distance</i>—20. <i>ibgp-distance</i>—200. <i>local-distance</i>—220. Local-distance is the administrative distance used for aggregate discard routes when they are installed in the RIB.
<p>log-neighbor-changes [disable]</p> <p>Example:</p>	<p>Generates a system message when this specific neighbor changes state.</p> <p>The disable option suppresses neighbor status</p>

```
switch(config-router-af)#
log-neighbor-changes disable
```

change messages for this specific neighbor.

To tune BGP, use the following optional commands in neighbor configuration mode:

Command	Purpose
<p>description <i>string</i></p> <p>Example:</p> <pre>switch(config-router-neighbor)# description main site</pre>	<p>Sets a descriptive string for this BGP peer. The string can be up to 80 alphanumeric characters.</p>
<p>low-memory exempt</p> <p>Example:</p> <pre>switch(config-router-neighbor)# low-memory exempt</pre>	<p>Exempts this BGP neighbor from a possible shutdown due to a low memory condition.</p>
<p>transport connection-mode passive</p> <p>Example:</p> <pre>switch(config-router-neighbor)# transport connection-mode passive</pre>	<p>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.</p>
<p>[no default] remove-private-as [all replace-as]</p> <p>Example:</p> <pre>switch(config-router-neighbor)# remove-private-as</pre>	<p>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.</p> <p>The optional parameters are as follows:</p> <ul style="list-style-type: none"> • no—Disables the command. • default—Moves the command to its default mode. • all—Removes all private-as numbers from the AS-path. • replace-as—Replaces all private AS numbers with the replace-as AS-path value.
<p>update-source <i>interface-type number</i></p> <p>Example:</p> <pre>switch(config-router-neighbor)# update-source ethernet 2/1</pre>	<p>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 fallover when update-source is configured.</p>

To tune BGP, use the following optional commands in neighbor address-family configuration mode:

Command	Purpose
---------	---------

Command	Purpose
allows in Example: <pre>switch(config-router-neighbor-af)# allows in</pre>	Allows routes that have their own AS in the AS path to be installed in the BRIB.
default-originate [route-map map-name] Example: <pre>switch(config-router-neighbor-af)# default-originate</pre>	Generates a default route to the BGP peer.
disable-peer-as-check Example: <pre>switch(config-router-neighbor-af)# disable-peer-as-check</pre>	Disables peer AS-number checking while the device advertises routes learned from one node to another node in the same AS path.
filter-list list-name {in out} Example: <pre>switch(config-router-neighbor-af)# filter-list BGPFilter in</pre>	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.
prefix-list list-name {in out} Example: <pre>switch(config-router-neighbor-af)# prefix-list PrefixFilter in</pre>	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.
send-community Example: <pre>switch(config-router-neighbor-af)# send-community</pre>	Sends the community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
send-community extended Example: <pre>switch(config-router-neighbor-af)# send-community extended</pre>	Sends the extended community attribute to this BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.
suppress-inactive Example: <pre>switch(config-router-neighbor-af)# suppress-inactive</pre>	Advertises the best (active) routes only to the BGP peer. This command triggers an automatic soft clear or refresh of BGP neighbor sessions.

10.6.29 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.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp as-number**

3. **neighbor ip-address remote-as as-number**
4. **address-family {ipv4 | ipv6} {unicast | multicast}**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp as-number Example: switch(config)# router bgp 65535 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	neighbor ip-address remote-as as-number Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65534 switch(config-router-neighbor)#	Places the router in neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 4	address-family {ipv4 ipv6} {unicast multicast} Example: switch(config-router-neighbor)# address-family ipv4 multicast switch(config-router-neighbor-af)#	Enters address family configuration mode.
Step 5	copy running-config startup-config Example: switch(config-router-neighbor-af)# copy running-config startup-config	(Optional) Saves this configuration change.

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
```

10.7 Configuring BMP

You can configure BMP on the device.

BEFORE YOU BEGIN

You must enable BGP.

SUMMARY STEPS

1. **configure terminal**
2. **router bgp** *as-number*
3. **bmp server** *server-number*
4. **address** *ip-address port-number port-number*
5. **description** *string*
6. **initial-refresh** {**skip** | **delay** *time*}
7. **initial-delay** *time*
8. **stats-reporting-period** *time*
9. **shutdown**
10. **neighbor** *ip-address remote-as as-number*
11. **bmp-activate-server** *server-number*
12. (Optional) **show bgp bmp server** [*server-number*] [**detail**]
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 200 switch(config-router)#	Enters BGP mode and assigns the autonomous system number to the local BGP speaker.
Step 3	bmp server <i>server-number</i> Example: switch(config-router)# bmp server 1	Configures the BMP server to which BGP should send information. The server number is used as a key. Note You can configure up to two BMP servers.
Step 4	address <i>ip-address port-number port-number</i> Example: switch(config-router)# address 10.1.1.1 port-number 2000	Configures the IPv4 or IPv6 address of the host and the port number on which the BMP speaker connects to the BMP server.
Step 5	description <i>string</i> Example: switch(config-router)# description BMPserver1	Configures the BMP server description. You can enter up to 256 alphanumeric characters.
Step 6	initial-refresh { skip delay <i>time</i> }	Configures the option to send a route refresh when BGP is converged and the BMP server connection is established later. The skip option specifies to not send a route refresh if the BMP server connection comes up later.

	Command	Purpose
		The delay option specifies the time in seconds after which the route refresh should be sent. The range is from 30 to 720 seconds, and the default value is 30 seconds.
Step 7	initial-delay <i>time</i> Example: switch(config-router)# initial-delay 120	Configures the delay after which a connection is attempted to the BMP server. The range is from 30 to 720 seconds, and the default value is 45 seconds.
Step 8	stats-reporting-period <i>time</i> Example: switch(config-router)# stats-reporting-period 50	Configures the time interval in which the BMP server receives the statistics report from BGP neighbors. The range is from 30 to 720 seconds, and the default is disabled.
Step 9	shutdown Example: switch(config-router)# shutdown	Disables the connection to the BMP server.
Step 10	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: switch(config-router)# neighbor 192.168.1.2 remote-as 65535 switch(config-router-neighbor)#	Enters neighbor configuration mode for BGP routing and configures the neighbor IP address.
Step 11	bmp-activate-server <i>server-number</i> Example: switch(config-router-neighbor)# bmp-activate-server 1	Configures the BMP server to which a neighbor's information should be sent.
Step 12	show bgp bmp server [<i>server-number</i>] [detail] Example: switch(config-router-neighbor)# show bgp bmp server	(Optional) Displays BMP server information.
Step 13	copy running-config startup-config Example: switch(config-router-neighbor)# copy running-config startup-config	(Optional) Saves this configuration change.

10.8 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 VRFs.

SUMMARY STEPS

1. **configure terminal**

2. **router bgp** *as-number*
3. **graceful-restart**
4. **graceful-restart** [**restart-time** *time* | **stalepath-time** *time*]
5. **graceful-restart-helper**
6. (Optional) **show running-config bgp**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Creates a new BGP process with the configured autonomous system number.
Step 3	graceful-restart Example: 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	graceful-restart [restart-time <i>time</i> stalepath-time <i>time</i>] Example: switch(config-router)# graceful-restart restart-time 300	Configures the graceful restart timers. The optional parameters are as follows: <ul style="list-style-type: none">• 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	graceful-restart-helper Example: 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	show running-config bgp Example: switch(config-router)# show running-config bgp	(Optional) Displays the BGP configuration.
Step 7	copy running-config startup-config	(Optional) Saves this configuration change.

Command	Purpose
Example: switch(config-router)# copy running-config startup-config	

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
```

10.8.1 Configuring Virtualization

You can configure one BGP process, create multiple VRFs, and use the same BGP process in each VRF.

BEFORE YOU BEGIN

You must enable BGP.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router bgp** *as-number*
5. **vrf** *vrf-name*
6. **neighbor** *ip-address* **remote-as** *as-number*
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	exit Example: switch(config-vrf)# exit switch(config)#	Exits VRF configuration mode.
Step 4	router bgp <i>as-number</i> Example: switch(config)# router bgp 65535 switch(config-router)#	Creates a new BGP process with the configured autonomous system number.

	Command	Purpose
Step 5	vrf <i>vrf-name</i> Example: <pre>switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#</pre>	Enters the router VRF configuration mode and associates this BGP instance with a VRF.
Step 6	neighbor <i>ip-address</i> remote-as <i>as-number</i> Example: <pre>switch(config-router-vrf)# neighbor 209.165.201.1 remote-as 65535 switch(config-router--vrf-neighbor)#</pre>	Configures the IP address and AS number for a remote BGP peer.
Step 7	copy running-config startup-config Example: <pre>switch(config-router-vrf-neighbor)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

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
```

10.9 Verifying the Advanced BGP Configuration

To display the BGP configuration, perform one of the following tasks:

Command	Purpose
show bgp all [summary] [vrf <i>vrf-name</i>]	Displays the BGP information for all address families.
show bgp bmp server [<i>server-number</i>] [detail]	Displays BMP server information.
show bgp convergence [vrf <i>vrf-name</i>]	Displays the BGP information for all address families.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] community { regex <i>expression</i> [community] [no-advertise] [no-export] [no-export-subconfed]} [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP community.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] community-list <i>list-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP community list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity { regex	Displays the BGP routes that match a BGP extended community.

Command	Purpose
<i>expression</i> generic [non-transitive transitive] <i>aa4:nn</i> [exact-match]} [vrf <i>vrf-name</i>]	
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] extcommunity-list <i>list-name</i> [exact-match]} [vrf <i>vrf-name</i>]	Displays the BGP routes that match a BGP extended community list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] { dampening dampened-paths [regex <i>expression</i>]} [vrf <i>vrf-name</i>]	Displays the information for BGP route dampening. Use the clear bgp dampening command to clear the route flap dampening information.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] history-paths [regex <i>expression</i>] [vrf <i>vrf-name</i>]	Displays the BGP route history paths.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] filter-list <i>list-name</i> [vrf <i>vrf-name</i>]	Displays the information for the BGP filter list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] neighbors [<i>ip-address</i> <i>ipv6-prefix</i>] [vrf <i>vrf-name</i>]	Displays the information for BGP peers. Use the clear bgp neighbors command to clear these neighbors.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] { nexthop nexthop-database } [vrf <i>vrf-name</i>]	Displays the information for the BGP route next hop.
show bgp paths	Displays the BGP path information.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] policy <i>name</i> [vrf <i>vrf-name</i>]	Displays the BGP policy information. Use the clear bgp policy command to clear the policy information.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] prefix-list <i>list-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the prefix list.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] received-paths [vrf <i>vrf-name</i>]	Displays the BGP paths stored for soft reconfiguration.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] regex <i>expression</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the AS_path regular expression.
show bgp { ipv4 ipv6 } { unicast multicast } [<i>ip-address</i> <i>ipv6-prefix</i>] route-map <i>map-name</i> [vrf <i>vrf-name</i>]	Displays the BGP routes that match the route map.
show bgp peer-policy <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer policies.
show bgp peer-session <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer sessions.
show bgp peer-template <i>name</i> [vrf <i>vrf-name</i>]	Displays the information about BGP peer templates. Use the clear bgp peer-template

Command	Purpose
	command to clear all neighbors in a peer template.
show bgp process	Displays the BGP process information.
show ip route <i>ip-address</i> detail vrf all i bw	Displays the link bandwidth EXTCOMM fields. <i>bw:xx</i> (such as <i>bw:40</i>) in the output indicates that BGP peers are sending BGP extended attributes with the bandwidth (for weighted ECMP).
show {ipv4 ipv6} bgp options	Displays the BGP status and configuration information.
show {ipv4 ipv6} mbgp options	Displays the BGP status and configuration 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} {unicast multicast} [<i>ip-address</i> <i>ipv6-prefix</i>] flap-statistics [<i>vrf vrf-name</i>]	Displays the BGP route flap statistics.
show bgp {ipv4 ipv6} unicast injected-routes	Displays injected routes in the routing table.
show bgp sessions [<i>vrf vrf-name</i>]	Displays the BGP sessions for all peers.
show bgp statistics	Displays the BGP statistics.

10.11 Configuration Examples

This example shows how to configure MD5 authentication for prefix-based neighbors:

```

template peer BasePeer-V6
  description BasePeer-V6
  password 3 f4200cfc725bbd28
  transport connection-mode passive
  address-family ipv6 unicast
template peer BasePeer-V4
  bfd
  description BasePeer-V4
  password 3 f4200cfc725bbd28
  address-family ipv4 unicast
--
neighbor fc00::10:3:11:0/127 remote-as
  65006 inherit peer BasePeer-V6
neighbor 10.3.11.0/31 remote-as 65006
  inherit peer BasePeer-V4

```

This example shows how to enable neighbor status change messages globally and suppress them for a specific neighbor:

```
router bgp 65100 log-  
neighbor-changes  
  neighbor 209.165.201.1 remote-as  
    65535 description test  
  address-family ipv4 unicast  
    soft-reconfiguration inbound  
    disable log-neighbor-changes
```

CHAPTER 11 Configuring RIP

This chapter describes how to configure the Routing Information Protocol (RIP) on the Inspur INOS-CN device.

11.1 About RIP

11.1.1 RIP Overview

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.

11.1.2 RIPv2 Authentication

You can configure authentication on RIP messages to prevent unauthorized or invalid routing updates in your network. Inspur INOS-CN supports a simple password or an MD5 authentication digest.

You can configure the RIP authentication per interface by using keychain management for the authentication keys. Keychain management allows you to control changes to the authentication keys used by an MD5 authentication digest or simple text password authentication. See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for more details about creating keychains.

To use an MD5 authentication digest, you configure a password that is shared at the local router and all remote RIP neighbors. Inspur INOS-CN 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.

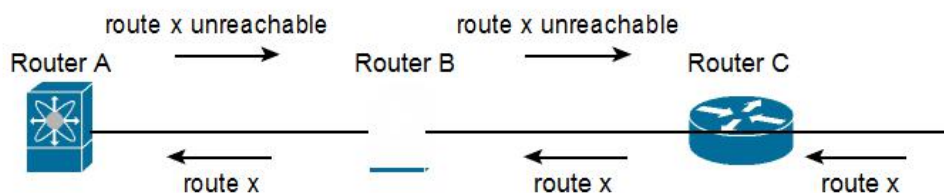
11.1.3 Split Horizon

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-CN 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 11-1 shows a sample RIP network with split horizon and poison reverse enabled.

Figure 11-1 RIP with Split Horizon Poison Reverse



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.

11.1.4 Route Filtering

You can configure a route policy on a RIP-enabled interface to filter the RIP updates. Inspur INOS-CN updates the route table with only those routes that the route policy allows.

11.1.5 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.

11.1.6 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-CN 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.

11.1.7 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-CN supports the Equal Cost Multiple Path (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.

11.1.8 High Availability

Inspur INOS-CN supports stateless restarts for RIP. After a reboot or supervisor switchover, Inspur INOS-CN applies the running configuration and RIP immediately sends request packets to repopulate its routing table.

11.1.9 Virtualization Support

Inspur INOS-CN supports multiple instances of the RIP protocol that run on the same system. RIP supports virtual routing and forwarding (VRF) instances.

11.2 Licensing Requirements for RIP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	RIP requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

11.3 Prerequisites for RIP

RIP has the following prerequisites:

- You must enable RIP (see the “Enabling RIP” section).

11.4 Guidelines and Limitations

RIP has the following configuration guidelines and limitations:

- Inspur INOS-CN does not support RIPv1. If Inspur INOS-CN receives a RIPv1 packet, it logs a message and drops the packet.
- Inspur INOS-CN does not establish adjacencies with RIPv1 routers.
- IPv6 is not supported for RIP.

11.5 Default Settings

Table 11-1 lists the default settings for RIP parameters.

Table 11-1 Default RIP Parameters

Parameters	Default
Maximum paths for load balancing	16
RIP feature	Disabled
Split horizon	Enabled

11.6 Configuring RIP

11.6.1 Enabling RIP

You must enable RIP before you can configure RIP.

SUMMARY STEPS

1. **configure terminal**
2. **feature rip**
3. (Optional) **show feature**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	feature rip Example: switch(config)# feature rip	Enables the RIP feature.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To disable the RIP feature and remove all associated configurations, use the following command in global configuration mode.

Command	Purpose
no feature rip Example: switch(config)# no feature rip	Disables the RIP feature and removes all associated configurations.

11.6.2 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.

SUMMARY STEPS

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. (Optional) **show ip rip** [*instance instance-tag*] [*vrf vrf-name*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.

	Command	Purpose
Step 2	<pre>router rip instance-tag</pre> <p>Example:</p> <pre>switch(config)# router RIP Enterprise switch(config-router)#</pre>	Creates a new RIP instance with the configured <i>instance-tag</i> .
Step 3	<pre>address-family ipv4 unicast</pre> <p>Example:</p> <pre>switch(config-router)# address-family ipv4 unicast switch(config-router-af)#</pre>	Configures the address family for this RIP instance and enters address-family configuration mode.
Step 4	<pre>show ip rip [instance instance-tag] [vrf vrf-name]</pre> <p>Example:</p> <pre>switch(config-router-af)# show ip rip</pre>	(Optional) Displays a summary of RIP information for all RIP instances.
Step 5	<pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config-router-af)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

To remove the RIP instance and the associated configurations, use the following command in global configuration mode.

Command	Purpose
<pre>no router rip instance-tag</pre> <p>Example:</p> <pre>switch(config)# no router rip Enterprise</pre>	Deletes the RIP instance and all associated configuration.

You can configure the following optional parameters for RIP in address-family configuration mode:

Command	Purpose
<pre>distance value</pre> <p>Example:</p> <pre>switch(config-router-af)# distance 30</pre>	Sets the administrative distance for RIP. The range is from 1 to 255. The default is 120.
<pre>maximum-paths number</pre> <p>Example:</p> <pre>switch(config-router-af)# maximum-paths 6</pre>	Configures the maximum number of equal-cost paths that RIP maintains in the route table. The range is from 1 to 64. The default is 16.

This example shows how to create a RIP instance for IPv4 and set 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
```

11.6.3 Restarting a RIP Instance

You can restart a RIP instance. This clears all neighbors for the instance.

To restart a RIP instance and remove all associated neighbors, use the following command:

Command	Purpose
<pre>restart rip instance-tag</pre> <p>Example:</p> <pre>switch(config)# restart rip Enterprise</pre>	Restarts the RIP instance and removes all neighbors.

11.6.4 Configuring RIP on an Interface

You can add an interface to a RIP instance.

BEFORE YOU BEGIN

You must enable RIP.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip router rip** *instance-tag*
4. (Optional) **show ip rip** [*instance instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*] [**detail**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<pre>configure terminal</pre> <p>Example:</p> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<pre>interface interface-type slot/port</pre> <p>Example:</p> <pre>switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 3	<pre>ip router rip instance-tag</pre> <p>Example:</p> <pre>switch(config-if)# ip router rip Enterprise</pre>	Associates this interface with a RIP instance.
Step 4	<pre>show ip rip [instance instance-tag] interface [interface-type slot/port] [vrf vrf-name] [detail]</pre> <p>Example:</p> <pre>switch(config-if)# show ip rip Enterprise ethernet 1/2</pre>	(Optional) Displays RIP information for an interface.
Step 5	<pre>copy running-config startup-config</pre> <p>Example:</p>	(Optional) Saves this configuration change.

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

This example shows how to add Ethernet 1/2 interface to a RIP instance:

```
switch# configure terminal
switch(config)# interface ethernet 1/2
switch(config-if)# ip router rip Enterprise
switch(config)# copy running-config startup-config
```

11.6.5 Configuring RIP Authentication

You can configure authentication for RIP packets on an interface.

BEFORE YOU BEGIN

You must enable RIP.

Configure a keychain if necessary before enabling authentication. See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for details on implementing keychains.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **ip rip authentication mode** {text | md5}
4. **ip rip authentication key-chain** *key*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ip rip authentication mode {text md5} Example: switch(config-if)# ip rip authentication mode md5	Sets the authentication type for RIP on this interface as cleartext or MD5 authentication digest.
Step 4	ip rip authentication key-chain <i>key</i> Example: switch(config-if)# ip rip authentication key-chain RIPKey	Configures the authentication key used for RIP on this interface.
Step 5	copy running-config startup-config Example:	(Optional) Saves this configuration change.

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

This example shows how to create a keychain and configure MD5 authentication on a RIP interface:

```
switch# configure terminal
switch(config)# key chain RIPKey
switch(config-keychain)# key 2
switch(config-keychain-key)# accept-lifetime 00:00:00 Oct 01 2018
infinite
switch(config-keychain-key)# send-lifetime 00:00:00 Oct 01 2018 infinite
switch(config-keychain-key)# exit
switch(config-keychain)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# ip rip authentication mode md5
switch(config-if)# ip rip authentication key-chain RIPKey
switch(config-if)# copy running-config startup-config
```

11.6.6 Configuring a Passive Interface

You can configure a RIP interface to receive routes but not send route updates by setting the interface to passive mode.

To configure a RIP interface in passive mode, use the following command in interface configuration mode:

Command	Purpose
ip rip passive-interface Example: switch(config-if)# ip rip passive-interface	Sets the interface into passive mode.

11.6.7 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.

To configure split horizon with poison reverse on an interface, use the following command in interface configuration mode:

Command	Purpose
ip rip poison-reverse Example: switch(config-if)# ip rip poison-reverse	Enables split horizon with poison reverse. Split horizon with poison reverse is disabled by default.

11.6.8 Configuring Route Summarization

You can create aggregate addresses that are represented in the routing table by a summary address. Inspur INOS-CN 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 following command in interface configuration mode:

Command	Purpose
ip rip summary-address <i>ip-prefix/mask-len</i> Example: <pre>switch(config-if)# ip rip summary-address 1.1.1.1/32</pre>	Configures a summary address for RIP for IPv4 addresses.

11.6.9 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.

Configure a route map before configuring redistribution.

SUMMARY STEPS

1. **configure terminal**
2. **router rip** *instance-tag*
3. **address-family ipv4 unicast**
4. **redistribute** {*bgp as* | *direct* | *eigrp* | *isis* | *ospf* | *ospfv3* | *rip*} *instance-tag* | **static** } **route-map** *map-name*
5. (Optional) **default-information originate** [*always*] [**route-map** *map-name*]
6. (Optional) **default-metric** *value*
7. (Optional) **show ip rip route** [*ip-prefix* [*longer-prefixes* | *shorter-prefixes*]] [*vrf vrf-name*] [**summary**]
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	router rip <i>instance-tag</i> Example: <pre>switch(config)# router rip Enterprise switch(config-router)#</pre>	Creates a new RIP instance with the configured <i>instance-tag</i> .
Step 3	address-family ipv4 unicast Example: <pre>switch(config-router)# address-family ipv4 unicast switch(config-router-af)#</pre>	Enters address-family configuration mode.
Step 4	redistribute { <i>bgp as</i> <i>direct</i> { <i>eigrp</i> <i>isis</i> <i>ospf</i> <i>ospfv3</i> <i>rip</i> } <i>instance-tag</i> static } route-map <i>map-name</i>	Redistributes routes from other protocols into RIP.

	Command	Purpose
	Example: <pre>switch(config-router-af)# redistribute eigrp 201 route-map RIPmap</pre>	
Step 5	default-information originate [always] [route-map map-name] Example: <pre>switch(config-router-af)# default-information originate always</pre>	(Optional) Generates a default route into RIP, optionally controlled by a route map.
Step 6	default-metric value Example: <pre>switch(config-router-af)# default-metric 2</pre>	(Optional) Sets the default metric for all redistributed routes. The range is from 1 to 15. The default is 1.
Step 7	show ip rip route [ip-prefix [longer-prefixes shorter-prefixes] [vrf vrf-name] [summary] Example: <pre>switch(config-router-af)# show ip rip route</pre>	(Optional) Shows the routes in RIP.
Step 8	copy running-config startup-config Example: <pre>switch(config-router-af)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This 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
```

11.6.10 Configuring Inspur INOS-CN RIP for Compatibility with Inspur INOS RIP

You can configure Inspur INOS-CN RIP to behave like Inspur INOS RIP in the way that routes are advertised and processed.

Directly connected routes are treated with cost 1 in Inspur INOS-CN RIP and with cost 0 in Inspur INOS RIP. When routes are advertised in Inspur INOS-CN RIP, the receiving device adds a minimum cost of +1 to all received routes and installs the routes in its routing table. In Inspur INOS 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-CN and Inspur INOS devices are working together. You can prevent these compatibility issues by configuring Inspur INOS-CN RIP to advertise and process routes like Inspur INOS RIP.

BEFORE YOU BEGIN

You must enable RIP.

SUMMARY STEPS

1. **configure terminal**
2. **router rip** *instance-tag*
3. **[no] metric direct 0**
4. (Optional) **show running-config rip**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router rip <i>instance-tag</i> Example: switch(config)# router rip 100 switch(config-router)#	Creates a new RIP instance with the configured instance tag. You can enter 100, 201, or up to 20 alphanumeric characters for the instance tag.
Step 3	[no] metric direct 0 Example: switch(config-router)# metric direct 0	Configures all directly connected routes with cost 0 instead of the default of cost 1 in order to make Inspur INOS-CN RIP compatible with Inspur INOS RIP in the way that routes are advertised and processed. Note This command must be configured on all Inspur INOS-CN devices that are present in any RIP network that also contains Inspur INOS devices.
Step 4	show running-config rip Example: switch(config-router)# show running-config rip	(Optional) Displays the current running RIP configuration.
Step 5	copy running-config startup-config Example: switch(config-router)# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to disable Inspur INOS-CN RIP compatibility with Inspur INOS 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
```

11.6.11 Configuring Virtualization

You can configure multiple RIP instances, create multiple VRFs, 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.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **exit**
4. **router rip** *instance-tag*
5. **vrf** *vrf-name*
6. (Optional) **address-family ipv4 unicast**
7. (Optional) **redistribute** {*bgp as* | *direct* | {*eigrp* | *isis* | *ospf* | *ospfv3* | *rip*} *instance-tag* | *static*} **route-map** *map-name*
8. **interface ethernet** *slot/port*
9. **vrf member** *vrf-name*
10. **ip-address** *ip-prefix/length*
11. **ip router rip** *instance-tag*
12. (Optional) **show ip rip** [**instance** *instance-tag*] **interface** [*interface-type slot/port*] [**vrf** *vrf-name*]
13. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context RemoteOfficeVRF switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode.
Step 3	exit Example: switch(config-vrf)# exit switch(config)#	Exits VRF configuration mode.
Step 4	router rip <i>instance-tag</i> Example: switch(config)# router rip Enterprise switch(config-router)#	Creates a new RIP instance with the configured instance tag.
Step 5	vrf <i>vrf-name</i> Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Creates a new VRF.
Step 6	address-family ipv4 unicast Example:	(Optional) Configures the VRF address family for this RIP instance.

	Command	Purpose
	<pre>switch(config-router-vrf)# address-family ipv4 unicast switch(config-router-vrf-af)#</pre>	
Step 7	<pre>redistribute {bgp as direct {eigrp isis ospf ospfv3 rip} instance-tag static} route-map map-name</pre> <p>Example:</p> <pre>switch(config-router-vrf-af)# redistribute eigrp 201 route-map RIPmap</pre>	(Optional) Redistributes routes from other protocols
Step 8	<pre>interface ethernet slot/port</pre> <p>Example:</p> <pre>switch(config-router-vrf-af)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 9	<pre>vrf member vrf-name</pre> <p>Example:</p> <pre>switch(config-if)# vrf member RemoteOfficeVRF</pre>	Adds this interface to a VRF.
Step 10	<pre>ip address ip-prefix/length</pre> <p>Example:</p> <pre>switch(config-if)# ip address 192.0.2.1/16</pre>	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 11	<pre>ip router rip instance-tag</pre> <p>Example:</p> <pre>switch(config-if)# ip router rip Enterprise</pre>	Associates this interface with a RIP instance.
Step 12	<pre>show ip rip [instance instance-tag] interface [interface-type slot/port] [vrf vrf-name]</pre> <p>Example:</p> <pre>switch(config-if)# show ip rip Enterprise ethernet 1/2</pre>	(Optional) Displays RIP information for an interface in a VRF.
Step 13	<pre>copy running-config startup-config</pre> <p>Example:</p> <pre>switch(config-if)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

This 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

```

11.6.12 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 internetwork needs.

You can use the following optional commands in address-family configuration mode to tune RIP:

Command	Purpose
<pre> timers basic <i>update timeout holddown</i> <i>garbage-collection</i> Example: switch(config-router-af)# timers basic 40 120 120 100 </pre>	<p>Sets the RIP timers in seconds. The parameters are as follows:</p> <ul style="list-style-type: none"> <i>update</i>—The range is from 5 to any positive integer. The default is 30. <i>timeout</i>—The time that Inspur INOS-CN waits before declaring a route as invalid. If Inspur INOS-CN does not receive route update information for this route before the timeout interval ends, Inspur INOS-CN declares the route as invalid. The range is from 1 to any positive integer. The default is 180. <i>holddown</i>—The time during which Inspur INOS-CN ignores better route information for an invalid route. The range is from 0 to any positive integer. The default is 180. <i>garbage-collection</i>—The time from when Inspur INOS-CN marks a route as invalid until Inspur INOS-CN removes the route from the routing table. The range is from 1 to any positive integer. The default is 120.

You can use the following optional commands in interface configuration mode to tune RIP:

Command	Purpose
<pre> ip rip metric-offset <i>value</i> Example: switch(config-if)# ip rip metric-offset 10 </pre>	<p>Adds a value to the metric for every route received on this interface. The range is from 1 to 15. The default is 1.</p>
<pre> ip rip route-filter [prefix-list <i>list-name</i> route-map <i>map-name</i>] [in out] Example: </pre>	<p>Specifies a route map to filter incoming or outgoing RIP updates.</p>

Command	Purpose
switch(config-if)# ip rip route-filter route-map InputMap in	

11.7 Verifying the RIP Configuration

To display RIP configuration, perform one of the following tasks:

Command	Purpose
show ip rip instance [<i>instance-tag</i>] [vrf <i>vrf-name</i>]	Displays the status for an instance of RIP.
show ip rip [instance <i>instance-tag</i>] interface <i>slot/port</i> detail [vrf <i>vrf-name</i>]	Displays the RIP status for an interface.
show ip rip [instance <i>instance-tag</i>] neighbor [<i>interface-type number</i>] [vrf <i>vrf-name</i>]	Displays the RIP neighbor table.
show ip rip [instance <i>instance-tag</i>] route [<i>ip-prefix/length</i>] [longer-prefixes shorter--prefixes] [summary] [vrf <i>vrf-name</i>]	Displays the RIP route table.
show running-configuration rip	Displays the current running RIP configuration.

11.8 Displaying RIP Statistics

To display RIP statistics, use the following commands:

Command	Purpose
show ip rip [instance <i>instance-tag</i>] policy statistics redistribute { bgp as direct { eigrp isis ospf ospfv3 rip } <i>instance-tag</i> static } [vrf <i>vrf-name</i>]	Displays the RIP policy statistics.
show ip rip [instance <i>instance-tag</i>] statistics <i>interface-type number</i>] [vrf <i>vrf-name</i>]	Displays the RIP statistics.

Use the **clear ip rip policy statistics redistribute** *protocol process-tag* command to clear policy statistics.

Use the **clear ip rip statistics** command to clear RIP statistics.

11.9 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 ipv4 unicast
          redistribute eigrp 201 route-map
            RIPmap max-paths 10
        !
interface ethernet 1/2
  vrf member NewVRF
  ip address 192.0.2.1/16
  ip router rip Enterprise
  ip rip authentication mode md5
  ip rip authentication key-chain RIPKey
```

CHAPTER 12 Configuring Static Routing

This chapter describes how to configure static routing on the Inspur INOS-CN device.

12.1 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).

12.1.1 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.

12.1.2 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 IPv4/IPv6 address.

12.1.3 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.

12.1.4 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.

12.1.5 Remote Next Hops for Static Routes

You can specify the next-hop address of a neighboring router that is not directly connected to the router for static routes with remote (nondirectly 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 hops that have reachability to the remote next hops.

12.1.6 BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4. 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 CN12900 Series INOS-CN Interfaces Configuration Guide for more information.

12.1.7 Virtualization Support

Static routes support virtual routing and forwarding (VRF) instances.

12.2 Licensing Requirements for Static Routing

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	Static routing requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

12.3 Prerequisites for Static Routing

Static routing has the following prerequisites:

- If the next-hop address for a static route is unreachable, the static route is not added to the unicast routing table.

12.4 Default Settings

Table 12-1 lists the default settings for static routing parameters.

Table 12-1 Default Static Routing Parameters

Parameters	Default
Administrative distance	1
RIP feature	Disabled

12.5 Configuring Static Routing

12.5.1 Configuring a Static Route

You can configure a static route on the router.

SUMMARY STEPS

1. **configure terminal**
2. **ip route** {*ip-prefix* | *ip-addr/ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**name** *nexthop-name*] [**tag** *tag-value*] [*pref*] [**track** *number*] [**vrf** *name*] or **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*] [**track** *number*] [**vrf** *name*]
3. (Optional) **show** {**ip** | **ipv6**} **static-route**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip route { <i>ip-prefix</i> <i>ip-addr/ip-mask</i> } {[<i>next-hop</i> <i>nh-prefix</i>] [<i>interface</i> <i>next-hop</i> <i>nh-prefix</i>]} [name <i>nexthop-name</i>] [tag <i>tag-value</i>] [<i>pref</i>] [track <i>number</i>] [vrf <i>name</i>] Example: switch(config)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4	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 . The <i>preference</i> value sets the administrative distance. The range is from 1 to 255. The default is 1. The track number specifies the object to be tracked. The range is from 1 to 500. The VRF name specifies the VRF for the next hop, if it is different from the present VRF.
	ipv6 route <i>ip6-prefix</i> { <i>nh-prefix</i> <i>link-local-nh-prefix</i> } (<i>nexthop</i> [<i>interface</i>] <i>link-local-nexthop</i> [<i>interface</i>]) [name <i>nexthop-name</i>] [tag <i>tag-value</i>] [<i>pref</i>] [track <i>number</i>] [vrf <i>name</i>] Example: switch(config)# ipv6 route 2001:0DB8::/48 6:::6 ethernet 2/1	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 . The <i>preference</i> value sets the administrative distance. The range is from 1 to 255. The default is 1. The track number specifies the object to be tracked. The range is from 1 to 500. The VRF name specifies the VRF for the next hop, if it is different from the present VRF.
Step 3	show { ip ipv6 } static-route Example: switch(config)# show ip static-route	(Optional) Displays information about static routes.
Step 4	copy running-config startup-config	(Optional) Copies the running configuration to the

Command	Purpose
Example: <pre>switch(config)# copy running-config startup-config</pre>	startup configuration.

This example shows how to configure a static route for a null interface:

```
switch# configure terminal
switch(config)# ip route 1.1.1.1/32 null 0
switch(config)# copy running-config startup-config
```

Use the **no {ip | ipv6} route** command to remove the static route.

12.5.2 Configuring a Static Route over a VLAN

You can configure a static route without next hop support over a VLAN.

BEFORE YOU BEGIN

Ensure that the access port is part of the VLAN.

SUMMARY STEPS

1. **configure terminal**
2. **feature interface-vlan**
3. **interface vlan** *vlan-id*
4. **ip address** *ip-addr/length*
5. **ip route** *ip-addr/length vlan-id*
6. (Optional) **show ip route**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	feature interface vlan Example: <pre>switch(config)# feature interface-vlan</pre>	Enables VLAN interface mode.
Step 3	interface-vlan <i>vlan-id</i> Example: <pre>switch(config)# interface-vlan 10</pre>	Creates an SVI and enters interface configuration mode. The range for the <i>vlan-id</i> argument is from 1 to 4094, except for the VLANs reserved for the internal switch.
Step 4	ip address <i>ip-addr/length</i> Example: <pre>switch(config)# ip address 192.0.2.1/8</pre>	Configures an IP address for the VLAN.
Step 5	ip route <i>ip-addr/length vlan-id</i> Example:	Adds an interface static route without a next hop on the switch virtual interface (SVI).

	Command	Purpose
	switch(config)# ip route 209.165.200.224/27 vlan 10	The IP address is the address that is configured on the interface that is connected to the switch.
Step 6	show ip route Example: switch(config)# show ip route	(Optional) Displays routes from the Unicast Route Information Base (URIB).
Step 7	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a static route without a next hop over an SVI:

```
switch# configure terminal
switch(config)# feature interface-vlan
switch(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
```

Use the **no ip route** command to remove the static route.

12.5.3 Configuring Virtualization

You can configure a static route in a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **ip route** {*ip-prefix* | *ip-addr ip-mask*} {*next-hop* | *nh-prefix* | *interface*} [**name** *nexthop-name*] [**tag** *tag-value*] [**pref**] [**track** *number*] [**vrf** *name*] or **ipv6 route** *ip6-prefix* {*nh-prefix* | *link-local-nh-prefix*} | {*next-hop* [*interface*] | *link-local-next-hop* [*interface*]} [**name** *nexthop-name*] [**tag** *tag-value*] [**pref**] [**track** *number*] [**vrf** *name*]
4. (Optional) **show** {**ip** | **ipv6**} **static-route** *vrf vrf-name*
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context StaticVrf	Creates a VRF and enters VRF configuration mode.
Step 3	ip route { <i>ip-prefix</i> <i>ip-addr ip-mask</i> }	Configures a static route and the interface for this

Command	Purpose
<pre>{next-hop nh-prefix interface} [name nexthop-name] [tag tag-value] [pref] [track number] [vrf name]</pre> <p>Example:</p> <pre>switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2</pre>	<p>static route. Use ? to display a list of supported interfaces. You can specify a null interface by using null 0.</p> <p>You can optionally configure the next-hop address.</p> <p>The <i>preference</i> value sets the administrative distance. The range is from 1 to 255. The default is 1.</p> <p>The track number specifies the object to be tracked.</p> <p>The range is from 1 to 500. The VRF name specifies the VRF for the next hop, if it is different from the present VRF.</p>
<pre>ipv6 route ip6-prefix {nh-prefix link-local-nh-prefix} (nexthop [interface] link-local-nexthop [interface]) [name nexthop-name] [tag tag-value] [pref] [track number] [vrf name]</pre> <p>Example:</p> <pre>switch(config)# ipv6 route 2001:0DB8::/48 6::6 ethernet 2/1</pre>	<p>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.</p> <p>You can optionally configure the next-hop address.</p> <p>The <i>preference</i> value sets the administrative distance. The range is from 1 to 255. The default is 1.</p> <p>The track number specifies the object to be tracked.</p> <p>The range is from 1 to 500. The VRF name specifies the VRF for the next hop, if it is different from the present VRF.</p>
<p>Step 4 <code>show {ip ipv6} static-route vrf vrf-name</code></p> <p>Example:</p> <pre>switch(config-vrf)# show ip static-route</pre>	<p>(Optional) Displays information on static routes.</p>
<p>Step 5 <code>copy running-config startup-config</code></p> <p>Example:</p> <pre>switch(config-vrf)# copy running-config startup-config</pre>	<p>(Optional) Copies the running configuration to the startup configuration.</p>

This example shows how to configure a static route:

```
switch# configure terminal
switch(config)# vrf context StaticVrf
switch(config-vrf)# ip route 192.0.2.0/8 192.0.2.10
switch(config-vrf)# copy running-config startup-config
```

12.6 Verifying the Static Routing Configuration

To display the static routing configuration, perform one of the following tasks:

Command	Purpose
<code>show {ip ipv6} static-route</code>	Displays the configured static routes.

Command	Purpose
<code>show ipv6 static-route vrf <i>vrf-name</i></code>	Displays static route information for each VRF.
<code>show {ip ipv6} static-route track-table</code>	Displays information about the IPv4 or IPv6 static-route track table.

12.7 Configuration Example for Static Routing

This example shows how to configure static routing:

```
configure terminal
ip route 192.0.2.0/8 192.0.2.10 copy
running-config startup-config
```

CHAPTER 13 Configuring Layer 3 Virtualization

This chapter describes how to configure Layer 3 virtualization on the Inspur INOS-CN device.

13.1 About Layer 3 Virtualization

Inspur INOS-CN supports multiple virtual routing and forwarding instances (VRFs). 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.

Each router has a default VRF and a management VRF:

Management VRF

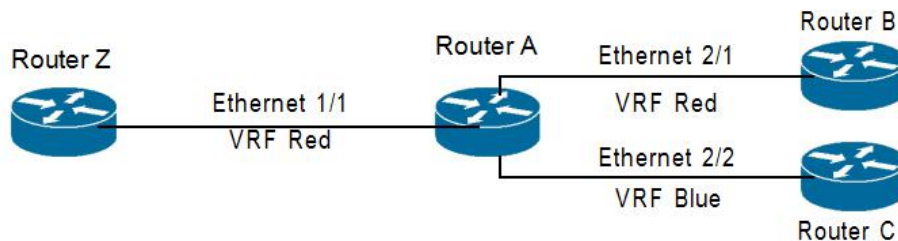
- The management VRF is for management purposes only.
- Only the mgmt 0 interface can be in the management VRF.
- The mgmt 0 interface cannot be assigned to another VRF.
- No routing protocols can run in the management VRF (static only). Default VRF
- All Layer 3 interfaces exist in the default VRF until 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 INOS.

13.1.1 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 13-1 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 13-1 VRFs in a Netwo



By default, Inspur INOS-CN 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-CN uses for incoming packets.

VRF Route Leaking

Inspur INOS-CN supports route leaking between VRFs.

You can import IP prefixes from the global routing table (the default VRF) into any other VRF by using an import policy or export IP prefixes from a non-default VRF into the default VRF using an export policy. The VRF

import and export policies uses a route map to specify the prefixes to be imported or exported into a VRF. The policies can import or export IPv4 and IPv6 unicast prefixes.

IP prefixes are defined as match criteria for the import or export route map through standard route policy filtering mechanisms. For example, you can create an IP prefix list or an as-path filter to define an IP prefix or IP prefix range and use that prefix list or as-path filter in a match clause for the route map. Prefixes that pass through the route map are imported or exported into the specified VRF using the import or export policy. Any route or path that is imported from another VRF cannot be imported or exported again.

13.1.2 VRF-Aware Services

A fundamental feature of the Inspur INOS-CN 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—See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for more information.
- Call Home—See the *Inspur CN12900 Series INOS-CN System Management Configuration Guide* for more information.
- DNS—See Chapter 4, “Configuring DNS” for more information.
- HTTP—See the *Inspur CN12900 Series INOS-CN Fundamentals Configuration Guide* for more information.
- HSRP—See Chapter 17, “Configuring HSRP,” for more information.
- NTP—See the *Inspur CN12900 Series INOS-CN System Management Configuration Guide* for more information.
- RADIUS—See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for more information.
- Ping and Traceroute —See the *Inspur CN12900 Series INOS-CN Fundamentals Configuration Guide* for more information.
- SSH—See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for more information.
- SNMP—See the *Inspur CN12900 Series INOS-CN System Management Configuration Guide* for more information.
- Syslog—See the *Inspur CN12900 Series INOS-CN System Management Configuration Guide* for more information.
- TACACS+—See the *Inspur CN12900 Series INOS-CN Security Configuration Guide* for more information.
- TFTP—See the *Inspur CN12900 Series INOS-CN Fundamentals Configuration Guide* for more information.
- VRRP—See Chapter 18, “Configuring VRRP,” for more information.

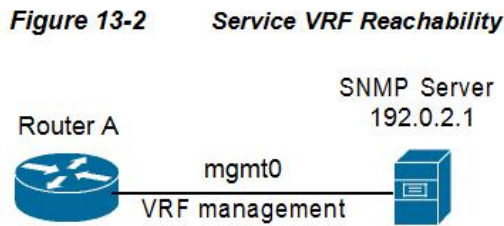
See the appropriate configuration guide for each service for more information on configuring VRF support in that service.

Reachability

Reachability indicates which VRF contains the routing information needed 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-CN must use to reach the server.

Figure 13-2 shows an SNMP server that is reachable over the management VRF. You configure Router A to use the management VRF for the SNMP server host 192.0.2.1.

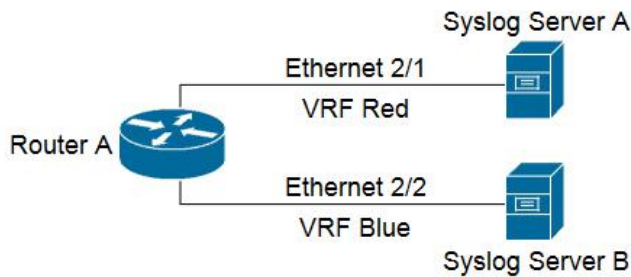
Figure 13-2 Service VRF Reachability



Filtering

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. Figure 13-3 shows two syslog servers with each server supporting one VRF. syslog server A is configured in VRF Red, so Inspur INOS-CN sends only system messages generated in VRF Red to syslog server A.

Figure 13-3 Service VRF Filtering

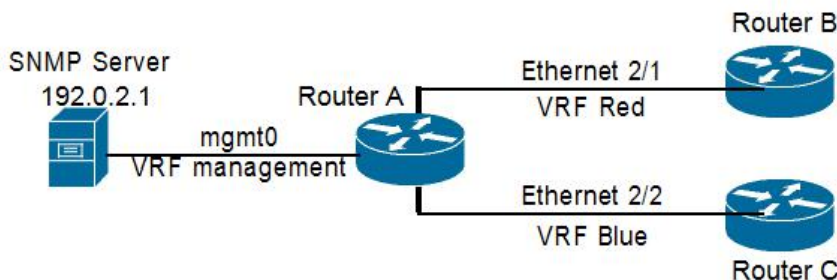


Combining Reachability and Filtering

You can combine reachability and filtering for VRF-aware services. You can configure the VRF that Inspur INOS-CN 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.

Figure 13-4 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 13-4 Service VRF Reachability Filtering



13.2 Licensing Requirements for VRFs

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	VRFs require no license. Any feature not included in a license package is bundled with the INOS-CN image and VRF route leaking requires an Essentials license package.

13.3 Guidelines and Limitations for VRFs

VRFs have the following configuration guidelines and limitations:

- When you make an interface a member of an existing VRF, Inspur INOS-CN 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-CN 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.

13.4 Guidelines and Limitations for VRF Route Leaking

VRF route leaking has the following configuration guidelines and limitations:

- Route leaking is supported between any two non-default VRFs and from the default VRF to a non-default VRF. Route leaking is also supported from a non-default VRF to the default VRF.
- You can restrict route leaking to specific routes using route map filters to match designated IP addresses.
- By default, the maximum number of IP prefixes that can be imported from the default VRF into a non-default VRF is 1000 routes.
- There is no limit on the number of routes that can be leaked between two non-default VRFs.
- VRF route leaking requires an Enterprise license, and BGP must be enabled.

13.5 Default Settings

Table 13-1 lists the default settings for VRF parameters.

Table 13-1 Default VRF Parameters

Parameters	Default
Configured VRFs	Default, management
Routing context	Default VRF
Prefix limit for VRF route leaking	1000

13.6 Configuring VRFs

13.6.1 Creating a VRF

You can create a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. (Optional) **ip route** {*ip-prefix* | *ip-addr ip-mask*} {[*next-hop* | *nh-prefix*] | [*interface next-hop* | *nh-prefix*]} [**tag** *tag-value* [*pref*]
4. (Optional) **show vrf** [*vrf-name*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>name</i> Example: switch(config)# vrf context Enterprise switch(config-vrf)#	Creates a new VRF and enters VRF configuration mode. The <i>name</i> can be any case-sensitive, alphanumeric string up to 32 characters.
Step 3	ip route { <i>ip-prefix</i> <i>ip-addr ip-mask</i> } {[<i>next-hop</i> <i>nh-prefix</i>] [<i>interface next-hop</i> <i>nh-prefix</i>]} [tag <i>tag-value</i> [<i>pref</i>] Example: switch(config-vrf)# ip route 192.0.2.0/8 ethernet 1/2 192.0.2.4	(Optional) Configures a static route and the interface for this static route. You can optionally configure the next-hop address. The <i>preference</i> value sets the administrative distance. The range is from 1 to 255. The default is 1.
Step 4	show vrf [<i>vrf-name</i>] Example: switch(config-vrf)# show vrf Enterprise	(Optional) Displays VRF information.
Step 5	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

To delete the VRF and the associated configuration, use the following command in global configuration mode:

Command	Purpose
no vrf context <i>name</i> Example: switch(config)# no vrf context Enterprise	Deletes the VRF and all associated configurations.

Any commands available in global configuration mode are also available in VRF configuration mode.

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
```

13.6.2 Assigning VRF Membership to an Interface

You can make an interface a member of a VRF.

BEFORE YOU BEGIN

Assign the IP address for an interface after you have configured the interface for a VRF.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrf member** *vrf-name*
4. **ip address** *ip-prefix/length*
5. (Optional) **show vrf** *vrf-name interface interface-type number*
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 4	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 5	show vrf <i>vrf-name interface interface-type number</i> Example: switch(config-vrf)# show vrf Enterprise interface ethernet 1/2	(Optional) Displays VRF information.
Step 6	copy running-config startup-config	(Optional) Saves this configuration change.

Command	Purpose
Example: switch(config)# copy running-config startup-config	

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
```

13.6.3 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 the routing protocol. This section uses OSPFv2 as an example protocol for the detailed configuration steps.

SUMMARY STEPS

1. **configure terminal**
2. **router ospf instance-tag**
3. **vrf vrf-name**
4. (Optional) **maximum-paths paths**
5. **interface interface-type slot/port**
6. **vrf member vrf-name**
7. **ip address ip-prefix/length**
8. **ip router ospf instance-tag area area-id**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	router ospf instance-tag Example: switch(config-vrf)# router ospf 201 switch(config-router)#	Creates a new OSPFv2 instance with the configured instance tag.
Step 3	vrf vrf-name Example: switch(config-router)# vrf RemoteOfficeVRF switch(config-router-vrf)#	Enters VRF configuration mode.
Step 4	maximum-paths paths Example: switch(config-router-vrf)# maximum-paths	(Optional) Configures the maximum number of equal OSPFv2 paths to a destination in the route table for this VRF. Used for load balancing.

	Command	Purpose
	4	
Step 5	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 6	vrf member <i>vrf-name</i> Example: switch(config-if)# vrf member RemoteOfficeVRF	Adds this interface to a VRF.
Step 7	ip address <i>ip-prefix/length</i> Example: switch(config-if)# ip address 192.0.2.1/16	Configures an IP address for this interface. You must do this step after you assign this interface to a VRF.
Step 8	ip router ospf <i>instance-tag area area-id</i> Example: switch(config-if)# ip router ospf 201 area 0	Assigns this interface to the OSPFv2 instance and area configured.
Step 9	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

This 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 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
```

13.6.4 Configuring Global VRF Route Leaking

You can configure route leaking from the default VRF to a non-default VRF or from a non-default VRF to the default VRF.

Route leaking between non-default VRFs is enabled automatically (with route target matching).

BEFORE YOU BEGIN

Make sure that the Enterprise license is installed and BGP is enabled.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** [*vrf-name*]
3. **address-family** {*ipv4* | *ipv6*} **unicast**
4. **{import | export} vrf default** [*prefix-limit*] **map** *route-map*
5. (Optional) **show bgp process vrf** [*vrf-name*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	vrf context <i>vrf-name</i> Example: switch(config)# vrf context vpn1 switch(config-vrf)#	Creates a new VRF.
Step 3	address-family { <i>ipv4</i> <i>ipv6</i> } unicast Example: switch(config-vrf)# address-family ipv4 unicast switch(config-vrf-af-ipv4)#	Enters global address family configuration mode for the IPv4 or IPv6 address family.
Step 4	{import export} vrf default <i>[prefix-limit] map route-map</i> Example: switch(config-vrf-af-ipv4)# import vrf default map importmap Example: switch(config-vrf-af-ipv4)# export vrf default map exportmap	Configures VRF route leaking. The following options are available: <ul style="list-style-type: none"> • import—Copies a route containing IPv4 or IPv6 unicast prefixes from the global routing table (the default VRF) into any other VRF. • export—Copies a route containing IPv4 or IPv6 unicast prefixes from a non-default VRF into the global routing table (the default VRF). • <i>prefix-limit</i>—Specifies the maximum number of routes that can be imported or exported. The range is from 1 to 2147483647, and the default value is 1000.
Step 5	show bgp process vrf [<i>vrf-name</i>] Example: switch(config-vrf-af-ipv4)# show bgp process vrf vpn1	(Optional) Displays the BGP process information for the specified VRF.
Step 6	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

13.6.5 Configuring a VRF-Aware Service

You can configure a VRF-aware service for reachability and filtering. This section uses SNMP and IP domain lists as example services for the detailed configuration steps.

SUMMARY STEPS

1. **configure terminal**
2. **snmp-server host** *ip-address* [**filter-vrf** *vrf-name*] [**use-vrf** *vrf-name*]
3. **vrf context** [*vrf-name*]
4. **ip domain-list** *domain-name* [**all-vrfs**][**use-vrf** *vrf-name*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# <code>configure terminal</code> switch(config)#	Enters global configuration mode.
Step 2	snmp-server host <i>ip-address</i> [filter-vrf <i>vrf-name</i>] [use-vrf <i>vrf-name</i>] Example: switch(config)# <code>snmp-server host 192.0.2.1 use-vrf Red</code> switch(config-vrf)#	Configures a global SNMP server and configures the VRF that Inspur INOS-CN uses to reach the service. Use the filter-vrf keyword to filter information from the selected VRF to this server.
Step 3	vrf context <i>vrf-name</i> Example: switch(config)# <code>vrf context Blue</code> switch(config-vrf)#	Creates a new VRF.
Step 4	ip domain-list <i>domain-name</i> [all-vrfs][use-vrf <i>vrf-name</i>] Example: switch(config-vrf)# <code>ip domain-list List all-vrfs use-vrf Blue</code> switch(config-vrf)#	Configures the domain list in the VRF and optionally configures the VRF that Inspur INOS-CN uses to reach the domain name listed.
Step 5	copy running-config startup-config Example: switch(config)# <code>copy running-config startup-config</code>	(Optional) Saves this configuration change.

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
```

13.6.6 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:

Command	Purpose
routing-context vrf <i>vrf-name</i> Example: <pre>switch# routing-context vrf red switch%red#</pre>	Sets the routing context for all EXEC commands. Default routing context is the default VRF.

To return to the default VRF scope, use the following command in EXEC mode:

Command	Purpose
routing-context vrf default Example: <pre>switch%red# routing-context vrf default switch#</pre>	Sets the default routing context.

13.7 Verifying the VRF Configuration

To display VRF configuration information, perform one of the following tasks:

Command	Purpose
show bgp process vrf [<i>vrf-name</i>]	Displays the BGP process information for the specified VRF.
show vrf [<i>vrf-name</i>]	Displays the information for all or one VRF.
show vrf [<i>vrf-name</i>] detail	Displays detailed information for all or one VRF.
show vrf [<i>vrf-name</i>] [interface <i>interface-type slot/port</i>]	Displays the VRF status for an interface.

13.8 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.

This example shows how to configure route leaking between two non-default VRFs and from the default VRF to a non-default VRF:

```

feature bgp
vrf context Green
ip route 33.33.33.33/32
35.35.1.254 address-family ipv4
unicast route-target import 3:3
route-target export 2:2
export map test
import map test
import vrf default map test

interface Ethernet1/7
vrf member Green
ip address 35.35.1.2/24

vrf context Shared
ip route 44.44.44.44/32
45.45.1.254 address-family ipv4
unicast route-target import 1:1
route-target import 2:2 route-
target export 3:3
export map test
import map test
import vrf default map test

interface Ethernet1/11
vrf member Shared
ip address 45.45.1.2/24

router bgp 100 address-
family ipv4 unicast
redistribute static route-map
test vrf Green
address-family ipv4 unicast
redistribute static route-map
test vrf Shared
address-family ipv4 unicast
redistribute static route-map test

ip prefix-list test seq 5 permit 0.0.0.0/0 le 32

route-map test permit 10
match ip address prefix-list test

ip route 100.100.100.100/32 55.55.55.1

switch# show ip route vrf all
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>

55.55.55.0/24, ubest/mbest: 1/0, attached *via
55.55.55.5, Lo0, [0/0], 00:07:59, direct
55.55.55.5/32, ubest/mbest: 1/0, attached *via

```

```

55.55.55.5, Lo0, [0/0], 00:07:59, local
100.100.100.100/32, ubest/mbest: 1/0
*via 55.55.55.1, [1/0], 00:07:42, static

IP Route Table for VRF "management"
'*' denotes best ucast next-hop '*'
denotes best mcast next-hop '[x/y]'
denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

0.0.0.0/0, ubest/mbest: 1/0
*via 10.29.176.1, [1/0], 12:53:54, static
10.29.176.0/24, ubest/mbest: 1/0, attached
*via 10.29.176.233, mgmt0, [0/0], 13:11:57, direct
10.29.176.233/32, ubest/mbest: 1/0, attached
*via 10.29.176.233, mgmt0, [0/0], 13:11:57, local

IP Route Table for VRF "Green" '*'
denotes best ucast next-hop '*'
denotes best mcast next-hop '[x/y]'
denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

33.33.33.33/32, ubest/mbest: 1/0
*via 35.35.1.254, [1/0], 00:23:44, static
35.35.1.0/24, ubest/mbest: 1/0, attached
*via 35.35.1.2, Eth1/7, [0/0], 00:26:46, direct
35.35.1.2/32, ubest/mbest: 1/0, attached
*via 35.35.1.2, Eth1/7, [0/0], 00:26:46,
local 44.44.44.44/32, ubest/mbest: 1/0
*via 45.45.1.254%Shared, [20/0], 00:12:08, bgp-100, external, tag
100 100.100.100.100/32, ubest/mbest: 1/0
*via 55.55.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100

IP Route Table for VRF "Shared" '*'
denotes best ucast next-hop '*'
denotes best mcast next-hop '[x/y]'
denotes [preference/metric]
'%<string>' in via output denotes VRF <string>

33.33.33.33/32, ubest/mbest: 1/0
*via 35.35.1.254%Green, [20/0], 00:12:34, bgp-100, external, tag
100 44.44.44.44/32, ubest/mbest: 1/0
*via 45.45.1.254, [1/0], 00:23:16, static
45.45.1.0/24, ubest/mbest: 1/0, attached
*via 45.45.1.2, Eth1/11, [0/0], 00:25:53,
direct 45.45.1.2/32, ubest/mbest: 1/0, attached
*via 45.45.1.2, Eth1/11, [0/0], 00:25:53,
local 100.100.100.100/32, ubest/mbest: 1/0
*via 55.55.55.1%default, [20/0], 00:07:41, bgp-100, external, tag 100

```

This example shows how to configure route leaking from a non-default VRF to the default VRF:

```

feature bgp
vrf context vpn1 address-
  family ipv4 unicast
    import vrf default map importmap
    export vrf default map exportmap

show bgp ipv4 unicast 123.123.123.123/32
BGP routing table information for VRF default, address family IPv4

```

```
Unicast BGP routing table entry for 123.123.123.123/32, version 6
Paths: (1 available, best #1)
Flags: (0x8008001a) on xmit-list, is in urib, is best urib route
```

```
Advertised path-id 1
Path type: redistrib, path is valid, is best path
      Imported from 100:1:123.123.123.123/32 (VRF vpn1)
AS-Path: NONE, path locally originated
  0.0.0.0 (metric 0) from 0.0.0.0 (1.1.1.1)
  Origin incomplete, MED 0, localpref 100, weight
32768 Extcommunity: RT:100:1

Path-id 1 not advertised to any peer
Path-id 1 scheduled to be advertised to peers:
  2.2.2.2
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

  None
  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 : importmap
  Export default limit : 1000
  Export default prefix count : 3
  Export default map : exportmap
```


13.8.1 Related Documents

Related Topic	Document Title
BGP	<i>Chapter 9, “Configuring Basic BGP”</i>
VRFs	<i>Inspur CN12900 Series INOS-CN Fundamentals Configuration Guide</i> <i>Inspur CN12900 Series INOS-CN System Management Configuration Guide</i>

13.8.2 Standards

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

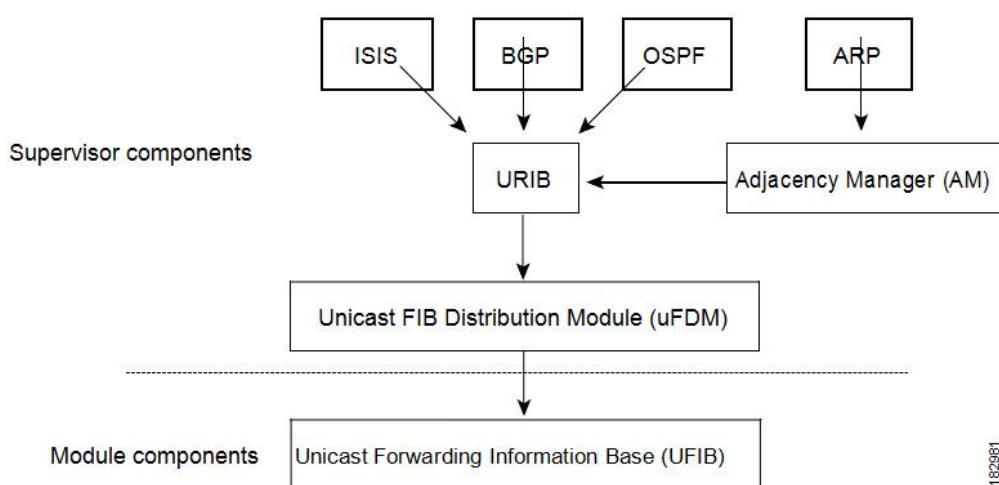
CHAPTER 14 Managing the Unicast RIB and FIB

This chapter describes how to manage routes in the unicast Routing Information Base (RIB) and the Forwarding Information Base (FIB) on the Inspur INOS-CN device.

14.1 About the Unicast RIB and FIB

The unicast RIB (IPv4 RIB and IPv6 RIB) and FIB are part of the Inspur INOS-CN forwarding architecture, as shown in Figure 14-1.

Figure 14-1 Inspur INOS-CN 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).

14.1.1 Layer 3 Consistency Checker

In rare instances, an inconsistency can occur between the unicast RIB and the FIB on each module. Inspur INOS-CN 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.

You can then manually clear any inconsistencies.

14.2 Licensing Requirements for the Unicast RIB and FIB

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	The unicast RIB and FIB require no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

14.3 Managing the Unicast RIB and FIB

14.3.1 Displaying Module FIB Information

You can display the FIB information on a module.

DETAILED STEPS

To display the FIB information on a module, use the following commands in any mode:

Command	Purpose
<pre>show forwarding {ipv4 ipv6} adjacency module slot Example: switch# show forwarding ipv6 adjacency module 2</pre>	Displays the adjacency information for IPv4 or IPv6.
<pre>show forwarding {ipv4 ipv6} route module slot Example: switch# show forwarding ipv6 route module 2</pre>	Displays the route table for IPv4 or IPv6.

14.3.2 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 Routing Information Base (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 Forwarding Information Base (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.

To configure the unicast FIB load-sharing algorithm, use the following command in global configuration mode:

Command	Purpose
<code>ip load-sharing address {destination}</code>	Configures the unicast FIB load-sharing algorithm for

Command	Purpose
<pre>port destination source-destination [port source-destination]} [universal-id seed][rotate rotate] [concatenation] Example: switch(config)# ip load-sharing address source-destination</pre>	<p>data traffic.</p> <ul style="list-style-type: none"> The universal-id option sets the random seed for the hash algorithm and shifts the flow from one link to another. <p>You do not need to configure the universal ID. Inspur INOS-CN chooses the universal ID if you do not configure it. The <i>universal-id</i> range is from 1 to 4294967295.</p> <ul style="list-style-type: none"> 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 <i>rotate</i> value, the 64-bit stream is interpreted starting from that bit position in a cyclic rotation. The <i>rotate</i> range is from 1 to 63, and the default is 32. <p>Note With multi-tier Layer 3 topology, polarization is possible. To avoid polarization, use a different rotate bit at each tier of the topology.</p> <p>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 <i>Inspur CN12900 Series INOS-CN Interfaces Configuration Guide</i>.</p> <ul style="list-style-type: none"> 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
<pre>show ip load-sharing Example: switch(config)# show ip load-sharing address source-destination</pre>	<p>Displays the unicast FIB load-sharing algorithm for data traffic.</p>

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
<pre>show routing hash source-addr dest-addr [source-port dest-port] [vrf vrf-name] Example: switch# show routing hash 192.0.2.1 10.0.0.1</pre>	<p>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.</p>

This example shows how to display the route selected for a source/destination pair:

```
switch# show routing hash 10.0.0.5 192.0.0.2
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 *172.0.0.2 (hash: 0x0e), for route:
```

14.3.3 Displaying Routing and Adjacency Information

You can display the routing and adjacency information.

To display the routing and adjacency information, use the following commands in any mode:

Command	Purpose
<pre>show {ip ipv6} route [route-type interface int-type number next-hop] Example: switch# show ip route</pre>	<p>Displays the unicast route table. The <i>route-type</i> argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the ? command to see the supported interfaces.</p>
<pre>show {ip ipv6} adjacency [prefix interface-type number [summary] non-best] [detail] [vrf vrf-id] Example: switch# show ip adjacency</pre>	<p>Displays the adjacency table. The argument ranges are as follows:</p> <ul style="list-style-type: none"> <i>prefix</i>—Any IPv4 or IPv6 prefix address. <i>interface-type number</i>—Use the ? command to see the supported interfaces. <i>vrf-id</i>—Any case-sensitive, alphanumeric string up to 64 characters.
<pre>show {ip ipv6} routing [route-type interface int-type number next-hop recursive-next-hop summary updated {since until} time] Example: switch# show routing summary</pre>	<p>Displays the unicast route table. The <i>route-type</i> argument can be a single route prefix, direct, static, or a dynamic route protocol. Use the ? command to see the supported interfaces.</p>

This example shows how to display 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

```

This example shows how to display 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

```

14.3.4 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
<pre> test forwarding [ipv4 ipv6] [unicast] inconsistency [vrf vrf-name] [module {slot all}] Example: switch(config)# test forwarding inconsistency </pre>	<p>Starts a Layer 3 consistency check. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters. The <i>slot</i> range is from 1 to 26.</p>

To stop the Layer 3 consistency checker, use the following commands in global configuration mode:

Command	Purpose
<pre> test forwarding [ipv4 ipv6] [unicast] inconsistency [vrf vrf-name] [module {slot all}] stop Example: switch# test forwarding inconsistency </pre>	<p>Stops a Layer 3 consistency check. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters. The <i>slot</i> range is from 1 to 26.</p>

 stop

To display the Layer 3 inconsistencies, use the following commands in any mode:

Command	Purpose
<pre>show forwarding [ipv4 ipv6] inconsistency [vrf vrf-name] [module (slot all)]</pre> <p>Example:</p> <pre>switch# show forwarding inconsistency</pre>	<p>Displays the results of a Layer 3 consistency check. The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters. The <i>slot</i> range is from 1 to 26.</p>

14.3.5 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.

To clear an entry in the FIB, including a Layer 3 inconsistency, use the following command in any mode:

Command	Purpose
<pre>clear forwarding {ipv4 ipv6} route {* prefix} [vrf vrf-name] module (slot all)</pre> <p>Example:</p> <pre>switch# clear forwarding ipv4 route * module 1</pre>	<p>Clears one or more entries from the FIB. The route options are as follows:</p> <ul style="list-style-type: none"> *—All routes. <i>prefix</i>—Any IP or IPv6 prefix. <p>The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters. The <i>slot</i> range is from 1 to 26.</p>

14.3.6 Configuring Maximum Routes for the Unicast RIB

You can configure the maximum number of routes allowed in the routing table.

SUMMARY STEPS

1. **configure terminal**
2. **vrf context** *vrf-name*
3. **ipv4 unicast**
4. **maximum routes** *max-routes* [*threshold* [*reinstall threshold*] | **warning-only**]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	<pre>configure terminal</pre> <p>Example:</p> <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	<pre>vrf context vrf-name</pre> <p>Example:</p>	Creates a VRF and enters VRF configuration mode.

	Command	Purpose
	<pre>switch(config)# vrf context Red switch(config-vrf)#</pre>	
Step 3	<pre>ipv4 unicast Example: switch(config-vrf)# ipv4 unicast switch(config-vrf-af-ipv4)#</pre>	Enters address-family configuration mode.
Step 4	<pre>maximum routes max-routes [threshold [reinstall threshold] warning-only] Example: switch(config-vrf-af-ipv4)# maximum routes 250 90</pre>	<p>Configures the maximum number of routes allowed in the routing table. The range is from 1 to 4294967295. You can optionally specify the following:</p> <ul style="list-style-type: none"> • <i>threshold</i>—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	<pre>copy running-config startup-config Example: switch(config-vrf-af-ipv4)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

14.3.7 Estimating Memory Requirements for Routes

You can estimate the memory that a number of routes and next-hop addresses will use.

To estimate the memory requirements for routes, use the following command in any mode:

Command	Purpose
<pre>show routing {ipv6} memory estimate routes num-routes next-hops num-nexthops Example: switch# show routing memory estimate routes 5000 next-hops 2</pre>	Displays the memory requirements for routes. The <i>num-routes</i> range is from 1000 to 1000000. The <i>num-nexthops</i> range is from 1 to 16.

14.3.8 Clearing Routes in the Unicast RIB

You can clear one or more routes from the unicast RIB.

Caution	The * keyword is severely disruptive to routing.
----------------	--

To clear one or more entries in the unicast RIB, use the following commands in any mode:

Command	Purpose
<pre>clear {ip ipv4 ipv6} route {* {route prefix/length}[next-hop interface]} [vrf vrf-name]</pre> <p>Example: switch(config)# clear ip route 10.2.2.2</p>	<p>Clears one or more routes from both the unicast RIB and all the module FIBs. The route options are as follows:</p> <ul style="list-style-type: none"> • *—All routes. • <i>route</i>—An individual IP or IPv6 route. • <i>prefix/length</i>—Any IP or IPv6 prefix. • <i>next-hop</i>—The next-hop address • <i>interface</i>—The interface to reach the next-hop address. <p>The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.</p>
<pre>clear routing [multicast unicast] [ip ipv4 ipv6] {* {route prefix/length}[next-hop interface]} [vrf vrf-name]</pre> <p>Example: switch(config)# clear routing ip 10.2.2.2</p>	<p>Clears one or more routes from the unicast RIB. The route options are as follows:</p> <ul style="list-style-type: none"> • *—All routes. • <i>route</i>—An individual IP or IPv6 route. • <i>prefix/length</i>—Any IP or IPv6 prefix. • <i>next-hop</i>—The next-hop address • <i>interface</i>—The interface to reach the next-hop address. <p>The <i>vrf-name</i> can be any case-sensitive, alphanumeric string up to 64 characters.</p>

14.4 Verifying the Unicast RIB and FIB

To display the unicast RIB and FIB information, perform one the following tasks:

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 ipv6} adjacency	Displays the adjacency table.
show {ip ipv6} route	Displays IPv4 or IPv6 routes from the unicast RIB.
show routing	Displays routes from the unicast RIB.

14.4.1 Related Documents

Related Topic	Document Title
Configuring EEM	<i>Inspur CN12900 Series INOS-CN System Management</i>

*Configuration
Guide*

CHAPTER 15 Configuring Route Policy Manager

This chapter describes how to configure the Route Policy Manager on the Inspur INOS-CN device.

15.1 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 or IPv6 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.

15.1.1 Prefix 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-CN assigns a sequence number automatically. Inspur INOS-CN evaluates prefix lists starting with the lowest sequence number. Inspur INOS-CN processes the first successful match for a given prefix. Once a match occurs, Inspur INOS-CN processes the permit or deny statement and does not evaluate the rest of the prefix list.

15.1.2 Prefix List Masks

Inspur INOS-CN supports masks for prefix lists. Masking uses the number 1 and the number 0 to specify how the software treats the corresponding IP address bits.

- A mask bit 0 means ignore the corresponding bit value.
- A mask bit 1 means check the corresponding bit value for an exact match.

You can use a prefix list to match the IP address in a route map, which in turn is used in routing protocols during redistribution. The IP address is matched against the prefix list where the bits corresponding to the mask bit 1 are the same as the subnet provided in the prefix list.

By carefully setting masks, you can select one or several IP addresses for permit or deny tests.

The prefix list mask allows noncontiguous bits in the mask. You can thus define a range of even- or odd-numbered IP addresses.

15.1.3 Route Maps

You can use route maps for route redistribution. 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

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-CN 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-CN 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:

- 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 a rendezvous point, groups, or sources.
- Other parameters—Match based on an 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, community, extended community, dampening, local preference, origin, or weight attributes.
- Metrics—Change the route-metric, the route-tag, or the route-type.
- 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

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.

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 user-defined network number.
- A regular expression.

Inspur INOS-CN 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.

15.1.4 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. If the router processes the route against all entries in a route map with no match, the router accepts the route (inbound route maps) or forwards the route (outbound route maps).

15.2 Licensing Requirements for Route Policy Manager

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	Route Policy Manager requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

15.3 Guidelines and Limitations

Route Policy Manager has the following configuration 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-CN fails the match statement and processes the next route-map entry.
- When you change a route map, Inspur INOS-CN holds all the changes until you exit from the route-map configuration submode. Inspur INOS-CN then sends all the changes to the protocol clients to take effect.
- Inspur recommends that you do not have both IPv4 and IPv6 match statements in the same route-map sequence. If both are required, they should be specified in different sequences in the same route-map.
- 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.
- Route Policy Manager does not support MAC lists.
- The maximum number of characters for ACL names in the **ip access-list name** command is 64. However, ACL names that are associated with RPM commands (such as **ip prefix-list** and **match ip address**) accept a maximum of only 63 characters.
- BGP supports only specific **match** commands.
- If you create an ACL named "prefix-list," it cannot be associated with a route map that is created using the **match ip address** command. The RPM command **match ip address prefix-list** makes the previous command (with the "prefix-list" ACL name) ambiguous.
- You can configure only one ACL when using the **match ip address** command.

15.4 Default Settings

Table 15-1 lists the default settings for Route Policy Manager.

Table 15-1 Default Route Policy Manager Parameters

Parameters	Default
Route Policy Manager	Enabled
Administrative distance	115

15.5 Configuring Route Policy Manager

15.5.1 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).

Use the **mask** keyword to define a range of possible contiguous or non-contiguous routes to be compared to the prefix address.

SUMMARY STEPS

1. **configure terminal**
2. **{ip | ipv6} prefix-list name description string**
3. **ip prefix-list name [seq number] [{permit | deny} prefix {[eq prefix-length] | [ge prefix-length] [le prefix-length]}] [mask mask]** or **ipv6 prefix-list name [seq number] [{permit | deny} prefix {[eq prefix-length] | [ge prefix-length] [le prefix-length]}] [mask mask]**
4. (Optional) **show {ip | ipv6} prefix-list name**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	{ip ipv6} prefix-list name description string Example: switch(config)# ip prefix-list AllowPrefix description allows engineering server	(Optional) Adds an information string about the prefix list.
Step 3	ip prefix-list name [seq number] [{permit deny} prefix {[eq prefix-length] [ge prefix-length] [le prefix-length]}] [mask mask] Example: switch(config)# ip prefix-list AllowPrefix seq 10 permit 192.0.2.0/24 eq 24 Example: switch(config)# ip prefix-list even permit 0.0.0.0/32 mask 0.0.0.1	Creates an IPv4 prefix list or adds a prefix to an existing prefix list. The prefix length is matched as follows: <ul style="list-style-type: none"> • eq—Matches the exact <i>prefix length</i>. • ge—Matches a prefix length that is equal to or greater than the configured <i>prefix length</i>. • le—Matches a prefix length that is equal to or less than the configured <i>prefix length</i>. • mask—Specifies the bits of a prefix address in a prefix list that are compared to the bits of the prefix address used in routing protocols during redistribution.
	ipv6 prefix-list name [seq number] [{permit deny} prefix {[eq	Creates an IPv6 prefix list or adds a prefix to an existing prefix list. The prefix length is configured as

Command	Purpose
<pre> prefix-length] [ge prefix-length] [le prefix-length]]] [mask mask] Example: switch(config)# ipv6 prefix-list AllowIPv6Prefix seq 10 permit 2001:0DB8:: 1e 32 </pre>	<p>follows:</p> <ul style="list-style-type: none"> • eq—Matches the exact <i>prefix length</i>. • ge—Matches a prefix length that is equal to or greater than the configured <i>prefix length</i>. • le—Matches a prefix length that is equal to or less than the configured <i>prefix length</i>. • mask—Specifies the bits of a prefix address in a prefix list that are compared to the bits of the prefix address used in routing protocols during redistribution.
<p>Step 4</p> <pre> show {ip ipv6} prefix-list name Example: switch(config)# show ip prefix-list AllowPrefix </pre>	<p>(Optional) Displays information about prefix lists.</p>
<p>Step 5</p> <pre> copy running-config startup-config Example: switch# copy running-config startup-config </pre>	<p>(Optional) Saves this configuration change.</p>

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 65535
switch(config-router)# neighbor 192.0.2.1/16 remote-as 65534
switch(config-router-neighbor)# address-family ipv4 unicast
switch(config-router-neighbor-af)# prefix-list allowprefix in

```

This example shows how to create an IPv4 prefix list with a match mask for all /24 odd IP addresses:

```

switch# configure terminal
switch(config)# ip prefix-list list1 seq 7 permit 22.1.1.0/24 mask 255.255.1.0
switch(config)# show route-map test
route-map test, permit, sequence
  7 Match clauses:
    ip address prefix-lists:
      list1 Set clauses:
        extcommunity COST:igp:10:20
switch(config)# show ip prefix-list list1
ip prefix-list list1: 1 entries
  seq 7 permit 22.1.1.0/24 mask 255.255.1.0

```

This example shows how to create an IPv4 prefix list that matches all subnets of 21.1.0.0/16 where the subnet prefix is 17 or greater. Due to the **mask** option, only those incoming prefixes where the first bit in the third octet is unset (even) will be matched.

```

switch# configure terminal

```



```
switch(config)# ip prefix-list list1 seq 10 permit 21.1.0.0/16 ge 17 mask 255.255.1.0
```

15.5.2 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.

SUMMARY STEPS

1. **configure terminal**
2. **ip as-path access-list** *name* {deny | permit} *expression*
3. (Optional) **show** {ip | ipv6} **as-path list** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip as-path access-list <i>name</i> {deny permit} <i>expression</i> Example: switch(config)# ip as-path access-list Allow40 permit 40	Creates a BGP AS-path list using a regular expression.
Step 3	show {ip ipv6} as-path-access-list <i>name</i> Example: switch(config)# show ip as-path-access-list Allow40	(Optional) Displays information about as-path access lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

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 65535:20
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
```

15.5.3 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.

SUMMARY STEPS

1. **configure terminal**
2. **ip community-list standard** *list-name* {deny | permit} [*community-list*] [internet] [local-AS] [no-advertise] [no-export] or **ip community-list expanded** *list-name* {deny | permit} *expression*
3. (Optional) **show ip community-list** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip community-list standard <i>list-name</i> {deny permit} [<i>community-list</i>] [internet] [local-AS] [no-advertise] [no-export] Example: switch(config)# ip community-list standard BGPCommunity permit no-advertise 65535:20	Creates a standard BGP community list. The <i>list-name</i> can be any case-sensitive, alphanumeric string up to 63 characters. The <i>community-list</i> can be one or more communities in the <i>aa:nn</i> format.
	ip community-list expanded <i>list-name</i> {deny permit} <i>expression</i> Example: switch(config)# ip community-list expanded BGPComplex deny 50000:[0-9][0-9]_	
Step 3	show ip community-list <i>name</i> Example: switch(config)# show ip community-list BGPCommunity	(Optional) Displays information about community lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

This example shows how to create a community list with two entries:

```
switch# configure terminal
switch(config)# ip community-list standard BGPCCommunity permit no-advertise
65535:20
switch(config)# ip community-list standard BGPCCommunity permit local-AS no-export
switch(config)# copy running-config startup-config
```

15.5.4 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.

SUMMARY STEPS

1. **configure terminal**
2. **ip extcommunity-list standard** *list-name* {deny | permit} 4bytegeneric {transitive | non-transitive} *aa4:nn* or **ip extcommunity-list expanded** *list-name* {deny | permit} *expression*
3. (Optional) **show ip extcommunity-list** *name*
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	ip extcommunity-list standard <i>list-name</i> {deny permit} 4bytegeneric {transitive nontransitive} <i>community1</i> [<i>community2...</i>] Example: switch(config)# ip extcommunity-list standard BGPExtCommunity permit 4bytegeneric transitive 65535:20	Creates a standard BGP extended community list. The <i>community</i> can be one or more extended communities in the <i>aa4:nn</i> format.
	ip extcommunity-list expanded <i>list-name</i> {deny permit} <i>expression</i> Example: switch(config)# ip extcommunity-list expanded BGPExtComplex deny 1.5:[0-9][0-9]_	
Step 3	show ip community-list <i>name</i>	(Optional) Displays information about extended

	Command	Purpose
	Example: switch(config)# show ip community-list BGPCommunity	community lists.
Step 4	copy running-config startup-config Example: switch# copy running-config startup-config	(Optional) Saves this configuration change.

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 65535:40 65535:60
switch(config)# copy running-config startup-config
```

15.5.5 Configuring Route Maps

You can use route maps for route redistribution or route filtering. Route maps can contain multiple match criteria and multiple set criteria.

Configuring a route map for BGP triggers an automatic soft clear or refresh of BGP neighbor sessions.

SUMMARY STEPS

1. **configure terminal**
2. **route-map** *map-name* [**permit** | **deny**] [*seq*]
3. (Optional) **continue** *seq*
4. (Optional) **exit**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	route-map <i>map-name</i> [permit deny] [<i>seq</i>] Example: switch(config)# route-map Testmap permit 10 switch(config-route-map)#	Creates a route map or enters route-map configuration mode for an existing route map. Use <i>seq</i> to order the entries in a route map.
Step 3	continue <i>seq</i> Example: switch(config-route-map)# continue 10	(Optional) Determines what sequence statement to process next in the route map. Used only for filtering and redistribution.
Step 4	exit Example: switch(config-route-map)# exit	(Optional) Exits route-map configuration mode.

Command	Purpose
Step 5 <code>copy running-config startup-config</code> Example: <pre>switch(config)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

You can configure the following optional match parameters for route maps in route-map configuration mode:

Command	Purpose
match as-path <i>name</i> [<i>name...</i>] Example: <pre>switch(config-route-map)# match as-path Allow40</pre>	Matches against one or more AS-path lists. Create the AS-path list with the ip as-path access-list command.
match as-number { <i>number</i> [, <i>number...</i>] as-path-list <i>name</i> [<i>name...</i>]} Example: <pre>switch(config-route-map)# match as-number 33,50-60</pre>	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.
match community <i>name</i> [<i>name...</i>][exact-match] Example: <pre>switch(config-route-map)# match community BGPCommunity</pre>	Matches against one or more community lists. Create the community list with the ip community-list command.
match extcommunity <i>name</i> [<i>name...</i>][exact-match] Example: <pre>switch(config-route-map)# match extcommunity BGPextCommunity</pre>	Matches against one or more extended community lists. Create the community list with the ip extcommunity-list command.
match interface <i>interface-type number</i> [<i>interface-type number...</i>] Example: <pre>switch(config-route-map)# match interface e 1/2</pre>	Matches any routes that have their next hop out one of the configured interfaces. Use ? to find a list of supported interface types. Note BGP does not support this command.
match ip address prefix-list <i>name</i> [<i>name...</i>] Example: <pre>switch(config-route-map)# match ip address prefix-list AllowPrefix</pre>	Matches against one or more IPv4 prefix lists. Use the ip prefix-list command to create the prefix list.
match ipv6 address prefix-list <i>name</i> [<i>name...</i>] Example: <pre>switch(config-route-map)# match ip address prefix-list AllowIPv6Prefix</pre>	Matches against one or more IPv6 prefix lists. Use the ipv6 prefix-list command to create the prefix list.
match ip multicast [source <i>ipsource</i>] 	Matches an IPv4 multicast packet based on the

Command	Purpose
<pre>[[group ipgroup] [rp iprp]]</pre> <p>Example:</p> <pre>switch(config-route-map)# match ip multicast rp 192.0.2.1</pre>	<p>multicast source, group, or rendezvous point.</p> <p>Note BGP does not support this command.</p>
<pre>match ipv6 multicast [source ipsource] [[group ipgroup] [rp iprp]]</pre> <p>Example:</p> <pre>switch(config-route-map)# match ip multicast source 2001:0DB8::1</pre>	<p>Matches an IPv6 multicast packet based on the multicast source, group, or rendezvous point.</p> <p>Note BGP does not support this command.</p>
<pre>match ip next-hop prefix-list name [name...]</pre> <p>Example:</p> <pre>switch(config-route-map)# match ip next-hop prefix-list AllowPrefix</pre>	<p>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.</p>
<pre>match ipv6 next-hop prefix-list name [name...]</pre> <p>Example:</p> <pre>switch(config-route-map)# match ipv6 next-hop prefix-list AllowIPv6Prefix</pre>	<p>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.</p>
<pre>match ip route-source prefix-list name [name...]</pre> <p>Example:</p> <pre>switch(config-route-map)# match ip route-source prefix-list AllowPrefix</pre>	<p>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.</p>
<pre>match ipv6 route-source prefix-list name [name...]</pre> <p>Example:</p> <pre>switch(config-route-map)# match ipv6 route-source prefix-list AllowIPv6Prefix</pre>	<p>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.</p>
<pre>match metric value [+ deviation.] [value..]</pre> <p>Example:</p> <pre>switch(config-route-map)# match metric 50 + 10</pre>	<p>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 within the range:</p> <p style="text-align: center;"><i>value - deviation to value + deviation.</i></p>
<pre>match ospf-area area-id</pre> <p>Example:</p> <pre>switch(config-route-map)# match ospf-area 1</pre>	<p>Matches the OSPFv2 or OSPFv3 area ID. The <i>area-id</i> range is from 0 to 4294967295.</p> <p>Note BGP does not support this command.</p>
<pre>match route-type route-type</pre> <p>Example:</p> <pre>switch(config-route-map)# match route-type level 1 level 2</pre>	<p>Matches against a type of route. The <i>route-type</i> can be one or more of the following:</p> <ul style="list-style-type: none"> • external—The external route (BGP, EIGRP, and OSPF type 1 or 2) • inter-area—The OSPF inter-area route

Command	Purpose
	<ul style="list-style-type: none"> • internal—The internal route (including the OSPF intra- or inter-area) • intra-area—The OSPF intra-area route • level-1—The IS-IS level 1 route • level-2—The IS-IS level 2 route • local—The locally generated 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 <p>Note BGP does not support this command.</p>
match tag <i>tagid</i> [<i>tagid...</i>] Example: <pre>switch(config-route-map)# match tag 2</pre>	<p>Matches a route against one or more tags for filtering or redistribution.</p> <p>Note BGP does not support this command.</p>
match vlan <i>vlan-id</i> [<i>vlan-range</i>] Example: <pre>switch(config-route-map)# match vlan 3, 5-10</pre>	<p>Matches against a VLAN.</p> <p>Note BGP does not support this command.</p>

You can configure the following optional set parameters for route maps in route-map configuration mode:

Command	Purpose
set as-path { <i>tag</i> prepend { <i>last-as number</i> <i>as-1</i> [<i>as-2...</i>]}} Example: <pre>switch(config-route-map)# set as-path prepend 10 100 110</pre>	<p>Modifies an AS-path attribute for a BGP route. You can prepend the configured <i>number</i> of last AS numbers or a string of particular AS-path values (<i>as-1 as-2...as-n</i>).</p>
set comm-list <i>name</i> delete Example: <pre>switch(config-route-map)# set comm-list BGPCommunity delete</pre>	<p>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.</p>
set community { <i>none</i> additive local-AS no-advertise no-export <i>community-1</i> [<i>community-2...</i>]} Example: <pre>switch(config-route-map)# set community local-AS</pre>	<p>Sets the community attribute for a BGP route update.</p> <p>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.</p> <p>Note Use the send-community command in BGP neighbor address-family configuration mode to propagate BGP</p>

Command	Purpose
	community attributes to BGP peers.
<pre>set dampening halflife reuse suppress duration Example: switch(config-route-map)# set dampening 30 1500 10000 120</pre>	<p>Sets the following BGP route dampening parameters:</p> <ul style="list-style-type: none"> <i>halflife</i>—The range is from 1 to 45 minutes. The default is 15. <i>reuse</i>—The range is from is 1 to 20000 seconds. The default is 750. <i>suppress</i>—The range is from is 1 to 20000. The default is 2000. <i>duration</i>—The range is from is 1 to 255 minutes. The default is 60.
<pre>set distance value Example: switch(config-route-map)# set distance 150</pre>	Sets the administrative distance of routes for OSPFv2 or OSPFv3. The range is from 1 to 255.
<pre>set extcomm-list name delete Example: switch(config-route-map)# set extcomm-list BGPextCommunity delete</pre>	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.
<pre>set extcommunity 4byteas-generic {transitive nontransitive} {none additive} community-1 [community-2...]} Example: switch(config-route-map)# set extcommunity generic transitive 1.0:30</pre>	<p>Sets the extended community attribute for a BGP route update.</p> <p>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.</p> <p>Note Use the send-community command in BGP neighbor address-family configuration mode to propagate BGP extended community attributes to BGP peers.</p>
<pre>set extcommunity cost community-id1 cost [igp pre-bestpath] [community-id2...]} Example: switch(config-route-map)# set extcommunity cost 33 1.0:30</pre>	<p>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 <i>community-id</i> range is from 0 to 255. The <i>cost</i> 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.</p>
<pre>set extcommunity rt community-1 [additive]</pre>	Sets the extended community route target attribute

Command	Purpose
<pre>[community-2...]} Example: switch(config-route-map)# set extcommunity rt 1.0:30</pre>	<p>for a BGP route update. The <i>community</i> 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.</p> <p>Use the additive keyword to add a route target to an existing extended community route target attribute.</p>
<pre>set forwarding-address Example: switch(config-route-map)# set forwarding-address</pre>	<p>Sets the forwarding address for OSPF.</p>
<pre>set ip next-hop unchanged Example: switch(config-route-map)# set ip next-hop unchanged</pre>	<p>Specifies an unchanged next-hop IP address. This command is required for BGP IPv6-over-IPv4 peering.</p>
<pre>set level {backbone level-1 level-1-2 level-2} Example: switch(config-route-map)# set level backbone</pre>	<p>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.</p>
<pre>set local-preference value Example: switch(config-route-map)# set local-preference 4000</pre>	<p>Sets the BGP local preference value. The range is from 0 to 4294967295.</p>
<pre>set metric [+ -]bandwidth-metric Example: switch(config-route-map)# set metric +100</pre>	<p>Adds or subtracts from the existing metric value. The metric is in Kb/s. The range is from 0 to 4294967295.</p>
<pre>set metric bandwidth [delay reliability load mtu] Example: switch(config-route-map)# set metric 33 44 100 200 1500</pre>	<p>Sets the route metric values. Metrics are as follows:</p> <ul style="list-style-type: none"> <i>metric0</i>—Bandwidth in Kb/s. The range is from 0 to 4294967295. <i>metric1</i>—Delay in 10-microsecond units. <i>metric2</i>—Reliability. The range is from 0 to 255 (100 percent reliable). <i>metric3</i>—Loading. The range is from 1 to 255 (100 percent loaded). <i>metric4</i>—MTU of the path. The range is from 1 to 16777215.
<pre>set metric-type {external internal type-1 type-2} Example: switch(config-route-map)# set metric-type internal</pre>	<p>Sets the metric type for the destination routing protocol. The options are as follows:</p> <p>external—IS-IS external metric</p> <p>internal—IGP metric as the MED for BGP</p>

Command	Purpose
	type-1—OSPF external type 1 metric type-2—OSPF external type 2 metric
set nssa-only Example: <pre>switch(config-route-map)# set nssa-only</pre>	Sets Type-7 LSA generated on ASBR with no P bit set. This prevents Type-7 to Type-5 LSA translation in OSPF.
set origin {egp as-number igp incomplete} Example: <pre>switch(config-route-map)# set origin incomplete</pre>	Sets the BGP origin attribute. The EGP <i>as-number</i> range is from 0 to 65535.
set tag name Example: <pre>switch(config-route-map)# set tag 33</pre>	Sets the tag value for the destination routing protocol. The <i>name</i> parameter is an unsigned integer. Note BGP does not support this command. Note For EIGRP internal routes in the default (narrow) metric version, 255 (0xFF) is the maximum tag value that can be assigned. In the wide (64-bit) metric version, the tag value can be as large as 0xFF FF FF FF.
set weight count Example: <pre>switch(config-route-map)# set weight 33</pre>	Sets the weight for the BGP route. The range is from 0 to 65535.

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, Inspur INOS-CN ignores the **metric-type internal** command.

15.6 Verifying the Route Policy Manager Configuration

To display route policy manager configuration information, perform one of the following tasks:

Command	Purpose
show ip community-list [<i>name</i>]	Displays information about a community list.
show ip extcommunity-list [<i>name</i>]	Displays information about an extended community list.
show [ip ipv6] prefix-list [<i>name</i>]	Displays information about an IPv4 or IPv6 prefix list.
show route-map [<i>name</i>]	Displays information about a route map.

15.7 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 172.16.0.1 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 172.16.201.0/27
```

CHAPTER 16 Configuring Policy-Based Routing

This chapter describes how to configure policy-based routing on the Inspur INOS-CN device.

16.1 About Policy-Based Routing

With policy-based routing, you can configure a defined policy for IPv4 and IPv6 traffic flows that lessens the 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 that determines where to forward packets.

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 CN12900 Series INOS-CN Quality of Service Configuration Guide*).
- Load sharing—Distributes traffic among multiple paths based on the traffic characteristics.

16.1.1 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. You can interpret the statements as follows:

- If the statement is marked as permit and the packets meet the match criteria, the set clause is 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.

16.1.2 Set Criteria for Policy-Based Routing

The Inspur CN12900 Series switches support the following **set** commands for route maps used in policy-based routing:

- **set {ip | ipv6} next-hop address1 [address2...] [load-share]**
- **set interface null0**

These **set** commands are mutually exclusive within the route-map sequence.

In the first command, the IP address specifies 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.

If the packets do not meet any of the defined match criteria, those packets are routed through the normal destination-based routing process.

16.1.3 Route-Map Processing Logic

When a packet is received on an interface that is configured with a route map, the forwarding logic processes each route-map statement according to the sequence number.

If the route-map statement encountered is a **route-map...permit** statement, the packet is matched against the criteria in the **match** command. This command may refer to an ACL that has one or more access control entries (ACEs). If the packet matches the permit ACEs in the ACL, the policy-based routing logic executes the action specified by the **set** command on the packet.

If the route-map statement encountered is a **route-map... deny** statement, the packet is matched against the criteria in the **match** command. This command may refer to an ACL that has one or more ACEs. If the packet matches the permit ACEs in the ACL, policy-based routing processing terminates, and the packet is routed using the default IP routing table.

If the route-map configuration does not contain a match statement, the policy-based routing logic executes the action specified by the **set** command on the packet. All packets are routed using policy-based routing.

If the route-map configuration references a match statement but the match statement references a non-existing ACL or an existing ACL without any access control entries (ACEs), the packet is routed using the default routing table.

If the next-hop specified in the **set {ip | ipv6} next-hop** command is down, is not reachable, or is removed, the packet is routed using the default routing table.

16.1.4 Policy-Based Routing Filtering Options

You can identify traffic by using additional options. The following list includes most but not all additional filtering options.

Policy-based routing ACLs support the following additional filtering options:

- Layer 3 source and/or destination address
- TCP and UDP ports
- Precedence level*
- Differentiated Services Code Point (DSCP) value*
- TCP packets with the ACK, FIN, PSH, RST, SYN, or URG bit set*
- Established TCP connections*
- Packet length*

*The Inspur CN12900 Series switches do not support these filtering options.

16.2 Licensing Requirements for Policy-Based Routing

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	Policy-based routing requires an Essentials license package. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

16.3 Prerequisites for Policy-Based Routing

Policy-based routing has the following prerequisites:

- 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.

16.4 Guidelines and Limitations

Policy-based routing has the following configuration guidelines and limitations:

- A policy-based routing route map can have only one match statement per route-map statement.
- A policy-based routing route map can have only one set statement per route-map statement, unless you are using IP SLA policy-based routing.
- 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.
- Using a prefix list as a match criteria is not supported. Do not use a prefix list in a policy-based routing route map.
- Policy-based routing supports only unicast traffic. Multicast traffic is not supported.
- The Inspur CN12900 Series switches support policy-based routing with Layer 3 port-channel subinterfaces.
- An ACL used in a policy-based routing route map cannot include deny access control entries (ACEs).
- Policy-based routing is supported only in the default system routing mode.
- The Inspur CN12900 Series switches do not support the **set vrf** and **set default next-hop** commands.
- Policy-based routing traffic cannot be balanced if the next hop is recursive over ECMP paths. Instead, use the **set {ip | ipv6} next-hop ip-address load-share** command to specify the adjacent next hops.
- When you configure multiple features on an interface (such as PBR and ingress ACL), the ACLs for those features are merged for TCAM optimization. As a result, statistics are not supported.
- The Inspur CN12900 Series switches do not support policy-based routing with VXLAN.
- The Inspur CN12900 Series switches do not support policy-based ACLs (PBACLs), also referred to as object-group ACLs.
- The Inspur CN12900 Series switches support IPv4 and IPv6 policy-based routing. PBR policy has a higher priority over attached and local routes. Explicit white listing might be required if protocol neighbors are directly attached.

16.5 Default Settings

Table 16-1 lists the default settings for policy-based routing.

Table 16-1 Default Policy-Based Routing Parameters

Parameters	Default
Policy-based routing	Disabled

16.6 Configuring Policy-Based Routing

This section includes the following topics:

- Enabling the Policy-Based Routing Feature
- Configuring a Route Policy

16.6.1 Enabling the Policy-Based Routing Feature

You must enable the policy-based routing feature before you can configure a route policy.

SUMMARY STEPS

1. **configure terminal**
2. **[no] feature pbr**
3. (Optional) **show feature**

4. (Optional) copy running-config startup-config

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] feature pbr Example: switch(config)# feature pbr	Enables the policy-based routing feature. Use the no form of this command to disable the policy-based routing feature. Note The no feature pbr command removes the policies applied under the interfaces. It does not remove the ACL or route-map configuration nor does it create a system checkpoint.
Step 3	show feature Example: switch(config)# show feature	(Optional) Displays enabled and disabled features.
Step 4	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

16.6.2 Configuring a Route Policy

You can use route maps in policy-based routing to assign routing policies to the inbound interface. Inspur INOS-CN routes the packets as soon as it finds a next hop and an interface.

BEFORE YOU BEGIN

The Inspur CN12900 Series switches have an IPv4 RACL TCAM region by default for IPv4 traffic.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type slot/port*
3. **{ip | ipv6} policy route-map** *map-name*
4. **route-map** *map-name* [**permit** | **deny**] [*seq*]
5. **match {ip | ipv6} address** *access-list-name name* [*name...*]
6. (Optional) **set ip next-hop** *address1* [*address2...*] [**load-share**]
7. (Optional) **set ipv6 next-hop** *address1* [*address2...*] [**load-share**]
8. (Optional) **set interface** **null0**
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal	Enters global configuration mode.

	Command	Purpose
	Example: <pre>switch# configure terminal switch(config)#</pre>	
Step 2	interface <i>type slot/port</i> Example: <pre>switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 3	{ip ipv6} policy route-map <i>map-name</i> Example: <pre>switch(config-if)# ip policy route-map Testmap</pre>	Assigns a route map for IPv4 or IPv6 policy-based routing to the interface.
Step 4	route-map <i>map-name</i> [permit deny] [<i>seq</i>] Example: <pre>switch(config-if)# route-map Testmap switch(config-route-map)#</pre>	Creates a route map or enters route-map configuration mode for an existing route map. Use <i>seq</i> to order the entries in a route map.
Step 5	match {ip ipv6} address access-list-name <i>name</i> <i>[name...]</i> Example: <pre>switch(config-route-map)# match ip address access-list-name ACL1</pre>	Matches an IPv4 or IPv6 address against one or more IP or IPv6 access control lists (ACLs). This command is used for policy-based routing and is ignored by route filtering or redistribution.
Step 6	set ip next-hop <i>address1</i> [<i>address2...</i>] <i>[load-share]</i> Example: <pre>switch(config-route-map)# set ip next-hop 192.0.2.1</pre>	(Optional) Sets the IPv4 next-hop address for policy-based routing. This command uses the first valid next-hop address if multiple addresses are configured. Use the optional load-share keyword to load balance traffic across a maximum of 32 next-hop addresses. For PBR with VXLAN, the load-share keyword is not required.
Step 7	set ipv6 next-hop <i>address1</i> [<i>address2...</i>] <i>[load-share]</i> Example: <pre>switch(config-route-map)# set ipv6 next-hop 2001:0DB8::1</pre>	(Optional) Sets the IPv6 next-hop address for policy-based routing. This command uses the first valid next-hop address if multiple addresses are configured. Use the optional load-share keyword to load balance traffic across a maximum of 32 next-hop addresses. For PBR with VXLAN, the load-share keyword is not required.
Step 8	set interface null0 Example: <pre>switch(config-route-map)# set interface null0</pre>	(Optional) Sets the interface used for routing. Use the null0 interface to drop packets.
Step 9	copy running-config startup-config Example:	(Optional) Saves this configuration change.

Command	Purpose
switch(config-route-map)# copy running-config startup-config	

16.7 Verifying the Policy-Based Routing Configuration

To display the 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 route-map [name] pbr-statistics	Displays policy statistics.

16.8 Configuration Examples for Policy-Based Routing

This example shows how to configure a simple route policy on an interface:

```
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:

```
switch# 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

switch# show route-map pbr-sample pbr-statistics
route-map pbr-sample, permit, sequence 10
Policy routing matches: 84 packets
Default routing: 233 packets

switch# show ip policy
Interface  Route-map      VRF-Name
Status
Ethernet1/2 pbr-sample  --
Active
```

CHAPTER 17 Configuring HSRP

This chapter describes how to configure the Hot Standby Router Protocol (HSRP) on the Inspur INOS-CN device.

17.1 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.

17.1.1 HSRP Overview

When you use HSRP, you configure the HSRP virtual IP address as the host's 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.

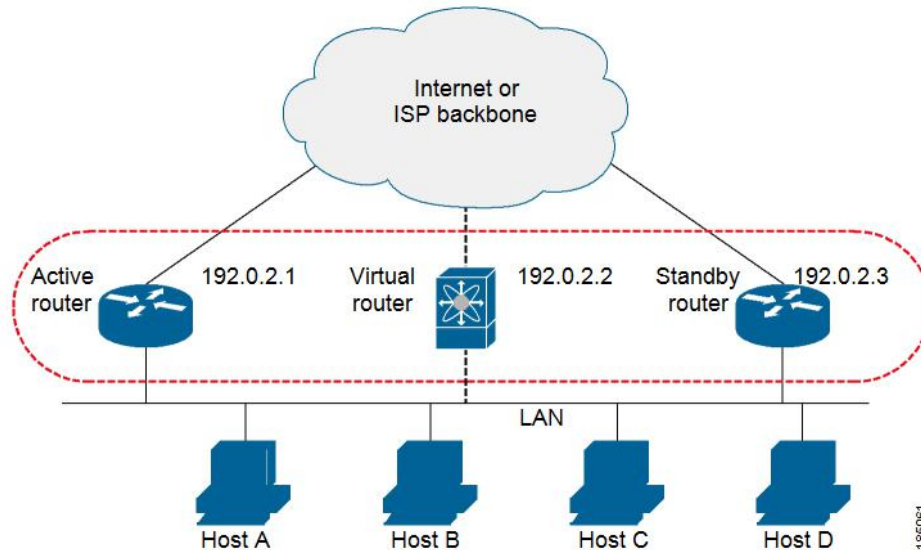
HSRP detects when the designated active router fails. At that point, a selected standby router assumes control of the virtual MAC and IP addresses of the HSRP group. HSRP also selects a new standby router at that time.

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.

Figure 17-1 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.

Figure 17-1 HSRP Topology with Two Enabled Routers

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.

17.1.2 HSRP Versions

Inspur INOS-CN 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-CN 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.

17.1.3 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.

17.1.4 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, which is 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 resign).

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 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.

Table 17-1 shows the MAC and IP addresses used for IPv6 neighbor discovery packets and HSRP packets.

Table 17-1 HSRP and IPv6 ND Addresses

Packet	MAC Source Address	IPv6 Source Address	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	Interface MAC	Interface IPv6 address	Virtual IPv6	HSRP virtual MAC address

(NA)	address		address	
Route advertisement (RA)	Interface MAC address	Virtual 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). Link-local addresses have no secondary virtual IP addresses.

For global unicast addresses, HSRP adds the virtual IPv6 address to the URIB and IPv6.

17.1.5 HSRP Subnet VIP

You can configure an HSRP subnet virtual IP (VIP) address in a different subnet than that of the interface IP address.

This feature enables you to conserve public IPv4 addresses by using a VIP as a public IP address and an interface IP as a private IP address. HSRP subnet VIPs are not needed for IPv6 addresses because a larger pool of IPv6 addresses is available and because routable IPv6 addresses can be configured on an SVI and used with regular HSRP.

This feature also enables periodic ARP synchronization to vPC peers and allows ARP to source with the VIP when an HSRP subnet VIP is configured for hosts in the VIP subnet.

17.1.6 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.

17.1.7 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.

17.1.8 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. Figure 17-2 shows an example of a load-sharing HSRP IPv4 configuration.

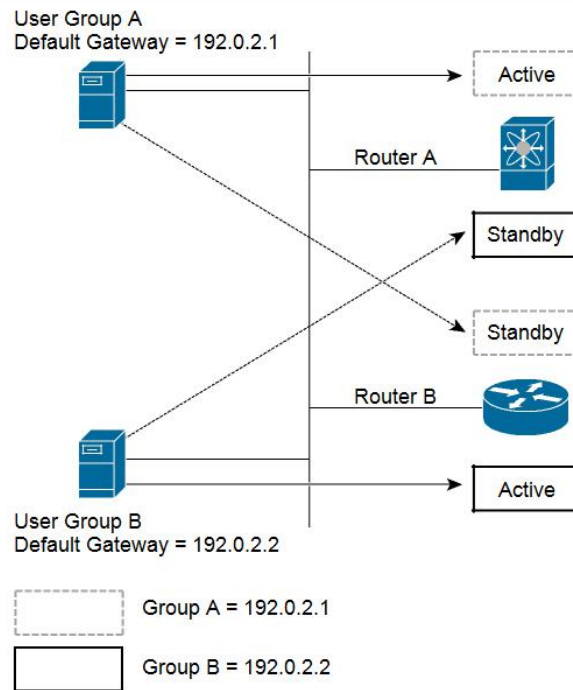
Figure 17-2 HSRP Load Sharing

Figure 17-2 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.

17.1.9 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-CN reduces the HSRP priority by the configured amount.

17.1.10 vPC and HSRP

HSRP interoperates with virtual port channels (vPCs). vPCs allow links that are physically connected to two different Inspur CN12900 Series devices to appear as a single port channel by a third device. See the *Inspur CN12900 Series INOS-CN 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 that use this source MAC address might 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 CN12900 Series INOS-CN Layer 2 Switching Configuration Guide* for more information on the vPC peer gateway.

17.1.11 BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4. 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information.

17.1.12 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-CN 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. HSRP supports extended nonstop forwarding (NSF) to temporarily extend these HSRP hold timers during a controlled switchover.

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. After the switchover, 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.

17.1.13 Virtualization Support

HSRP supports virtual routing and forwarding (VRF) instances.

17.2 Licensing Requirements for HSRP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	HSRP requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

17.3 Prerequisites for HSRP

- You must enable the HSRP feature in a device before you can configure and enable any HSRP groups.

17.4 Guidelines and Limitations for HSRP

HSRP has the following configuration guidelines and limitations:

- You must configure an IP address for the interface that you configure HSRP on and enable that interface before HSRP becomes active.
- Inspur CN12900 Series switches running in max-host routing mode do not support four-way HSRP.
- 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.
- 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.
- If HSRP IPv4 and IPv6 use the same virtual MAC address on an SVI, the HSRP state should be the same for both HSRP IPv4 and IPv6. The priority and preemption should be configured to result in the same state after failovers.
- Inspur INOS-CN removes all Layer 3 configurations on an interface when you change the interface VRF membership, port channel membership, or 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.
- You cannot use the HSRP MAC address burned-in option on a VLAN interface that is a vPC member.
- Inspur INOS-CN supports having the same HSRP groups on all nodes in a double-sided vPC.
- If you have not configured authentication, the **show hsrp** command displays the following string:

```
Authentication text "inspur"
```

The default behavior of HSRP is as defined in RFC 2281:

```
If no authentication data is configured, the RECOMMENDED
default value is 0x63 0x69 0x73 0x63 0x6F 0x00 0x00 0x00.
```

- The HSRP subnet VIP feature has the following guidelines and limitations:
 - This feature is supported for Inspur CN12900 Series switches.
 - This feature is supported only for IPv4 addresses and only in a vPC topology.
 - Primary or secondary VIPs can be subnet VIPs, but subnet VIPs must not overlap any interface subnet.
 - Regular host VIPs use a mask length of 0 or 32. If you specify a mask length for a subnet VIP, it must be greater than 0 and less than 32.
 - uRPF is not supported with this feature.
 - DHCP sourcing with VIPs is also not supported.
 - This feature does not support using a DHCP relay agent to relay DHCP packets with a VIP as the source.
 - VIP direct routes must be explicitly advertised to routing protocols using redistribute commands and route maps.
 - Supervisor-generated traffic (pings, trace routes, and so on) destined for VIP subnets will continue to source with SVI IP addresses and not with the VIP.
 - If the subnet VIP is configured with /32 as the length, you must use the **no** command with /32 to remove the IP address (for example, **no ip ip-address/32**).

17.5 Default Settings

Table 17-2 lists the default settings for HSRP parameters.

Table 17-2 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

17.6 Configuring HSRP

17.6.1 Enabling HSRP

You must globally enable HSRP before you can configure and enable any HSRP groups.

DETAILED STEPS

To enable the HSRP feature, use the following command in global configuration mode:

Command	Purpose
feature hsrp Example: switch(config)# feature hsrp	Enables HSRP.

To disable the HSRP feature and remove all associated configurations, use the following command in global configuration mode:

Command	Purpose
no feature hsrp Example: switch(config)# no feature hsrp	Disables HSRP for all groups.

17.6.2 Configuring the HSRP Version

You can configure the HSRP version. If you change the version for existing groups, Inspur INOS-CN reinitializes HSRP for those groups because the virtual MAC address changes. The HSRP version applies to all groups on the interface.

To configure the HSRP version, use the following command in interface configuration mode:

Command	Purpose
hsrp version {1 2} Example: switch(config-if)# hsrp version 2	Configures the HSRP version. Version 1 is the default.

17.6.3 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

Ensure that you have enabled the HSRP feature.

Inspur INOS-CN enables an HSRP group once you configure its virtual IP address. You should configure HSRP attributes such as authentication, timers, and priority before you enable the HSRP group.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ip** *ip-address*[/length]
4. **hsrp** *group-number* [ipv4]
5. **ip** [*ip-address* [secondary]]
6. **exit**
7. **no shutdown**
8. (Optional) **show hsrp** [group *group-number*] [ipv4]
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type number</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	ip <i>ip-address</i> [/length] Example: switch(config-if)# ip 192.0.2.2/8	Configures the IPv4 address of the interface. Note If you provide a length, the IP address should not be in a subnet of the interface. If you do not provide a length, the IP address should be in a subnet of the interface.
Step 4	hsrp <i>group-number</i> [ipv4] Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#	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	ip [<i>ip-address</i> [secondary]] Example: switch(config-if-hsrp)# ip 192.0.2.1	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	exit Example: switch(config-if-hsrp)# exit	Exits HSRP configuration mode.
Step 7	no shutdown Example: switch(config-if)# no shutdown	Enables the interface.
Step 8	show hsrp [group <i>group-number</i>] [ipv4] Example: switch(config-if)# show hsrp group 2	(Optional) Displays HSRP information.
Step 9	copy running-config startup-config Example:	(Optional) Copies the running configuration to the startup configuration.

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

This 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
```

17.6.4 Configuring an HSRP Group for IPv6

You can configure an HSRP group on an IPv6 interface and configure the 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.

BEFORE YOU BEGIN

You must enable HSRP.

Ensure that you have enabled HSRP version 2 on the interface that you want to configure an IPv6 HSRP group on.

Ensure that you have configured HSRP attributes such as authentication, timers, and priority before you enable the HSRP group.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *type number*
3. **ipv6 address** *ipv6-address/length*
4. **hsrp version 2**
5. **hsrp group-number** **ipv6**
6. **ip** *ipv6-address*
7. **ip autoconfig**
8. **exit**
9. **no shutdown**
10. (Optional) **show hsrp** [**group** *group-number*] [**ipv6**]
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>type number</i>	Enters interface configuration mode.

	Example: switch(config)# interface ethernet 3/2 switch(config-if)#	
Step 3	ipv6 address <i>ipv6-address/length</i> Example: switch(config-if)# ipv6 address 2001:0DB8::0001:0001/64	Configures the IPv6 address of the interface.
Step 4	hsrp version 2 Example: switch(config-if-hsrp)# hsrp version 2	Configures this group for HSRP version 2.
Step 5	hsrp group-number ipv6 Example: switch(config-if)# hsrp 10 ipv6 switch(config-if-hsrp)#	Creates an IPv6 HSRP group and enters HSRP configuration mode. The range for HSRP version 2 is from 0 to 4095. The default value is 0.
Step 6	ip <i>ipv6-address</i> Example: switch(config-if-hsrp)# ip 2001:DB8::1	Configures the virtual IPv6 address for the HSRP group and enables the group.
Step 7	ip autoconfig Example: 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 8	exit Example: switch(config-if-hsrp)# exit switch(config-if)#	Exits HSRP configuration mode.
Step 9	no shutdown Example: switch(config-if)# no shutdown	Enables the interface.
Step 10	show hsrp [group <i>group-number</i>] [ipv6] Example: switch(config-if)# show hsrp group 10	(Optional) Displays HSRP information.
Step 11	copy running-config startup-config Example: switch(config-if)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure an IPv6 HSRP group on Ethernet 3/2:

```
switch# configure terminal
switch(config)# interface ethernet 3/2
switch(config-if)# ipv6 address
2001:0DB8::0001:0001/64
switch(config-if-hsrp)# hsrp version 2
switch(config-if)# hsrp 2 ipv6
switch(config-if-hsrp)# ip
```

```
2001:DB8::1
switch(config-if-hsrp)# exit
switch(config-if)# no shutdown
switch(config-if)# copy running-config startup-config
```

17.6.5 Configuring the HSRP Virtual MAC Address

You can override the default virtual MAC address that HSRP derives from the configured group number.

To manually configure the virtual MAC address for an HSRP group, use the following command in hsrp configuration mode:

Command	Purpose
mac-address <i>string</i> Example: switch(config-if-hsrp)# mac-address 5000.1000.1060	Configures the virtual MAC address for an HSRP group. The string uses the standard MAC address format (xxxx.xxxx.xxxx).

To configure HSRP to use the burned-in MAC address of the interface for the virtual MAC address, use the following command in interface configuration mode:

Command	Purpose
hsrp use-bia [<i>scope interface</i>] Example: switch(config-if)# hsrp use-bia	Configures HSRP to use the burned-in MAC address of the interface for the HSRP virtual MAC address. You can optionally configure HSRP to use the burned-in MAC address for all groups on this interface by using the scope interface keyword.

17.6.6 Authenticating HSRP

You can configure HSRP to authenticate the protocol using cleartext or MD5 digest authentication. MD5 authentication uses a key chain (see the *Inspur CN12900 Series INOS-CN 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **hsrp** *group-number* [**ipv4** | **ipv6**]
4. **authentication text** *string* or **authentication md5** {**key-chain** *key-chain* | **key-string** {**0** | **7**} *text* [**timeout** *seconds*]}
5. (Optional) **show hsrp** [**group** *group-number*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	hsrp group-number [ipv4 ipv6] Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#	Creates an HSRP group and enters HSRP configuration mode.
Step 4	authentication text string Example: switch(config-if-hsrp)# authentication text mypassword	Configures cleartext authentication for HSRP on this interface.
	authentication md5 {key-chain key-chain key-string {0 7} text [timeout seconds]} Example: switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys	Configures MD5 authentication for HSRP on this interface. 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	show hsrp [group group-number] Example: switch(config-if-hsrp)# show hsrp group 2	(Optional) Displays HSRP information.
Step 6	copy running-config startup-config Example: switch(config-if-hsrp)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure MD5 authentication for HSRP on Ethernet 1/2 after creating the key chain:

```
switch# configure terminal
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 2013 23:59:59 Sep 12
2013
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
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 2013 23:59:59 Dec 12
2013
switch(config-keychain-key) send-lifetime 00:00:00 Sep 12 2013 23:59:59 Nov 12 2013
switch(config-keychain-key)# interface ethernet 1/2
```

```
switch(config-if)# hsrp 2
switch(config-if-hsrp)# authentication md5 key-chain hsrp-keys
switch(config-if-hsrp)# copy running-config startup-config
```

17.6.7 Authenticating HSRP

You can configure HSRP to authenticate the protocol using cleartext or MD5 digest authentication. MD5 authentication uses a key chain (see the *Inspur CN12900 Series INOS-CN 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.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **hsrp** *group-number [ipv4 | ipv6]*
4. **authentication text** *string* or **authentication md5** {*key-chain key-chain | key-string {0 | 7} text [timeout seconds]*}
5. (Optional) **show hsrp** [*group group-number*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#	Enters interface configuration mode.
Step 3	hsrp <i>group-number [ipv4 ipv6]</i> Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#	Creates an HSRP group and enters HSRP configuration mode.
Step 4	authentication text <i>string</i> Example: switch(config-if-hsrp)# authentication text mypassword	Configures cleartext authentication for HSRP on this interface.
	authentication md5 { <i>key-chain key-chain key-string {0 7} text [timeout seconds]</i> }	Configures MD5 authentication for HSRP on this interface. 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.

	Command	Purpose
	show hsrp [<i>group group-number</i>] Example: <pre>switch(config-if-hsrp)# show hsrp group 2</pre>	(Optional) Displays HSRP information.
Step 6	copy running-config startup-config Example: <pre>switch(config-if-hsrp)# copy running-config startup-config</pre>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure MD5 authentication for HSRP on Ethernet 1/2 after creating the key chain:

```
switch# configure terminal
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 2013 23:59:59 Sep 12 2013
switch(config-keychain-key) send-lifetime 00:00:00 Jun 01 2013 23:59:59 Aug 12 2013
switch(config-keychain-key) key 1
```

17.6.8 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 an HSRP group 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. **configure terminal**
2. **track object-id interface interface-type slot/port {line-protocol | ip routing | ipv6 routing}**
3. **track object-id {ip | ipv6} route ip-prefix/length reachability**
4. **exit**
5. **interface interface-type slot/port**
6. **hsrp group-number [ipv4 | ipv6]**
7. **priority [value]**
8. **track object-id [decrement value]**
9. **preempt [delay [minimum seconds] [reload seconds] [sync seconds]]**
10. (Optional) **show hsrp interface interface-type slot/port**
11. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.

	Command	Purpose
	<pre>track <i>object-id</i> interface <i>interface-type</i> <i>slot/port</i> {line-protocol ip routing ipv6 routing} Example: switch(config)# track 1 interface ethernet 2/2 line-protocol switch(config-track)#</pre>	<p>Configures the interface that the track object tracks. Changes in the state of the interface affect the track object status as follows:</p> <ul style="list-style-type: none"> You configure the interface and corresponding object number that you use with the track command in global configuration mode. The line-protocol keyword tracks whether the interface is up. The ip routing or ipv6 routing keyword also checks that IP routing is enabled on the interface and an IP address is configured.
Step 3	<pre>track <i>object-id</i> {ip ipv6} route <i>ip-prefix/length</i> reachability Example: switch(config-track)# track 2 ip route 192.0.2.0/8 reachability</pre>	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500.
Step 4	<pre>exit Example: switch(config-track)# exit switch(config)#</pre>	Exits track configuration mode.
Step 5	<pre>interface <i>interface-type</i> <i>slot/port</i> Example: switch(config)# interface ethernet 1/2 switch(config-if)#</pre>	Enters interface configuration mode.
Step 6	<pre>hsrp <i>group-number</i> [ipv4 ipv6] Example: switch(config-if)# hsrp 2 switch(config-if-hsrp)#</pre>	Creates an HSRP group and enters HSRP configuration mode.
Step 7	<pre>priority [<i>value</i>] Example: switch(config-if-hsrp)# priority 254</pre>	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 8	<pre>track <i>object-id</i> [decrement <i>value</i>] Example: switch(config-if-hsrp)# track 1 decrement 20</pre>	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 9	<pre>preempt [delay [<i>minimum seconds</i>] [reload <i>seconds</i>] [sync <i>seconds</i>]] Example: switch(config-if-hsrp)# preempt delay minimum 60</pre>	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. Optionally, a delay can be configured that delays the HSRP group preemption by the configured time. The range is from 0 to 3600 seconds.
Step 10	<pre>show hsrp interface <i>interface-type</i> <i>slot/port</i></pre>	(Optional) Displays HSRP information for an interface.

Command	Purpose
Example: <pre>switch(config-if-hsrp)# show hsrp interface ethernet 1/2</pre>	
Step 11 copy running-config startup-config Example: <pre>switch(config-if-hsrp)# copy running-config startup-config</pre>	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure HSRP object tracking on Ethernet interface 1/2:

```
switch# configure terminal
switch(config)# track 1 interface ethernet 2/2 line-protocol
switch(config-track)# track 2 ip route 192.0.2.0/8
reachability
switch(config-track)# exit
switch(config)# interface ethernet 1/2
switch(config-if)# hsrp 2
switch(config-if-hsrp)# priority 254
switch(config-if-hsrp)# track 1 decrement 20
switch(config-if-hsrp)# preempt delay minimum
60
switch(config-if-hsrp)# copy running-config startup-config
```

17.6.9 Configuring the HSRP Priority

You can configure the priority of an HSRP group. 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.

To configure the HSRP priority, use the following command in the HSRP group configuration mode:

Command	Purpose
<pre>priority level [forwarding-threshold lower lower-value upper upper-value]</pre> Example: <pre>switch(config-if-hsrp)# priority 60 forwarding-threshold lower 40 upper 50</pre>	Sets the priority level used to select the active router in an HSRP group. The level range is from 0 to 255. The default is 100. Optionally, this command 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.

17.6.10 Customizing HSRP

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:

Command	Purpose
name <i>string</i> Example: <pre>switch(config-if-hsrp)# name HSRP-1</pre>	Specifies the IP redundancy name for an HSRP group. The <i>string</i> is from 1 to 255 characters. The default string has the following format: <i>hsrp-interface short-name group-id</i> . For example, hsrp-Eth2/1-1.
preempt [delay [minimum <i>seconds</i>] reload <i>seconds</i>] [sync <i>seconds</i>] Example: <pre>switch(config-if-hsrp)# preempt delay minimum 60</pre>	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. Optionally, a delay can be configured that delays the HSRP group preemption by the configured time. The range is from 0 to 3600 seconds.
timers [msec] <i>hellotime</i> [msec] <i>holdtime</i> Example: <pre>switch(config-if-hsrp)# timers 5 18</pre>	Configures the hello and hold time for this HSRP member as follows: <ul style="list-style-type: none"> <i>hellotime</i>—The interval between successive hello packets sent. The range is from 1 to 254 seconds. <i>holdtime</i>—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: <ul style="list-style-type: none"> <i>hellotime</i>—The interval between successive hello packets sent. The range is from 255 to 999 milliseconds. <i>holdtime</i>—The interval before the information in the hello packet is considered invalid. The range is from 750 to 3000 milliseconds.

To customize HSRP, use the following commands in interface configuration mode:

Command	Purpose
hsrp delay minimum <i>seconds</i> Example: <pre>switch(config-if)# hsrp delay minimum 30</pre>	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.
hsrp delay reload <i>seconds</i> Example: <pre>switch(config-if)# hsrp delay reload 30</pre>	Specifies the minimum amount of time that HSRP waits after a reload and before participating in the group. The range is from 0 to 10000 seconds. The default is 0.

17.6.11 Configuring Extended Hold Timers for HSRP

You can configure HSRP to use extended hold timers to support extended NSF during a controlled (graceful) switchover. You should configure extended hold timers on all HSRP routers.

To configure HSRP extended hold timers, use the following command in global configuration mode:

Command	Purpose
hsrp timers extended-hold [<i>timer</i>] Example: switch(config)# hsrp timers extended-hold	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.

Use the **show hsrp** command or the **show running-config hsrp** command to display the extended hold time.

17.7 Verifying the HSRP Configuration

To display HSRP configuration information, perform one of the following tasks:

Command	Purpose
show hsrp [<i>group group-number</i>]	Displays the HSRP status for all groups or one group.
show hsrp [<i>group group-number</i>] [<i>interface interface-type slot/port</i>] [active] [all] [init] [learn] [listen] [speak] [standby] [detail]	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 [<i>group group-number</i>] [<i>interface interface-type slot/port</i>] [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 delay [<i>interface interface-type slot/port</i>]	Displays the HSRP delay value for all interfaces or one interface.
show hsrp [<i>interface interface-type slot/port</i>]	Displays the HSRP status for an interface.
show ip local-pt	Displays whether the netstack has programmed a subnet route for the VIP subnet.

17.8 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 Oct 01 2018 23:59:59 Sep 12 2013
    send-lifetime 00:00:00 Oct 01 2018 23:59:59 Aug 12 2013
  key 1
    key-string 7 uaeqdyito
    accept-lifetime 00:00:00 Oct 01 2018 23:59:59 Nov 12 2013
    send-lifetime 00:00:00 Oct 01 2018 23:59:59 Nov 12 2013
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

```

This example shows how to configure an HSRP subnet VIP address, which is configured in a different subnet than that of the interface IP address.

```

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

```

This example shows a VIP mismatch error when a VIP without a subnet is configured in a different subnet than the interface IP address.

```

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: VIP subnet mismatch with interface IP!

```

This example shows a VIP mismatch error when the HSRP subnet VIP address is configured in the same subnet as the interface IP address.

```

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: Subnet VIP cannot be in same subnet as interface IP!

```

Related Topic	Document Title
Configuring the Virtual Router Redundancy	Chapter 18, "Configuring VRRP"

Protocol	
Configuring high availability	<i>Inspur CN12900 Series INOS-CN High Availability and Redundancy Guide</i>

CHAPTER 18 Configuring VRRP

This chapter describes how to configure the Virtual Router Redundancy Protocol (VRRP) on the Inspur INOS-CN device.

18.1 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.

18.1.1 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:

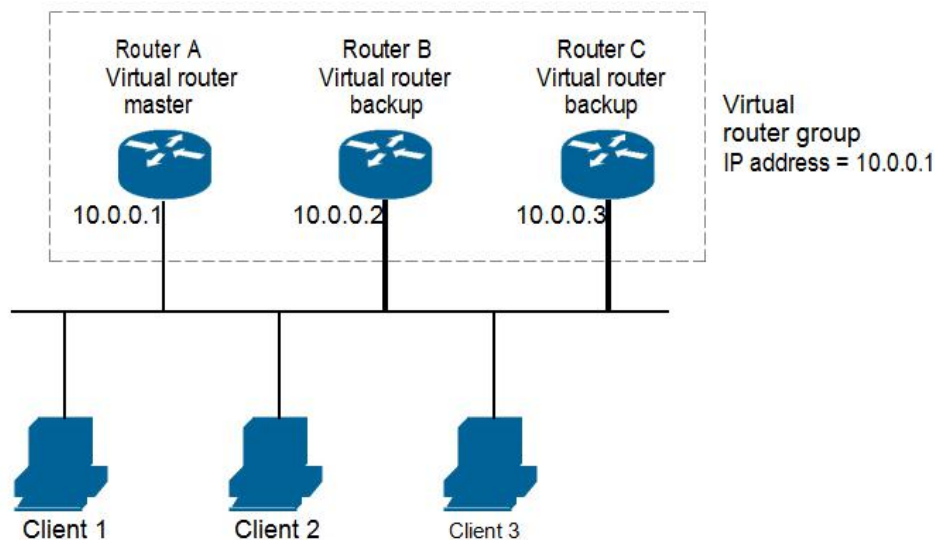
- Proxy ARP—The client uses the Address Resolution Protocol (ARP) to get the destination it wants to reach, and a router responds to the ARP request with its own MAC address.
- Routing protocol—The client listens to dynamic routing protocol updates (for example, from Routing Information Protocol [RIP]) 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 18-1 shows a basic VLAN topology. In this example, Routers A, B, and C form a VRRP group. The IP address of the group is the same address that was configured for the Ethernet interface of Router A (10.0.0.1).

Figure 18-1 Basic VRRP Topology

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.

18.1.2 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 multiple 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.

18.1.3 Multiple VRRP Groups

You can configure multiple VRRP groups on a physical interface. For the number of supported VRRP groups, see

the *Inspur CN12900 Series INOS-CN Verified Scalability Guide*.

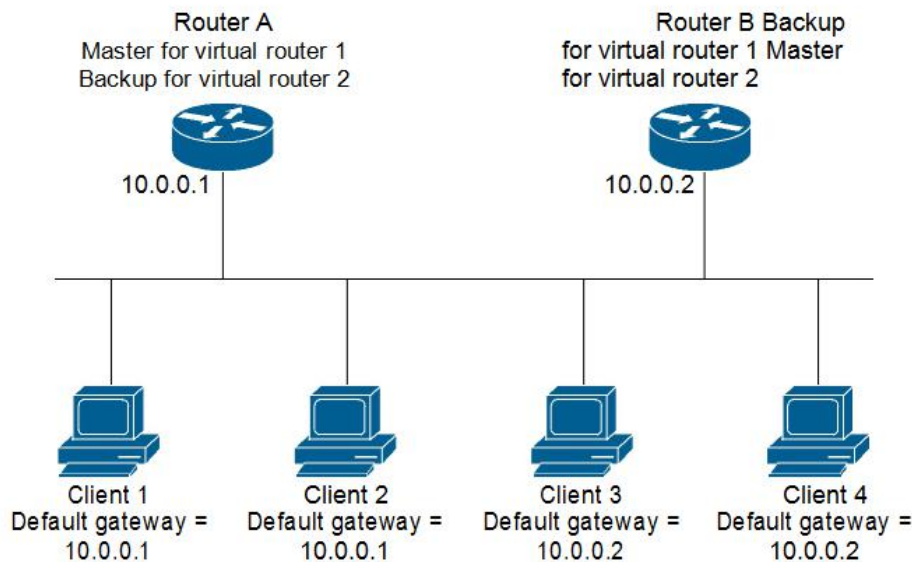
The number of VRRP groups that a router interface can support depends on the following factors:

- 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 18-2 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.

Figure 18-2 Load Sharing and Redundancy VRRP Topology



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.

18.1.4 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.

The 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.

18.1.5 vPC and VRRP

VRRP interoperates with virtual port channels (vPCs). vPCs allow links that are physically connected to two different Inspur CN12900 Series devices to appear as a single port channel by a third device. See the *Inspur CN12900 Series INOS-CN 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. See the “Configuring VRRP Priority”.

18.1.6 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-CN encapsulates the VRRP advertisements in IP packets and sends them to the IP multicast address assigned to the VRRP group. Inspur INOS-CN sends the advertisements once every second by default, but you can configure a different advertisement interval.

18.1.7 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.

18.1.8 VRRP Tracking

VRRP supports the following 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. See Chapter 19, “Configuring Object Tracking” for more information on object tracking.

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 might 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. See the “Configuring VRRP Interface State Tracking” section for more information.

18.1.9 BFD

This feature supports bidirectional forwarding detection (BFD) for IPv4. 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 CN12900 Series INOS-CN Interfaces*

Configuration Guide for more information.

18.2 Information About 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 information 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.

18.2.1 VRRPv3 Benefits

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

18.2.2 VRRPv3 Object Tracking

Beginning with Inspur INOS-CN Release 9.2(2), VRRPv3 supports object tracking, which tracks the state of a configured object and uses that state to determine the priority of the VRRPv3 router in a VRRPv3 group. See the “Configuring Object Tracking” chapter for more information on object tracking.

If the tracked object goes down, VRRPv3 decrements the priority by the configured value. The default value is 10. If the same tracked object goes down again, no action is taken. When the tracked object comes up, VRRPv3 increments the priority by the configured value.

18.3 High Availability

VRRP supports high availability through stateful restarts and stateful switchovers. A stateful restart occurs when the VRRP process fails and is restarted. A stateful switchover occurs when the active supervisor switches to the standby supervisor. Inspur INOS-CN applies the run-time configuration after the switchover.

18.4 Virtualization Support

VRRP supports virtual routing and forwarding (VRF) instances.

18.5 Licensing Requirements for VRRP

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	VRRP requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

18.6 Guidelines and Limitations for VRRP

VRRP has the following configuration guidelines and limitations:

- 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 that you configure VRRP on and enable that interface before VRRP becomes active.
- Inspur INOS-CN 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 reenble 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.

18.7 Guidelines and Limitations for VRRPv3

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 (BVI), Gigabit Ethernet interfaces, 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.
- Unlike VRRPv2, VRRPv3 does not support bidirectional forwarding for faster failure detection.
- Unlike VRRPv2, VRRPv3 does not support native interface tracking.

- You must create the object before configuring object tracking.
- The following guidelines and limitations apply to VRRPv3 object tracking:
 - Beginning with Inspur INOS-CN Release 9.2(2), all Inspur CN12900 Series switches support VRRPv3 object tracking.
 - We recommend that you do not use VRRPv3 object tracking in a vPC domain.

18.8 Default Settings for VRRP Parameters

Table 18-1 lists the default settings for VRRP parameters.

Table 18-1 Default VRRP Parameters

Parameters	Default
VRRP	Disabled
Advertisement interval	1 seconds
Authentication	No authentication
Preemption	Enabled
Priority	100

18.9 Default Settings for VRRPv3 Parameters

Table 18-1 lists the default settings for VRRPv3 parameters.

Table 18-2 Default VRRPv3 Parameters

Parameters	Default
VRRPv3	Disabled
VRRS	Disabled
VRRPv3 secondary address matching	Enabled
Priority of a VRRPv3 group	100
VRRPv3 advertisement timer	1000 milliseconds

18.10 Configuring VRRP

18.10.1 Enabling the VRRP Feature

You must globally enable the VRRP feature before you can configure and enable any VRRP groups.

To enable the VRRP feature, use the following command in global configuration mode:

Command	Purpose
<pre>feature vrrp</pre> <p>Example:</p> <pre>switch(config)# feature vrrp</pre>	Enables VRRP.

To disable the VRRP feature and remove all associated configurations, use the following command in global

configuration mode:

Command	Purpose
no feature vrrp Example: switch(config)# no feature vrrp	Disables the VRRP feature.

18.10.2 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-CN 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

Ensure that you configure an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **address** *ip-address* [**secondary**]
5. **no shutdown**
6. (Optional) **show vrrp**
7. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	vrrp <i>number</i> Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group. The range is from 1 to 255.
Step 4	address <i>ip-address</i> [secondary] Example: switch(config-if-vrrp)# address 192.0.2.8	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.

	Command	Purpose
		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	no shutdown Example: <pre>switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#</pre>	Enables the VRRP group. Disabled by default.
Step 6	show vrrp Example: <pre>switch(config-if-vrrp)# show vrrp</pre>	(Optional) Displays VRRP information.
Step 7	copy running-config startup-config Example: <pre>switch(config-if-vrrp)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

18.10.3 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

You must enable VRRP.

Ensure that you have configured an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **priority** *level* [**forwarding-threshold** **lower** *lower-value* **upper** *upper-value*]
6. **no shutdown**
7. (Optional) **show vrrp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal</pre>	Enters global configuration mode.

	Command	Purpose
	<code>switch(config)#</code>	
Step 2	interface <i>interface-type slot/port</i> Example: <code>switch(config)# interface ethernet 2/1</code> <code>switch(config-if)#</code>	Enters interface configuration mode.
Step 3	vrrp <i>number</i> Example: <code>switch(config-if)# vrrp 250</code> <code>switch(config-if-vrrp)#</code>	Creates a virtual router group.
Step 4	shutdown Example: <code>switch(config-if-vrrp)# shutdown</code> <code>switch(config-if-vrrp)#</code>	Disables the VRRP group. Disabled by default.
Step 5	priority <i>level</i> [forwarding-threshold <i>lower lower-value</i> upper upper-value] Example: <code>switch(config-if-vrrp)# priority 60</code> <code>forwarding-threshold lower 40 upper 50</code>	Sets the priority level used to select the active router in an VRRP group. The <i>level</i> 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 <i>lower-value</i> range is from 1 to 255. The default is 1. The <i>upper-value</i> range is from 1 to 255. The default is 255.
Step 6	no shutdown Example: <code>switch(config-if-vrrp)# no shutdown</code> <code>switch(config-if-vrrp)#</code>	Enables the VRRP group. Disabled by default.
Step 7	show vrrp Example: <code>switch(config-if-vrrp)# show vrrp</code>	(Optional) Displays a summary of VRRP information.
Step 8	copy running-config startup-config Example: <code>switch(config-if-vrrp)# copy running-config startup-config</code>	(Optional) Saves this configuration change.

18.10.4 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.
- Ensure that you have enabled VRRP.
- Ensure that you have configured an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp number**
4. **shutdown**
5. **authentication text password**
6. **no shutdown**
7. (Optional) **show vrrp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	vrrp number Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 4	shutdown Example: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	Disables the VRRP group. Disabled by default.
Step 5	authentication text password Example: switch(config-if-vrrp)# authentication text aPassword	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	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 7	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 8	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

18.10.5 Configuring Time Intervals for Advertisement Packets

You can configure the time intervals for advertisement packets.

BEFORE YOU BEGIN

You must enable VRRP.

Ensure that you have configured an IP address on the interface .

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **advertisement-interval** *seconds*
6. **no shutdown**
7. (Optional) **show vrrp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	vrrp <i>number</i> Example: switch(config-if)# vrrp 250 switch(config-if-vrrp)#	Creates a virtual router group.
Step 4	shutdown Example: switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#	Disables the VRRP group. Disabled by default.
Step 5	advertisement-interval <i>seconds</i> Example: switch(config-if-vrrp)# advertisement-interval 15	Sets the interval time in seconds between sending advertisement frames. The range is from 1 to 255. The default is 1 second.
Step 6	no shutdown Example: switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#	Enables the VRRP group. Disabled by default.
Step 7	show vrrp Example:	(Optional) Displays a summary of VRRP information.

	Command	Purpose
	<code>switch(config-if-vrrp)# show vrrp</code>	
Step 8	copy running-config startup-config Example: <code>switch(config-if-vrrp)# copy running-config startup-config</code>	(Optional) Saves this configuration change.

18.10.6 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

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **no preempt**
6. **no shutdown**
7. (Optional) **show vrrp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <code>switch# configure terminal</code> <code>switch(config)#</code>	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: <code>switch(config)# interface ethernet 2/1</code> <code>switch(config-if)#</code>	Enters interface configuration mode.
Step 3	vrrp <i>number</i> Example: <code>switch(config-if)# vrrp 250</code> <code>switch(config-if-vrrp)#</code>	Creates a virtual router group.
Step 4	no shutdown Example: <code>switch(config-if-vrrp)# no shutdown</code>	Enables the VRRP group. Disabled by default.
Step 5	no preempt	Disables the preempt option and allows the master to remain

	Command	Purpose
	Example: switch(config-if-vrrp)# no preempt	when a higher-priority backup appears.
Step 6	no shutdown Example: switch(config-if-vrrp)# no shutdown	Enables the VRRP group. Disabled by default.
Step 7	show vrrp Example: switch(config-if-vrrp)# show vrrp	(Optional) Displays a summary of VRRP information.
Step 8	copy running-config startup-config Example: switch(config-if-vrrp)# copy running-config startup-config	(Optional) Saves this configuration change.

18.10.7 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-CN assigns the tracking priority value to the virtual router. When the tracked interface comes up and an IP address is configured on this interface,

BEFORE YOU BEGIN

You must enable VRRP.

Ensure that you have configured an IP address on the interface.

Ensure that you have enabled the virtual router.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrp** *number*
4. **shutdown**
5. **track interface** *type number priority value*
6. **no shutdown**
7. (Optional) **show vrrp**
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example:	Enters interface configuration mode.

	Command	Purpose
	<pre>switch(config)# interface ethernet 2/1 switch(config-if)#</pre>	
Step 3	<p>vrrp number</p> <p>Example:</p> <pre>switch(config-if)# vrrp 250 switch(config-if-vrrp)#</pre>	Creates a virtual router group.
Step 4	<p>shutdown</p> <p>Example:</p> <pre>switch(config-if-vrrp)# shutdown switch(config-if-vrrp)#</pre>	Disables the VRRP group. Disabled by default.
Step 5	<p>track interface type number priority value</p> <p>Example:</p> <pre>switch(config-if-vrrp)# track interface ethernet 2/10 priority 254</pre>	Enables interface priority tracking for a VRRP group. The priority range is from 1 to 254.
Step 6	<p>no shutdown</p> <p>Example:</p> <pre>switch(config-if-vrrp)# no shutdown switch(config-if-vrrp)#</pre>	Enables the VRRP group. Disabled by default.
Step 7	<p>show vrrp</p> <p>Example:</p> <pre>switch(config-if-vrrp)# show vrrp</pre>	(Optional) Displays a summary of VRRP information.
Step 8	<p>copy running-config startup-config</p> <p>Example:</p> <pre>switch(config-if-vrrp)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

18.10.8 Configuring VRRP Object Tracking

You can track an IPv4 object using VRRP.

BEFORE YOU BEGIN

Make sure that VRRP is enabled.

Configure object tracking using the commands in the “Configuring Object Tracking” chapter.

SUMMARY STEPS

1. **configure terminal**
2. **interface interface-type slot/port**
3. **vrrp number address-family ipv4**
4. **track object-number decrement number**
5. (Optional) **show running-config vrrp**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: <pre>switch# configure terminal switch(config)#</pre>	Enters global configuration mode.
Step 2	interface interface-type slot/port Example: <pre>switch(config)# interface ethernet 2/1 switch(config-if)#</pre>	Specifies an interfaces and enters interface configuration mode.
Step 3	vrrp number address-family ipv4 Example: <pre>switch(config-if)# vrrp 5 address-family ipv4 switch(config-if-vrrp-group)#</pre>	Creates a VRRP group for IPV4 and enters VRRP group configuration mode. The range is from 1 to 255.
Step 4	track object-number decrement number Example: <pre>switch(config-if-vrrp-group)# track 1 decrement 3</pre>	Configures the process to track the state of the IPv4 object using the VRRP group. VRRP on the interface registers with the tracking process to be informed of any changes to the object in the VRRP group. If the object state on the interface goes down, the priority of the VRRP group is reduced by the decrement number specified.
Step 5	show running-config vrrp Example: <pre>switch(config-if-vrrp-group)# show running- config vrrp</pre>	(Optional) Displays the running configuration for VRRP.
Step 6	copy running-config startup-config Example: <pre>switch(config-if-vrrp-group)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

18.11 Configuring VRRPv3

18.11.1 Enabling VRRPv3 and VRRS

You must globally enable VRRPv3 before you can configure and enable any VRRPv3 groups.

SUMMARY STEPS

1. **configure terminal**
2. **[no] feature vrrpv3**
3. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	[no] feature vrrpv3 Example: switch(config)# feature vrrpv3	Enables VRRP version 3 and Virtual Router Redundancy Service (VRRS). The no form of this command disables VRRPv3 and VRRS. 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.
Step 3	copy running-config startup-config Example: switch(config)# copy running-config startup-config	(Optional) Saves this configuration change.

18.11.2 Creating VRRPv3 Groups

You can create a VRRPv3 group, assign the virtual IP address, and enable the group.

BEFORE YOU BEGIN

Make sure that VRRPv3 is enabled.

Make sure that you have configured an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet slot/port**
3. **vrrpv3 number address-family [ipv4 | ipv6]**
4. (Optional) **address ip-address [primary | secondary]**
5. (Optional) **description description**
6. (Optional) **match-address**
7. (Optional) **preempt [delay minimum seconds]**
8. (Optional) **priority level**
9. (Optional) **timers advertise interval**
10. (Optional) **vrrp2**
11. (Optional) **vrrs leader vrrs-leader-name**
12. (Optional) **shutdown**
13. (Optional) **show fhrp [interface-type interface-number] [verbose]**
14. (Optional) **show vrrpv3 interface-type interface-number**
15. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal	Enters global configuration mode.

	Command	Purpose
	switch(config)#	
Step 2	interface ethernet slot/port Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	vrrpv3 number address-family [ipv4 ipv6] Example: switch(config-if)# vrrpv3 5 address-family ipv4 switch(config-if-vrrpv3-group)#	Creates a VRRPv3 group and enters VRRPv3 group configuration mode. The range is from 1 to 255.
Step 4	address ip-address [primary secondary] Example: switch(config-if-vrrpv3-group)# address 100.0.1.10 primary	(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	description description Example: switch(config-if-vrrpv3-group)# description group3	(Optional) Specifies a description for the VRRPv3 group. You can enter up to 80 alphanumeric characters.
Step 6	match-address Example: switch(config-if-vrrpv3-group)# match-address	(Optional) Matches the secondary address in the advertisement packet against the configured address.
Step 7	preempt [delay minimum seconds] Example: switch(config-if-vrrpv3-group)# preempt delay minimum 30	(Optional) Enables preemption of a lower priority master switch with an optional delay. The range is from 0 to 3600.
Step 8	priority level Example: switch(config-if-vrrpv3-group)# priority 3	(Optional) Specifies the priority of the VRRPv3 group. The range is from 1 to 254.
Step 9	timers advertise interval Example: switch(config-if-vrrpv3-group)# timers advertise 1000	(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	vrrp2 Example: 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

	Command	Purpose
		perform an upgrade.
Step 11	vrrs leader <i>vrrs-leader-name</i> Example: <pre>switch(config-if-vrrpv3-group)# vrrs leader leader1</pre>	(Optional) Specifies a leader's name to be registered with VRRS.
Step 12	shutdown Example: <pre>switch(config-if-vrrpv3-group)# shutdown</pre>	(Optional) Disables the VRRP configuration for the VRRPv3 group.
Step 13	show fhrp [<i>interface-type</i> <i>interface-number</i>] [verbose] Example: <pre>switch(config-if-vrrpv3-group)# show fhrp ethernet 2/1 verbose</pre>	(Optional) Displays First Hop Redundancy Protocol (FHRP) information. Use the verbose keyword to view detailed information.
Step 14	show vrrpv3 <i>interface-type</i> <i>interface-number</i> Example: <pre>switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 2/1</pre>	(Optional) Displays the VRRPv3 configuration information for the specified interface.
Step 15	copy running-config startup-config Example: <pre>switch(config-if-vrrpv3-group)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

18.11.3 Configuring VRRPv3 Control Groups

You can configure VRRPv3 control groups.

BEFORE YOU BEGIN

Make sure that VRRPv3 is enabled.

Make sure that you have configured an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip address** *ip-address mask* [**secondary**]
4. **vrrpv3 number address-family** [**ipv4** | **ipv6**]
5. (Optional) **address** *ip-address* [**primary** | **secondary**]
6. (Optional) **shutdown**
7. (Optional) **show fhrp** [*interface-type interface-number*] [**verbose**]
8. (Optional) **show vrrpv3** *interface-type interface-number*
9. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet slot/port Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	ip address ip-address mask [secondary] Example: switch(config-if)# ip address 209.165.200.230 255.255.255.224	Configures the IP address on the interface. You can use the secondary keyword to configure additional IP addresses on the interface.
Step 4	vrrpv3 number address-family [ipv4 ipv6] Example: switch(config-if)# vrrpv3 5 address-family ipv4 switch(config-if-vrrpv3-group)#	Creates a VRRPv3 group and enters VRRPv3 group configuration mode. The range is from 1 to 255.
Step 5	address ip-address [primary secondary] Example: switch(config-if-vrrpv3-group)# address 209.165.200.227 primary	(Optional) Specifies a primary or secondary IPv4 or IPv6 address for the VRRPv3 group.
Step 6	shutdown Example: switch(config-if-vrrpv3-group)# shutdown	(Optional) Disables the VRRP configuration for the VRRPv3 group.
Step 7	show fhrp [interface-type interface-number] [verbose] Example: switch(config-if-vrrpv3-group)# show fhrp ethernet 2/1 verbose	(Optional) Displays First Hop Redundancy Protocol (FHRP) information. Use the verbose keyword to view detailed information.
Step 8	show vrrpv3 interface-type interface-number Example: switch(config-if-vrrpv3-group)# show vrrpv3 ethernet 2/1	(Optional) Displays the VRRPv3 configuration information for the specified interface.
Step 9	copy running-config startup-config Example: switch(config-if-vrrpv3-group)# copy running-config startup-config	(Optional) Saves this configuration change.

18.11.4 Configuring VRRPv3 Object Tracking

You can track an IPv4 or IPv6 object using VRRPv3.

BEFORE YOU BEGIN

Make sure that VRRPv3 is enabled.

Configure object tracking using the commands in the “Configuring Object Tracking” chapter.

SUMMARY STEPS

1. **configure terminal**
2. **interface** *interface-type slot/port*
3. **vrrpv3** *number address-family [ipv4 | ipv6]*
4. **object-track** *object-number decrement number*
5. (Optional) **show running-config vrrpv3**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface <i>interface-type slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Specifies an interfaces and enters interface configuration mode.
Step 3	vrrpv3 <i>number address-family [ipv4 ipv6]</i> Example: switch(config-if)# vrrpv3 5 address-family ipv6 switch(config-if-vrrpv3-group)#	Creates a VRRPv3 group for IPV4 or IPV6 and enters VRRPv3 group configuration mode. The range is from 1 to 255.
Step 4	object-track <i>object-number decrement number</i> Example: switch(config-if-vrrpv3-group)# object- track 1 decrement 2	Configures the process to track the state of the IPv4 or IPv6 object using the VRRPv3 group. VRRPv3 on the interface registers with the tracking process to be informed of any changes to the object in the VRRPv3 group. If the object state on the interface goes down, the priority of the VRRPv3 group is reduced by the decrement number specified.
Step 5	show running-config vrrpv3 Example: switch(config-if-vrrpv3-group)# show running-config vrrpv3	(Optional) Displays the running configuration for VRRPv3.
Step 6	copy running-config startup-config Example: switch(config-if-vrrpv3-group)# copy running-config startup-config	(Optional) Saves this configuration change.

18.11.5 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

Make sure that VRRPv3 is enabled.

Make sure that you have configured an IP address on the interface.

SUMMARY STEPS

1. **configure terminal**
2. **interface ethernet** *slot/port*
3. **ip address** *ip-address mask* [**secondary**]
4. **vrrs pathway** *vrrs-tag*
5. **mac address** {*mac-address* | **inherit**}
6. **address** *ip-address*
7. (Optional) **show vrrs pathway** *interface-type interface-number*
8. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	interface ethernet <i>slot/port</i> Example: switch(config)# interface ethernet 2/1 switch(config-if)#	Enters interface configuration mode.
Step 3	ip address <i>ip-address mask</i> [secondary] Example: switch(config-if)# ip address 209.165.200.230 255.255.255.224	Configures the IP address on the interface. You can use the secondary keyword to configure additional IP addresses on the interface.
Step 4	vrrs pathway <i>vrrs-tag</i> Example: switch(config-if)# vrrs pathway path1 switch(config-if-vrrs-pw)#	Defines the VRRS pathway for a VRRS group and enters VRRS pathway configuration mode. The <i>vrrs-tag</i> argument specifies the name of the VRRS tag that is being associated with the pathway.
Step 5	mac address { <i>mac-address</i> 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	address <i>ip-address</i> Example: switch(config-if-vrrs-pw)# address 209.165.201.10	Defines the virtual IPv4 or IPv6 address for a pathway. A VRRPv3 group is capable of controlling more than one pathway.
Step 7	show vrrs pathway <i>interface-type</i>	(Optional) Displays the VRRS pathway information

Command	Purpose
<i>interface-number</i> Example: <pre>switch(config-if-vrrs-pw)# show vrrs pathway ethernet 1/2</pre>	for different pathway states, such as active, inactive, and not ready.
Step 8 copy running-config startup-config Example: <pre>switch(config-if-vrrs-pw)# copy running-config startup-config</pre>	(Optional) Saves this configuration change.

18.12 Verifying the VRRP Configuration

To display VRRP configuration information, perform one of the following tasks:

Command	Purpose
show vrrp	Displays the VRRP status for all groups.
show fhrp [<i>interface-type</i> <i>interface-number</i>] [verbose]	Displays First Hop Redundancy Protocol (FHRP) information.
show interface <i>interface-type</i>	Displays the virtual router configuration for an interface.

18.13 Verifying the VRRPv3 Configuration

To display VRRPv3 configuration information, perform one of the following tasks:

Command	Purpose
show vrrpv3 [all brief detail]	Displays the VRRPv3 configuration information.
show vrrpv3 <i>interface-type</i> <i>interface-number</i>	Displays the VRRPv3 configuration information for a specific interface.
show vrrs client [<i>client-name</i>]	Displays the VRRS client information.
show vrrs pathway [<i>interface-type</i> <i>interface-number</i>]	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 [<i>tag-name</i>]	Displays the VRRS tag information.

18.14 Monitoring and Clearing VRRP Statistics

To display VRRP statistics, use the following commands:

Command	Purpose
show vrrp statistics	Displays the VRRP statistics.

Use the **clear vrrp statistics** command to clear all the VRRP statistics for all interfaces on the device.

18.15 Monitoring and Clearing VRRPv3 Statistics

To display VRRPv3 statistics, use the following commands:

Command	Purpose
<code>show vrrpv3 statistics</code>	Displays the VRRPv3 statistics.

Use the `clear vrrpv3 statistics` command to clear the VRRPv3 statistics for all interfaces on the device.

18.16 Configuration Examples 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 of 1 second.
 - Preemption is disabled.

Router A

```
interface ethernet 1/0
 ip address 10.1.0.2/16
 no shutdown
 vrrp 1
  priority 120
  authentication text inspur
  advertisement-interval 3
  address 10.1.0.10
 no shutdown
 vrrp 5
  priority 100
  advertisement-interval
  30 address 10.1.0.50
 no shutdown
 vrrp 100
  no preempt
  address 10.1.0.100
 no shutdown
```

Router B

```
interface ethernet 1/0
 ip address 10.2.0.1/24
 no shutdown
 vrrp 1
```

```

priority 100
authentication text inspur
advertisement-interval 3
address 10.2.0.10
no shutdown

vrrp 5
priority 200
advertisement-interval
30 address 10.2.0.50
no shutdown
vrrp 100
no preempt
address 10.2.0.100
no shutdown

```

18.17 Configuration Examples for VRRPv3

This example shows how to enable VRRPv3 and create and customize a VRRPv3 group:

```

switch# configure terminal
switch(config)# feature vrrpv3
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
switch(config-if-vrrp3-group)# show fhrp ethernet 4/6
verbose
switch(config-if-vrrp3-group)# show vrrpv3 ethernet 4/6

```

This 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(config-if-vrrp3-group)# address 209.165.200.227
primary
switch(config-if-vrrp3-group)# vrrs leader leader1
switch(config-if-vrrp3-group)# shutdown
switch(config-if-vrrp3-group)# show fhrp ethernet 1/2 verbose
switch(config-if-vrrp3-group)# show vrrpv3 ethernet 1/2

```

This example shows how to configure object tracking for VRRPv3:

```

track 1 interface Ethernet1/12 ip routing
track 2 interface Ethernet1/12 ipv6 routing
track 3 interface Ethernet1/12 line-protocol
track 4 interface Ethernet1/12.1 ip routing
track 5 interface Ethernet1/12.1 ipv6 routing
track 6 interface Ethernet1/12.1 line-protocol
track 7 interface loopback1 ip routing
track 8 interface loopback1 ipv6 routing
track 9 interface loopback1 line-protocol
track 10 interface port-channel1 ip routing
track 11 interface port-channel1 ipv6 routing
track 12 interface port-channel1 line-protocol

```

```

track 13 ip route 170.10.10.10/24 reachability
track 14 ip route 180.10.10.0/24 reachability hmm
track 15 ipv6 route 2001::170:10:10:10/128 reachability
track 16 list boolean and
    object 1
    object 2
interface Vlan10
 vrrpv3 10 address-family ipv4
   timers advertise 100
   priority 200
   object-track 1 decrement 2
   object-track 2 decrement 2
   object-track 3 decrement 2
   object-track 4 decrement 2
   object-track 5 decrement 2
   object-track 6 decrement 2
   object-track 7 decrement 2
   object-track 8 decrement 2
   object-track 9 decrement 2
   object-track 10 decrement 2
   address 10.10.10.3 primary
interface Vlan10
 vrrpv3 10 address-family ipv6
   timers advertise 100
   priority 200
   object-track 1 decrement 4
   object-track 2 decrement 4
   object-track 3 decrement 4
   object-track 4 decrement 4
   object-track 5 decrement 4
   object-track 6 decrement 4
   object-track 7 decrement 4
   object-track 8 decrement 4
   object-track 9 decrement 4
   object-track 10 decrement 4
   object-track 11 decrement 4
   object-track 12 decrement 4
   object-track 13 decrement 4
   address fe80::10:10:10:3 primary
show running-config vrrpv3

```

This 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 inherit
switch(config-if-vrrs-pw)# address 209.165.201.10
switch(config-if-vrrs-pw)# show vrrs pathway ethernet 1/2

```

18.18 Additional References

18.18.1 Related Documents

Related Topic	Document Title
Configuring high availability	<i>Inspur CN12900 Series INOS-CN High Availability and Redundancy Guide</i>

CHAPTER 19 Configuring Object Tracking

This chapter describes how to configure object tracking on the Inspur INOS-CN device.

19.1 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 tracked object's state changes. This feature allows you to increase the availability of the network and shorten recovery time if an object state goes down.

19.1.1 Object Tracking Overview

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)
- Hot Standby Redundancy Protocol (HSRP)
- Virtual port channel (vPC)
- Virtual Router Redundancy Protocol (VRRP) and VRRPv3

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-CN 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.

19.1.2 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 CN12900 Series INOS-CN Interfaces Configuration Guide* for more information on vPCs.

19.1.3 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-CN 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.

19.1.4 Virtualization Support

Object tracking supports virtual routing and forwarding (VRF) instances. By default, Inspur INOS-CN 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.

19.2 Licensing Requirements for Object Tracking

The following table shows the licensing requirements for this feature:

Product	License Requirement
Inspur INOS-CN	Object tracking requires no license. Any feature not included in a license package is bundled with the INOS-CN image and is provided at no extra charge to you.

19.3 Guidelines and Limitations

Object tracking has the following configuration guidelines and limitations:

- Supports Ethernet, subinterfaces, port channels, loopback interfaces, and VLAN interfaces.
- Supports one tracked object per HSRP group.
- VRRP and VRRPv3 support object tracking.

19.4 Default Settings

Table 19-1 lists the default settings for object tracking parameters.

Table 19-1 Default Object Tracking Parameters

Parameters	Default
Tracked object VRF	Member of default VRF

19.5 Configuring Object Tracking

19.5.1 Configuring Object Tracking for an Interface

You can configure Inspur INOS-CN to track the line protocol or IPv4 or IPv6 routing state of an interface.

SUMMARY STEPS

1. **configure terminal**
2. **track *object-id* interface *interface-type* *number* {ip | ipv6 | routing | line-protocol}**
3. (Optional) **show track [*object-id*]**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>object-id</i> interface <i>interface-type</i> <i>number</i> {ip routing ipv6 routing line-protocol} Example: switch(config)# track 1 interface ethernet 1/2 line-protocol switch(config-track)#	Creates a tracked object for an interface and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500.
Step 3	show track [<i>object-id</i>] Example: switch(config-track)# show track 1	(Optional) Displays object tracking information.
Step 4	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This 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-track)# copy running-config startup-config
```

This 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-track)# copy running-config startup-config
```

This 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-track)# copy running-config startup-config
```

19.5.2 Deleting a Tracked Object

You can delete a tracked object.

SUMMARY STEPS

1. **configure terminal**

2. no track *object-id*

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	no track <i>object-id</i> Example: switch(config)# no track 1 switch(config-track)#	Deletes a tracked object for an interface. The <i>object-id</i> range is from 1 to 500.

This example shows how to delete a tracked object:

```
switch# configure terminal
switch(config)# no track 1
switch(config-track)# copy running-config startup-config
```

19.5.3 Configuring Object Tracking for Route Reachability

You can configure Inspur INOS-CN to track the existence and reachability of an IP route or IPv6 route.

SUMMARY STEPS

1. **configure terminal**
2. **track *object-id* {ip | ipv6} route *prefix/length* reachability**
3. (Optional) **show track [*object-id*]**
4. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>object-id</i> {ip ipv6} route <i>prefix/length</i> reachability Example: switch(config)# track 3 ipv6 route 2::5/64 reachability switch(config-track)#	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500. The prefix format for IPv4 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	show track [<i>object-id</i>] Example: switch(config-track)# show track 1	(Optional) Displays object tracking information.
Step 4	copy running-config startup-config Example:	(Optional) Copies the running configuration to the startup configuration.

Command	Purpose
switch(config-track)# copy running-config startup-config	

This 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
```

This 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
```

19.5.4 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.

SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list boolean** {and | or}
3. **object** *object-id* [not]
4. (Optional) **show track** [*object-id*]
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>track-number</i> list boolean {and or} Example: switch(config)# track 1 list boolean and switch(config-track)#	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: <ul style="list-style-type: none"> • 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.

	Command	Purpose
		The <i>track-number</i> range is from 1 to 500.
Step 3	object <i>object-id</i> [not] Example: switch(config-track)# object 10	Adds a tracked object to the track list. The <i>object-id</i> 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	show track [<i>object-id</i>] Example: switch(config-track)# show track	(Optional) Displays object tracking information.
Step 5	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This 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
```

19.5.5 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.

SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list threshold percentage**
3. **threshold percentage up** *up-value* **down** *down-value*
4. **object** *object-id*
5. (Optional) **show track** [*object-id*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track <i>track-number</i> list threshold percentage Example:	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.

	Command	Purpose
	switch(config)# track 1 list threshold percentage switch(config-track)#	The <i>track-number</i> range is from 1 to 500.
Step 3	threshold percentage up <i>up-value</i> down <i>down-value</i> Example: switch(config-track)# threshold percentage up 70 down 30	Configures the threshold percentage for the tracked list. The range from 0 to 100 percent.
Step 4	object <i>object-id</i> Example: switch(config-track)# object 10	Adds a tracked object to the track list. The <i>object-id</i> range is from 1 to 500.
Step 5	show track [<i>object-id</i>] Example: switch(config-track)# show track	(Optional) Displays object tracking information.
Step 6	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a track list with an up threshold of 70 percent and a down threshold of 30 percent:

```
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
```

19.5.6 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.

SUMMARY STEPS

1. **configure terminal**
2. **track** *track-number* **list threshold weight**
3. **threshold weight up** *up-value* **down** *down-value*
4. **object** *object-id* **weight** *value*
5. (Optional) **show track** [*object-id*]
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track track-number list threshold weight Example: switch(config)# track 1 list threshold weight switch(config-track)#	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 <i>track-number</i> range is from 1 to 500.
Step 3	threshold weight up up-value down down-value Example: switch(config-track)# threshold weight up 30 down 10	Configures the threshold weight for the tracked list. The range from 1 to 255.
Step 4	object object-id weight value Example: switch(config-track)# object 10 weight 15	Adds a tracked object to the track list. The <i>object-id</i> range is from 1 to 500. The <i>value</i> range is from 1 to 255. The default weight value is 10.
Step 5	show track [object-id] Example: switch(config-track)# show track	(Optional) Displays object tracking information.
Step 6	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This example shows how to configure a track list with an up weight threshold of 30 and a down threshold of 10:

```
switch# configure terminal
switch(config)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
```

In this example, the track list is up if object 10 and object 20 are up, and the track list goes to the down state if all three objects are down.

19.5.7 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-CN 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 an 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 follows:

- 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.

SUMMARY STEPS

1. **configure terminal**
2. **track object-id {parameters}**
3. **track track-number list {parameters}**
4. **delay {up up-time [down down-time] | down down-time [up up-time]}**
5. (Optional) **show track [object-id]**
6. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track object-id {parameters} Example: switch(config)# track 2 ip route 192.0.2.0/8 reachability switch(config-track)#	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500. The prefix format for IPv4 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	track track-number list {parameters} Example: switch(config)# track 1 list threshold weight switch(config-track)#	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 <i>track-number</i> range is from 1 to 500.
Step 4	delay {up up-time [down down-time] down down-time [up up-time]} Example: switch(config-track)# delay up 20 down 30	Configures the object delay timers. The range is from 0 to 180 seconds.
Step 5	show track [object-id] Example: switch(config-track)# show track 3	(Optional) Displays object tracking information.
Step 6	copy running-config startup-config Example:	(Optional) Copies the running configuration to the startup configuration.

Command	Purpose
switch(config-track)# copy running-config startup-config	

This 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
```

This 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)# track 1 list threshold weight
switch(config-track)# threshold weight up 30 down 10
switch(config-track)# object 10 weight 15
switch(config-track)# object 20 weight 15
switch(config-track)# object 30
switch(config-track)# delay up 20 down 30
```

This 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
```

19.5.8 Configuring Object Tracking for a Nondefault VRF

You can configure Inspur INOS-CN to track an object in a specific VRF.

BEFORE YOU BEGIN

Ensure that nondefault VRFs are created first.

SUMMARY STEPS

1. **configure terminal**
2. **track object-id {ip | ipv6} route prefix/length reachability**
3. **vrf member vrf-name**
4. (Optional) **show track [object-id]**
5. (Optional) **copy running-config startup-config**

DETAILED STEPS

	Command	Purpose
Step 1	configure terminal Example: switch# configure terminal switch(config)#	Enters global configuration mode.
Step 2	track object-id {ip ipv6} route prefix/length reachability Example: switch(config)# track 3 ipv6 route 1::2/64 reachability switch(config-track)#	Creates a tracked object for a route and enters tracking configuration mode. The <i>object-id</i> range is from 1 to 500. The prefix format for IPv4 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	vrf member vrf-name Example: switch(config-track)# vrf member Red	Configures the VRF to use for tracking the configured object.
Step 4	show track [object-id] Example: switch(config-track)# show track 3	(Optional) Displays object tracking information.
Step 5	copy running-config startup-config Example: switch(config-track)# copy running-config startup-config	(Optional) Copies the running configuration to the startup configuration.

This 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
```

This 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
```

This example shows how to modify tracked object 2 to use VRF Blue instead of VRF Red to look up reachability information for this object:

```
switch# configure terminal
switch(config)# track 2
switch(config-track)# vrf member Blue
switch(config-track)# copy running-config startup-config
```

19.6 Verifying the Object Tracking Configuration

To display object tracking configuration information, perform one of the following tasks:

Command	Purpose
<code>show track [object-id] [brief]</code>	Displays the object tracking information for one or more objects.
<code>show track [object-id] interface [brief]</code>	Displays the interface-based object tracking information.
<code>show track [object-id] {ip ipv6} route [brief]</code>	Displays the IPv4 or IPv6 route-based object tracking information.

19.7 Configuration Examples 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
switch(config-track)# copy running-config startup-config
```

19.8 Related Topics

See the following topics for information related to object tracking:

- Chapter 13, “Configuring Layer 3 Virtualization”
- Chapter 17, “Configuring HSRP”

19.8.1 Related Documents

Related Topic	Document Title
Configuring the Embedded Event Manager	<i>Inspur CN12900 Series INOS-CN System Management Configuration Guide</i>

APPENDIX A IETF RFCs Supported by Inspur INOS-CN Unicast Features

This appendix lists the IETF RFCs supported in Inspur INOS-CN. For information on each RFC, see www.ietf.org.

BGP RFCs

RFCs	Title
RFC 1997	<i>BGP Communities Attribute</i>
RFC 2385	<i>Protection of BGP Sessions via the TCP MD5 Signature Option</i>
RFC 2439	<i>BGP Route Flap Damping</i>
RFC 2519	<i>A Framework for Inter-Domain Route Aggregation</i>
RFC 2545	<i>Use of BGP-4 Multiprotocol Extensions for IPv6 Inter-Domain Routing</i>
RFC 2858	<i>Multiprotocol Extensions for BGP-4</i>
RFC 2918	<i>Route Refresh Capability for BGP-4</i>
RFC 3065	<i>Autonomous System Confederations for BGP</i>
RFC 3392	<i>Capabilities Advertisement with BGP-4</i>
RFC 4271	<i>A Border Gateway Protocol 4 (BGP-4)</i>
RFC 4273	<i>Definitions of Managed Objects for BGP-4</i>
RFC 4456	<i>BGP Route Reflection: An Alternative to Full Mesh Internal BGP (IBGP)</i>
RFC 4486	<i>Subcodes for BGP Cease Notification Message</i>
RFC 4724	<i>Graceful Restart Mechanism for BGP</i>
RFC 4893	<i>BGP Support for Four-octet AS Number Space</i>
RFC 5004	<i>Avoid BGP Best Path Transitions from One External to Another</i>
RFC 5396 ¹	<i>Textual Representation of Autonomous System (AS) Numbers</i>
RFC 5549	<i>Advertising IPv4 Network Layer Reachability Information with an IPv6 Next Hop</i>
RFC 5668	<i>4-Octet AS Specific BGP Extended Community</i>
draft-ietf-idr-add-paths-08.txt	<i>Advertisement of Multiple Paths in BGP</i>
draft-ietf-idr-bgp4-mib-15.txt	<i>BGP4-MIB</i>
draft-kato-bgp-ipv6-link-local-00.txt	<i>BGP4+ Peering Using IPv6 Link-local Address</i>

1. RFC 5396 is partially supported. The asplain and asdot notations are supported, but the asdot+ notation is not.

First-Hop Redundancy Protocol RFCs

RFCs	Title
RFC 2281	<i>Hot Standby Redundancy Protocol</i>
RFC 3768	<i>Virtual Router Redundancy Protocol</i>

IP Services RFCs

RFCs	Title
RFC 786	<i>UDP</i>
RFC 791	<i>IP</i>
RFC 792	<i>ICMP</i>
RFC 793	<i>TCP</i>
RFC 826	<i>ARP</i>
RFC 1027	<i>Proxy ARP</i>
RFC 1591	<i>DNS Client</i>
RFC 1812	<i>IPv4 routers</i>
RFC 4022	<i>TCP-MIB</i>
RFC 4292	<i>IP-FORWARDING-TABLE-MIB</i>
RFC 4293	<i>IP-MIB</i>

IPv6 RFCs

RFCs	Title
RFC 1981	<i>Path MTU Discovery for IP version 6</i>
RFC 2373	<i>IP Version 6 Addressing Architecture</i>
RFC 2374	<i>An Aggregatable Global Unicast Address Format</i>
RFC 2460	<i>Internet Protocol, Version 6 (IPv6) Specification</i>
RFC 2461	<i>Neighbor Discovery for IP Version 6 (IPv6)</i>
RFC 2462	<i>IPv6 Stateless Address Autoconfiguration</i>
RFC 2464	<i>Transmission of IPv6 Packets over Ethernet Networks</i>
RFC 3152	<i>Delegation of IP6.ARPA</i>
RFC 3162	<i>RADIUS and IPv6</i>
RFC 3513	<i>Internet Protocol Version 6 (IPv6) Addressing Architecture</i>
RFC 3596	<i>DNS Extensions to Support IP version 6</i>
RFC 4193	<i>Unique Local IPv6 Unicast Addresses</i>

IS-IS RFCs

RFCs	Title
RFC 1142	<i>OSI 10589 Intermediate system to intermediate system intra-</i>

RFCs	Title
	<i>domain routing exchange protocol</i>
RFC 1195	<i>Use of OSI IS-IS for routing in TCP/IP and dual environment</i>
RFC 2763, RFC 5301	<i>Dynamic Hostname Exchange Mechanism for IS-IS</i>
RFC 2966, RFC 5302	<i>Domain-wide Prefix Distribution with Two-Level IS-IS</i>
RFC 2972	<i>IS-IS Mesh Groups</i>
RFC 3277	<i>IS-IS Transient Blackhole Avoidance</i>
RFC 3373, RFC 5303	<i>Three-Way Handshake for IS-IS Point-to-Point Adjacencies</i>
RFC 3567, RFC 5304	<i>IS-IS Cryptographic Authentication</i>
RFC 3784, RFC 5305	<i>IS-IS Extensions for Traffic Engineering</i>
RFC 3847, RFC 5306	<i>Restart Signaling for IS-IS</i>
draft-ietf-isis-igp-p2p-over-lan-06.txt	<i>Internet Draft Point-to-point operation over LAN in link-state routing protocols</i>

OSPF RFCs

RFCs	Title
RFC 2328	<i>OSPF Version 2</i>
RFC 2370	<i>The OSPF Opaque LSA Option</i>
RFC 2740	<i>OSPF for IPv6</i>
RFC 3101	<i>The OSPF Not-So-Stubby Area (NSSA) Option</i>
RFC 3137	<i>OSPF Stub Router Advertisement</i>
RFC 3623	<i>Graceful OSPF Restart</i>
RFC 4552 (partial support)	<i>Authentication/Confidentiality for OSPFv3</i>
RFC 5709	<i>OSPFv2 HMAC-SHA Cryptographic Authentication</i>
draft-ietf-ospf-ospfv3-graceful-restart-04.txt	<i>OSPFv3 Graceful Restart</i>

Per-Hop Behavior RFCs

RFCs	Title
RFC 2597	<i>Assured Forwarding PHB Group</i>
RFC 3246	<i>An Expedited Forwarding PHB</i>

RIP RFCs

RFCs	Title
RFC 2082	<i>RIP-2 MD5 Authentication</i>
RFC 2453	<i>RIP Version 2</i>