Contents

Configuring basic IPv6 settings

About IPv6

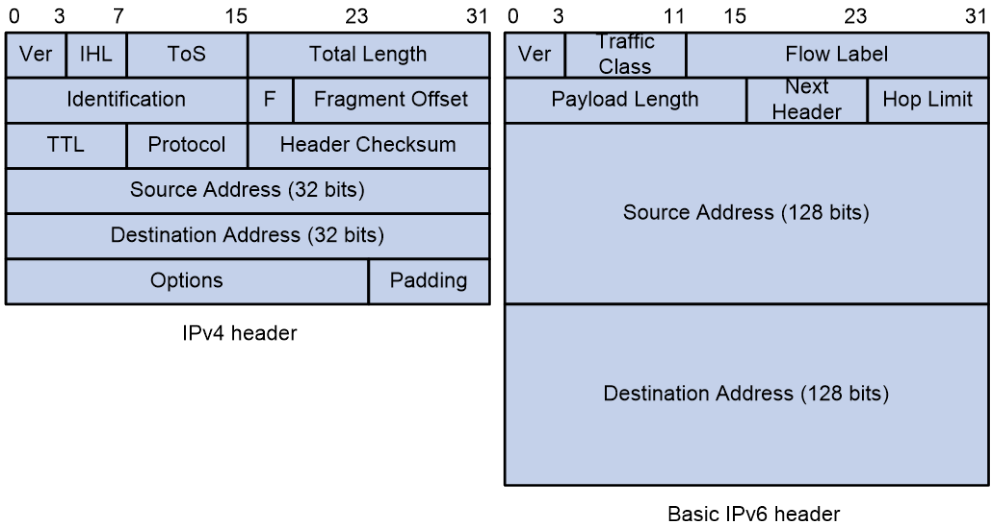
IPv6, also called IP next generation (IPng), was designed by the IETF as the successor to IPv4. One significant difference between IPv6 and IPv4 is that IPv6 increases the IP address size from 32 bits to 128 bits.

IPv6 features

Simplified header format

IPv6 removes several IPv4 header fields or moves them to the IPv6 extension headers to reduce the length of the basic IPv6 packet header. The basic IPv6 packet header has a fixed length of 40 bytes to simplify IPv6 packet handling and improve forwarding efficiency. Although the IPv6 address size is four times the IPv4 address size, the basic IPv6 packet header size is only twice the size of the option-less IPv4 packet header.

Figure 1 IPv4 packet header format and basic IPv6 packet header format



Larger address space

IPv6 can provide 3.4 x 10^{38} addresses to meet the requirements of hierarchical address assignment for both public and private networks.

Hierarchical address structure

IPv6 uses a hierarchical address structure to speed up route lookup and reduce the IPv6 routing table size through route aggregation.

Address autoconfiguration

To simplify host configuration, IPv6 supports stateful and stateless address autoconfiguration.

- Stateful address autoconfiguration enables a host to acquire an IPv6 address and other configuration information from a server (for example, a DHCPv6 server). For more information about DHCPv6 server, see "Configuring the DHCPv6 server."

- Stateless address autoconfiguration enables a host to automatically generate an IPv6 address and other configuration information by using its link-layer address and the prefix information advertised by a router.

To communicate with other hosts on the same link, a host automatically generates a link-local address based on its link-layer address and the link-local address prefix (FE80::/10).

Built-in security

IPv6 defines extension headers to support IPsec. IPsec provides end-to-end security and enhances interoperability among different IPv6 applications.

QoS support

The Flow Label field in the IPv6 header allows the device to label the packets of a specific flow for special handling.

Enhanced neighbor discovery mechanism

The IPv6 neighbor discovery protocol uses a group of ICMPv6 messages to manage information exchange among neighboring nodes on the same link. The group of ICMPv6 messages replaces ARP messages, ICMPv4 router discovery messages, and ICMPv4 redirect messages and provides a series of other functions.

Flexible extension headers

IPv6 eliminates the Options field in the header and introduces optional extension headers to provide scalability and improve efficiency. The Options field in the IPv4 packet header contains a maximum of 40 bytes, whereas the IPv6 extension headers are restricted to the maximum size of IPv6 packets.

IPv6 addresses

IPv6 address format

An IPv6 address is represented as a set of 16-bit hexadecimals separated by colons (:). An IPv6 address is divided into eight groups, and each 16-bit group is represented by four hexadecimal numbers, for example, 2001:0000:130F:0000:0000:09C0:876A:130B.

To simplify the representation of IPv6 addresses, you can handle zeros in IPv6 addresses by using the following methods:

- The leading zeros in each group can be removed. For example, the above address can be represented in a shorter format as 2001:0:130F:0:0:9C0:876A:130B.
- If an IPv6 address contains one or more consecutive groups of zeros, they can be replaced by a double colon (::). For example, the above address can be represented in the shortest format as 2001:0:130F::9C0:876A:130B.

IMPORTANT:

A double colon can appear once or not at all in an IPv6 address. This limit allows the device to determine how many zeros the double colon represents and correctly convert it to zeros to restore a 128-bit IPv6 address.

An IPv6 address consists of an address prefix and an interface ID, which are equivalent to the network ID and the host ID of an IPv4 address.

An IPv6 address prefix is written in IPv6-address/prefix-length notation. The prefix-length is a decimal number indicating how many leftmost bits of the IPv6 address are in the address prefix.

IPv6 address types

IPv6 addresses include the following types:

- **Unicast address**—An identifier for a single interface, similar to an IPv4 unicast address. A packet sent to a unicast address is delivered to the interface identified by that address.

- **Multicast address**—An identifier for a set of interfaces (typically belonging to different nodes), similar to an IPv4 multicast address. A packet sent to a multicast address is delivered to all interfaces identified by that address.

  Broadcast addresses are replaced by multicast addresses in IPv6.
- **Anycast address**—An identifier for a set of interfaces (typically belonging to different nodes). A packet sent to an anycast address is delivered to the nearest interface among the interfaces identified by that address. The nearest interface is chosen according to the routing protocol's measure of distance.

The type of an IPv6 address is designated by the first several bits, called the format prefix.

Table 1 Mappings between address types and format prefixes

Type		Format prefix (binary)	IPv6 prefix ID
Unicast address	Unspecified address	00...0 (128 bits)	::/128
	Loopback address	00...1 (128 bits)	::1/128
	Link-local address	1111111010	FE80::/10
	Global unicast address	Other forms	N/A
Multicast address		11111111	FF00::/8
Anycast address		Anycast addresses use the unicast address space and have the identical structure of unicast addresses.	

Unicast addresses

Unicast addresses include global unicast addresses, link-local unicast addresses, the loopback address, and the unspecified address.

- **Global unicast addresses**—Equivalent to public IPv4 addresses, global unicast addresses are provided for Internet service providers. This type of address allows for prefix aggregation to restrict the number of global routing entries.
- **Link-local addresses**—Used for communication among link-local nodes for neighbor discovery and stateless autoconfiguration. Packets with link-local source or destination addresses are not forwarded to other links.
- **A loopback address**—0:0:0:0:0:0:0:1 (or ::1). It has the same function as the loopback address in IPv4. It cannot be assigned to any physical interface. A node uses this address to send an IPv6 packet to itself.
- **An unspecified address**—0:0:0:0:0:0:0:0 (or ::). It cannot be assigned to any node. Before acquiring a valid IPv6 address, a node fills this address in the source address field of IPv6 packets. The unspecified address cannot be used as a destination IPv6 address.

Multicast addresses

IPv6 multicast addresses listed in Table 2 are reserved for special purposes.

Table 2 Reserved IPv6 multicast addresses

Address	Application
FF01::1	Node-local scope all-nodes multicast address.
FF02::1	Link-local scope all-nodes multicast address.
FF01::2	Node-local scope all-routers multicast address.
FF02::2	Link-local scope all-routers multicast address.

Multicast addresses also include solicited-node addresses. A node uses a solicited-node multicast address to acquire the link-layer address of a neighboring node on the same link and to detect

duplicate addresses. Each IPv6 unicast or anycast address has a corresponding solicited-node address. The format of a solicited-node multicast address is FF02:0:0:0:0:1:FFXX:XXXX. FF02:0:0:0:0:1:FF is fixed and consists of 104 bits, and XX:XXXX is the last 24 bits of an IPv6 unicast address or anycast address.

EUI-64 address-based interface identifiers

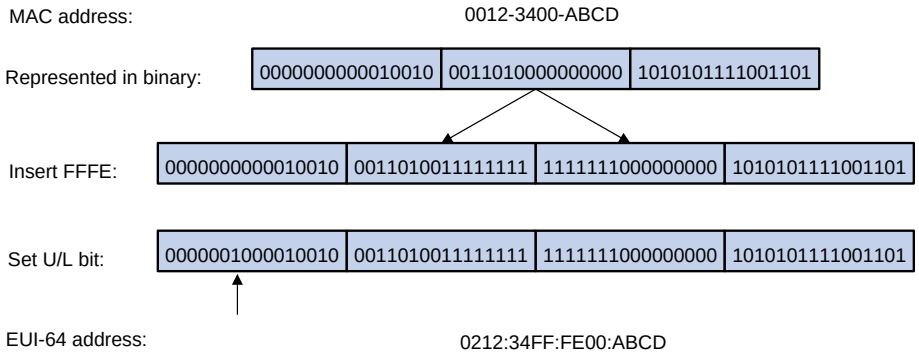
An interface identifier is 64 bits long and uniquely identifies an interface on a link.

On an IEEE 802 interface (such as a VLAN interface), the interface identifier is derived from the link-layer address (typically a MAC address) of the interface. The MAC address is 48 bits long.

To obtain an EUI-64 address-based interface identifier, follow these steps:

1. Insert the 16-bit binary number 1111111111111110 (hexadecimal value of FFFE) behind the 24th high-order bit of the MAC address.
2. Invert the universal/local (U/L) bit (the seventh high-order bit). This operation makes the interface identifier have the same local or global significance as the MAC address.

Figure 2 Converting a MAC address into an EUI-64 address-based interface identifier



On a tunnel interface, the lower 32 bits of the EUI-64 address-based interface identifier are the source IPv4 address of the tunnel interface. The higher 32 bits of the EUI-64 address-based interface identifier of an ISATAP tunnel interface are 0000:5EFE, whereas those of other tunnel interfaces are all zeros. For more information about tunnels, see "Configuring tunneling."

On an interface of another type (such as a serial interface) the EUI-64 address-based interface identifier is generated randomly by the device.

IPv6 path MTU discovery

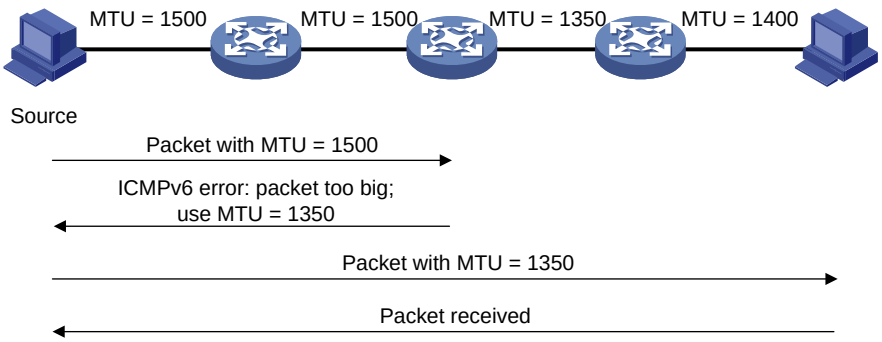
The links that a packet passes from a source to a destination can have different MTUs, among which the minimum MTU is the path MTU. If a packet exceeds the path MTU, the source end fragments the packet to reduce the processing pressure on intermediate devices and to use network resources effectively.

A source end uses path MTU discovery to find the path MTU to a destination, as shown in Figure 3.

1. The source host sends a packet no larger than its MTU to the destination host.
2. If the MTU of an intermediate device's output interface is smaller than the packet, the device performs the following operations:
   - Discards the packet.
   - Returns an ICMPv6 error message containing the interface MTU to the source host.
3. Upon receiving the ICMPv6 error message, the source host performs the following operations:
   - Uses the returned MTU to limit the packet size.
   - Performs fragmentation.
   - Sends the fragments to the destination host.

4. Step 2 and step 3 are repeated until the destination host receives the packet. In this way, the source host finds the minimum MTU of all links in the path to the destination host.

Figure 3 Path MTU discovery process



IPv6 transition technologies

IPv6 transition technologies enable communication between IPv4 and IPv6 networks.

Dual stack

Dual stack is the most direct transition approach. A network node that supports both IPv4 and IPv6 is a dual-stack node. A dual-stack node configured with an IPv4 address and an IPv6 address can forward both IPv4 and IPv6 packets. An application that supports both IPv4 and IPv6 prefers IPv6 at the network layer.

Dual stack is suitable for communication between IPv4 nodes or between IPv6 nodes. It is the basis of all transition technologies. However, it does not solve the IPv4 address depletion issue because each dual-stack node must have a globally unique IPv4 address.

Tunneling

Tunneling uses one network protocol to encapsulate the packets of another network protocol and transfers them over the network. For more information about tunneling, see "Configuring tunneling."

NAT-PT

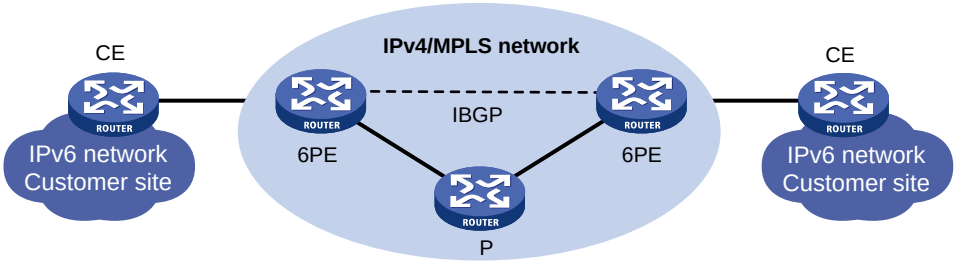
Network Address Translation – Protocol Translation (NAT-PT) enables communication between IPv4 and IPv6 nodes by translating between IPv4 and IPv6 packets. It performs IP address translation, and according to different protocols, performs semantic translation for packets. This technology is only suitable for communication between a pure IPv4 node and a pure IPv6 node.

6PE

6PE enables communication between isolated IPv6 networks over an IPv4 backbone network.

6PE adds labels to the IPv6 routing information about customer networks and advertises the information into the IPv4 backbone network over internal Border Gateway Protocol (IBGP) sessions. IPv6 packets are labeled and forwarded over tunnels on the backbone network. The tunnels can be GRE tunnels or MPLS LSPs.

Figure 4 Network diagram



6PE is a highly efficient solution. When an ISP wants to utilize the existing IPv4/MPLS network to provide IPv6 traffic switching, it only needs to upgrade the PE routers. In addition, the operation risk of 6PE is very low.

Protocols and standards

- RFC 1881, *IPv6 Address Allocation Management*
- RFC 1887, *An Architecture for IPv6 Unicast Address Allocation*
- RFC 1981, *Path MTU Discovery for IP version 6*
- RFC 2375, *IPv6 Multicast Address Assignments*
- RFC 2460, *Internet Protocol, Version 6 (IPv6) Specification*
- RFC 2464, *Transmission of IPv6 Packets over Ethernet Networks*
- RFC 2526, *Reserved IPv6 Subnet Anycast Addresses*
- RFC 3307, *Allocation Guidelines for IPv6 Multicast Addresses*
- RFC 4191, *Default Router Preferences and More-Specific Routes*
- RFC 4291, *IP Version 6 Addressing Architecture*
- RFC 4443, *Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification*

IPv6 basics tasks at a glance

To configure basic IPv6 settings, perform the following tasks:

1. Configuring an IPv6 address

   Choose the following tasks as needed:
   - Configuring an IPv6 global unicast address
   - Configuring an IPv6 link-local address
   - Configuring an IPv6 anycast address
2. (Optional.) Configuring path MTU discovery
   - Setting the interface MTU for IPv6 packets
   - Setting a static path MTU for an IPv6 address
   - Setting the aging time for dynamic path MTUs
3. (Optional.) Controlling sending ICMPv6 messages
   - Configuring the rate limit for ICMPv6 error messages
   - Enabling replying to multicast echo requests
   - Enabling sending ICMPv6 destination unreachable messages
   - Enabling sending ICMPv6 time exceeded messages

- Enabling sending ICMPv6 redirect messages
- Specifying the source address for ICMPv6 packets

4. (Optional.) Enabling Layer 3 packet statistics counting
5. (Optional.) Enabling IPv6 local fragment reassembly
6. (Optional.) Configuring IPv6 bandwidth-based load sharing
7. (Optional.) Enabling discarding IPv6 packets that contain extension headers

Configuring an IPv6 global unicast address

About IPv6 global unicast address

Use one of the following methods to configure an IPv6 global unicast address for an interface:

- **EUI-64 IPv6 address**—The IPv6 address prefix of the interface is manually configured, and the interface ID is generated automatically by the interface.
- **Manual configuration**—The IPv6 global unicast address is manually configured.
- **Stateless address autoconfiguration**—The IPv6 global unicast address is generated automatically based on the address prefix information contained in the RA message.
- **Prefix-specific address autoconfiguration**—The IPv6 global unicast address is generated automatically based on the prefix specified by its ID. The prefix can be manually configured or obtained through DHCPv6.

You can configure multiple IPv6 global unicast addresses on an interface.

Manually configured global unicast addresses (including EUI-64 IPv6 addresses) take precedence over automatically generated ones. If you manually configure a global unicast address with the same address prefix as an existing global unicast address on an interface, the manually configured one takes effect. However, it does not overwrite the automatically generated address. If you delete the manually configured global unicast address, the device uses the automatically generated one.

Generating an EUI-64 IPv6 address

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Configure an EUI-64 IPv6 address on the interface.
   ipv6 address { *ipv6-address prefix-length* | *ipv6-address*/*prefix-length* } **eui-64**
   By default, no EUI-64 IPv6 address is configured on an interface.

Manually assigning an IPv6 global unicast address

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Assign an IPv6 global unicast address to the interface.

ipv6 address { *ipv6-address prefix-length* | *ipv6-address*/*prefix-length* }

By default, no IPv6 global unicast address is configured on an interface.

Stateless address autoconfiguration

About stateless address autoconfiguration and temporary address

Stateless address autoconfiguration enables an interface to automatically generate an IPv6 global unicast address by using the address prefix in the received RA message and the interface ID. On an IEEE 802 interface (such as an Ethernet interface or a VLAN interface), the interface ID is generated based on the interface's MAC address and is globally unique. An attacker can exploit this rule to identify the sending device easily.

To fix the vulnerability, you can configure the temporary address feature. With this feature, an IEEE 802 interface generates the following addresses:

- **Public IPv6 address**—Includes the address prefix in the RA message and a fixed interface ID generated based on the MAC address of the interface.
- **Temporary IPv6 address**—Includes the address prefix in the RA message and a random interface ID generated through MD5.

You can also configure the interface to preferentially use the temporary IPv6 address as the source address of sent packets. When the valid lifetime of the temporary IPv6 address expires, the interface deletes the address and generates a new one. This feature enables the system to send packets with different source addresses through the same interface. If the temporary IPv6 address cannot be used because of a DAD conflict, the public IPv6 address is used.

The preferred lifetime and valid lifetime for a temporary IPv6 address are determined as follows:

- The preferred lifetime of a temporary IPv6 address takes the smaller of the following values:
  - The preferred lifetime of the address prefix in the RA message.
  - The preferred lifetime configured for temporary IPv6 addresses minus DESYNC_FACTOR (a random number in the range of 0 to 600 seconds).
- The valid lifetime of a temporary IPv6 address takes the smaller of the following values:
  - The valid lifetime of the address prefix.
  - The valid lifetime configured for temporary IPv6 addresses.

Restrictions and guidelines

If the IPv6 prefix in the RA message is not 64 bits long, stateless address autoconfiguration fails to generate an IPv6 global unicast address.

To generate a temporary address, an interface must be enabled with stateless address autoconfiguration. Temporary IPv6 addresses do not overwrite public IPv6 addresses, so an interface can have multiple IPv6 addresses with the same address prefix but different interface IDs.

If an interface fails to generate a public IPv6 address because of a prefix conflict or other reasons, it does not generate any temporary IPv6 address.

Executing the **undo ipv6 address auto** command on an interface deletes all IPv6 global unicast addresses and link-local addresses that are automatically generated on the interface.

Enabling stateless address autoconfiguration

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Enable stateless address autoconfiguration on an interface, so that the interface can automatically generate a global unicast address.

   ipv6 address auto

   By default, the stateless address autoconfiguration feature is disabled on an interface.

Configuring the temporary address feature and preferentially using the temporary IPv6 address as the source address of outgoing packets

1. Enter system view.

   system-view

2. Enable the temporary IPv6 address feature.

   ipv6 temporary-address [*valid-lifetime preferred-lifetime*]

   By default, the temporary IPv6 address feature is disabled.

3. Enable the system to preferentially use the temporary IPv6 address as the source address of the outgoing packets.

   ipv6 prefer temporary-address

   By default, the system does not preferentially use the temporary IPv6 address as the source address of the outgoing packets.

Configuring prefix-specific address autoconfiguration

1. Enter system view.

   system-view

2. Configure an IPv6 prefix.

   Choose one option as needed:

   - Configure a static IPv6 prefix.

     ipv6 prefix *prefix-number ipv6-prefix*/*prefix-length*

     By default, no static IPv6 prefixes exist.

   - Use DHCPv6 to obtain a dynamic IPv6 prefix.

     For more information about IPv6 prefix acquisition, see "Configuring the DHCPv6 client."

3. Enter interface view.

   interface *interface-type interface-number*

4. Specify an IPv6 prefix for an interface to automatically generate an IPv6 global unicast address and advertise the prefix.

   ipv6 address *prefix-number sub-prefix/prefix-length*

   By default, no IPv6 prefix is specified for the interface to automatically generate an IPv6 global unicast address.

Configuring an IPv6 link-local address

About IPv6 link-local address

Configure IPv6 link-local addresses using one of the following methods:

- **Automatic generation**—The device automatically generates a link-local address for an interface according to the link-local address prefix (FE80::/10) and the link-layer address of the interface.
- **Manual assignment**—Manually configure an IPv6 link-local address for an interface.

Restrictions and guidelines

After you configure an IPv6 global unicast address for an interface, the interface automatically generates a link-local address. This link-local address is the same as the one generated by using the **ipv6 address auto link-local** command. If a link-local address is manually assigned to an interface, this manual assigned link-local address takes effect. If the manually assigned link-local address is deleted, the automatically generated link-local address takes effect.

Using the **undo ipv6 address auto link-local** command on an interface deletes only the link-local address generated by the **ipv6 address auto link-local** command. If the interface has an IPv6 global unicast address, it still has a link-local address. If the interface has no IPv6 global unicast address, it has no link-local address.

An interface can have only one link-local address. As a best practice, use the automatic generation method to avoid link-local address conflicts. If both the automatic generation and manual assignment methods are used, the manual assignment takes precedence.

- If you first use automatic generation and then manual assignment, the manually assigned link-local address overwrites the automatically generated one.
- If you first use manual assignment and then automatic generation, both of the following occur:
  - The link-local address is still the manually assigned one.
  - The automatically generated link-local address does not take effect. If you delete the manually assigned address, the automatically generated link-local address takes effect.

Configuring automatic generation of an IPv6 link-local address for an interface

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Configure the interface to automatically generate an IPv6 link-local address.
   ipv6 address auto link-local
   By default, no link-local address is configured on an interface.
   After an IPv6 global unicast address is configured on the interface, a link-local address is generated automatically.

Manually assigning an IPv6 link-local address to an interface

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Manually assign an IPv6 link-local address to the interface.
   ipv6 address *ipv6-address* **link-local**
   By default, no link-local address is configured on an interface.

Configuring an IPv6 anycast address

1. Enter system view.

system-view

2. Enter interface view.
   interface *interface-type interface-number*
3. Configure an IPv6 anycast address.
   ipv6 address { *ipv6-address prefix-length* | *ipv6-address*/*prefix-length* } **anycast**

   By default, no IPv6 anycast address is configured on an interface.

Configuring path MTU discovery

Setting the interface MTU for IPv6 packets

About interface MTU for IPv6 packets

If the size of a packet exceeds the MTU of the sending interface, the device discards the packet. If the device is an intermediate device, it also sends the source host an ICMPv6 Packet Too Big message with the MTU of the sending interface. The source host fragments the packets according to the MTU. To avoid this situation, set a proper interface MTU.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Set the interface MTU for IPv6 packets.
   ipv6 mtu *size*

   By default, no interface MTU is set.

Setting a static path MTU for an IPv6 address

About static path MTU for an IPv6 address

You can set a static path MTU for an IPv6 address. Before sending a packet to the IPv6 address, the device compares the output interface MTU with the static path MTU. If the packet size exceeds the smaller one of the two values, the device fragments the packet according to the smaller value. After sending the fragmented packets, the device dynamically finds the path MTU to a destination host (see "IPv6 path MTU discovery").

Procedure

1. Enter system view.
   system-view
2. Set a static path MTU for an IPv6 address.
   ipv6 pathmtu [**vpn-instance** *vpn-instance-name*] *ipv6-address value*

   By default, no path MTU is set for any IPv6 address.

Setting the aging time for dynamic path MTUs

About the aging time for dynamic path MTUs

After the device dynamically discovers the path MTU to a destination host (see "IPv6 path MTU discovery"), it performs the following operations:

- Sends packets to the destination host based on this path MTU.
- Starts the aging timer for this path MTU.

When the aging timer expires, the device removes the dynamic path MTU and discovers the path MTU again.

Restrictions and guidelines

The aging time is invalid for a static path MTU.

Procedure

1. Enter system view.
   system-view
2. Set the aging time for dynamic path MTUs.
   ipv6 pathmtu age *age-time*
   The default setting is 10 minutes.

Controlling sending ICMPv6 messages

Configuring the rate limit for ICMPv6 error messages

About the rate limit for ICMPv6 error messages

To avoid sending excessive ICMPv6 error messages within a short period that might cause network congestion, you can limit the rate at which ICMPv6 error messages are sent. A token bucket algorithm is used with one token representing one ICMPv6 error message.

A token is placed in the bucket at intervals until the maximum number of tokens that the bucket can hold is reached.

A token is removed from the bucket when an ICMPv6 error message is sent. When the bucket is empty, ICMPv6 error messages are not sent until a new token is placed in the bucket.

Procedure

1. Enter system view.
   system-view
2. Set the bucket size and the interval for tokens to arrive in the bucket for ICMPv6 error messages.
   ipv6 icmpv6 error-interval *interval* [*bucketsize*]
   By default, the bucket allows a maximum of 10 tokens. A token is placed in the bucket at an interval of 100 milliseconds.
   To disable the ICMPv6 rate limit, set the interval to 0 milliseconds.

Enabling replying to multicast echo requests

1. Enter system view.
   system-view

2. Enable replying to multicast echo requests.

   ipv6 icmpv6 multicast-echo-reply enable

   By default, this feature is disabled.

Enabling sending ICMPv6 destination unreachable messages

About sending ICMPv6 destination unreachable messages

The device sends the source the following ICMPv6 destination unreachable messages:

- **ICMPv6 No Route to Destination message**—A packet to be forwarded does not match any route.
- **ICMPv6 Communication with Destination Administratively Prohibited message**—An administrative prohibition is preventing successful communication with the destination. This is typically caused by a firewall or an ACL on the device.
- **ICMPv6 Beyond Scope of Source Address message**—The destination is beyond the scope of the source IPv6 address. For example, a packet's source IPv6 address is a link-local address, and its destination IPv6 address is a global unicast address.
- **ICMPv6 Address Unreachable message**—The device fails to resolve the link layer address for the destination IPv6 address of a packet.
- **ICMPv6 Port Unreachable message**—No port process on the destination device exists for a received UDP packet.

Restrictions and guidelines

An ICMPv6 destination unreachable message indicates that the destination is not reachable from the source device. Attackers can launch malicious attacks to make the device generate incorrect ICMPv6 destination unreachable messages, which will affect the function of the network. To protect the network from malicious attacks and decrease unnecessary network traffic, you can disable the sending of ICMPv6 destination unreachable messages.

Procedure

1. Enter system view.

   system-view

2. Enable sending ICMPv6 destination unreachable messages.

   ipv6 unreachables enable

   By default, this feature is disabled.

Enabling sending ICMPv6 time exceeded messages

About sending ICMPv6 time exceeded messages

The device sends the source ICMPv6 time exceeded messages as follows:

- If a received packet is not destined for the device and its hop limit is 1, the device sends an ICMPv6 hop limit exceeded in transit message to the source.
- Upon receiving the first fragment of an IPv6 datagram destined for the device, the device starts a timer. If the timer expires before all fragments arrive, the device sends an ICMPv6 fragment reassembly time exceeded message to the source.

Restrictions and guidelines

If the device receives large numbers of malicious packets, its performance degrades greatly because it must send back ICMP time exceeded messages. To prevent such attacks, disable sending ICMPv6 time exceeded messages.

Procedure

1. Enter system view.
   system-view
2. Enable sending ICMPv6 time exceeded messages.
   ipv6 hoplimit-expires enable
   By default, sending ICMPv6 time exceeded messages is enabled.

Enabling sending ICMPv6 redirect messages

About sending ICMPv6 redirect messages

Upon receiving a packet from a host, the device sends an ICMPv6 redirect message to inform the host of a better next hop when the following conditions are met:

- The interface receiving the packet is the interface forwarding the packet.
- The selected route is not created or modified by any ICMPv6 redirect messages.
- The selected route is not a default route.
- The forwarded packet does not contain the routing extension header.

The ICMPv6 redirect feature simplifies host management by enabling hosts that hold few routes to optimize their routing table gradually. However, to avoid adding too many routes on hosts, this feature is disabled by default.

Procedure

1. Enter system view.
   system-view
2. Enable sending ICMPv6 redirect messages.
   ipv6 redirects enable
   By default, sending ICMPv6 redirect messages is disabled.

Specifying the source address for ICMPv6 packets

About specifying the source address for ICMPv6 packets

Perform this task to specify the source IPv6 address for outgoing ping echo requests and ICMPv6 error messages. It is a good practice to specify the IPv6 address of the loopback interface as the source IPv6 address. This feature helps users to easily locate the sending device.

Restrictions and guidelines

If you specify an IPv6 address in the **ping** command, ping echo requests use the specified address as the source IPv6 address. If you do not specify an IPv6 address in the **ping** command, ping echo requests use the IPv6 address specified by the **ipv6 icmpv6 source** command.

Procedure

1. Enter system view.
   system-view
2. Specify an IPv6 address as the source address for outgoing ICMPv6 packets.
   ipv6 icmpv6 source [**vpn-instance** *vpn-instance-name*] *ipv6-address*
   By default, the device uses the IPv6 address of the sending interface as the source IPv6 address for outgoing ICMPv6 packets.

Enabling Layer 3 packet statistics counting

About enabling Layer 3 packet statistics counting

The Layer 3 packet statistics counting feature counts statistics about incoming and outgoing IPv6 packets on an interface. To display the collected statistics, execute the **display ipv6 statistics** command.

Restrictions and guidelines

When the interface is processing a large number of packets, the Layer 3 packet statistics counting will cause high CPU usage and degrade the forwarding performance. If the statistics are not necessary, disable this feature to ensure the device performance.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable Layer 3 packet statistics counting.
   statistics l3-packet enable [**inbound** | **outbound**]
   By default, Layer 3 packet statistics counting is disabled.

Enabling IPv6 local fragment reassembly

About IPv6 local fragment reassembly

Use this feature on a multichassis IRF fabric to improve fragment reassembly efficiency. This feature enables a subordinate to reassemble the IPv6 fragments of a packet if all the fragments arrive at it. If this feature is disabled, all IPv6 fragments are delivered to the master device for reassembly.

Restrictions and guidelines

The IPv6 local fragment reassembly feature applies only to fragments destined for the same subordinate.

Procedure

1. Enter system view.
   system-view
2. Enable IPv6 local fragment reassembly.
   ipv6 reassemble local enable
   By default, IPv6 local fragment reassembly is disabled.

Configuring IPv6 bandwidth-based load sharing

About IPv6 bandwidth-based load sharing

This feature shares IPv6 traffic among multiple output interfaces based on their expected load percentages. The device calculates the load percentage for each output interface in terms of the interface expected bandwidth.

For devices that run load sharing protocols such as Locator/ID Separation Protocol (LISP), they implement load sharing based on the ratios defined by these protocols.

Procedure

1. Enter system view.
   system-view
2. Enable IPv6 bandwidth-based load sharing.
   ipv6 bandwidth-based-sharing
   By default, IPv6 bandwidth-based load sharing is disabled.
3. Enter interface view.
   interface *interface-type interface-number*
4. Set the expected bandwidth of an interface.
   bandwidth *bandwidth*
   By default, the expected bandwidth of an interface equals the absolute bandwidth of the link.

Enabling discarding IPv6 packets that contain extension headers

About discarding IPv6 packets that contain extension headers

This feature enables a device to discard a received IPv6 packet in which the extension headers cannot be processed by the device.

Procedure

1. Enter system view
   system-view
2. Enable the device to discard IPv6 packets that contain extension headers.
   ipv6 extension-header drop enable
   By default, the device does not discard IPv6 packets that contain extension headers.

Display and maintenance commands for IPv6 basics

Execute **display** commands in any view and **reset** commands in user view.

For information about the **display tcp statistics**, **display udp statistics**, **reset tcp statistics**, and **reset udp statistics** command, see the IP performance commands in *Layer 3—IP Services Command Reference*.

Task	Command
Display IPv6 FIB entries.	**display ipv6 fib** [**vpn-instance** *vpn-instance-name*] [*ipv6-address* [*prefix-length*]]
Display ICMPv6 traffic statistics.	**display ipv6 icmp statistics** [**slot** *slot-number*]
Display IPv6 information about the	**display ipv6 interface** [*interface-type*

Task	Command
interface.	[*interface-number*]] [**brief**]
Display IPv6 prefix information about the interface.	**display ipv6 interface** *interface-type interface-number* **prefix**
Display the IPv6 path MTU information.	**display ipv6 pathmtu** [**vpn-instance** *vpn-instance-name*] { *ipv6-address* | { **all** | **dynamic** | **static** } [**count**] }
Display the IPv6 prefix information.	**display ipv6 prefix** [*prefix-number*]
Display brief information about IPv6 RawIP connections.	**display ipv6 rawip** [**slot** *slot-number*]
Display detailed information about IPv6 RawIP connections.	**display ipv6 rawip verbose** [**slot** *slot-number* [**pcb** *pcb-index*]]
Display IPv6 and ICMPv6 packet statistics.	**display ipv6 statistics** [**slot** *slot-number*]
Display brief information about IPv6 TCP connections.	**display ipv6 tcp** [**slot** *slot-number*]
Display detailed information about IPv6 TCP connections.	**display ipv6 tcp verbose** [**slot** *slot-number* [**pcb** *pcb-index*]]
Display brief information about IPv6 UDP connections.	**display ipv6 udp** [**slot** *slot-number*]
Display detailed information about IPv6 UDP connections.	**display ipv6 udp verbose** [**slot** *slot-number* [**pcb** *pcb-index*]]
Display IPv6 TCP traffic statistics.	**display tcp statistics** [**slot** *slot-number*]
Display IPv6 UDP traffic statistics.	**display udp statistics** [**slot** *slot-number*]
Clear path MTUs.	**reset ipv6 pathmtu** { **all** | **dynamic** | **static** }
Clear IPv6 and ICMPv6 packet statistics.	**reset ipv6 statistics** [**slot** *slot-number*]
Clear IPv6 TCP traffic statistics.	**reset tcp statistics**
Clear IPv6 UDP traffic statistics.	**reset udp statistics**

Basic IPv6 settings configuration examples

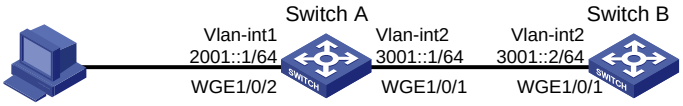
Example: Configuring basic IPv6 settings

Network configuration

As shown in Figure 5, a host, Switch A, and Switch B are connected through Ethernet ports. Add the Ethernet ports to corresponding VLANs. Configure IPv6 addresses for the VLAN interfaces and verify that they are connected. Switch B can reach the host.

Enable IPv6 on the host to automatically obtain an IPv6 address through IPv6 ND.

Figure 5 Network diagram



Procedure

This example assumes that the VLAN interfaces have been created on the switches.

1. Configure Switch A:

   # Specify a global unicast address for VLAN-interface 2.

   ```
   <SwitchA> system-view
   [SwitchA] interface vlan-interface 2
   [SwitchA-Vlan-interface2] ipv6 address 3001::1/64
   [SwitchA-Vlan-interface2] quit
   ```

   # Specify a global unicast address for VLAN-interface 1, and allow it to advertise RA messages (no interface advertises RA messages by default).

   ```
   [SwitchA] interface vlan-interface 1
   [SwitchA-Vlan-interface1] ipv6 address 2001::1/64
   [SwitchA-Vlan-interface1] undo ipv6 nd ra halt
   [SwitchA-Vlan-interface1] quit
   ```

2. Configure Switch B:

   # Configure a global unicast address for VLAN-interface 2.

   ```
   <SwitchB> system-view
   [SwitchB] interface vlan-interface 2
   [SwitchB-Vlan-interface2] ipv6 address 3001::2/64
   [SwitchB-Vlan-interface2] quit
   ```

   # Configure an IPv6 static route with destination IPv6 address 2001::/64 and next hop address 3001::1.

   ```
   [SwitchB] ipv6 route-static 2001:: 64 3001::1
   ```

3. Configure the host:

   Enable IPv6 for the host to automatically obtain an IPv6 address through IPv6 ND.

   # Display neighbor information for Twenty-FiveGigE 1/0/2 on Switch A.

   ```
   [SwitchA] display ipv6 neighbors interface twenty-fivegige 1/0/2
   Type: S-Static    D-Dynamic    O-Openflow     R-Rule    IS-Invalid static
   IPv6 address              MAC address    VLAN/VSI   Interface     State T  Aging
   FE80::215:E9FF:FEA6:7D14  0015-e9a6-7d14 1          WGE1/0/2      STALE D  1238
   2001::15B:E0EA:3524:E791  0015-e9a6-7d14 1          WGE1/0/2      STALE D  1248
   ```

   The output shows that the IPv6 global unicast address that Host obtained is 2001::15B:E0EA:3524:E791.

Verifying the configuration

Display the IPv6 interface settings on Switch A. All IPv6 global unicast addresses configured on the interface are displayed.

```
[SwitchA] display ipv6 interface vlan-interface 2
Vlan-interface2 current state: UP
Line protocol current state: UP
IPv6 is enabled, link-local address is FE80::20F:E2FF:FE00:2
  Global unicast address(es):
    3001::1, subnet is 3001::/64
```

```
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF00:2
  MTU is 1500 bytes
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
  InReceives:                   25829
  InTooShorts:                  0
  InTruncatedPkts:              0
  InHopLimitExceeds:            0
  InBadHeaders:                 0
  InBadOptions:                 0
  ReasmReqds:                   0
  ReasmOKs:                     0
  InFragDrops:                  0
  InFragTimeouts:               0
  OutFragFails:                 0
  InUnknownProtos:              0
  InDelivers:                   47
  OutRequests:                  89
  OutForwDatagrams:             48
  InNoRoutes:                   0
  InTooBigErrors:               0
  OutFragOKs:                   0
  OutFragCreates:               0
  InMcastPkts:                  6
  InMcastNotMembers:            25747
  OutMcastPkts:                 48
  InAddrErrors:                 0
  InDiscards:                   0
  OutDiscards:                  0
[SwitchA] display ipv6 interface vlan-interface 1
Vlan-interface1 current state: UP
Line protocol current state: UP
IPv6 is enabled, link-local address is FE80::20F:E2FF:FE00:1C0
  Global unicast address(es):
    2001::1, subnet is 2001::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:1
    FF02::1:FF00:1C0
  MTU is 1500 bytes
```

```
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  ND advertised reachable time is 0 milliseconds
  ND advertised retransmit interval is 0 milliseconds
  ND router advertisements are sent every 600 seconds
  ND router advertisements live for 1800 seconds
  Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
  InReceives:                   272
  InTooShorts:                  0
  InTruncatedPkts:              0
  InHopLimitExceeds:            0
  InBadHeaders:                 0
  InBadOptions:                 0
  ReasmReqds:                   0
  ReasmOKs:                     0
  InFragDrops:                  0
  InFragTimeouts:               0
  OutFragFails:                 0
  InUnknownProtos:              0
  InDelivers:                   159
  OutRequests:                  1012
  OutForwDatagrams:             35
  InNoRoutes:                   0
  InTooBigErrors:               0
  OutFragOKs:                   0
  OutFragCreates:               0
  InMcastPkts:                  79
  InMcastNotMembers:            65
  OutMcastPkts:                 938
  InAddrErrors:                 0
  InDiscards:                   0
  OutDiscards:                  0
```

Display the IPv6 interface settings on Switch B. All IPv6 global unicast addresses configured on the interface are displayed.

```
[SwitchB] display ipv6 interface vlan-interface 2
Vlan-interface2 current state :UP
Line protocol current state :UP
IPv6 is enabled, link-local address is FE80::20F:E2FF:FE00:1234
  Global unicast address(es):
    3001::2, subnet is 3001::/64
  Joined group address(es):
    FF02::1
    FF02::2
    FF02::1:FF00:2
    FF02::1:FF00:1234
  MTU is 1500 bytes
```

```
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND retransmit interval is 1000 milliseconds
  Hosts use stateless autoconfig for addresses
IPv6 Packet statistics:
  InReceives:                   117
  InTooShorts:                  0
  InTruncatedPkts:              0
  InHopLimitExceeds:            0
  InBadHeaders:                 0
  InBadOptions:                 0
  ReasmReqds:                   0
  ReasmOKs:                     0
  InFragDrops:                  0
  InFragTimeouts:               0
  OutFragFails:                 0
  InUnknownProtos:              0
  InDelivers:                   117
  OutRequests:                  83
  OutForwDatagrams:             0
  InNoRoutes:                   0
  InTooBigErrors:               0
  OutFragOKs:                   0
  OutFragCreates:               0
  InMcastPkts:                  28
  InMcastNotMembers:            0
  OutMcastPkts:                 7
  InAddrErrors:                 0
  InDiscards:                   0
  OutDiscards:                  0
```

Ping Switch A and Switch B on the host, and ping Switch A and the host on Switch B to verify that they are connected.

NOTE:

When you ping a link-local address, use the **-i** parameter to specify an interface for the link-local address.

```
[SwitchB] ping ipv6 -c 1 3001::1
Ping6(56 data bytes) 3001::2 --> 3001::1, press CTRL_C to break
56 bytes from 3001::1, icmp_seq=0 hlim=64 time=4.404 ms

--- Ping6 statistics for 3001::1 ---
1 packet(s) transmitted, 1 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 4.404/4.404/4.404/0.000 ms
[SwitchB] ping ipv6 -c 1 2001::15B:E0EA:3524:E791
Ping6(56 data bytes) 3001::2 --> 2001::15B:E0EA:3524:E791, press CTRL_C to break
56 bytes from 2001::15B:E0EA:3524:E791, icmp_seq=0 hlim=64 time=5.404 ms

--- Ping6 statistics for 2001::15B:E0EA:3524:E791 ---
```

```
1 packet(s) transmitted, 1 packet(s) received, 0.0% packet loss
round-trip min/avg/max/std-dev = 5.404/5.404/5.404/0.000 ms
```

The output shows that Switch B can ping Switch A and the host. The host can also ping Switch B and Switch A.

Configuring IPv6 neighbor discovery

About IPv6 neighbor discovery

ICMPv6 messages used by IPv6 neighbor discovery

The IPv6 neighbor discovery (ND) process uses ICMP messages for address resolution, neighbor reachability verification, and neighboring device tracking.

Table 3 describes the ICMPv6 messages used by the IPv6 ND protocol.

Table 3 ICMPv6 messages used by ND

ICMPv6 message	Type	Function
Neighbor Solicitation (NS)	135	Acquires the link-layer address of a neighbor on the local link.
		Verifies the reachability of a neighbor.
		Detects duplicate addresses.
Neighbor Advertisement (NA)	136	Responds to an NS message.
		Notifies the neighboring nodes of link layer changes.
Router Solicitation (RS)	133	Requests an address prefix and other configuration information for autoconfiguration after startup.
Router Advertisement (RA)	134	Responds to an RS message.
		Advertises information, such as the Prefix Information options and flag bits.
Redirect	137	Informs the source host of a better next hop on the path to a particular destination when certain conditions are met.

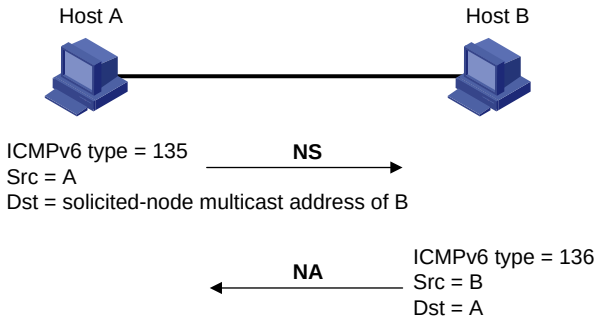
Address resolution

This function is similar to ARP in IPv4. An IPv6 node acquires the link-layer addresses of neighboring nodes on the same link through NS and NA messages.

Figure 6 shows how Host A acquires the link-layer address of Host B on the same link. The address resolution procedure is as follows:

1. Host A multicasts an NS message. The source address of the NS message is the IPv6 address of the sending interface of Host A. The destination address is the solicited-node multicast address of Host B. The NS message body contains the link-layer address of Host A and the target IPv6 address.
2. After receiving the NS message, Host B determines whether the target address of the packet is its IPv6 address. If it is, Host B learns the link-layer address of Host A, and then unicasts an NA message containing its link-layer address.
3. Host A acquires the link-layer address of Host B from the NA message.

Figure 6 Address resolution



Neighbor reachability detection

After Host A acquires the link-layer address of its neighbor Host B, Host A can use NS and NA messages to test the reachability of Host B as follows:

1. Host A sends an NS message whose destination address is the IPv6 address of Host B.
2. If Host A receives an NA message from Host B, Host A decides that Host B is reachable. Otherwise, Host B is unreachable.

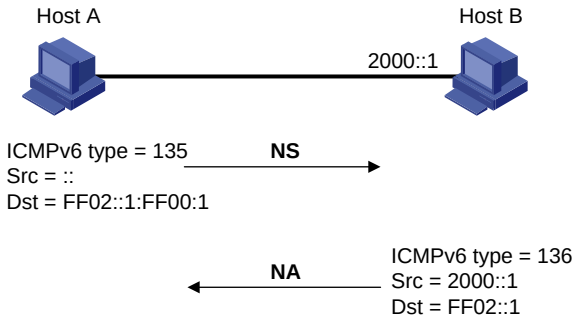
Duplicate address detection

After Host A acquires an IPv6 address, it performs Duplicate Address Detection (DAD) to check whether the address is being used by any other node. This is similar to gratuitous ARP in IPv4. DAD is accomplished through NS and NA messages.

The DAD procedure is as follows:

1. Host A sends an NS message. The source address is the unspecified address and the destination address is the corresponding solicited-node multicast address of the IPv6 address to be detected. The NS message body contains the detected IPv6 address.
2. If Host B uses this IPv6 address, Host B returns an NA message that contains its IPv6 address.
3. Host A knows that the IPv6 address is being used by Host B after receiving the NA message from Host B. If receiving no NA message, Host A decides that the IPv6 address is not in use and uses this address.

Figure 7 Duplicate address detection



Router/prefix discovery and stateless address autoconfiguration

Router/prefix discovery allows an IPv6 node to find the neighboring routers and learn the prefix and network configuration parameters of the network from receiving RA messages.

Stateless address autoconfiguration allows an IPv6 node to automatically generate an IPv6 address based on the information learned through router/prefix discovery.

A node performs router/prefix discovery and stateless address autoconfiguration as follows:

1. At startup, a node sends an RS message to request configuration information from a router.
2. The router returns an RA message containing the Prefix Information option and other configuration information. (The router also periodically sends an RA message.)
3. The node automatically generates an IPv6 address and other configuration parameters according to the configuration information in the RA message.

The Prefix Information option contains an address prefix and the preferred lifetime and valid lifetime of the address prefix. A node updates the preferred lifetime and valid lifetime upon receiving a periodic RA message.

The generated IPv6 address is valid within the valid lifetime and becomes invalid when the valid lifetime expires.

After the preferred lifetime expires, the node cannot use the generated IPv6 address to establish new connections, but can receive packets destined for the IPv6 address. The preferred lifetime cannot be greater than the valid lifetime.

Redirection

Upon receiving a packet from a host, the gateway sends an ICMPv6 redirect message to inform the host of a better next hop when the following conditions are met:

- The interface receiving the packet is the same as the interface forwarding the packet.
- The selected route is not created or modified by an ICMPv6 redirect message.
- The selected route is not a default route on the device.
- The forwarded IPv6 packet does not contain the routing extension header.

Protocols and standards

- RFC 4861, *Neighbor Discovery for IP Version 6 (IPv6)*
- RFC 4862, *IPv6 Stateless Address Autoconfiguration*
- RFC 8106, *IPv6 Router Advertisement Options for DNS Configuration*

IPv6 neighbor discovery tasks at a glance

All IPv6 neighbor discovery tasks are optional.

- Configuring ND entry related features
  - Configuring a static neighbor entry
  - Setting the dynamic neighbor learning limit on an interface
  - Enabling unsolicited NA learning
  - Setting the aging timer for ND entries in stale state
  - Minimizing link-local ND entries

Configuring a static neighbor entry

About static neighbor entries

A neighbor entry stores information about a link-local node. The entry can be created dynamically through NS and NA messages, or configured statically.

The device uniquely identifies a static neighbor entry by using the neighbor's IPv6 address and the number of the Layer 3 interface that connects to the neighbor. You can configure a static neighbor entry by using one of the following methods:

- **Method 1**—Associate a neighbor's IPv6 address and link-layer address with the local Layer 3 interface.
- **Method 2**—Associate a neighbor's IPv6 address and link-layer address with a Layer 2 port in a VLAN.
- **Method 3**—Specify a neighbor IPv6 address, MAC address, input interface (VSI interface), output interface (tunnel interface), and VSI name.
- **Method 4**—Specify a neighbor IPv6 address, MAC address, input interface (VSI interface), output interface (determined by a Layer 2 interface and Ethernet service instance), and VSI name.

Method 3 and Method 4 are supported in Release 6555P02 and later.

Restrictions and guidelines

To configure a static neighbor entry for a VLAN interface, use Method 1 or Method 2.

- If you use Method 1, the device is required to resolve the Layer 2 port in the related VLAN.
- If you use Method 2, make sure the Layer 2 port belongs to the specified VLAN and the corresponding VLAN interface already exists. After the configuration, the device associates the VLAN interface with the neighbor IPv6 address to identify the static neighbor entry.

If the device and its neighbor are connected through a VSI interface, use Method 3 or Method 4 to configure the neighbor entry.

- If Method 3 is used, the neighbor entry is in REACH state. This method is applicable to the network where VXLAN gateways are connected through tunnel interfaces. In the network, a VXLAN gateway is identified by both the VSI and VSI interface. A VSI interface is associated with multiple tunnel interfaces. To create a neighbor entry, you must specify the VSI interface, VSI, and tunnel interface.
- If Method 4 is used, the neighbor entry is in REACH state. This method is applicable to the network where VXLAN gateways are associated with local sites. A VXLAN gateway is identified by both the VSI and VSI interface. One VXLAN gateway might have multiple local sites. Local sites access the VXLAN network through Layer 2 interfaces where Ethernet service instance

and VSI mappings are configured. To create a neighbor entry, you must specify the VSI interface, Layer 2 interface connected to the local site, Ethernet service instance, and VSI.

For more information about VSI, VSI interfaces, and Ethernet service instances, see VXLAN overview in *VXLAN Configuration Guide*.

For more information about tunnel interfaces, see tunneling configuration in *Layer 3—IP Services Configuration Guide*.

To delete a static neighbor entry for a VSI interface, specify only the VSI interface.

To delete a static neighbor entry for a VLAN interface, specify only the VLAN interface.

You can use the **undo ipv6 neighbor** command to delete both static and dynamic neighbor entries.

Procedure

1. Enter system view.

   system-view

2. Configure a static neighbor entry.

   ipv6 neighbor *ipv6-address mac-address* { *vlan-id port-type port-number* | **interface** *interface-type interface-number* | **vsi-interface** *vsi-interface-id* **tunnel** *number* **vsi** *vsi-name* | **vsi-interface** *vsi-interface-id interface-type interface-number* **service-instance** *instance-id* **vsi** *vsi-name* } [**vpn-instance** *vpn-instance-name*]

   By default, no static neighbor entries exist.

Setting the dynamic neighbor learning limit on an interface

About the dynamic neighbor learning limit on an interface

The device can dynamically acquire the link-layer address of a neighboring node through NS and NA messages and add it into the neighbor table. When the number of dynamic neighbor entries reaches the limit, the interface stops learning neighbor information.

This feature limits the neighbor table size. A large neighbor table will degrade the forwarding performance.

Procedure

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Set the dynamic neighbor learning limit on the interface.

   ipv6 neighbors max-learning-num *max-number*

   By default, the maximum number of dynamic neighbor entries that an interface can learn equals the ND table capacity of the device. You can use the **hardware-resource switch-mode** command to set the ND table capacity. For information about the **hardware-resource switch-mode** command, see device management commands in *Fundamentals Command Reference*.

Enabling unsolicited NA learning

IMPORTANT:

This feature is available in Release 6555P02 and later.

About this task

On some networks, a server multicasts NA messages to two peer devices for link backup. The peer devices cannot learn ND entry for the server from these NA messages by default. If no ND learning is triggered by data exchange between the server and peer devices, the peer devices learn the entry for the server only when the server unicasts messages to them.

This feature enables an interface to learn ND entries from unsolicited NA messages. The ND entries generated by using this method are in stale state.

Restrictions and guidelines

To ensure that the device learns ND entries from trusted NA messages, enable this feature only on a secure network.

This feature might cause the device to learn excessive ND entries that consume too many system resources. As a best practice, execute the **ipv6 neighbor stale-aging** command to set a smaller aging timer before you enable this feature. The smaller aging timer accelerates the aging of ND entries in stale state.

Procedure

1. Enter system view.

   system-view
2. Enter Layer 3 interface view.

   interface *interface-type interface-number*
3. Enable unsolicited NA learning.

   ipv6 nd unsolicited-na-learning enable

   By default, unsolicited NA learning is disabled.

Setting the aging timer for ND entries in stale state

About the aging timer for ND entries in stale state

ND entries in stale state have an aging timer. If an ND entry in stale state is not refreshed before the timer expires, the ND entry changes to the delay state. If it is still not refreshed in 5 seconds, the ND entry changes to the probe state, and the device sends an NS message three times. If no response is received, the device deletes the ND entry.

Restrictions and guidelines

You can set the aging timer for ND entries in stale state in system view and interface view. For ND entries in stale state on an interface, the aging timer in interface view has higher priority than the aging timer in system view.

Procedure

1. Enter system view.

   system-view
2. Set the aging timer for ND entries in stale state.
   - Set the aging timer for ND entries in stale state in system view.

     ipv6 neighbor stale-aging { *aging-minutes* | **second** *aging-seconds* }

The default setting is 240 minutes.

- Execute the following commands in sequence to set the aging timer in interface view.

  interface *interface-type interface-number*

  ipv6 neighbor timer stale-aging { *aging-minutes* | **second** *aging-seconds* }

  By default, the aging timer of ND entries in stale state is not configured on an interface. The aging timer is determined by the configuration of the **ipv6 neighbor stale-aging** command in system view.

Minimizing link-local ND entries

About minimizing link-local ND entries

Perform this task to minimize link-local ND entries assigned to the hardware. Link-local ND entries refer to ND entries that contain link-local addresses.

By default, the device assigns all ND entries to the hardware. With this feature enabled, the newly learned link-local ND entries are not assigned to the hardware if the link-local addresses of the entries are not the next hops of any routes. This feature saves hardware resources.

This feature takes effect only on newly learned link-local ND entries.

Procedure

1. Enter system view.

   system-view

2. Minimize link-local ND entries.

   ipv6 neighbor link-local minimize

   By default, the device assigns all ND entries to the hardware.

Setting the hop limit

About hop limit

You can set the hop limit value to fill in the Hop Limit field for IPv6 packets to be sent.

Procedure

1. Enter system view.

   system-view

2. Set the value for the Hop Limit field in the IP header.

   ipv6 hop-limit *value*

   The default setting is 64.

Configuring RA message sending and parameters

About RA message parameters

You can enable an interface to send RA messages, and configure the interval for sending RA messages and parameters in RA messages. After receiving an RA message, a host can use these parameters to perform corresponding operations. Table 4 describes the configurable parameters in an RA message.

Table 4 Parameters in an RA message and their descriptions

Parameter	Description
Hop Limit	Maximum number of hops in RA messages. A host receiving the RA message fills the value in the Hop Limit field of sent IPv6 packets.
Prefix information	After receiving the prefix information, the hosts on the same link can perform stateless autoconfiguration.
MTU	Guarantees that all nodes on the link use the same MTU.
Boot file URL	Specifies the URL address for downloading the boot file in RA messages. The device can use the ND protocol to obtain both the IPv6 address and the boot file URL for automatic configuration instead of using DHCPv6.
M flag	Determines whether a host uses stateful autoconfiguration to obtain an IPv6 address. If the M flag is set to 1, the host uses stateful autoconfiguration (for example, from a DHCPv6 server) to obtain an IPv6 address. Otherwise, the host uses stateless autoconfiguration to generate an IPv6 address according to its link-layer address and the prefix information in the RA message.
O flag	Determines whether a host uses stateful autoconfiguration to obtain configuration information other than the IPv6 address. If the O flag is set to 1, the host uses stateful autoconfiguration (for example, from a DHCPv6 server) to obtain configuration information other than the IPv6 address. Otherwise, the host uses stateless autoconfiguration.
Router Lifetime	Tells the receiving hosts how long the advertising router can live. If the lifetime of a router is 0, the router cannot be used as the default gateway.
Retrans Timer	If the device does not receive a response message within the specified time after sending an NS message, it retransmits the NS message.
Reachable Time	If the neighbor reachability detection shows that a neighbor is reachable, the device considers the neighbor reachable within the specified reachable time. If the device needs to send a packet to the neighbor after the specified reachable time expires, the device reconfirms whether the neighbor is reachable.
Router Preference	Specifies the router preference in an RA message. A host selects a router as the default gateway according to the router preference. If router preferences are the same, the host selects the router from which the first RA message is received.
DNS server option	DNS server information for IPv6 hosts. Hosts can obtain DNS server information from received RA messages instead of using DHCPv6.
DNS suffix information in DNS Search List (DNSSL) option	DNS suffix information for IPv6 hosts. Hosts can obtain DNS suffix information from received RA messages instead of using DHCPv6.

Restrictions and guidelines

The maximum interval for sending RA messages should be less than (or equal to) the router lifetime in RA messages. In this way, the router can be updated by an RA message before expiration.

The values of the NS retransmission timer and the reachable time configured for an interface are sent in RA messages to hosts. This interface sends NS messages at the interval of the NS retransmission timer and considers a neighbor reachable within the reachable time.

Enabling the sending of RA messages

1. Enter system view.

system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Enable the sending of RA messages.

   undo ipv6 nd ra halt

   The default setting is disabled.

4. Set the maximum and minimum intervals for sending RA messages.

   ipv6 nd ra interval *max-interval min-interval*

   By default, the maximum interval for sending RA messages is 600 seconds, and the minimum interval is 200 seconds.

   The device sends RA messages at random intervals between the maximum interval and the minimum interval.

   The minimum interval should be less than or equal to 0.75 times the maximum interval.

Configuring parameters for RA messages

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Configure the prefix information in RA messages.

   ipv6 nd ra prefix { *ipv6-prefix prefix-length* | *ipv6-prefix*/*prefix-length* } [*valid-lifetime preferred-lifetime* [**no-autoconfig** | **off-link**] * | **no-advertise**]

   By default, no prefix information is configured for RA messages, and the IPv6 address of the interface sending RA messages is used as the prefix information. If the IPv6 address is manually configured, the prefix uses a fixed valid lifetime of 2592000 seconds (30 days) and a preferred lifetime of 604800 seconds (7 days). If the IPv6 address is automatically obtained, the prefix uses the valid lifetime and preferred lifetime configured for the IPv6 address.

4. Configure the default settings for prefixes advertised in RA messages.

   ipv6 nd ra prefix default [*valid-lifetime preferred-lifetime* [**no-autoconfig** | **off-link**] * | **no-advertise**]

   By default, no default settings are configured for prefixes advertised in RA messages.

5. Turn off the MTU option in RA messages.

   ipv6 nd ra no-advlinkmtu

   By default, RA messages contain the MTU option.

6. Specify unlimited hops in RA messages.

   ipv6 nd ra hop-limit unspecified

   By default, the maximum number of hops in RA messages is 64.

7. Specify the URL of the boot file in RA messages.

   ipv6 nd ra boot-file-url *url-string*

   By default, RA messages do not contain the URL of the boot file.

8. Set the M flag bit to 1.

   ipv6 nd autoconfig managed-address-flag

   By default, the M flag bit is set to 0 in RA advertisements. Hosts receiving the advertisements will obtain IPv6 addresses through stateless autoconfiguration.

9. Set the O flag bit to 1.

   ipv6 nd autoconfig other-flag

   By default, the O flag bit is set to 0 in RA advertisements. Hosts receiving the advertisements will acquire other configuration information through stateless autoconfiguration.

10. Set the router lifetime in RA messages.

    ipv6 nd ra router-lifetime *time*

    By default, the router lifetime is 1800 seconds.

11. Set the NS retransmission timer.

    ipv6 nd ns retrans-timer *value*

    By default, an interface sends NS messages every 1000 milliseconds, and the value of the Retrans Timer field in RA messages is 0.

12. Set the router preference in RA messages.

    ipv6 nd router-preference { **high** | **low** | **medium** }

    By default, the router preference is medium.

13. Set the reachable time.

    ipv6 nd nud reachable-time *time*

    By default, the neighbor reachable time is 30000 milliseconds, and the value of the Reachable Time field in sent RA messages is 0.

Specifying DNS server information in RA messages

About specifying DNS server information in RA messages

The DNS server options in RA messages provide DNS server information for IPv6 hosts. The RA messages allow hosts to obtain their IPv6 addresses and the DNS server through stateless autoconfiguration. This method is useful in a network where DHCPv6 infrastructure is not provided.

One DNS server option contains one DNS server. All DNS server options are sorted in ascending order of the DNS server sequence number.

After you execute the **ipv6 nd ra dns server** command, the device immediately sends an RA message with the existing and newly specified DNS server information.

After you execute the **undo ipv6 nd ra dns server** command, the device immediately sends two RA messages.

- The first RA message contains information about all DNS servers, including the DNS servers specified in the **undo** command with their lifetime set to 0 seconds.
- The second RA message contains information about remaining DNS servers.

Each time the device sends an RA message from an interface, it immediately refreshes the RA message advertisement interval for that interface.

Restrictions and guidelines

You can configure a maximum of eight DNS servers on an interface.

The default lifetime of a DNS server is three times the maximum interval for advertising RA messages. To set the maximum interval, use the **ipv6 nd ra interval** command.

Procedure

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Specify DNS server information to be advertised in RA messages.

   ipv6 nd ra dns server *ipv6-address* [*seconds* | **infinite**] **sequence** *seqno*

   By default, no DNS server information is specified and RA messages do not contain DNS server options.

Specifying DNS suffix information in RA messages

About specifying DNS suffix information in RA messages

The DNSSL option in RA messages provides suffix information for IPv6 hosts. The RA messages allow hosts to obtain their IPv6 addresses and the DNS suffix through stateless autoconfiguration. This method is useful in a network where DHCPv6 infrastructure is not provided.

One DNSSL option contains one DNS suffix. All DNSSL options are sorted in ascending order of the sequence number of the DNS suffix.

After you execute the **ipv6 nd ra dns search-list** command, the device immediately sends an RA message with the existing and newly specified DNS suffix information.

After you execute the **undo ipv6 nd ra dns search-list** command, the device immediately sends two RA messages.

- The first RA message contains information about all DNS suffixes, including DNS suffixes specified in the **undo** command with their lifetime set to 0 seconds.
- The second RA message contains information about remaining DNS suffixes.

Each time the device sends an RA message from an interface, it immediately refreshes the RA message advertisement interval for that interface.

Restrictions and guidelines

You can configure a maximum of eight DNS suffixes on an interface.

The default lifetime of a DNS suffix is three times the maximum interval for advertising RA messages. To set the maximum interval, use the **ipv6 nd ra interval** command.

Procedure

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Specify DNS suffix information to be advertised in RA messages.

   ipv6 nd ra dns search-list *domain-name* [*seconds* | **infinite**] **sequence** *seqno*

   By default, no DNS suffix information is specified and RA messages do not contain DNS suffix options.

Suppressing advertising DNS information in RA messages

About suppressing advertising DNS information in RA messages

Perform this task to suppress the device from advertising information about DNS server addresses and DNS suffixes in RA messages.

Whether enabling this feature on an interface will trigger sending RA message immediately depends on the interface configuration:

- If the interface has DNS server information configured, the device immediately sends two RA messages. In the first message, the lifetime for DNS server addresses is 0 seconds. The second RA message does not contain any DNS server options.
- If the interface has no DNS server information specified, no RA messages are triggered.

If you specify a new DNS server or remove a DNS server on the interface after enabling DNS server suppression, the device immediately sends an RA message without any DNS server options.

Whether disabling this feature on an interface will trigger sending RA message immediately depends on the interface configuration:

- If the interface has DNS server information configured, the device immediately sends an RA message containing the DNS server information.
- If the interface has no DNS server information specified, no RA messages are triggered.

Each time the device sends an RA message from an interface, it immediately refreshes the RA message advertisement interval for that interface.

The same suppression mechanism applies when you enable or disable DNS suffix suppression in RA messages.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable DNS server suppression in RA messages.
   ipv6 nd ra dns server suppress
   By default, DNS server suppression in RA messages is disabled.
4. Enable DNS suffix suppression in RA messages.
   ipv6 nd ra dns search-list suppress
   By default, DNS suffix suppression in RA messages is disabled.

Setting the maximum number of attempts to send an NS message for DAD

About the maximum number of attempts to send an NS message for DAD

An interface sends an NS message for DAD for an obtained IPv6 address. The interface resends the NS message if it does not receive a response within the time specified by the **ipv6 nd ns retrans-timer** command. If the interface receives no response after making the maximum attempts specified by the **ipv6 nd dad attempts** command, the interface uses the IPv6 address.

Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Set the number of attempts to send an NS message for DAD.
   ipv6 nd dad attempts *times*
   The default setting is 1. When the *times* argument is set to 0, DAD is disabled.

Configuring ND snooping in a VLAN

About ND snooping in a VLAN

The ND snooping feature is used in Layer 2 switching networks. It learns the source MAC addresses, source IPv6 addresses, input interfaces, and VLANs of arriving ND messages and data packets to build the ND snooping table.

ND snooping entries can be used by ND detection to prevent spoofing attacks.

ND detection processes the ND messages received on ND trusted and untrusted interfaces as follows:

- ND detection forwards all ND messages received on an ND trusted interface.
- ND detection compares all ND messages received on an ND untrusted interface with the ND snooping entries except for RA and redirect messages.

You can use the **ipv6 nd detection trust** command to specify a Layer 2 Ethernet or aggregate port as an ND trusted interface. For more information about the **ipv6 nd detection trust** command, see *Security Command Reference*.

ND snooping entries can be used by IPv6 source guard to prevent spoofing attacks. For more information about IPv6 source guard, see *Security Configuration Guide*.

ND snooping provides device liveness tracking so that the ND snooping table can be updated in a timely manner. After ND snooping is enabled for a VLAN, the device uses the following mechanisms to create, update, and delete ND snooping entries. The following example uses ND messages for illustration.

Creation of ND snooping entries

Upon receiving an ND message or data packet from an unknown source, the device creates an ND snooping entry in INVALID status and performs DAD for the source IPv6 address. The device sends NS messages out of the ND trusted interfaces in the receiving VLAN twice. The sending interval is set by the **ipv6 nd snooping dad retrans-timer** command.

- If the device does not receive an NA message within the invalid entry lifetime (set by the **ipv6 nd snooping lifetime invalid** command), the entry becomes valid.
- If the device receives an NA message within the invalid entry lifetime, it deletes this entry.

Updating of ND snooping entries

When the ND untrusted interface that receives an ND message is different from that in the entry for an IPv6 address, the device performs DAD for the entry. It sends NS messages twice. The sending interval is set by the **ipv6 nd snooping dad retrans-timer** command.

- If the device does not receive an NA message within the invalid entry lifetime, it updates the entry with the new receiving interface.
- If the device receives an NA message within the invalid entry lifetime, the ND snooping entry remains unchanged.

Deletion of ND snooping entries

- When an ND trusted interface in the VLAN receives an ND message from the IPv6 address in a learned ND snooping entry, it performs DAD for the entry. The device sends NS messages twice. The sending interval is set by the **ipv6 nd snooping dad retrans-timer** command.
  - If the device does not receive an NA message within the invalid entry lifetime, it deletes the entry.
  - If the device receives an NA message within the invalid entry lifetime, the ND snooping entry remains unchanged.

- If an ND snooping entry has no matching ND messages within the valid entry lifetime (set by the **ipv6 nd snooping lifetime valid** command), the entry becomes invalid. The device then performs DAD for the entry by sending NS messages out of the interface in the entry twice. The sending interval is set by the **ipv6 nd snooping dad retrans-timer** command.
  - If the device does not receive an NA message within the invalid entry lifetime, it deletes the entry.
  - If the device receives an NA message within the invalid entry lifetime, the ND snooping entry remains unchanged and becomes valid.

Procedure

1. Enter system view.

   system-view
2. Enter VLAN view.

   vlan *vlan-id*
3. Enable ND snooping for IPv6 addresses. Choose the options to configure as needed:
   - Enable ND snooping for global unicast addresses.

     ipv6 nd snooping enable global
   - Enable ND snooping for link-local addresses.

     ipv6 nd snooping enable link-local

   By default, ND snooping is disabled for IPv6 global unicast addresses and link-local addresses.
4. (Optional.) Enable ND snooping for data packets from unknown sources.

   ipv6 nd snooping glean source

   By default, ND snooping is disabled for data packets from unknown sources.

   Before executing this command for a VLAN, you must configure IPv6 source guard on all untrusted interfaces in the same VLAN. This operation ensures correct forwarding of the data packets received by all these interfaces.
5. Return to system view.

   quit
6. Enter Layer 2 Ethernet or Layer 2 aggregate interface view.

   interface *interface-type interface-number*
7. (Optional.) Set the maximum number of ND snooping entries that an interface can learn.

   ipv6 nd snooping max-learning-num *max-number*

   By default, an interface can learn a maximum of 8192 ND snooping entries.
8. (Optional.) Configure the port as an ND snooping uplink port. The ND snooping uplink port cannot learn ND snooping entries.

   ipv6 nd snooping uplink

   By default, the port is not an ND snooping uplink port. After ND snooping is enabled, the port can learn ND snooping entries.
9. Return to system view.

   quit
10. (Optional.) Set timeout timers for ND snooping entries.

    ipv6 nd snooping lifetime { **invalid** *invalid-lifetime* | **valid** *valid-lifetime* }

    The default settings are as follows:
    - The timeout timer for ND snooping entries in INVALID status (TENTATIVE, TESTING_TPLT, or TESTING_VP) is 500 milliseconds.

- The timeout timer for ND snooping entries in VALID status is 300 seconds.

11. (Optional.) Set the interval for retransmitting an NS message for DAD.
    ipv6 nd snooping dad retrans-timer *interval*
    The default value is 250 milliseconds.

Configuring ND snooping in a VXLAN

About ND snooping in a VXLAN

The ND snooping feature in a VXLAN learns the source MAC addresses, source IPv6 addresses, VSI name, and link IDs to build the ND snooping table. For more information about VXLAN, see VXLAN overview in *VXLAN Configuration Guide*.

ND snooping entries can be used by ND detection to prevent spoofing attacks. For more information about the ND attack detection, see *Security Configuration Guide*.

ND snooping entries can be used by IPv6 source guard to prevent spoofing attacks. For more information about IPv6 source guard, see *Security Configuration Guide*.

After ND snooping is enabled in a VSI, the device uses the following mechanisms to create and maintain ND snooping entries.

Creation of ND snooping entries

With this feature enabled in a VXLAN, all receiving ND messages are delivered to the CPU. The CPU obtains the source IPv6 addresses, source MAC addresses, VSI name, and link IDs to create ND snooping entries.

Aging of ND snooping entries

The aging time of an ND snooping entry is 5 minutes. If no matching packet is received within 30 seconds before the end of the entry aging time, the entry is set to TENTATIVE status. At the same time, the device sends a NS message for the entry. If the source IP and source MAC in the received ND message matches the ND snooping entry, the device sets the entry to VALID and resets the aging time.

When the aging time of an ND snooping entry is reached, the entry is deleted.

Procedure

1. Enter system view.
   system-view
2. Create a VSI and enter its view.
   vsi *vsi-name*
3. Enable ND snooping for IPv6 addresses. Choose the options to configure as needed:
   - Enable ND snooping for global unicast addresses.
     ipv6 nd snooping enable global
   - Enable ND snooping for link-local addresses.
     ipv6 nd snooping enable link-local

   By default, ND snooping is disabled for IPv6 global unicast addresses and link-local addresses.

Enabling ND proxy

About ND proxy

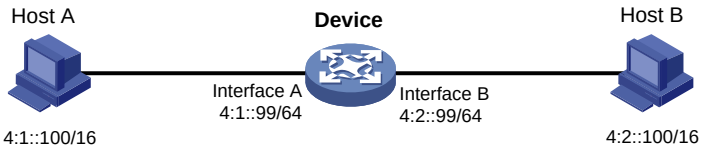
ND proxy enables a device to answer an NS message requesting the hardware address of a host on another network. With ND proxy, hosts in different broadcast domains can communicate with each other as they would on the same network.

ND proxy includes common ND proxy and local ND proxy.

Common ND proxy

As shown in Figure 8, Interface A with IPv6 address 4:1::99/64 and Interface B with IPv6 address 4:2::99/64 belong to different subnets. Host A and Host B reside on the same network but in different broadcast domains.

Figure 8 Application environment of ND proxy



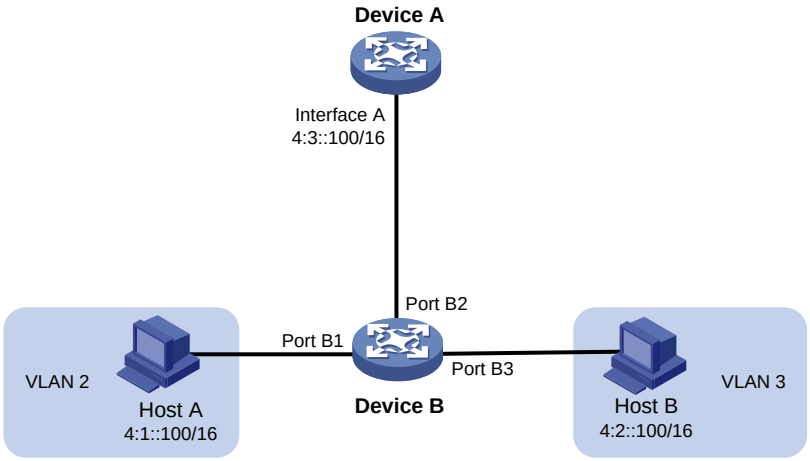
Because Host A's IPv6 address is on the same subnet as Host B's, Host A directly sends an NS message to obtain Host B's MAC address. However, Host B cannot receive the NS message because they belong to different broadcast domains.

To solve this problem, enable common ND proxy on Interface A and Interface B of the device. The device replies to the NS message from Host A, and forwards packets from other hosts to Host B.

Local ND proxy

As shown in Figure 9, Host A belongs to VLAN 2 and Host B belongs to VLAN 3. Host A and Host B connect to Port B1 and Port B3, respectively.

Figure 9 Application environment of local ND proxy



Because Host A's IPv6 address is on the same subnet as Host B's, Host A directly sends an NS message to obtain Host B's MAC address. However, Host B cannot receive the NS message because they belong to different VLANs.

To solve this problem, enable local ND proxy on Interface A of Device A so that Device A can forward messages between Host A and Host B.

Local ND proxy implements Layer 3 communication for two hosts in the following cases:

- The two hosts connect to ports of the same device and the ports must be in different VLANs.
- The two hosts connect to isolated Layer 2 ports in the same isolation group of a VLAN.
- If super VLAN is used, the two hosts must belong to different sub VLANs.
- If Private VLAN is used, the two hosts must belong to different secondary VLANs.

Enabling common ND proxy

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable common ND proxy.
   proxy-nd enable
   By default, common ND proxy is disabled.

Enabling local ND proxy

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable local ND proxy.
   local-proxy-nd enable
   By default, local ND proxy is disabled.

Configuring IPv6 ND direct route advertisement

About IPv6 ND direct route advertisement

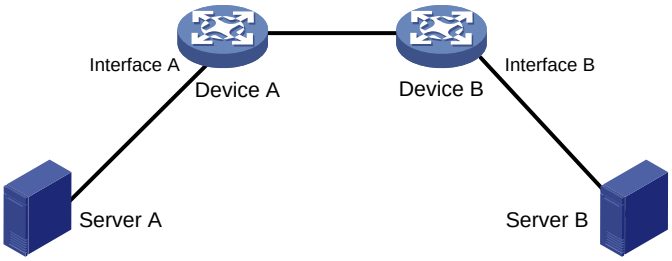
This feature generates 128-bit host routes based on ND entries for packet forwarding and route advertisement.

After you enable this feature, the routing table might be populated with excessive host routes. To reduce the routing table size, execute the **ipv6 nd route-direct prefix convert-length** command for the device to generate network routes for identified ND entries instead of host routes.

Application in Layer 3 access networks

As shown in Figure 10, ND direct route advertisement is enabled on Interface A and Interface B. This feature generates a host route to Server A and a host route to Server B for the routing protocols to advertise. So each device forwards only the traffic destined to the server within the network, which saves bandwidth.

Figure 10 Application in a Layer 3 access network



Procedure

1. Enter system view.
   system-view
2. Enter interface view.
   interface *interface-type interface-number*
3. Enable ND direct route advertisement.
   ipv6 nd route-direct advertise
   By default, ND direct route advertisement is disabled.
4. (Optional.) Specify a prefix length for generating a network route for identified ND entries.
   ipv6 nd route-direct prefix *ipv6-prefix prefix-length* **convert-length** *convert-length*
   By default, no prefix length is specified for generating a network route for identified ND entries.

Enabling recording user IPv6 address conflicts

About recording user IPv6 address conflicts

This feature detects and records user IPv6 address conflicts. A conflict occurs if an incoming NA packet has the same source IP address as an existing ND entry but a different source MAC address. The device generates a user IPv6 address conflict record, logs the conflict, and sends the log to the information center. For information about the log destination and output rule configuration in the information center, see the information center in *Network Management and Monitoring Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Enable recording user IPv6 address conflicts.
   ipv6 nd user-ip-conflict record enable
   By default, recording user IPv6 address conflicts is disabled.

Enabling recording user port migrations

About recording user port migrations

This feature enables the device to detect and record user port migrations. A user port migrates if an incoming NA packet has the same source IPv6 address and source MAC address as an existing ND entry but a different ingress port. The device generates a user port migration record, logs the migration event, and sends the log to the information center. For information about the log

destination and output rule configuration in the information center, see the information center in *Network Management and Monitoring Configuration Guide*.

Procedure

1. Enter system view.
   system-view
2. Enable recording user port migrations.
   ipv6 nd user-move record enable
   By default, recording user port migrations is disabled.

Enabling ND logging for user online and offline events

About ND logging for user online and offline events

This feature enables the device to generate user online or offline logs upon such events and send these logs to the information center. For information about the log destination and output rule configuration in the information center, see the information center in *Network Management and Monitoring Configuration Guide*.

Restrictions and guidelines

A higher log output rate consumes more CPU resources. Adjust the log output rate based the CPU performance and usage.

Procedure

1. Enter system view.
   system-view
2. Enable ND logging for user online and offline events.
   ipv6 nd online-offline-log enable [**rate** *rate*]
   By default, ND logging for user online and offline events is disabled.

Display and maintenance commands for IPv6 ND

Execute **display** commands in any view and **reset** commands in user view.

Task	Command
Display the number of IPv6 ND snooping entries in VLANs.	**display ipv6 nd snooping count vlan** [**interface** *interface-type interface-number*]
Display the number of IPv6 ND snooping entries in a VSI.	**display ipv6 nd snooping count vsi** [*vsi-name*]
Display IPv6 ND snooping entries in VLANs.	**display ipv6 nd snooping vlan** [[*vlan-id* | **interface** *interface-type interface-number*] [**global** | **link-local**] | *ipv6-address*] [**verbose**]
Display IPv6 ND snooping entries in a VSI.	**display ipv6 nd snooping vsi** [*vsi-name*] [**slot** *slot-number*
Display user IPv6 address conflict records.	**display ipv6 nd user-ip-conflict record** [**slot** *slot-number*]

Display user port migration records.	**display ipv6 nd user-move record** [**slot** *slot-number*]
Display the total number of neighbor entries.	**display ipv6 neighbors** { { **all** | **dynamic** | **static** } [**slot** *slot-number*] | **interface** *interface-type interface-number* | **vlan** *vlan-id* } **count**
Display neighbor information.	**display ipv6 neighbors** { { *ipv6-address* | **all** | **dynamic** | **static** } [**slot** *slot-number*] | **interface** *interface-type interface-number* | **vlan** *vlan-id* } [**verbose**]
Display the maximum number of ND entries that a device supports.	**display ipv6 neighbors entry-limit**
Display neighbor information for a VPN.	**display ipv6 neighbors vpn-instance** *vpn-instance-name* [**count**]
Clear IPv6 ND snooping entries in a VLAN.	**reset ipv6 nd snooping vlan** { [*vlan-id*] [**global** | **link-local**] | *vlan-id ipv6-address* }
Clear IPv6 ND snooping entries in a VSI.	**reset ipv6 nd snooping vsi** [*vsi-name*]
Clear IPv6 neighbor information.	**reset ipv6 neighbors** { **all** | **dynamic** | **interface** *interface-type interface-number* | **slot** *slot-number* | **static** }

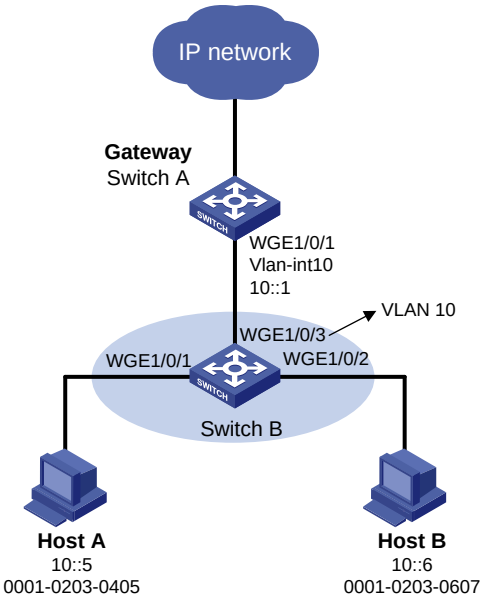
IPv6 ND configuration examples

Example: Configuring ND snooping

Network configuration

As shown in Figure 11, Host A and Host B are connected to the gateway through Device B. Enable ND snooping on Device B to learn ND snooping entries about Host A and Host B.

Figure 11 Network diagram



Procedure

1. Configure Device A:

   # Create VLAN 10.

   ```
   <DeviceA> system-view
   [DeviceA] vlan 10
   [DeviceA-vlan10] quit
   ```

   # Configure Twenty-FiveGigE 1/0/3 to trunk VLAN 10.

   ```
   [DeviceA] interface twenty-fivegige 1/0/3
   [DeviceA-Twenty-FiveGigE1/0/3] port link-type trunk
   [DeviceA-Twenty-FiveGigE1/0/3] port trunk permit vlan 10
   [DeviceA-Twenty-FiveGigE1/0/3] quit
   ```

   # Assign IPv6 address 10::1/64 to VLAN-interface 10.

   ```
   [DeviceA] interface vlan-interface 10
   [DeviceA-Vlan-interface10] ipv6 address 10::1/64
   [DeviceA-Vlan-interface10] quit
   ```

2. Configure Device B:

   # Create VLAN 10.

   ```
   [DeviceB] vlan 10
   [DeviceB-vlan10] quit
   ```

   # Configure Twenty-FiveGigE 1/0/1, Twenty-FiveGigE 1/0/2, and Twenty-FiveGigE 1/0/3 to trunk VLAN 10.

   ```
   [DeviceB] interface twenty-fivegige 1/0/1
   [DeviceB-Twenty-FiveGigE1/0/1] port link-type access
   [DeviceB-Twenty-FiveGigE1/0/1] port access vlan 10
   [DeviceB-Twenty-FiveGigE1/0/1] quit
   [DeviceB] interface twenty-fivegige 1/0/2
   [DeviceB-Twenty-FiveGigE1/0/2] port link-type access
   [DeviceB-Twenty-FiveGigE1/0/2] port access vlan 10
   [DeviceB-Twenty-FiveGigE1/0/2] quit
   ```

```
[DeviceB] interface twenty-fivegige 1/0/3
[DeviceB-Twenty-FiveGigE1/0/3] port link-type trunk
[DeviceB-Twenty-FiveGigE1/0/3] port trunk permit vlan 10
[DeviceB-Twenty-FiveGigE1/0/3] quit
```

Enable ND snooping for global unicast addresses and link-local addresses in VLAN 10.

```
[DeviceB] vlan 10
[DeviceB-vlan10] ipv6 nd snooping enable global
[DeviceB-vlan10] ipv6 nd snooping enable link-local
```

Enable ND snooping for data packets from unknown sources in VLAN 10.

```
[DeviceB-vlan10] ipv6 nd snooping glean source
[DeviceB-vlan10] quit
```

Configure Twenty-FiveGigE 1/0/3 as ND trusted interface.

```
[DeviceB] interface twenty-fivegige 1/0/3
[DeviceB-Twenty-FiveGigE1/0/3] ipv6 nd detection trust
[DeviceB-Twenty-FiveGigE1/0/3] quit
```

Configure Twenty-FiveGigE 1/0/1 to learn a maximum number of 200 ND snooping entries.

```
[DeviceB] interface twenty-fivegige 1/0/1
[DeviceB-Twenty-FiveGigE1/0/1] ipv6 nd snooping max-learning-num 200
[DeviceB-Twenty-FiveGigE1/0/1] quit
```

Configure Twenty-FiveGigE 1/0/2 to learn a maximum number of 200 ND snooping entries.

```
[DeviceB] interface twenty-fivegige 1/0/2
[DeviceB-Twenty-FiveGigE1/0/2] ipv6 nd snooping max-learning-num 200
[DeviceB-Twenty-FiveGigE1/0/2] quit
```

Verifying the configuration

Verify that Device B has learned ND snooping entries for Host A and Host B.

```
[DeviceB] display ipv6 nd snooping vlan 10
IPv6 address              MAC address     VID  Interface     Status       Age
10::5                     0001-0203-0405  10   WGE1/0/1      VALID        157
10::6                     0001-0203-0607  10   WGE1/0/2      VALID        105
```