IPv6 Configuration

Table of Contents

Chapter 1 **IPv6 Protocol's Configuration**

1.1 IPv6 Protocol's Configuration

The configuration of the IPv6 address of the router only takes effect on the VLAN interface, not on the physical interface.

The IPv6 protocol is disabled in default state. If the IPv6 protocol need be used on a VLAN interface, this protocol should be first enabled in VLAN interface configuration mode. To enable the IPv6 protocol, users have to set the IPv6 address. If on a VLAN interface at least one IPv6 address is set, the VLAN interface can handle the IPv6 packets and communicates with other IPv6 devices.

To enable the IPv6 protocol, users should finish the following task:

- Setting at least one IPv6 address in VLAN interface configuration mode

1.2 Enabling IPv6

1.2.1 Setting the IPv6 Address

The IPv6 address is used to determine the destination address to which the IPv6 packets can be sent. There are three kinds of IPv6 addresses.

Kind	Referred Format	Remarks
Unicast address	2001:0:0:0:0DB8:800:200C:417A/64	**2001:0:0:0:0DB8:800:200C:417A** stands for a unicast address, while **64** stands for the length of the prefix of this address.
Multicast address	FF01:0:0:0:0:0:0:101	All multicast addresses begin with FF.
Any address	2002:0:0:0:0DB8:800:200C:417A/64	The format of this address is the same as that of the unicast address. Different VLAN interfaces can be set to have the same address, no matter it is a unicast/broadcast/multicast address.

In order to enable IPv6, users must set a unicast address in VLAN interface configuration mode. The set unicast address must be one or multiple addresses of the following type:

- IPv6 link-local address
- Global IPv6 address

To set an IPv6 link-local address in VLAN interface configuration mode, run the following commands.

Command	Purpose
ipv6 enable	Sets a link-local address automatically.
ipv6 address fe80::x link-local	Sets a link-local address manually.

Note:

- The link-local address must begin with fe80. The default length of the prefix is 64 bit. At manual settings only the values at the last 64 bits can be designated.
- On a VLAN interface can only one link-local address be set.
- After IPv6 is enabled through the configuration of the link-local address, IPv6 only takes effect on the local link.

To set a global IPv6 address in VLAN interface configuration mode, run the following commands.

Command	Purpose
ipv6 address autoconfig	Sets a global address automatically.
ipv6 address [ipv6-address/prefix-length | prefix-name sub-bits/prefix-length] | [eui-64]	Sets a global address.
ipv6 address X:X:X:X::X/<0-128> anycast	Sets an address of unicast/broadcast/multicast.

Note:

- When IPv6 is enabled through the configuration of a global address, all interconnected IPv6 device can be handled by IPv6.
- If a link-local address has not been set before the configuration of the global address, the system will set a link-local address automatically.

Chapter 2 **Setting the IPv6 Services**

2.1 Setting the IPv6 Services

After IPv6 is enabled, all services provided by IPv6 can be set. The configurable IPv6 service is shown below:

- Managing the IPv6 link

2.1.1 Managing the IPv6 Link

IPv6 provides a series of services to control and manage the IPv6 link. This series of services includes:

- Setting the transmission frequency of the ICMPv6 packet
- Setting the source IPv6 route
- Setting the MTU of IPv6
- Setting IPv6 redirection
- Setting IPv6 destination unreachablity
- Setting IPv6 ACL
- Setting IPv6 Hop-Limit

1. Setting the transmission frequency of the ICMPv6 packet

If you want to limit the transmission frequency of the ICMPv6 packet, run the command in the following table. If the ICMPv6 transmission frequency is larger than the set value, the transmission frequency will be limited.

The default transmission frequency is 1000us.

Command	Purpose
ipv6 icmp6-ratelimit *ratelimit*	Sets the transmission frequency of the ICMPv6 packet.

2. Setting the source IPv6 route

IPv6 allows a host to designate the route of an IPv6 network, that is, the source route. The host can realize the source route through using the routing header in the IPv6 packets. The router can forward packets through the routing header, or desert this kind of packets considering security.

The router supports the source route by default. If the source route is closed, users can run the following command in global configuration mode to open the source route.

Command	Purpose
ipv6 source-route	Allows the source IPv6 route.

3. Setting the MTU of IPv6

All interfaces have a default IPv6 MTU. If the length of an IPv6 packet exceeds MTU, the router will fragment this IPv6 packet.

To set IPv6 MTU on a specific interface, run the following command in interface configuration mode:

Command	Purpose
ipv6 mtu bytes	Sets IPv6 MTU on an interface.

4. Setting IPv6 redirection

IPv6 redirection is opened by default. However, if a hot standby router protocol is configured on an interface, IPv6 redirection is automatically closed. If the hot standby router protocol is canceled, this function will not automatically opened.

To open IPv6 redirection, run the following command:

Command	Purpose
ipv6 redirects	Allows IPv6 to transmit the redirection packets.

5. Setting IPv6 destination unreachablity

In many cases, the system will automatically transmit the destination-unreachable packets. Users can close this function. If this function is closed, the system will not transmit the ICMP unreachable packets.

To enable this function, run the following command:

Command	Purpose
ipv6 unreachables	Allows IPv6 to transmit the destination unreachable packets.

6. Setting IPv6 ACL

Users can use ACL to control the reception and transmission of packets on a VLAN interface. If you introduce ACL on a VLAN interface in global configuration mode and designate the filtration's direction, the IPv6 packets will be filtered on this VLAN interface.

To filter the IPv6 packets, run the following command in interface configuration mode.

Command	Purpose
ipv6 traffic-filter *WORD* { in | out }	Filters the IPv6 packets in the reception or transmission direction (in: receive; out: transmit) on

	a VLAN interface.

7. Setting IPv6 Hop-Limit

Users can designate a router to transmit the value of the hop-limit field in the packets (except those forwarded packets). All those packets that this router transmits out, if the upper-level application does not apparently designate a hop-limit value, use the set value of hop-limit. At the same time, the value of the hop-limit field is added to the RA packets that this router transmits.

The default hop-limit value is 64. If you want to change this value, you can run the following command in interface configuration mode.

Command	Purpose
ipv6 cur-hoplimit *value*	Designates a router to transmit the hop-limit field of the packets.

Chapter 3 **Configuring RIPNG**

3.1 Overview

Routing Information Protocol of next generation (RIPng) is the RIP of version 6. In the equipment RIPng and RIP are two completely independent modules that are in charge of the learning and management of the routing information in version 6 and version 4 respectively.

RIPng is same to RIP in the internal working mechanism. RIPng switches the routing information through the UDP broadcast. In a router the update of the routing information is transmitted every 30 seconds. If a router has not received the routing update from its neighboring router in 180 seconds, the router will label this route unavailable in its routing table. And in the following 120 second this router will remote this route from its routing table.

RIPng can also be applied in small-scale networks. It uses the hop count to weigh the weights of different routes. This hop count means the number of routers that a packet has passed from a signal source to another signal source. The routing weight of the directly connected network is 0 and that of the unreachable network is 16. Since the route weight used by RIPng has a small range, it is unsuitable for the large-scale networks.

If a router has a default route, RIPng declares the route to the fake network 0::0/0. In fact, network **0::0/0** does not exist and it is just used to realize the default route in RIPng. If RIPng learns a default route or a router sets the default gateway and the default weight, the router will declare the default network.

RIPng sends the route update to the interface that is covered by instances. If an interface is not set to be an IPv6 interface, it will not be covered by an RIPng instance.

The RIPng protocol in our routers supports multiple instances. On an interface up to four instances can be set and one instance can cover up to 4 interfaces.

3.2 Setting RIPng Configuration Task List

Before setting RIPng, you have finished the following tasks. Among these tasks, you have to activate RIPng, but to other tasks, you can choose to do them according actual requirements.

- Allowing to set the unicast routing protocol
- Enabling RIPng
- Allowing the RIPng route to update the unicasting broadcast of a packet
- Applying the offset on the routing weight
- Filtering the received or transmitted routes
- Setting the management distance
- Adjusting the timer
- Summarizing the routes manually
- Maximum Number of Routes
- Activating or forbidding horizontal fragmentation
- Monitoring and maintaining RIPng

3.3 RIPng Configuration Tasks

3.3.1 Allowing to Set the Unicast Routing Protocol

To set the RIPng, you must first run the following command to allow to set the switch of a unicast route.

Command	Purpose
Ipv6 unicast-routing	Enables to set the unicast routing protocol on a device.

3.3.2 Enabling a RIPng Case

To enable the RIPng instance, run the following command in interface configuration mode:

Command	Purpose
ipv6 rip *instance-name* **enable**	Enables RIPng on an interface.

To enter the RIPng instance, run the following command in global configuration mode:

Command	Purpose
Ipv6 router rip *instance-name*	Enters the RIPng instance and its configuration mode.

Note: Users can enable a RIPng instance on an interface. If the RIPng instance does not exist, a RIPng instance will be generated. The system can directly enter the RIPng instance in global configuration mode and a RIPng instance will be generated if this RIPng instance does not exist.

Users can enable up to 4 RIPng instances on an interface and a RIPng instance can cover up to 4 interfaces.

3.3.3 Redistributing the Routes of an Unlocal Instance

RIPng can redistribute the routing information of an unlocal instance to the routing information database of the local instance, and then conducts route interaction with other devices through the routes in the routing database of this instance. To reach the aim above, run the following command in RIPng configuration mode:

Command	Purpose
Redistribute *protocol [instance-name | process-id]*	Redistributes static routes, other ospfv6 processes, and other RIPng instances.

3.3.4 Allowing the RIPng Route to Update the Unicasting Broadcast of a Packet

RIPng is generally a multicast protocol. To enable RIPng routing updates to reach the non-broadcast network, users must make configuration on a router to allow the switching of routing information. To reach the aim above, run the following command in RIPng configuration mode:

Command	Purpose
neighbor *ipv6-address*	Defines a neighboring router and switches the routing information with this neighboring router.

3.3.5 Applying the Offset on the Routing Weight

The offset list is used to add an offset for an incoming or outgoing route which RIPng learns. In this case, a local mechanism is provided to add the routing weight. Additionally, you can also use the access list or an interface to limit the offset list. To add the routing weight, run the following command in RIPng configuration mode:

Command	Purpose
offset {[interface-type number]|* } {**in**|**out**} *access-list-name* offset *value*	Adds an offset to a routing weight.

3.3.6 Filtering the Received or Transmitted Routes

Through settings the RIPng instance can filter the received or transmitted routes on the corresponding interface, in which flexible configuration policies can be flexibly realized. Run the following command in RIPng configuration mode:

Command	Purpose
filter *interface-type interface-number* **{in | out}** access-list | gateway | prefix-list	Filters the received or transmitted routing information.

3.3.7 Setting the Management Distance

Trough setting the management distance, you can change the credibility of the route of RIPng instance. In general, the bigger the value is, the more incredible the value is. To set the management distance, run the following command in RIPng configuration mode:

Command	Purpose
distance *weight* [*X:X:X:X::X/*<0-128> [Acc-list_name]	Sets the management distance of the RIPng instance's route.

3.3.8 Adjusting the Timer

The routing protocol needs several timers to judge the transmission frequency of routing updates and how long it takes for a route to become invalid. You can adjust these timers to make the performance of a routing protocol more suitable for the requirements of network interconnecting.

You can also adjust the routing protocols to speed up the convergence time of the IPv6 algorithm and make fast backup of the redundancy router, guaranteeing a maximum breakup for a terminal user when quick recovery is needed. To adjust the timer, run the following command in RIPng configuration mode:

Command	Purpose

timers holddown *value*	Means how long it takes for a route to be removed from the routing table.
timers garbage *value*	Means how long it takes for a route to be declared invalid.
timers update *value*	Means the transmission frequency of routing updates, whose unit is second.

3.3.9 Summarizing the Routes Manually

RIPng must summarize the routing information manually to reduce the number of the routes that interact with neighbors. To summarize the routing information, run the following command in the RIPng configuration mode:

Command	Purpose
aggregate-address *ipv6-prefix/prefixlen*	Summarizes the routing information.

3.3.10 Maximum Number of Routes

By default, the local RIPng routing table can contain up to 8192 routes. When the number of routes in the routing table exceeds the maximum number, the route will not be added to the routing table any more. Meanwhile, the user will be notified that the number of routes in the routing table has reached the maximum number. You can run the commands in the following table to configure the maximum number of the routes in the local RIPng routing table in router configuration mode.

Command	Purpose
max-routes *number*	Configures the maximum number of routes for the local RIPng routing table.
no max-routes	Resumes the default maximum number of the routes in the local RIP routing table.

3.3.11 Activating or Forbidding Horizontal Fragmentation

In normal cases, a router that connects the broadcast IPv6 network and uses the distance vector routing protocol takes the horizontal fragmentation to reduce the possibility of route loopback. The horizontal fragmentation blocks the routing information from being declared to the interface that receives this routing information. In this way the communication between multiple routers can be optimized, especially when the loopback is broken. However, this solution is not so good to those un-broadcast networks. In these networks, you have to forbid horizontal fragmentation.

To activate or disable horizontal fragmentation, run the following commands in VLAN configuration mode:

Command	Purpose
Ipv6 rip *instance-name* **split-horizon**	Activates horizontal fragmentation.
no ipv6 rip *instance-name* **split-horizon**	Forbids horizontal fragmentation.

By default, horizontal fragmentation is activated on those point-to-point interfaces and forbidden on those point-to-multipoint interfaces.

Note:

In normal cases, you are not recommended to change the default state unless you are sure that the routes can be correctly declared after the state of your application program is changed. If horizontal fragmentation is forbidden on a serial interface that connects a packet switching network, you have to disable horizontal fragmentation on routers of any related multicast group on a network.

3.3.12 Monitoring and Maintaining RIPng

Through monitoring and maintaining RIPng, you can get the statistic information of a network, including the parameters of RIPng, the network usage information and the real communication-tracing information. This kind of information can help users to judge the usage of network resources and solve network problems. From the statistics information, you can also know the reachablity of a network node.

To display all kinds of statistics information, run the following commands in EXEC mode:

Command	Purpose
show ipv6 rip *instance-name* summary	Displays the total routing information about a RIPng instance.
show ipv6 rip *instance-name* database	Displays all routes of a RIPng instance.
show ipv6 rip *instance-name* interface	Displays all interfaces that a RIPng instance covers.

To trace the information about the routing protocols, run the following commands in EXEC mode:

Command	Purpose
debug ipv6 rip *instance-name* database	Traces that a route of a RIPng instance is added to, removed from or changed in a routing table.
debug ipv6 rip *instance-name* event	Traces the abnormity that occurs in the running of a RIPng instance and the whole process of redistributing a RIPng instance.
debug ipv6 rip *instance-name* send	Traces the process that a RIPng instance transmits packets.
debug ipv6 rip *instance-name* recv	Traces the process that a RIPng instance receives packets.
debug ipv6 rip *instance-name* msg	Traces the important events that lead to the termination of the startup of a RIPng instance.
debug ipv6 rip *instance-name* all	Traces all the information about a RIPng instance.

3.4 RIPng Configuration Example

This section shows some RIPng configuration example:

Connect device A and device B directly and make the following settings:

Device A:

```
interface VLAN2
 no ip address
 no ip directed-broadcast
 ipv6 address 4444::4444/64
 ipv6 enable
 ipv6 rip dang2 enable
 ipv6 rip dang2 split-horizon
!
ipv6 router rip dang2
 redistribute static
 exit
!
!
```

Device B:

```
interface Ethernet1/1
 no ip address
 no ip directed-broadcast
 duplex half
 ipv6 address 4444::2222/64
 ipv6 enable
 ipv6 rip dang enable
 ipv6 rip dang split-horizon
!
ipv6 router rip dang
 redistribute static
 exit
!
```

In this way both device A and device B learns the static routing information from each other.

Chapter 4 **OSPFv3 Configuration**

4.1 Overview

OSPFv3 is an IGP routing protocol developed by the OSPF working group of IETF for the IPv6 network. OSPFv3 supports the IPv6 subnet, the mark of the external routing information and the packet's authentication.

OSPFv3 and OSPFv2 have a lot in common:

1. Both router ID and area ID are 32 bit.
2. The following are the same type of packets: Hello packets, DD packets, LSR packets, LSU packets and LSAck packets.

   Having the same neighbor discovery mechanism and the same neighborhood generation mechanism

   Having the same LSA expansion mechanism and the same LSA aging mechanism

The main differences of both OSPFv3 and OSPFv2 are shown below:

OSPFv3 is running on the basis of link, while OSPFv2 is running on the basis of network segment.

OSPFv3can run multiple instances on the same link.

OSPFv3 labels its neighbor through router ID, while OSPFv2 labels its neighbor through IP.

OSPFv3 defines 7 classes of LSAs.

The realization of the OSPFv3 functions on our routers complies with the requirements of OSPFv3. The following table shows some key functions in the realization of the OSPFv3 functions.

Key attributes	Description
Stub domain	Supports the stub domain.
Route forwarding	Means that routes that are learned or generated by any routing protocol can be forwarded to the domains of other routing protocols.
Parameters of a routing interface	The following are configurable interface parameters: output cost, retransmission interval, interface's transmission delay, router's priority, interface for judging the shutdown of a router, hello interval, and authentication key.
Virtual link	Supports the virtual link.

4.2 OSPFv3 Configuration Task List

OSPFv3 demands the switchover of routing data between in-domain router, ABR and ASBR. In order to simplify the settings, you can make related configuration to enable them to work under the default parameters without any authentication; if you want to change some parameters, you must guarantee that the parameters on all routers are identical.

To set OSPFv3, you must perform the following tasks. Except that the task of activating OSPFv3 is mandatory, other settings are optional.

- Enabling OSPFv3
- Setting the parameters of the OSPFv3 interface
- Setting OSPFv3 on different physical networks
- Setting the parameters of the OSPFv3 domain
- Setting the route summary in the OSPFv3 domain
- Setting the summary of the forwarded routes
- Generating a default route
- Choosing the route ID on the loopback interface
- Setting the timer of routing algorithm
- Monitoring and maintaining OSPFv3

4.3 OSPFv3 Configuration Tasks

4.3.1 Enabling OSPFv3

Before OSPFv3 is enabled, the function to forward the IPv6 packets must be enabled.

Run the following commands in global configuration mode:

Command	Purpose
ipv6 router ospf *process-id*	Activates OSPFv3 and enters the router configuration mode.
router-id *router-id*	Sets the router ID of a router on which OSPFv3 is running.

Run the following command in interface configuration mode:

Command	Purpose
ipv6 ospf *process-id* **area** *area-id* **[instance** *instance-id***]**	Enables OSPFv3 on an interface.

Note: If the OSPFv3 process is still not created before OSPFv3 is enabled on an interface, the OSPFv3 process will be automatically created.

4.3.2 Setting the Parameters of the OSPFv3 Interface

During OSPFv3 realization, related OSPFv3 parameters on an interface are allowed to be modified according to actual requirements. Of cause you have no need to change every parameter, but you have to make sure that some parameters are consistent on all routers in the connected networks.

Run the following commands in interface configuration mode to do relevant configurations:

Command	Purpose
ipv6 ospf cost *cost*	Sets the cost of the packet that is transmitted from the OSPFv6 interface.
ipv6 ospf retransmit-interval *seconds*	Sets the LSA retransmission interval between

	neighbors.
ipv6 ospf transmit-delay *seconds*	Sets the delay time for transmitting LSA on an OSPFv3 interface.
ipv6 ospf priority *number*	Sets a router to be the priority of the OSPFv3 DR router.
ipv6 ospf hello-interval *seconds*	Sets the interval for the OSPFv3 interface to transmit the Hello packets.
ipv6 ospf dead-interval *seconds*	Means that in a regulated interval if the OSPFv3 packets are not received from a neighboring router, this neighboring router is viewed to be shut down.

4.3.3 Setting OSPFv3 on Different Physical Networks

OSPFv3 divides physical network media into the following three kinds:

- Broadcast networks (Ethernet, Token Ring, FDDI)
- Non-broadcast and multi-access networks (SMDS, Frame Relay, X.25)
- Point-to-point networks (HDLC, PPP)

4.3.4 Setting the OSPF Network Type

No matter what physical media type the network is, you can configure your network to be a broadcast network, a non-broadcast network or a multi-access network. So you can set your network flexibly and your network can be set to be a non-broadcast and multi-access one, or a broadcast network such as the X.25, Frame Relay or SMDS network. Also the neighbor's settings will be simplified.

To set an un-broadcast and multi-access network is to suppose that every two routers have a virtual link or suppose a full-mesh network. It is unrealistic due to unbearable cost. But you set this network to be a point-to-multipoint one. Between those routers which are not adjacent the routing information can be switched through the virtual link.

The OSPFv3 point-to-multipoint interface can be set to be multipoint-to-point interface, through which multiple routes of a host can be established. The OSPFv3 point-to-multipoint network, comparing with the non-broadcast and multi-access network or the point-to-point network, has the following advantages:

- The point-to-multipoint network is easy to be set without generating DR.
- This kind of network do not require the full-mesh topology, so the construction cost is relatively low.
- This kind of networks are more reliable. Even if its virtual link fails, the connection can be maintained.

The network type of the routers is the broadcast type.

4.3.5 Setting the Parameters of the OSPFv3 Domain

The configurable domain parameters include: authentication, designating a stub area and specifying a weight for a default summary route. Its authentication is based on password protection.

The stub area means that external routes cannot be distributed to this area. Instead, ABR generates a default external route that enters the stub area, enabling the stub

area to communicate with external networks of an autonomous area. In order to make use of the attributes supported by the OSPF stub, the default route must be used in the stub area. To further reduce LSAs that are forwarded to the stub area, you can forbid the summary function on ABR.

Run the following command in router configuration mode to set the domain's parameters:

Command	Purpose
area *area-id* **stub** [no-summary]	Defines a stub area.
area *area-id* **default-cost** *cost*	Sets the weight of the default route of the stub area.

As to those areas that are not backbone areas and does not connect the backbone areas directly or as to those discontinuous areas, the OSPFv3 virtual link can be used to establish a logic connectivity. In order to create a virtual link, you have to perform configuration at the two terminals of the virtual link. If only one terminal is configured, the virtual link cannot work.

Run the following command in router configuration mode to set the domain's parameters:

Command	Purpose
area *area-id* **virtual-link** *neighbor-ID* [**dead-interval** *dead-value*][**hello-interval** *hello-value*][**retransmit-interval** *retrans-value*][**transdly** *dly-value*]	Establishes the virtual link.

4.3.6 Setting the Route Summary in the OSPFv3 Domain

With this function ABR can broadcast a summary route to other areas. In OSPFv3 ABR will broadcast each network to other areas. If network IDs are distributed to be continuous, you can set ABR to broadcast a summary route to other areas. The summary route can cover all networks in a certain range.

Run the following command in router configuration mode to set the address' range:

Command	Purpose
area *area-id* **range** *ipv6-prefix /prefix-length*	Sets the address' range of the summary route.

4.3.7 Setting the Summary of the Forwarded Routes

When routes are distributed from other routing areas to the OSPFv3 routing area, each route is singularly broadcasted as an external LSA. However, you can set a route on a router to make this route cover an address range. In this way, the size of the OSPFv3 link-state database can be reduced.

Run the following command in router configuration mode to set a summary route:

Command	Purpose
summary-prefix *ipv6-prefix /prefix-length*	Broadcasts only one summary route.

4.3.8 Generating a Default Route

ASBR should generate a default route to enter the OSPFv3 routing area. Whenever it is, you make configuration to enable a router to distribute a route to the OSPFv3 routing area and this route becomes ASBR automatically. However, ASBR does not generate a default route by default to enter the OSPFv3 routing area.

4.3.9 Choosing the Route ID on the Loopback Interface

OSPFv3 uses the maximum IPv4 address as its router ID. If the interface that connects the IPv4 address is down or the IPv4 address is deleted, the OSPF process will recalculate the ID of this new router and retransmit the routing information from all interfaces.

If an IPv4 address is configured on a loopback interface, the router will first use the IPv4 address of loopback as its ID. Because the loopback interface will never be down, the routing table is greatly stable.

The router can first select the loopback interface as its ID or select the maximum IPv4 address in all loopback interfaces as its ID. If there is no loopback interface, the IPv4 address of a router will be used as the router ID. You cannot specify OSPFv3 to use any specific interface.

Run the following commands in global configuration mode to set the IP loopback interface:

Command	Purpose
interface loopback *num*	Creates a loopback interface and enters the interface configuration mode.
ip address *ip-address mask*	Distributes an IPv4 address for an interface.

4.3.10 Setting the Timer of Routing Algorithm

You can set the delay between receiving the topology change information and calculating SPF. You can also set the interval between two continuous SFP algorithm. Run the following command in router configuration mode:

Command	Purpose
timers delay *delaytime*	Set a delay for routing algorithm in an area.
timers hold *holdtime*	Sets a minimum interval for routing algorithm in an area.

4.3.11 Monitoring and Maintaining OSPFv3

The network statistics information which can be displayed includes the content of the IP routing table, caching and database. This kind of information can help users to judge the usage of network resources and solve network problems.

You can run the following commands to display all kinds of routing statistics information:

Command	Purpose
show ipv6 ospf [*process-id*]	Displays the general information about the OSPFv3 routing process.

show ipv6 ospf [*process-id*] **database** **show ipv6 ospf** [*process-id*] **database [router] [adv-router** *router-id***]** **show ipv6 ospf** [*process-id*] **database [network] [adv-router** *router-id***]** **show ipv6 ospf** [*process-id*] **database [inter-prefix] [adv-router** *router-id***]** **show ipv6 ospf** [*process-id*] **database [inter-router] [adv-router** *router-id***]** **show ipv6 ospf** [*process-id*] **database [external] [adv-router** *router-id***]** **show ipv6 ospf** [*process-id*] **database [link] [adv-router** *router-id***]** **show ipv6 ospf** [*process-id*] **database [intra-prefix] [adv-router** *router-id***]**	Displays the information about the OSPFv3 database.
show ipv6 ospf interface	Displays the information about the OSPFv3 interface.
show ipv6 ospf neighbor	Displays the information about OSPFv3 neighbors.
show ipv6 ospf route	Displays the routing information about OSPFv3.
show ipv6 ospf topology	Displays the OSPFv3 topology.
show ipv6 ospf virtual-links	Displays the virtual links of OSPFv3.
debug ipv6 ospf	Monitors all OSPFv3 behaviors.
debug ipv6 ospf events	Monitors the OSPFv3 events.
debug ipv6 ospf ifsm	Monitors the state machine of the OSPFv3 interface.
debug ipv6 ospf lsa	Monitors related behaviors about OSPFv3 LSA.
debug ipv6 ospf nfsm	Monitors the state machine of the OSPFv3 neighbors.
debug ipv6 ospf nsm	Monitors the information of which the management module notifies OSPFv3.
debug ipv6 ospf packet	Monitors the OSPFv3 packets.
debug ipv6 ospf route	Monitors the routing information about OSPFv3.

4.4 OSPFv3 Configuration Example

4.4.1 Example for OSPFv3 Route Learning Settings

OSPFv3 requires switching information among many internal routers, ABR and ASBR. In the minimum settings, the OSPFv3-based router works under the case that all its parameters take their default values and there is no authentication.

The following are three configuration examples:

The first example shows the commands for basic OSPFv3 settings.

The second example shows multiple OSPFv3 processes can be set on a router.

The third example shows how to use OSPFv3 to learn routes.

The fourth example shows how to set the OSPFv3 virtual link.

a. Basic OSPFv3 Configuration Example

The following example shows a simple OSPFv3 settings. In this example, you have to activate process 90, connect Ethernet interface 0 to area 0.0.0.0, distribute RIPng to OSPFv3 and OSPFv3 to RIPng.

```
ipv6 unicast-routing
!
interface vlan 10
ipv6 address 2001::1/64
ipv6 enable
ipv6 rip admin enable
ipv6 rip admin split-horizon

ipv6 ospf 90 area 0
ipv6 ospf cost 1
!
ipv6 router ospf 90
router-id 1.1.1.1
redistribute rip
!
ipv6 router rip admin
redistribute ospf 90
```

b. Configuring multiple OSPFv3 processes

The following example shows that two OSPFv3 processes are created.

```
ipv6 unicast-routing
!
!
interface vlan 10
  ipv6 address 2001::1/64
   ipv6 enable

   ipv6 ospf 109 area 0 instance 1
   ipv6 ospf 110 area 0 instance 2
!
!
interface vlan 11
```

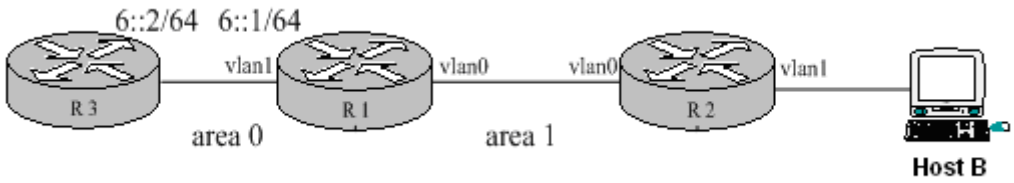
```
  ip address 2002::1/64
   ipv6 enable

 ipv6 ospf 109 area 1 instance 1
 ipv6 ospf 110 area 1 instance 2
!
!
ipv6 router ospf 109
  router-id 1.1.1.1
   redistribute static
!
ipv6 router ospf 110
  router-id 2.2.2.2
!
```

Each interface can belong to many OSPFv3 processes, but if an interface belongs to multiple OSPFv3 processes each OSPFv3 process must correspond to different instances.

c. Complicated configuration example

The following example shows how to configure multiple routers in a single OSPFv3 autonomous system. The following figure shows the network topology of the configuration example:



Configure the router according to the figure above:

```
R1:
interface vlan 0
     ipv6 enable

    ipv6 ospf 1 area 1
!
interface vlan 1
     ipv6 enable

    ipv6 ospf 1 area 0
!
ipv6 route 2001::/64 6::2
!
ipv6 router ospf 1
   router-id 1.1.1.1
    redistribute static
!

R2:
interface vlan 0
```

```
 ipv6 enable

 ipv6 ospf 1 area 1
!
!
ipv6 router ospf 1
 router-id 2.2.2.2
!
```

Browsing the routing table of R2:

```
R2#show ipv6 route
O       6::/64[1]
         [110,20] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
O       2001::/64[1] （route forwarding）
         [110,150] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
C       fe80::/10[1]
         is directly connected, L,Null0
C       fe80::/64[1]
         is directly connected, C, VLAN0
C       fe80::2e0:fff:fe26:a8/128[1]
         is directly connected, L, VLAN0
C       ff00::/8[1]
         is directly connected, L,Null0
```

From the command sentences above, we can see that R2 has learned route forwarding.

Setting area 1 to be the stub area:

R1:

```
interface vlan 0
 ipv6 enable

 ipv6 ospf 1 area 1
!
interface vlan 1
 ipv6 enable

 ipv6 ospf 1 area 0
!
ipv6 route 2001::/64 6::2
!
ipv6 router ospf 1
 router-id 1.1.1.1
 area 1 stub
 redistribute static
!
```

R2:

```
interface vlan 0
 ipv6 enable

 ipv6 ospf 1 area 1
!
!
ipv6 router ospf 1
 router-id 2.2.2.2
```

```
   area 1 stub
!
```

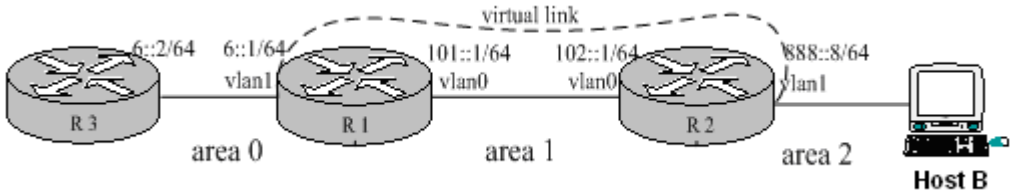
Browsing the routing table of R2:

```
R2#show ipv6 route
O        ::/0[1]
          [110,11] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
O        6::/64[1]
          [110,20] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
C        fe80::/10[1]
          is directly connected, L,Null0
C        fe80::/64[1]
          is directly connected, C, VLAN0
C        fe80::2e0:fff:fe26:a8/128[1]
          is directly connected, L, VLAN0
C        ff00::/8[1]
          is directly connected, L,Null0
```

It can be judged that ABR in the stub area can generate a default route normally and notify other routers in this area without importing ASE LSA into the stub area.

d. Configuring the virtual link

The following example shows how to configure a virtual link in a single autonomous OSPFv3 system. The following figure shows the network topology of the configuration example:



Configure the router according to the figure above:

R1:

```
interface vlan 0
    ipv6 address 101::1/64
     ipv6 enable

    ipv6 ospf 1 area 1
!
interface vlan 1
    ipv6 address 6::1/64
     ipv6 enable

    ipv6 ospf 1 area 0
!
ipv6 route 2001::/64 6::2
!
ipv6 router ospf 1
   router-id 200.200.200.1
   area 1 virtual-link 200.200.200.2
```

```
    redistribute static
!

R2:
interface vlan 0
    ipv6 address 101::2/64
     ipv6 enable

    ipv6 ospf 1 area 1
!
interface vlan 1
    ipv6 address 888::8/64
     ipv6 enable

    ipv6 ospf 1 area 2
!
!
ipv6 router ospf 1
  router-id 200.200.200.2
  area 1 virtual-link 200.200.200.1
!
```

Browsing the state of the OSPFv3 neighbor:

```
R1#show ipv6 ospf neighbor
OSPFv3 Process (1)
Neighbor ID      Pri    State           Dead Time    Interface    Instance ID
200.200.200.2      1    Full/DR         00:00:35     VLAN0          0
200.200.200.2      1    Full/ -         00:00:36     VLINK1         0

R2#show ipv6 ospf neighbor
OSPFv3 Process (1)
OSPFv3 Process (1)
Neighbor ID      Pri    State           Dead Time    Interface    Instance ID
200.200.200.1      1    Full/Backup     00:00:36     VLAN0          0
200.200.200.1      1    Full/ -         00:00:37     VLINK1         0
```

Browsing the information in the routing table:

```
R1#show ipv6 route
C       6::/64[1]
         is directly connected, C,VLAN1
C       6::1/128[1]
         is directly connected, L, VLAN1
C       101::/64[2]
         is directly connected, C, VLAN0
C       101::1/128[2]
         is directly connected, L, VLAN0
O       101::2/128[2]
         [110,10] via fe80:4::2e0:fff:fe26:a8(on VLAN0)
O       888::/64[2]
         [110,20] via fe80:4::2e0:fff:fe26:a8(on VLAN0)
S       2001::/64[1]
         [1,0] via 6::2(on VLAN1)
C       fe80::/10[2]
         is directly connected, L,Null0
```

```
C       fe80::/64[2]
         is directly connected, C, VLAN0
C       fe80::2e0:fff:fe26:2d98/128[2]
         is directly connected, L, VLAN0
C       fe80::/64[1]
         is directly connected, C, VLAN1
C       fe80::2e0:fff:fe26:2d99/128[1]
         is directly connected, L, VLAN1
C       ff00::/8[2]
         is directly connected, L,Null0

R2#show ipv6 route
O       6::/64[1]
         [110,20] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
C       101::/64[1]
         is directly connected, C, VLAN0
O       101::1/128[1]
         [110,10] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
C       101::2/128[1]
         is directly connected, L, VLAN0
C       888::/64[1]
         is directly connected, C, VLAN1
C       888::8/128[1]
         is directly connected, L, VLAN1
O       2001::/64[1]
         [110,150] via fe80:4::2e0:fff:fe26:2d98(on VLAN0)
C       fe80::/10[1]
         is directly connected, L,Null0
C       fe80::/64[1]
         is directly connected, C, VLAN0
C       fe80::2e0:fff:fe26:a8/128[1]
         is directly connected, L, VLAN0
C       fe80::/64[1]
         is directly connected, C, VLAN1
C       fe80::2e0:fff:fe26:a9/128[1]
         is directly connected, L, VLAN1
C       ff00::/8[1]
         is directly connected, L,Null0
```

Chapter 5 **Configuring the Routing Management Modules**

5.1 Overview

The static route is a special route and configured by the administrator manually; after a static route is set the packets with a designated destination will be forwarded along the path that is designated by the administrator.

In those networks with simple networking structures, only the configuration of the static routes can realize network interconnection. Properly setting and using static routes can improve the performance of networks and guarantee the bandwidth for important network application.

The shortage of the static route is that it cannot automatically adapt to the change of the network topology. When the network has trouble or the topology changes, the static routes are unreachable and the network then interrupts. In this case, the network administrator has to change the settings of static routes manually.

If the data packets that reach a designated network cannot find the corresponding items in the routing table in a device, the device will then discard these data packets.

After a default route is configured on the current device, those data packets that have no corresponding items in the routing table will not be discarded by the current device but forwarded to another device, which will forward these data packets.

The default route is used only when a device has not found a matching entry in the routing table.

If the destination address of a packet does not match up any entry in the routing table, this packet will select the default route.

If there is no default route and the destination of the packet is not in the routing table, this packet will be discarded and an ICMPv6 packet will be sent back to the source terminal, reporting that the destination address and the network are unreachable.

The default routes can be generated in two ways:

The first way is that the network administrator sets a static route to network 0::0/0. As to an incoming data packet, if the current device cannot find the corresponding routing item in the routing table, it will forward this packet to the designated next-hop port that is set in the static route.

The second way is that the default route is generated by the dynamic routing protocols. A device with strong routing ability forwards the default route to other devices, and the other devices generate in their own routing tables the default route that is oriented to the device with strong routing ability.

5.2 Configuration Task List of Routing Management Module

The routing management module has the following configuration tasks:

- Setting the static route
- Setting the threshold of routes in a routing table
- Checking whether the next hop of the static route is reachable

5.3 Routing Management Module's Configuration Tasks

5.3.1 Setting the Static Route

To set the static route, run the following command in global configuration mode:

Command	Purpose
Ipv6 route *prefix / prefixlen* {*ipv6-address* | *interface-type interface-number* [*ipv6-address*]} [*distance*]	Sets the static route.

When setting the static route, you can designate the type and number of the outgoing interfaces, and also the address of the next hop. It depends on actual requirements whether to designate an outgoing interface or the next-hop address. The next-hop address cannot be the IPv6 address of the local address, or the static route is invalid.

When you run **IPv6 route** to set the static route, if the destination address and the mask are set to 0::0/0, the configured route is a default one. The **prefixlen** parameter in the configured prefix should be less than or equal to 64, or be equal to 128 (host's route).

Different management distances can be set for different static routes and therefore these static routes can be flexibly applied on the routing management modules.

The next hop of the configured static route must be activated, otherwise the static route cannot be activated. When the next hop is an interface or an interface VLAN, the interface must be a v6 one; when the next hop is a gateway, this gateway must be in the directly-connected network segment.

5.3.2 Setting the Threshold of Routes in a Routing Table

To a maximum number of routes in a routing table, that is, to set a threshold for routes in a routing table, run the following command in global configuration mode:

Command	Purpose
Ipv6 route max-number <*64-640000*>	Sets the total number of routes, distributes systematic resources reasonably and provides a set speed.

5.3.3 Monitoring and Maintaining the State of the Routing Table

To display all kinds of statistics information about routes, run the following commands in EXEC mode:

Command	Purpose
show ipv6 route	Displays the information in the main routing table.
show ipv6 route [*protocol*]	Displays the routing information of the corresponding routing protocol in the routing table.

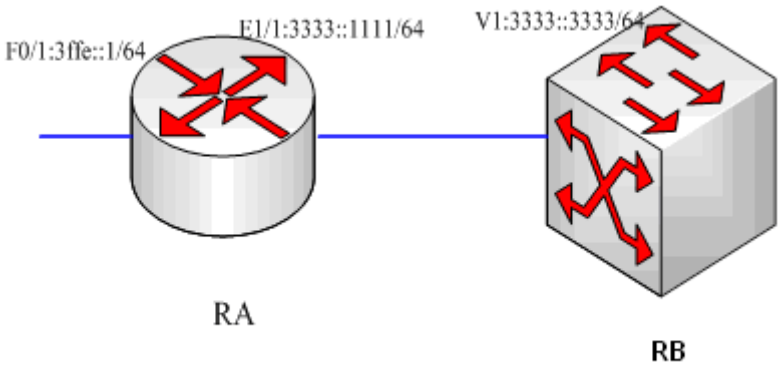
show ipv6 route summary	Displays the statistics information about the main routing table.
show ipv6 fib route	Displays the items in the main forwarding table, FIB.
show ipv6 fib summary	Displays the statistics information about the forwarding table.
Show ipv6 route summary (line card)	Displays the statistics information about the routing table on the wire card.
Show ipv6 route information (line card)	Displays the information about the current state on the wire card.
Show ipv6 route [delete | stale | un-lpm | no-lla | ipv6-address] (line card)	Displays all kinds of routing information on the wire card.

To trace all related events and information on the route management module, run the following commands in EXEC mode.

Command	Purpose
debug ipv6 routing message	Traces the reception and transmission of the IPv6 information.
debug ipv6 routing search	Traces routing search.
debug ipv6 routing timer	Traces the IPv6 clock.
Debug ipv6 routing redis	Traces the IPv6-redistribute information.
debug ipv6 fib cache	Traces the IPv6-cache information.
Debug ipv6 fib message	Traces the information in the IPv6 forwarding table.
Debug ipv6 routing exf (line card)	Traces the EXF information that is added to the line card.
Debug ipv6 routing packet (line card)	Traces the packet interaction information between the wire card and the main-control line card.
Debug ipv6 routing message (line card)	Traces the interaction information between the line card and the main control unit.
Debug ipv6 routing cache (line card)	Traces the information about the cache on the line card.
Debug ipv6 routing route (line card)	Traces the information about route change on the line card.
Debug ipv6 routing search (line card)	Traces the routing search information on the line card.

5.4 Static Route's Configuration Example

As shown in the following figure, RA directly connects router RB.



RA configuration:

```
interface FastEthernet0/1
 no ip address
 no ip directed-broadcast
 ipv6 address 3FFE::1/64
!
interface Ethernet1/1
 no ip address
 no ip directed-broadcast
 duplex half
 ipv6 address 3333::1111/64
!
```

RB configuration:

```
interface VLAN1
 no ip address
 no ip directed-broadcast
 ipv6 address 3333::3333/64
!
!
```

Browsing the address of the local link of RA:

```
RA_config#show ipv6 route
Codes: C - Connected, L - Local, S - Static, R - Ripng, B - BGP
       ON1 - OSPF NSSA external type 1, ON2 - OSPF NSSA external type 2
       OE1 - OSPF external type 1, OE2 - OSPF external type 2
       DHCP - DHCP type

VRF ID: 0

C       3333::/64[1]
         is directly connected, C,Ethernet1/1
C       3333::1111/128[1]
         is directly connected, L,Ethernet1/1
C       3ffe::/64[1]
         is directly connected, C,FastEthernet0/1
```

```
C       3ffe::1/128[1]
         is directly connected, L,FastEthernet0/1
C       fe80::/10[1]
         is directly connected, L,Null0
C       fe80::/64[1]
         is directly connected, C,FastEthernet0/1
C       fe80::a00:3eff:fed5:effc/128[1]
         is directly connected, L,FastEthernet0/1
C       fe80::/64[1]
         is directly connected, C,Ethernet1/1
C       fe80::a00:3eff:fed5:effd/128[1]
         is directly connected, L,Ethernet1/1
C       ff00::/8[1]
         is directly connected, L,Null0
!
```

Setting a static route, which leads to subnet **3ffe::/64** on RB:

```
!
ipv6 route 3ffe::/64 3333::1111
!
```

Or:

```
!
ipv6 route 3ffe::/64 VLAN1 fe80::a00:3eff:fed5:effd
!
```

Chapter 6 IPv6 ACL Configuration on the Physical Port

6.1 Setting IPv6 ACL Based on the Physical Port

6.1.1 Filtrating the IPv6 Packets

The IPv6 access control list is a set of ordered permit and deny conditions to apply the IPv6 address. The ROS software of our routers tests the addresses one by one in ACL. The first matchup decides whether the software accepts or denies the address. Because the ROS software stops matchup regulations after the first matchup, the ordering of conditions is important. If no regulation matches, this address is denied.

There are two steps in ACL:

(1) Establishing ACL through designating the ACL's name and the access conditions

(2) Applying ACL on a port

6.1.2 Establishing the IPv6 ACL

You can use a character string to establish the IPv6 ACL.

To establish the IPv6 ACL, run the following commands in global configuration mode.

Command	Purpose
Ipv6 access-list *name*	Uses a name to define a standard IPv6 ACL.
{**deny	permit**} *protocol* {*source-ipv6-prefix*/*prefix-length*	**any**	**host** *source-ipv6-address*} [*operator* [*port-number*]] {*destination-ipv6-prefix*/*prefix-length*	**any**	**host** *destination-ipv6-address*} [*operator* [*port-number*]] [**dscp** *value*] [**flow-label** *value*] [**sequence** *value*] {**deny	permit**} *protocol* **any any**	Specifies one or multiple deny or permit conditions in IPv6 ACL configuration mode.
Exit	Exits the IPv6 ACL configuration mode.

After an access control list is originally established, any added content is put at the end of the list. In other words, you cannot choose to add an ACL command line from a specified ACL. However, you can run **no permit** and **no deny** to remove a command line from ACL.

Note:

When an ACL is established, remember that the end of the default ACL contains the implicit deny sentence.

After ACL is established, it must be applied on the lines or ports.

6.1.3 Applying ACL on Ports

When ACL is established, it will be applied on one or multiple ports, or the egress port.

Run the following command to apply IPv6 ACL on a port:

Command	Purpose
ipv6 access-group *name in/out*	Ipv6 access list applied to the port

The access control list is used on the incoming interface. When a packet is received, the source address and destination address of the packet will be checked in ACL. If the access control list permits the destination address, the system will continue handling the packet. If ACL denies the address, the software discards the packet.

If the designated access control list does not exist, all packets are allowed to pass through.

6.1.4 ACL Example

In the following example, the first command line allows the new TCP to connect the Telnet port of host **2001:1::2E0:FFF:FE8E:700B**.

```
ipv6 access-list aaa
   permit tcp any host 2001:1::2E0:FFF:FE8E:700B eq 23

interface   g1/10
   ipv6 access-group aaa in
```

Chapter 7 **IPv6 Tunnel Configuration**

IPv6 tunnel is a protocol in which the IPv6 packets are used as IPv4 load and work in the IPv4 network to connect each isolate IPv6. The settings must be conducted on a tunnel port. The basic configuration flow is shown below:

- Setting the tunnel port
- Setting the IPv6 address of the tunnel port
- Setting the source Ipv4 address of a tunnel
- Setting the destination IPv4 address of the tunnel (optional)
- Setting the encapsulation ipv6 tuuel pto
- Setting routes

The IPv6 tunnel protocol is used set a tunnel, a 6-to-4 tunnel or an ISTAP tunnel.

7.1 Setting the IPv6 Tunnel Manually

7.1.1 Configuration Condition

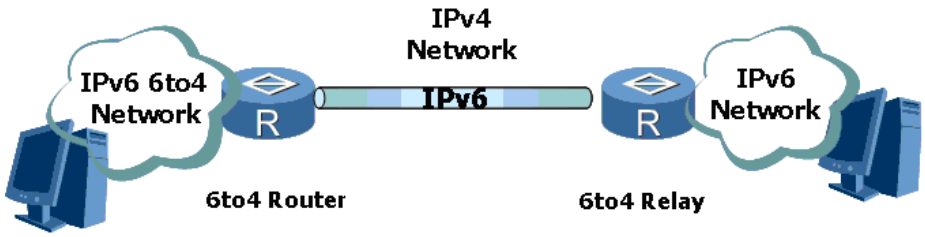
Manual IPv6 tunnel configuration need be conducted on a tunnel port with an IPv6 address. Moreover, the IPv4 addresses of the source terminal and destination terminal must be configured on the tunnel port manually. The two terminals of the manually configured tunnel must support the IPv4 stack and IPv6 stack at the same time. Because the tunnel's terminals need be expressly configured, this tunnel mode is suitable for a tunnel between routers.

7.2 IPv6 6to4 Tunnel

7.2.1 Configuration Condition

The IPv6 6to4 tunnel is a means to interconnect isolate IPv6 island in the IPv4 network. Each IPv6 isolated island has at least a unique IPv4 address of the whole network. This IPv4 address is taken by the edge router of the IPv6 isolated island as the source IPv4 address of the 6to4 tunnel and hence the prefix of the IPv6 address, 2002:border-router-IPv4-address::/48, is constructed. That's why this prefix is also unique in the whole network. The destination IPv4 address of the 6to4 tunnel need not be configured manually, but be extracted from the IPv6 destination address. The two terminals of the 6to4 tunnel must support the IPv4 stack and IPv6 stack at the same time.

7.2.2 6to4 Relay



In the 6to4 tunnel technology, the 6to4 network must use the 6to4 address. However, the 6to4 network, as a part of the whole network, need connect other pure IPv6 networks. In order to solve this problem, the 6to4 relay need be used.

Between a 6to4 domain and a pure IPv6 domain, at least one 6to4 relay is required to guarantee the connectivity.

7.3 IPv6 ISATAP Tunnel

7.3.1 Configuration Condition

The IPv6 ISATAP tunnel is mainly used to connect some isolated IPv6 hosts and the IPv6 network. In this kind of connection, one end supports the IPv6-IPv4-stack host of ISATAP and the other end supports the ISATAP router. When the ISATAP tunnel is used, the destination address of the IPv6 packet and the IPv6 address of the tunnel interface adopt the special address—the ISATAP address. The format of the ISATAP address is **Prefix（64bit）:0:5EFE:IPv4ADDR**, among which IPv4ADDR is the IPv4 source address of the tunnel interface. Through this embedded IPv4 address, the tunnel can be automatically established to finish the transmission of the IPv6 packets. Hence, the IPv6 address of the ISATAP tunnel must be a modified EUI-64 address.

It is to be noted that the current router 8000 series can only run as the ISATAP routers, not as the ISATAP hosts.

7.3.2 ISATAP Host

One end of the ISATAP tunnel is a host, whose IPv6 address has no state. The ISATAP host generates the local IPv6 link address automatically through the IPv4 address of the source terminal, sends an RS message to an ISATAP router through the ISATAP tunnel, and then the ISATAP router transmits a RA message, which contains the ISATAP prefix. After receiving the ISATAP prefix, the ISATAP host will obtain the global IPv6 address and add the ISATAP router to the potential routing table.

Because the ISATAP router cannot send an RA message positively to the host at the other terminal of the ISATAP tunnel, the lifetime of RA, which is set on the router, must be longer than the interval of RS transmission on the host.

Configuration of the ISATAP host:

Enter the following commands on the command-line window:

ipv6 install

netsh

interface ipv6 isatap

set router *ipv4-addr* *(ipv4-addr is the IPv4 address of the source terminal of the ISATAP tunnel.)*

Chapter 8 **NATPT Configuration**

8.1 NATPT Configuration

The NATPT protocol is disabled in default state. If NATPT is needed to transform the packets in domain 4 to the packets in domain 6, NATPT should be enabled in interface configuration mode and the correct configuration transformation regulation should be set. To enable the IPv6 protocol, you must set the **ipv6 nat** command on the port on which the packets are received and transmitted.

To enable the NATPT protocol, users should finish the following task:

Configuring the ipv6 nat command on at least one IPv4 port and at least one IPv6 port in port configuration mode

8.2 Enabling NATPT

8.2.1 Setting **ipv6 nat** on a Port

You should set the **ipv6 nat** command on the port on which the IPv4/IPv6 packets are transmitted or received. If this command is not set, the transformed packets cannot be transmitted out even if the transformation regulation is found.

Chapter 9 **Setting NATPT Services**

9.1 Setting NATPT Services

To transform the packets from v4 domain to v6 domain by using NATPT, you have to set the corresponding regulations besides enabling the NATPT protocol. The currently provided NATPT services mainly include:

- Managing the transformation regulation
- Managing the address pool
- Managing the transformation mapping that are generated by the transformation regulation

9.1.1 Managing the Transformation Regulation

The transformation regulations include two kinds: static transformation regulations and dynamic transformation regulation. The priority of static transformation regulations is higher than that of dynamic transformation regulations, that is, the static transformation regulations will be first matched when the packets are transformed. The main services can also be divided into three kinds:

- Managing static transformation regulations
- Managing dynamic transformation regulations
- Managing the prefix of the IPv6 matchup address

- **Managing Static Transformation Regulations**

  To add a static transformation regulation to a router, run the following commands in global configuration mode:

Command	Purpose
ipv6 nat v4v6 source *ipv4-address ipv6-address*	Sets a static transformation regulation for a source address from v4 domain to v6 domain. When the source address of the IPv4 packet is ***ipv4-address***, the source address of the transformed IPv6 packet is ***ipv6-address***.
ipv6 nat v6v4 source *ipv6-address ipv4-address*	Sets a static transformation regulation for a source address from v4 domain to v6 domain. When the source address of the IPv4 packet is ***ipv6-address***, the source address of the transformed IPv6 packet is ***ipv4-address***.

  It is to be noted that the static transformation regulations are bidirectional. Take **ipv6 nat v4v6 source 1.1.1.1 2001::1** as an example. On one side when the packets are transformed from IPv4 to IPv6, the source address will be replaced; on the other side, when an IPv6 packet with a destination address **2001::1** will also be transformed to be an IPv4 packet with a destination address **1.1.1.1**.

- **Managing Dynamic Transformation Regulations**

To add a dynamic transformation regulation to a router, run the following commands in global configuration mode:

Command	Purpose
ipv6 nat v4v6 source list *access- name* **pool** *name*	Sets a dynamic transformation regulation of the source address from v4 domain to v6 domain. When the IPv4 packet passes the checkup of the **access-name** IPv4 ACL, an IPv6 packet from the IPv6 address pool, name, is used as the source address of the transformed IPv6 packet.
ipv6 nat v6v4 source list *access- name* **pool** *name* **[overload]**	Sets a dynamic transformation regulation of the source address from v6 domain to v4 domain. When the IPv6 packet passes the checkup of the **access-name** IPv6 ACL, an IPv6 packet from the IPv4 address pool, **name**, is used as the source address of the transformed IPv4 packet.
ipv6 nat v6v4 source list *access- name* **interface interface**-*name* **overload**	Sets a dynamic transformation regulation of the source address from v6 domain to v4 domain. When the IPv6 packet passes the checkup of the **access-name** IPv6 ACL, it will automatically use the main IPv4 address of the IPv4 packet as the source address of the transformed packet.

Different from the static transformation regulations, the dynamic transformation regulations are unidirectional.

- **Managing the Prefix of the IPv6 Matchup Address**

  When packets are transformed from the v6 domain to the v4 domain, those IPv6 packets whose destination addresses match the prefix of the designated IPv6 address can be transformed.

  To set the prefix of a global IPv6 matchup address, run the following commands in global configuration mode; to set the prefix of the IPv6 matchup address on a specific port, run the following commands on port configuration mode:

Command	Purpose
ipv6 nat prefix *ipv6-prefix/prefix-length*	Sets the prefix of the IPv6 matchup address. The format of **ipv6-prefix** should comply with the requirements of RFC2373. **prefix-length** means the length of the prefix and its decimal value can only be 96.
ipv6 nat prefix *ipv6-prefix/prefix-length* **v4-mapped** *access-list-name*	Sets the prefix of the IPv6 matchup address.

9.1.2 Managing the Address Pool

The address pools can be divided into two kinds: one kind providing the IPv6 addresses while the other kind providing the IPv4 addresses. The configuration commands for two kinds of address pools are similar. The management of the address pool provides the following services:

- Managing the IPv4 address pool

- Managing the IPv6 address pool

Managing the IPv4 Address Pool

To add an IPv4 address pool to a router, run the following command in global configuration mode:

Command	Purpose
ipv6 nat v6v4 pool *name start-ipv4 end-ipv4* **prefix-length** *prefix-length*	Sets an IPv4 address pool whose name is **name**. The start address and end address of the address pool are **start-ipv4** and **end-ipv4** respectively. The **prefix-length** parameter of the start address or end address ranges between 1 and 32.

The IPv4 address pool is used when packets are transformed from the v6 domain to the v4 domain. The transformed IPv4 packet will be distributed with a dynamic IPv4 source address.

Managing the IPv6 Address Pool

To add an IPv6 address pool to a router, run the following command in global configuration mode:

Command	Purpose
ipv6 nat v4v6 pool *name start-ipv6 end-ipv6* **prefix-length** *prefix-length*	Sets an IPv6 address pool whose name is **name**. The start address and end address of the address pool are **start-ipv6** and **end-ipv6** respectively. The **prefix-length** parameter of the start address or end address ranges between 1 and 128.

The IPv6 address pool is used when packets are transformed from the v4 domain to the v6 domain. The transformed IPv6 packet will be distributed with a dynamic IPv6 source address.

9.1.3 Transforming Mapping Management

The transformation mapping is generated by the packet according to the transformation regulation.