Руководство пользователя

QSR-2830

Оглавление

1 CONFIGURING IPV6

1.1 Understanding IPv6

1.2 Overview

As the Internet is growing rapidly and the IPv4 address space is exhausting, the limitation of the IPv4 is more obvious. The research and practice of the next generation Internet Protocol (IPng) become popular. Furthermore, the IPng working group of the IETF has determined the protocol specification of IPng referred to as IPv6. See RFC 2460 for details.

Key Features

- More address space

The length of an address is extended to 128 bits from 32 bits of IPv4. Namely, there are 2^128-1 addresses for IPv6. IPv6 adopts hierarchical address mode and supports multiple-level IP address assignment, for example, from the Internet backbone network to the internal subnet of enterprises.

- Simplified format of packet header

The design principle of the new IPv6 packet header is to minimize the overhead. For this reason, some non-critical fields and optional fields are removed from the packet header and placed into the extended packet header. The length of an IPv6 address is 4 times the length of an IPv4 address; the size of the IPv6 packet header is only 2 times the size of the IPv4 packet header. The improved IPv6 packet header is more efficient for forwarding, for example, there is no checksum in the IPv6 packet header and it is not necessary for an IPv6 device to process the fragments during forwarding (the fragments are completed by the originator).

- High-efficient hierarchical addressing and routing structure

IPv6 adopts the aggregation mechanism and defines a flexible hierarchical addressing and routing structure, and several networks at the same level are represented with a unified network prefix at a higher-layer device. So it obviously reduces the routing entries that the device must maintain and greatly minimizes the routing and storage overhead.

- Simple management: plug and play

The management and maintenance of network nodes are simplified by the implementation of a series of auto-discovery and auto-configuration functions. For example, the neighbor discovery, MTU discovery, router advertisement (RA), router solicitation (RS) and auto-configuration technologies provide the related service for plug and play. It should be mentioned that IPv6 supports such address configuration methods as stateful configuration and stateless configuration. In IPv4, the Dynamical Host Configuration Protocol (DHCP) implements the automatic configuration of a host IP address and related configuration, while IPv6 inherits this auto-configuration service of IPv4 and refers to it as the stateful auto-configuration. Furthermore, IPv6 also adopts an auto-configuration service, referred to as stateless auto-configuration. During the stateless auto-configuration, the host obtains the link-local address, the address prefix of the local device and some other related configuration information automatically.

- Security

IPSec is an optional extended protocol of IPv4, but it is only a component of IPv6 used to provide security. At present, IPv6 implements the authentication header (AH) and encapsulated security payload (ESP) mechanisms. The former authenticates the integrity of data and the source of an IP packet to ensure that the packet does come from the node marked by the source address, while the latter provides the data encryption function to implement end-to-end encryption.

- More excellent QoS support

A new field in the IPv6 packet header defines how to identify and process a data flow. The Flow Label field in an IPv6 packet header is used to identify the data flow ID, by which IPv6 allows users to put forward the requirement for the QoS of communication. The device can identify all packets of a specified data flow by this field and provide special processing for these packets as required.

- New protocol for interactions between neighbor nodes

The Neighbor Discovery Protocol of IPv6 uses a series of IPv6 control information messages (ICMPv6) to manage the interactions between neighbor nodes (the nodes on the same link). The Neighbor Discovery Protocol and high-efficient multicast and unicast neighbor discovery messages replace the previous broadcast-based Address Resolution Protocol (ARP) and the ICMPv4 router discovery messages.

- Extensibility

IPv6 provides powerful extensibility and the new features can be added to the extended packet header after the IPv6 packet header. Unlike the IPv4 packet header, the IPv6 packet header can only support the options of up to 40 bytes, while the size of the IPv6 extended packet header is only limited by the maximum number of bytes of the whole IPv6 packet.

IPv6 supports the following features:
- IPv6 protocol
- IPv6 address format
- Type of IPv6 address
- ICMPv6
- IPv6 neighbor discovery
- Path MTU discovery
- ICMPv6 redirection
- Duplicate address detection
- IPv6 stateless auto-configuration
- IPv6 address configuration
- IPv6 route forwarding (supporting static route configuration)
- Configuration of various IPv6 parameters
- Diagnosis tool ping IPv6

1.2.1 IPv6 Address Format

The basic format of an IPv6 address is X : X : X : X : X : X : X : X, where X is a 4-digit hexadecimal integer (16 bits). Each digit contains 4 bits, each integer contains 4 hexadecimal digits, and each address contains 8 integers, so the address includes a total of 128 bits. Some legal IPv6 addresses are as follows:

2001:ABCD:1234:5678:AAAA:BBBB:1200:2100

800 : 0 : 0 :0 : 0 : 0 : 0 : 1

1080 : 0 : 0 : 0 : 8 : 800 : 200C : 417A

These integers are hexadecimal integers, where A to F denote 10 to 15 respectively. Each integer in the address must be denoted and the starting 0 need not be denoted. Some IPv6 addresses may contain a series of 0s (such as the second and third examples). In this case, colons (: :) are allowed to denote this series of 0s. Namely, the address 800:0:0:0:0:0:0:1 can be denoted as: 800 :: 1.

These two colons denote that this address can be extended to a complete 128-bit address. In this way, the 16-bit group can be replaced with two colons only when they are all 0s and the two colons can only be present once.

In the hybrid environment of IPv4 and IPv6, there is a hybrid denotation method. The lowest 32 bits in an IPv6 address can be used to denote an IPv4 address. The address can be expressed in a hybrid mode, that is, X: X : X : X : X : X : d . d . d . d, where, X denotes a 16-bit integer, while d denotes an 8-bit decimal integer. For instance, the address 0 : 0 : 0 : 0 : 0 : 0 : 192 .168 . 20 : 1 is a legal IPv6 address. After the abbreviated expression method is used, this address can be denoted as follows: : : 192.168. 20. 1. One typical example is an IPv4-compatible IPv6 address, which is expressed as “::A.B.C.D”, with the first 96 bits being all 0s, such as “::1.1.1.1”, but this expression method is revoked. Another typical example is an IPv4-mapped IPv6 address, which is expressed as “::FFFF:A.B.C.D” and used to express an IPv4 address as an IPv6 address, that is, map the IPv4 address”1.1.1.1” to the IPv6 address”::FFFF:1.1.1.1”.

Because the IPv6 address is divided into two parts, the subnet prefix and the interface identifier, it can be denoted as an address including an additional numeric value by the method like the CIDR address. This numeric value indicates how many bits represent the network part (the network prefix). Namely the IPv6 node address indicates the length of the prefix, and the length is differentiated from the IPv6 address by a slash. For instance: 12AB::CD30:0:0:0:0/60. The length of the prefix used for routing in this address is 60 bits.

1.2.2 Type of IPv6 Address

RFC 4291 defines three types of IPv6 addresses:
- Unicast: Identifier of a single interface. The packet to be sent to a unicast address will be transmitted to the interface identified by this address.
- Anycast: Identifiers of a set of interfaces. The packet to be sent to an anycast address will be transmitted to one of the interfaces identified by this address (the nearest one is selected according to the routing protocol).
- Multicast: Identifiers of a set of interfaces (In general, these interfaces belong to different nodes). The packet to be sent to a multicast address will be transmitted to all the interfaces that join this multicast address.

Caution The broadcast address is not defined in IPv6.

The following describes these types of addresses one by one.

1.2.2.1 Unicast Addresses

The unicast addresses are divided into unspecified address, loopback address, link-local address, site-local address and global unicast address. Now the site-local address has been revoked. The unicast addresses excepting the unspecified address, loopback address and link-local address are all global unicast addresses.

1) Unspecified address

The unspecified address is 0:0:0:0:0:0:0:0, generally abbreviated as :: and used for the following purposes.

- If there is no unicast address when a host is started, use the unspecified address as the source address, send an RS, and obtain the prefix information from the gateway to automatically generate the unicast address.
- When configuring an IPv6 address for the host, check whether the IPv6 address conflicts with the address of any other host in the same network segment or not. If so, use the unspecified address as the source address to send a neighbor solicitation (NS) message, same as free ARP.

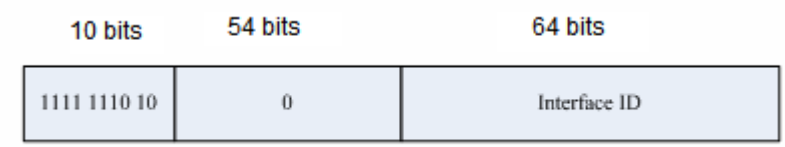
2) Loopback address

The loopback address is 0:0:0:0:0:0:0:1, abbreviated as ::1, which is equal to the IPv4 address 127.0.0.1 and used when the node sends the packets to itself.

3) Link–local address

The format of link-local address:

Figure 1

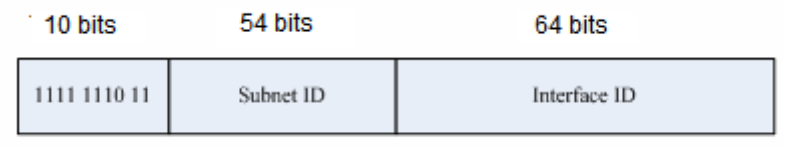


The link-local address is used to number the host on the single network link. The address identified by the first 10 bits of the prefix is the link-local address. The device will never forward the packet of the source address or the destination address with the link-local address. The intermediate 54 bits are all 0s. The last 64 bits indicate the interface identifier, and this part allows the single network to connect up to 2^{64}-1 hosts.

4) Site-local address

The format of site-local address:

Figure 2

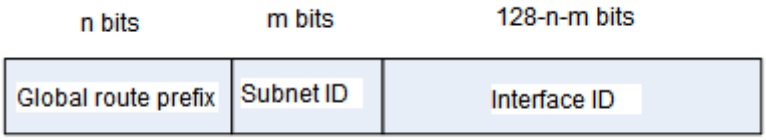


The site-local address can be used to transmit data within the site, and the device will not forward the packet of the source address or the destination address with the site-local address to the Internet. Namely, such packet can only be forwarded within the site, but cannot be forwarded out of the site. The site may be deemed as the LAN of a company, and the site-local address is similar to a private IPv4 address, for example, 192.168.0.0/16. RFC 3879 has revoked the site-local address. In new implementations, this prefix is no longer supported and is uniformly deemed as a global unicast address. In existing implementations and deployments, this prefix may be still used.

5) Global unicast address

The format of global unicast address:

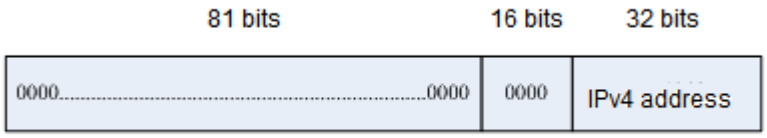
Figure 3



One class of the global unicast address is the IPv6 address embedded with an IPv4 address, which is used to interconnect the IPv4 nodes and the IPv6 nodes and divided into IPv4-compatible IPv6 address and IPv4-mapped IPv6 address.

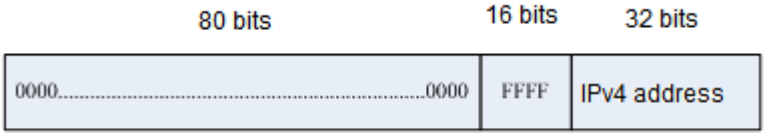
The format of IPv4-compatible IPv6 address:

Figure 4



The format of IPv4-mapped IPv6 address:

Figure 5



The IPv4-compatible IPv6 address is mainly used for automatic tunneling, which supports both IPv4 and IPv6. The IPv4-compatible IPv6 address is used to transmit an IPv6 packet via an IPv4 device in the tunneling way. Now the IPv4-compatible IPv6 address has been revoked. The IPv4-mapped IPv6 address is used by IPv6 nodes to access the nodes that only support IPv4. For example, when one IPv6 application of the IPv4/IPv6 host requests the resolution of a host name (the host only supports IPv4), the name server will internally generate an IPv4-mapped IPv6 address dynamically and return it to the IPv6 application.

1.2.2.2 Multicast Addresses

The format of the IPv6 multicast address is as follows:

```
|  8   | 4 | 4 |              112 bits              |
+----------+----+----+------------------------------------------------------------------+
|11111111|flgs|scop|              group ID              |
+----------+----+----+------------------------------------------------------------------+
```

The first byte of the address format is all 1s, which denote a multicast address.

- Flag field:

It consists of 4 bits. At present, only the fourth bit is specified. The bit is used to indicate whether the address is a known multicast address specified by the Internet Assigned Numbers Authority (IANA) or a temporary multicast address used on a specific occasion. If this flag bit is 0, it indicates this address is a known multicast address. If this bit is 1, it indicates that this address is a temporary one. Other 3 flag bits are reserved for future use.

- Range field:

The Range field is composed of 4 bits and used to denote the range of multicast, namely, whether the multicast group contains the local node, the local link and the local site or nodes in any positions in the IPv6 global address space.

- Group ID field:

This field is 112 bits long and used to identify a multicast group. Depending on whether a multicast address is temporary or known and the range of the address, a multicast identifier can denote different groups.

The multicast address of IPv6 is this type of address using FF00::/8 as the prefix. One multicast address of IPv6 usually identifies the interfaces of a serial of different nodes. When one packet is sent to one multicast address, this packet will be distributed to the interfaces of each node with this multicast address. One node (host or device) should join the following multicast addresses:

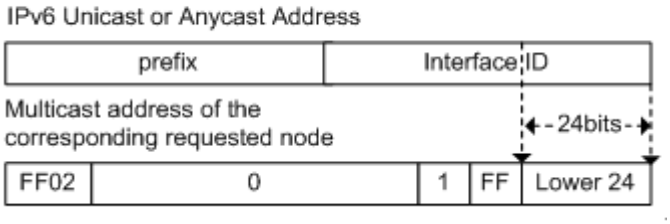
- The multicast address of all nodes on the local link, that is, FF02::1
- The multicast address of the solicited node, with the prefix of FF02:0:0:0:0:1:FF00:0000/104

For the device, it is necessary to join the multicast address FF02::2 of all devices on the local link.

If the multicast address of the solicited node corresponds to the IPv6 unicast and anycast address, it is necessary for the IPv6 node to join the corresponding multicast address of the solicited node for each configured unicast address and anycast address. The prefix of the multicast address of the solicited node is FF02:0:0:0:0:1:FF00:0000/104, another 24 bits are comprised of the unicast address or the lower 24 bits of the anycast address, for example, the multicast address of the solicited node corresponding to the unicast address FE80::2AA:FF:FE21:1234 is FF02::1:FF21:1234.

The multicast address of the solicited node is usually used in an NS message. The format of the solicited node is as follows:

Figure 6



1.2.2.3 Anycast Addresses

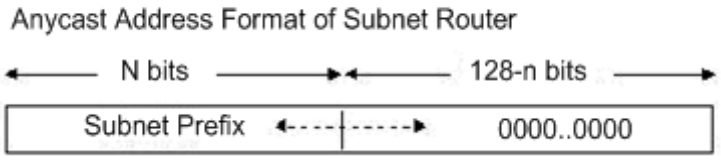
The anycast address is similar to the multicast address as more than one node shares an anycast address. The difference is that only one node expects to receive the data packet of the anycast address, while all nodes of the multicast group members expect to receive all packets sent to this address. The anycast address is assigned to the normal IPv6 unicast address space, so the anycast address cannot be differentiated from the unicast address from the style. For this reason, each member of an anycast group represented by an anycast address must be configured explicitly to identify the anycast address.

Caution The anycast address can only be assigned to the device, but cannot be assigned to the host. Furthermore, the anycast address cannot be used as the source address of the packet.

RFC 2373 predefines an anycast address, referred to as the anycast address of a subnet router. The following figure shows the anycast address format of the subnet router, which consists of the subnet prefix followed by a series of 0s (as the interface identifier).

The subnet prefix identifies a specified link (subnet) and the packet to be sent to the anycast address of the subnet router will be distributed to a device of this subnet. The anycast address of the subnet router is usually used for a node which needs to communicate with one device of a remote subnet.

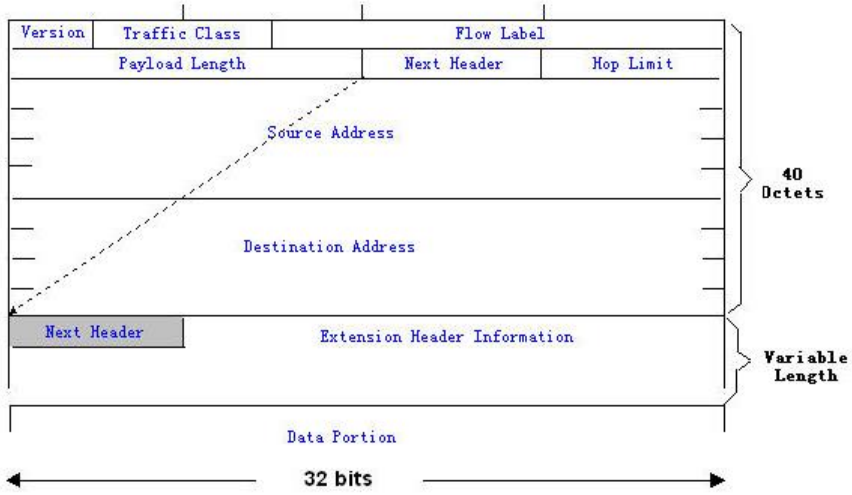
Figure 7



IPv6 Packet Header Structure

The format of the IPv6 packet header is shown in the following figure.

Figure 8



In IPv4, the packet header is measured in units of 4 bytes; in IPv6, the packet header is measured in units of 8 bytes, and the total size of the packet header is 40 bytes. In the IPv6 packet header, the following fields are defined:

- Version:

The length is 4 bits. For IPv6, the field must be 6.

- Traffic Class:

The length is 8 bits. It indicates a type of service provided to the packet and is equal to the “TOS” in IPv4.

- Flow Label:

The length is 20 bits. This field is used to identify the packets of the same service flow. One node can be used as the source of several service flows. The flow label and source node IP address identify a service flow uniquely.

- Payload Length:

The length is 16 bits, including the byte length of the payload and the length of various IPv6 extension options (if any). In other words, it includes the length of an IPv6 packet except for the IPv6 header.

- Next Header:

This field indicates the protocol type in the header field following the IPv6 header. Similar to the IPv4 protocol field, the Next Header field can be used to indicate whether the upper layer protocol is TCP or UDP. It can also be used to indicate whether an extended IPv6 header exists.

- Hop Limit:

The length is 8 bits. When the device forwards the packet for one time, the value of this field will decrease by 1. When the value of this field is 0, this packet will be discarded. It is similar to the lifetime field in the IPv4 packet header.

- Source Address:

The length is 128 bits. It indicates the sender address of an IPv6 packet.

- Destination Address:

The length is 128 bits. It indicates the receiver address of an IPv6 packet.

At present, the following extended headers are defined in IPv6:

- Hop-by-Hop Options:

This extended header must immediately follow an IPv6 header. It contains the option data that must be checked by each node along the path.

- Routing Header (Routing (Type 0)):

This extended header indicates the nodes that a packet will go through before reaching the destination. It contains the addresses of various nodes that the packet goes through. The initial destination address of the IPv6 header is the first one of a series of addresses in the routing header, other than the final destination address of the packet. After receiving this packet, the node of this address will process the IPv6 header and the routing header, and send the packet to the second address in the routing header. This process continues until the packet reaches the final destination.

- Fragment:

This extended header is used to fragment the packets longer than the MTU of the path between the source node and destination node.

- Destination Options:

This extended header replaces the IPv4 option field. At present, the only defined destination option is an option to be filled with an integral multiple of 64 bits (8 bytes) when necessary. This extended header can be used to carry the information checked by the destination node.

- Upper-layer header:

It indicates the upper layer transmission protocol, such as TCP(6) and UDP(17).

Furthermore, the extended header of Authentication and Encapsulating Security Payload will be described in the IPSec section. At present, the IPv6 implemented by the device does not support IPSec.

IPv6 Path MTU Discovery

Similar to the path MTU discovery of IPv4, the path MTU discovery of IPv6 allows one host to discover and adjust the size of the MTU in the data transmission path.

Furthermore, when the data packet to be sent is larger than the MTU of the data transmission path, the host will fragment the packet by itself. This behavior makes it not necessary for the device to process the fragment, and thus save resources and improve the efficiency of the IPv6 network.

Caution

The minimum link MTU is 68 bytes in IPv4, indicating that the links along the path over which the packets are transmitted should support at least the link MTU of 68 bytes. The minimum link MTU is 1280 bytes in IPv6. It is strongly recommended that the link MTU of 1500 bytes should be used for the link in IPv6.

1.2.3 IPv6 Neighbor Discovery

The main functions of the IPv6 Neighbor Discovery Protocol include router discovery, prefix discovery, parameter discovery, address auto-configuration, address resolution (ARP), next-hop determination, neighbor unreachability detection, duplicate address detection, and redirection. Neighbor discovery defines 5 types of ICMP messages, which are router solicitation (ICMP type133), RA (ICMP type134), NS or ARP request (ICMP type135), neighbor advertisement or APR response (ICMP type136) and ICMP redirection message (ICMP type137).

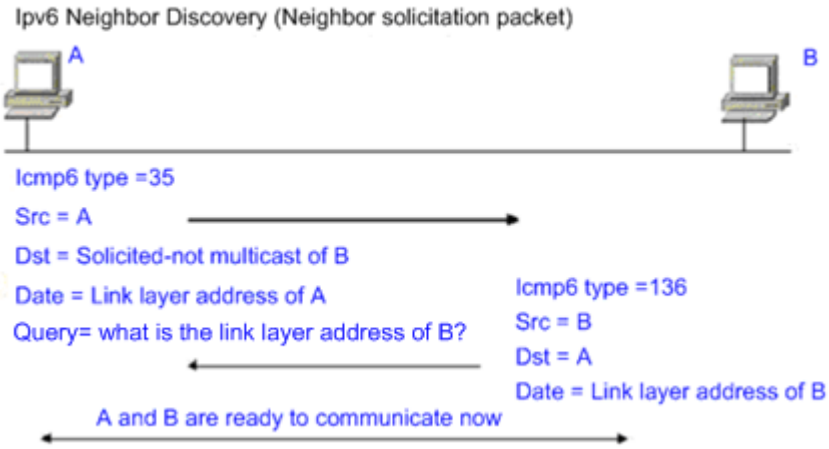
The following describes the neighbor discovery function in detail:

1.2.3.1 Address Resolution

A node must obtain the link layer address of another node before communicating with it. At this time, the node should send an NS message to the solicited multicast address, that is, the IPv6 address of the destination node. The NS message also contains the link layer address of itself. After receiving this NS message, the destination node responds with a message, referred to as neighbor advertisement (NA), with its link layer address. After receiving the response message, the source node can communicate with the destination node.

The following is the address resolution procedure:

Figure 9



1.2.3.2 Neighbor Unreachability Detection

When the reachable time of a neighbor expires, neighbor unreachability detection is performed if an IPv6 unicast packet needs to be sent to this neighbor.

Neighbor unreachability detection and sending the IPv6 packet to the neighbor can be performed concurrently. During the detection, the device continues to forward the IPv6 packet to the neighbor.

1.2.3.3 Duplicate Address Detection

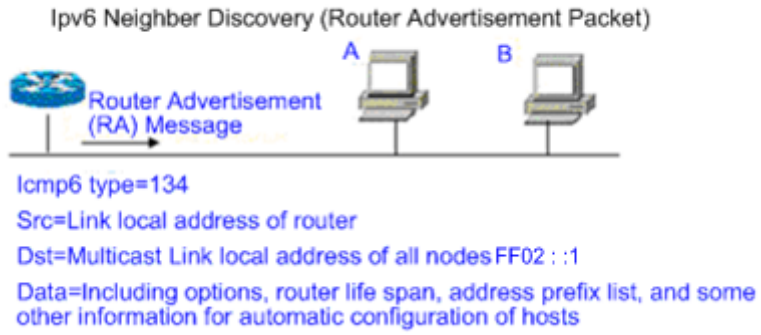
After an IPv6 address is configured for the host, duplicate address detection may be performed to know whether the IPv6 address is unique on the link, by sending an NS message with the source IPv6 address being an unspecified address.

1.2.3.4 Router, Prefix and Parameter Discovery

The router sends an RA to all the local nodes of the link periodically.

The following figure shows the process of sending the RA.

Figure 10



In general, the RA contains the following contents:

- One or more IPv6 address prefixes used for the on-link determination or the stateless address auto-configuration.
- Effective period of the IPv6 address prefix.
- Host auto-configuration mode (stateful or stateless).
- Information for the default device (namely, the device determines whether it is used as the default device. If yes, it will announce the time to act as the default device).
- Other information for host configuration such as the hop limit, the MTU and the NS retransmission interval.

The RA is also used to respond to the RS message sent by the host. The RS message allows the host to obtain the auto-configuration information immediately without waiting for the device to send the RA. If there is no unicast address when the host is started, the RS message sent by the host will use the unspecified address (0:0:0:0:0:0:0:0) as the source address of the RS message. Otherwise, the existing unicast address is used as the source address, while the RS message uses the multicast address (FF02::2) of all devices on the local link as the destination address. The RA message, in response to the RS message, will use the source address of the RS message as the destination address (if the source address is the unspecified address, it will use the multicast address FF02::1) of all nodes on the local link.

The following parameters can be configured in the RA message.

Ra-interval: interval for sending the RA

Ra-lifetime: router lifetime, namely, whether the device acts as the default router of the local link and the time to act this role

Prefix: IPv6 address prefix of the local link, which can be used for the on-link determination or the stateless address auto-configuration, including the configuration of other parameters for the prefix

Rs-initerval: interval for sending the NS message

Reachabletime: time maintained after the neighbor is considered reachable

The above parameters are configured in the IPv6 interface properties.


Caution

1. No RA message is sent actively on the interface by default. To allow the device to send the RA message, you can use the **no ipv6 nd suppress-ra** command in interface configuration mode.
2. In order to enable normal stateless address auto-configuration of the node, the length of the prefix for the RA message should be 64 bits.

1.2.3.5 Redirection

After receiving the IPv6 packets, the router discovers an optimal next hop and sends an ICMP redirection message to notify the host of the optimal next hop. Next time the host sends the IPv6 packets to the optimal next hop directly.

1.3 Configuring IPv6

The following will describe the configuration of various functional modules of IPv6 respectively.

1.3.1 Configuring an IPv6 Address

This section describes how to configure an IPv6 address on an interface. No IPv6 address is configured by default.


Caution

Once an interface is created and its link state is UP, the system will automatically generate the link-local address for the interface. At present, IPv6 does not support anycast address.
For S57 and S76 series, the range of the length of the prefix of the interface IPv6 address is [0, 64] or [128, 128], because the range of the length of the routing prefix supported by the hardware forwarding table of the chip is [0, 64] or [128, 128]. For S86 and S96 series, the range of the length of the prefix of the interface IPv6 address is not limited, but the total number of IPv6 routes within the range [65, 127] of the length of the routing prefix supported by switches is 512.

To configure an IPv6 address, use the following commands.

Command	Function
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Qtech(config-if)#**ipv6 enable**	Enables the IPv6 protocol on an interface. If this command is not run, the system automatically enables the IPv6 protocol when you

Command	Function
	configure an IPv6 address for an interface.
Qtech(config-if)#**ipv6 address** *ipv6-address/prefix-length* Qtech(config-if)#**ipv6 address** *ipv6-prefix/prefix-length* [**eui-64**]	Configures an IPv6 unicast address for this interface. When the command includes the keyword **Eui-64**, only the prefix must be specified, and the interface ID is automatically generated in the EUI-64 format. The generated IPv6 address consists of the configured address prefix and the 64-bit interface ID. Note: Whether the keyword **eui-64** is used, it is necessary to enter the complete address format to delete an IPv6 address (prefix + interface ID or prefix length). When you configure an IPv6 address on an interface, the IPv6 protocol is automatically enabled on the interface. Even if you use **no ipv6 enable**, you cannot disable the IPv6 protocol.
Qtech(config-if)#**end**	Returns to privileged EXEC mode.
Qtech#show **ipv6 interface** *interface-id*	Displays the IPv6 interface information.
Qtech#copy **running-config startup-config**	Saves the configuration.

To delete the configured IPv6 address, use the **no ipv6 address** *ipv6-prefixIprefix-length* **[eui-64]** command.

The following example configures an IPv6 address.

```
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if)# ipv6 enable
Qtech(config-if)# ipv6 address fec0:0:0:1::1/64
Qtech(config-if)# end
Qtech(config-if)# show ipv6 interface GigabitEthernet 0/1
Interface GigabitEthernet 0/1 is Up, ifindex: 1
address(es):
Mac Address: 00:00:00:00:00:01
INET6: fe80::200:ff:fe00:1 , subnet is fe80::/64
INET6: fec0:0:0:1::1 , subnet is fec0:0:0:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

1.3.2 Configuring ICMPv6 Redirection

This section describes how to configure the ICMPv6 redirection function on the interface. The redirection function of the IPv6 on the interface is enabled by default. The device needs to send a redirection message to the initiator during packet forwarding in the following cases:

- The destination address of the message is not a multicast address.
- The destination address of the message is not the device itself.
- The output interface of the next hop determined by the device for this message is the same as the interface that receives this message, namely, the next hop and the initiator are on the same link.
- The node identified by the source IP address of the packet is a neighbor of the local device. Namely, this node exists in the device's neighbor table.

Caution

The device other than the host can generate the redirection message, and the device will not update its routing table when it receives the redirection message.

To enable redirection on the interface, use the following commands.

Command	Function
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Qtech(config-if)#**ipv6 redirects**	Enables the IPv6 redirection function.
Qtech(config-if)#**end**	Returns to privileged EXEC mode.
Qtech#**show ipv6 interface** *interface-id*	Displays the interface configuration information.
Qtech#**copy running-config startup-config**	Saves the configuration.

To disable the redirection function, use the **no ipv6 redirects** command.

The example configures the redirection function.

```
Qtech(config)# interface GigabitEthernet 0/1
Qtech (config-if)# ipv6 redirects
Qtech (config-if)# end
Qtech # show ipv6 interface GigabitEthernet 0/1
Interface GigabitEthernet 0/1 is Up, ifindex: 1
address(es):
Mac Address: 00:d0:f8:00:00:01
INET6: fe80::2d0:f8ff:fe00:1 , subnet is fe80::/64
INET6: fec0:0:0:1::1 , subnet is fec0:0:0:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 10 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

1.3.3 Configuring a Static Neighbor

This section describes how to configure a static neighbor. No static neighbor is configured by default. In general, a neighbor learns and maintains its status by the Neighbor Discovery Protocol (NDP) dynamically. Moreover, you can configure the static neighbor manually.

To configure the static neighbor, use the following commands.

Command	Function
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 neighbor** *ipv6-address interface-id hardware-address*	Configure a static neighbor on the interface.
Qtech(config)#**end**	Returns to privileged EXEC mode.
Qtech#**show ipv6 neighbors**	View the neighbor list.
Qtech#**copy running-config startup-config**	Saves the configuration.

To delete the specified neighbor, use the **no ipv6 neighbor** *ipv6-address interface-id* command.

The following example configures a static neighbor on the GigabitEthernet 0/1 interface.

```
Qtech(config)# ipv6 neighbor fec0:0:0:1::100 GigabitEthernet 0/1 00d0.f811.1234
Qtech (config)# end
Qtech# show ipv6 neighbors verbose fec0:0:0:1::100
IPv6 Address          Linklayer Addr   Interface
fec0:0:0:1::100       00d0.f811.1234   GigabitEthernet 0/1
State: REACH/H Age: - asked: 0
```

Caution When you configure a static neighbor, the configuration takes effect only when the neighbor prefix matches the interface. Specifically, the configured static neighbor prefix belongs to the network segment of an address configured for the interface, and does not conflict with the address. An invalid static neighbor is in the inactive state. Data sent to the destination is not sent to the MAC address specified by the static neighbor, but the MAC address is learned based on routes in dynamic learning mode. To view the validity status of a static neighbor, run the **show ipv6 neighbor static** command.

1.3.4 Configuring Duplicate Address Detection

This section describes how to configure duplicate address detection times. Duplicate address detection is mandatory to assign unicast addresses to interfaces. The purpose is to detect the uniqueness of an address. Duplicate address detection should be performed for addresses that are configured in manual configuration mode, stateless auto-configuration mode, and stateful auto-configuration mode. However, it is not necessary to perform duplicate address detection under the following two conditions:

- The management prohibits the duplicate address detection, namely, the number of the NS messages sent for the duplicate address detection is set to 0.
- Duplicate address detection cannot be performed for a configured anycast address.

Furthermore, if the duplicate address detection function is not disabled on the interface, the system will restart the duplicate address detection process for the configured address when the interface changes to the Up state from the Down state.

To configure the number of NS messages sent for duplicate address detection, use the following commands.

Command	Function
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Qtech(config-if)#**ipv6 nd dad attempts** *attempts*	Configures the number of NS messages sent for duplicate address detection. When it is set to 0, the duplicate address detection function is disabled on the interface.
Qtech(config-if)#**end**	Returns to privileged EXEC mode.
Qtech#**show ipv6 interface vlan** *1*	Displays the IPv6 information on the interface.
Qtech#**copy running-config startup-config**	Saves the configuration.

To restore the default value, use the **no ipv6 nd dad attempts** command.

The following example configures the number of NS messages sent for duplicate address detection on the SVI1.

```
Qtech(config)# interface GigabitEthernet 0/1
Qtech(config-if)# ipv6 nd dad attempts 3
Qtech(config-if)# end
Qtech# show ipv6 interface GigabitEthernet 0/1
Qtech(config)# interface vlan 1
Qtech(config-if)# ipv6 nd dad attempts 3
Qtech(config-if)# end
Qtech# show ipv6 interface vlan 1
Interface GigabitEthernet 0/1 is Up, ifindex: 1
address(es):
Mac Address: 00:d0:f8:00:00:01
INET6: fe80::2d0:f8ff:fe00:1 , subnet is fe80::/64
INET6: fec0:0:0:1::1 , subnet is fec0:0:0:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 10 milliseconds
ICMP redirects are enabled
```

```
ND DAD is enabled, number of DAD attempts: 3
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

1.3.5 Configuring the IPv6 MTU

If an IPv6 packet exceeds the interface MTU size, the RGOS software splits the packet. For all devices in the same physical network segment, the IPv6 MTU of interconnected interfaces must be the same. The IPv6 MTU of interfaces automatically keeps consistent with the link MTU of interfaces.

Use the following commands to configure the IPv6 MTU.

Command	Function
Qtech(config-if)# **ipv6 mtu** *bytes*	Configures the interface MTU value. The range is from 1,280 to 1,500.
Qtech(config-if)# **no ipv6 mtu**	Restores the default MTU value.

1.3.6 Configuring Other Interface Parameters

The IPv6 parameters on an interface are divided into two parts. One is used to control the behavior of the device itself, and the other is used to control the contents of the RA sent by the device to determine what action should be taken by the host when the host receives this RA.

The following table describes these commands.

Command	Function
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**interface** *interface-id*	Enters interface configuration mode. Note that the **no switchport** command must be used to switch the layer-2 interface to the layer-3 interface.
Qtech(config-if)#**ipv6 enable**	Enables the IPv6 function.
Qtech(config-if)#**ipv6 nd ns-interval** *milliseconds*	(Optional) Defines the retransmission interval of the NS message, in milliseconds. The default value is 1000 milliseconds.
Qtech(config-if)#**ipv6 nd reachable-time** *milliseconds*	(Optional) Defines the time during which the neighbor is considered as reachable, in milliseconds. The default value is 30000 milliseconds.
Qtech(config-if)# **ipv6 nd prefix** {*ipv6-prefix*/*prefix-length* | **default**} [[*valid-lifetime preferred-lifetime*] | [**at** *valid-date preferred-date*] | [**infinite** {**infinite** | *preferred-lifetime*}]] [**no-advertise**] | [[**off-link**] [**no-autoconfig**]]	(Optional) Sets the address prefix to be advertised in the RA message.
Qtech(config-if)#**ipv6 nd ra-lifetime** *seconds*	(Optional) Sets the lifetime of the router in the RA message, namely, the time to act as the default device. The value 0 indicates that the device will not act as the default device of the directly-connected network. The default value is 1800 seconds.
Qtech(config-if)#**ipv6 nd ra-interval** {*seconds* **min-max** *min_value max_value*}	(Optional) Sets the time interval for the device to send the RA message periodically, in seconds. The default value is 200 seconds. With **min-max** specified, the actual interval for sending the RA message is a random value between the minimum value and the maximum value. Without **min-max** specified, the actual interval for sending the RA message is the configured value multiplied by 1.2 or 0.8.
Qtech(config-if)#**ipv6 nd managed-config-flag**	(Optional) Sets the **managed address configuration** flag bit of the RA message, and determines whether the host receiving the RA message will use the stateful auto-configuration to obtain the address. The flag bit is not configured for the RA message by default.
Qtech(config-if)#**ipv6 nd other-config-flag**	(Optional) Sets the **other stateful configuration** flag bit of the RA message, and determines whether the host receiving the RA message will use the stateful auto-configuration to obtain

Command	Function
	information other than the address. The flag bit is not configured for the RA message by default.
Qtech(config-if)#**ipv6 nd suppress-ra**	(Optional) Sets whether to suppress the RA message on this interface. The flag bit is not configured for the RA message by default.
Qtech(config-if)#**end**	Returns to privileged EXEC mode.
Qtech#**show ipv6 interface** [*interface-id*] [**ra-info**]	Displays the IPv6 information or the information of the RA sent by this interface.
Qtech#**copy running-config startup-config**	(Optional) Saves the configuration.

To restore the default value, use the **no** commands of above commands. For details, see the *IPv6 Command Reference*.

1.4 Monitoring and Maintaining IPv6

Use the following commands to display some internal information of the IPv6 protocol, such as the IPv6 information, the neighbor table, and the routing table information of an interface.

Command	Function
show ipv6 interface [*interface-id*] [**ra-info**]	Displays the IPv6 information of the interface.
Show ipv6 neighbors [vrf *vrf-name***] [verbose]** [*interface-id*] [*ipv6-address*]	Displays the neighbor information.
Show ipv6 route [vrf *vrf-name***] [static | local | connected | bgp | rip | ospf | isis**]	Displays the information of the IPv6 routing table.

- Display the IPv6 information of an interface.

```
Qtech# show ipv6 interface
interface GigabitEthernet 0/1 is Down, ifindex: 1
address(es):
Mac Address: 00:d0:f8:00:00:01
INET6: fe80::2d0:f8ff:fe00:1 , subnet is fe80::/64
INET6: fec0:1:1:1::1 , subnet is fec0:1:1:1::/64
Joined group address(es):
ff01:1::1
ff02:1::1
ff02:1::2
ff02:1::1:ff00:1
MTU is 1500 bytes
ICMP error messages limited to one every 10 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<160--240>
ND router advertisements live for 1800 seconds
```

- Display the information of the RA message to be sent on an interface.

```
Qtech# show ipv6 interface ra-info
GigabitEthernet 0/1: DOWN
RA timer is stopped
waits: 0, initcount: 3
statistics: RA(out/in/inconsistent): 4/0/0, RS(input): 0
Link-layer address: 00:00:00:00:00:01
Physical MTU: 1500
ND router advertisements live for 1800 seconds
ND router advertisements are sent every 200 seconds<160--240>
Flags: !M!O, Adv MTU: 1500
ND advertised reachable time is 0 milliseconds
ND advertised retransmit time is 0 milliseconds
ND advertised CurHopLimit is 64
Prefixes: (total: 1)
fec0:1:1:1::/64(Def, Auto, vltime: 2592000, pltime: 604800, flags: LA)
```

■ Display the neighbor table information of IPv6.

```
Qtech# show ipv6 neighbors
IPv6 Address                Linklayer Addr  Interface
fe80::200:ff:fe00:1         0000.0000.0001 GigabitEthernet 0/1
State: REACH/H Age: - asked: 0
fec0:1:1:1::1               0000.0000.0001 GigabitEthernet 0/1 State: REACH/H Age: -
asked: 0
```

2 CONFIGURING IPV6 TUNNELS

2.1 Overview

IPv6 is designed to inherit and replace IPv4. However, the evolution from IPv4 to IPv6 is a gradual process. Therefore, it is inevitable that these two protocols coexist for a period before IPv6 completely replaces IPv4. At the beginning of this transition stage, IPv4 networks are still main networks. IPv6 networks are similar to isolated islands in IPv4 networks. The problems about transition can be divided into the following two types:

6) Communication among isolated IPv6 networks via IPv4 networks
7) Communication between IPv6 networks and IPv4 networks

This article discusses the tunnel technology that is used to solve problem 1. The solution to problem 2 is Network Address Translation-Protocol Translation (NAT-PT), which is not covered in this article.

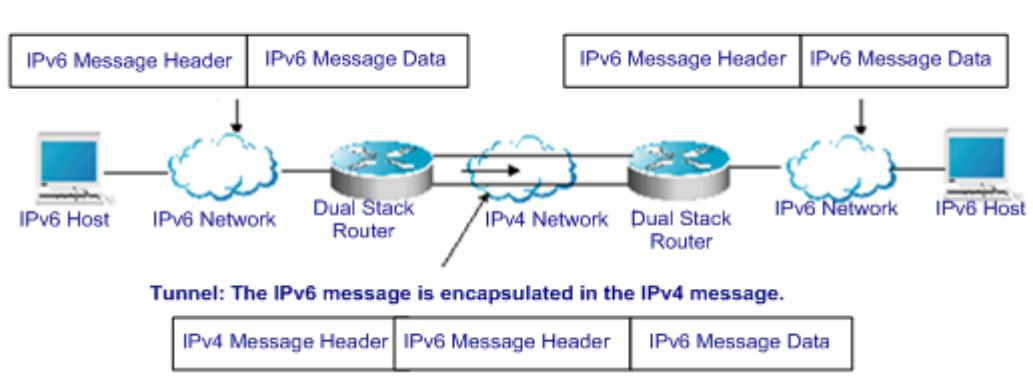
The IPv6 tunnel technology encapsulates IPv6 packets in IPv4 packets. In this way, IPv6 packets can communicate via IPv4 networks. Therefore, with the IPv6 tunnel technology, isolated IPv6 networks can communicate with each other via existing IPv4 networks, avoiding any modification and upgrade to existing IPv4 networks. An IPv6 tunnel can be configured between area border routers (ABRs) or between an ABR and a host. However, all the nodes at the two ends of the tunnel must support the IPv4 and IPv6 protocol stacks. At present, the following tunnel technologies are supported.

Tunnel Type	Reference
Manually Config Tunnel	RFC 2893
Automatic 6to4 Tunnel	RFC 3056
Intra-Site Automatic Tunnel Addressing Protocol(ISATAP)	draft-ietf-ngtrans-isatap-22

Caution Interconnecting the isolated IPv6 networks through the IPv6 tunnel technology is not the ultimate IPv6 network architecture. Instead, it is a transitional technology.

The model using the tunnel technology is shown in the following figure:

Figure 11



The features of various tunnels are respectively described below.

2.1.1 Manually Configured IPv6 Tunnel

One manually configured tunnel is similar to one permanent link set up between two IPv6 domains via the IPv4 backbone network. It is applicable to the relatively fixed connections that have a higher requirement on security between two ABRs or between an ABR and a host.

On a tunnel interface, you must manually configure the IPv6 address, source IPv4 address (tunnel source) and destination IPv4 address (tunnel destination) of the tunnel. The nodes at the two ends of the tunnel must support the IPv6 and IPv4 protocol stacks. In practical applications, tunnels are always manually configured in pairs. You can think it as a point-to-point tunnel.

2.1.2 Configuring GRE Tunnel

A GRE tunnel allows a user to use a transport protocol (such as IP) to transmit network packets of any protocol. Our products support four types of GRE tunnels: IPv4 over IPv4, IPv6 over IPv4, IPv6 over IPv6, and IPv4 over IPv6.

On the tunnel interface, the IP address of the tunnel source and the IP address of the tunnel destination must be configured manually, and nodes at both ends of the tunnel must support IPv6 and IPv4 protocol stacks. The GRE tunnel is always configured simultaneously on two edge devices, and can be considered as a point-to-point tunnel.

Note

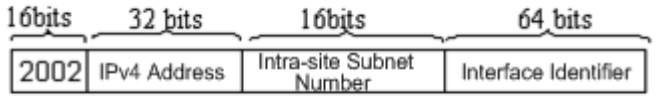
1. IPv4 over IPv6 GRE tunnel and IPv6 over IPv6 GRE tunnel are evaluation indicators.
2. IPv4 over IPv4 GRE is an evaluation indicator on the switch.
3. IPv6 over IPv4 GRE is an evaluation indicator on S5750 series switches.

2.1.3 Configuring Automatic 6to4 Tunnel

The automatic 6to4 tunnel technology allows isolated IPv6 networks to be interconnected via IPv4 networks. The difference between the automatic 6to4 tunnel and manually configured tunnel technologies is that the manual configured tunnel is a point-to-point tunnel, while a 6to4 tunnel is a point -to-multipoint tunnel.

The 6to4 tunnel uses an IPv4 network as a nonbroadcast multi-access (NBMA) link. Therefore, the devices of 6to4 need not be configured in pairs. The IPv4 address embedded in an IPv6 address will be used to look for the other end of the automatic tunnel. The 6to4 tunnel can be deemed as a point -to-multipoint tunnel. The automatic 6to4 tunnel can be configured on an ABR of one isolated IPv6 network. For each packet, it will automatically set up a tunnel to an ABR in another IPv6 network. The destination address of the tunnel is the IPv4 address of an ABR in the IPv6 network at the other end. The IPv4 address will be extracted from the destination IPv6 address of the packet. The destination IPv6 address begins with the prefix 2002::/16 in the following format.

Figure 12



IPv6 6to4 Address Format

The 6to4 address is an address for the automatic 6to4 tunnel technology. The IPv4 address embedded in it is usually the global IPv4 address of the egress of the ABR of the site. When the automatic tunnel is set up, the address is used as the destination IPv4 address for tunnel packet encapsulation. All the routers at the two ends of the 6ot4 tunnel must support the IPv6 and IPv4 protocol stacks. A 6to4 tunnel is usually configured between ABRs.

For example, if the global IPv4 address of the egress of the ABR of the site is 211.1.1.1 (D301:0101 in hexadecimal notation), a subnet number in the site is 1 and the interface identifier is 2e0:ddff:fee0:e0e1, then the corresponding 6to4 address can be denoted as follows:

2002: D301:0101:1: 2e0:ddff:fee0:e0e1

Caution

The IPv4 address embedded in the 6to4 address cannot be a private IPv4 address (i.e., the address of the network interface segment 10.0.0.0/8, 172.16.0.0/12 or 192.168.0.0/16) and must be the global IPv4 address.

Common application models of 6to4 tunnels:

- Simple application models

The simplest and most common application of 6to4 tunnels is used to interconnect multiple IPv6 sites. Each of the sites must have one connection to one of their shared IPv4 networks at least. This IPv4 network can be the Internet or a internal backbone network of an organization. The key is that each site must have a unique global IPv4 address. The 6to4 tunnel will use the address to form the IPv6 prefix of 6to4/48: 2002:IPV4 address/48.

- Hybrid application models

Based on the application described above, other 6to4 networks access the IPv6-only network through 6to4 relay devices at the edge. The router used to implement the function is called a 6to4 relay router.

2.1.4 Configuring ISATAP Automatic Tunnel

The Intra-site Automatic Tunnel Addressing Protocol (ISATAP) is a type of IPv6 tunnel technology by which an intra-site IPv6 architecture uses an IPv4 network as one nonbroadcast multi-access (NBMA) link layer, namely, using an IPv4 network as the virtual link layer of IPv6.

ISATAP is applicable to the case where the IPv6-only network inside a site is not ready for use yet and an IPv6 packet needs to be transferred internally in the site. For example, a few IPv6 hosts for test need to communicate with each other inside the site. By using an ISATAP tunnel, the IPv4/IPv6 dual stack hosts on a same virtual link can communicate with each other inside the site.

At the ISATAP site, the ISATAP device provides a standard router advertisement message, allowing the ISATAP host to be automatically configured inside the site. At the same time, the ISATAP device forwards the packets between an intra-site ISATAP host and an external IPv6 host.

The IPv6 address prefix used by ISATAP can be any legal 64-bit prefix for IPv6 unicast, including the global address prefix, link-local prefix and site-local prefix. The IPv4 address is placed as the last 32 bits of the IPv6 address, allowing a tunnel to be automatically set up.

ISATAP can be easily used with other transition technologies. Especially when used with the 6to4 tunnel technology, it can enable the dual-stack host of an intranet access an IPv6 backbone network very easily.

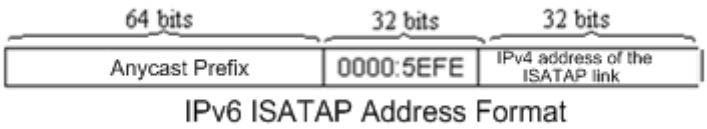
- ISATAP interface identifier

The unicast address used by ISATAP is in the form of a 64-bit IPv6 prefix plus a 64-bit interface identifier. The 64-bit interface identifier is generated in the revised EUI-64 address format. The value of the first 32 bits of the interface identifier is 0000:5EFE, an interface identifier of ISATAP.

- ISATAP address structure

An ISATAP address refers to a unicast address containing an ISATAP interface identifier in its interface identifier. An ISATAP address structure is shown in the following figure.

Figure 13



IPv6 ISATAP Address Format

The above figure shows that the interface identifier contains an IPv4 address. The address is the IPv4 address of a dual-stack host and will be used when an automatic tunnel is automatically set up.

For example, the IPv6 prefix is 2001::/64 and the embedded IPv4 address is 192.168.1.1. In the ISATAP address, the IPv4 address is denoted as the hexadecimal numeral C0A8:0101. Therefore, its corresponding ISATAP address is as follows:

2001::0000:5EFE:C0A8:0101

2.1.5 Configuring 6RD Tunnel

If you want to configure a IPv6 rapid development (6RD) tunnel, configure the tunnel interface with both the source IPv4 address and destination IPv4 address. The host or device at the peer end of the tunnel should be configured in the same way.

Command	Description

Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**interface tunnel** *tunnel-number*	Configures a tunnel by specifying the tunnel number and enters interface configuration mode.
Qtech(config-if-Tunnel id)#**tunnel mode ipv6ip 6rd**	Sets the tunnel type to 6RD tunnel,
Qtech(config-if-Tunnel id)#**ipv6 enable**	Enables the IPv6 function on the interface. You can also enable the IPv6 function directly by configuring an IPv6 address.
Qtech(config-if-Tunnel id)#**tunnel 6rd prefix** *ipv6-prefix* **/** *prefix-length*	Sets the 6RD prefix. If 6RD prefix is not configured, the 6RD tunnel cannot be up. If the prefix length is set to 0, it indicates that the prefix is deleted.
Qtech(config-if-Tunnel id)#**tunnel 6rd ipv4 prefix-length** *length* **suffix-length** *length*	Sets the IPv4 prefix and suffix length for the 6RD domain. The valid range is from 0 to 31. The sum of the prefix and suffix length cannot be greater than 31. The default is 0.
Qtech(config-if-Tunnel id)#**tunnel source** { *ipv4-address* | *interface-type interface-number* }	Sets the IPv4 source address of the tunnel or the referenced source interface number. The interface that is specified must be configured with an IPv4 address.
Qtech(config-if-Tunnel id)#**tunnel 6rd br** *ipv4-address*	Sets the border relay (BR) IPv4 address for the 6RD customer edge (CE). This command is configured only on the 6RD CE. Configuring this command allows the 6RD router to disable security check on the tunnel packet containing this source address.
Qtech(config-if-Tunnel id)#**exit**	Returns to privileged EXEC mode.
Qtech(config)#**ipv6 route** *prefix::/prefix-length next-hop*	Configures the tunnel route.

The following example configures the 6RD tunnel on the device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  Exit with CNTL/Z.
Qtech(config)#interface Tunnel 100
Qtech(config-if-Tunnel 100)#tunnel mode ipv6ip 6rd
Qtech(config-if-Tunnel 100)#tunnel source 10.1.1.1
Qtech(config-if-Tunnel 100)#tunnel 6rd prefix 2001:DA8::/32
Qtech(config-if-Tunnel 100)#tunnel 6rd ipv4 prefix-length 16 suffix-length 0
Qtech(config-if-Tunnel 100)#tunnel 6rd br 10.1.4.1
Qtech(config-if-Tunnel 100)#ipv6 enable
Qtech(config)#ipv6 address 2004::1/128
Qtech(config)#ipv6 route 2001:da8::/32 Tunnel 100
Qtech(config)#ipv6 route ::/0 Tunnel8 2001:DA8:401::1
Qtech(config)#ipv6 route 2001:da8:101::/48 Null 0
```

2.1.6 Configuring 6RD Tunnel via DHCP Automatic Configuration

You can configure the 6RD parameter for the DHCP client via the DHCP option on the DHCP server. The 6RD parameter includes the generic IPv4 prefix and suffix length, 6RD prefix length, 6RD prefix, and IPv4 address of the 6RD BR for a given 6RD domain. If you want to create a 6RD tunnel for the DHCP client, you can configure the DHCP option 212 for the client to obtain the 6RD parameter. Use the following command in DHCP address pool configuration mode to configure the 6RD parameter available for the DHCP client.

Command	**Function**
Qtech(dhcp-config)# **option 6rd ipv4masklen** <*mask-length*> **ipv6prefixlen** <*prefix-length*> **ipv6prefix** <*ipv6-prefix*> **br-addr** <*ipv4-address*>	Configures the 6RD parameter.
Use the following command to enable the DHCP 6RD client interface to obtain IP address.**Command**	**Function**

Command	Function
Qtech(config-if-GigabitEthernet 0/0)#ip address dhcp 6rd	Enables the DHCP client to obtain the IP address.

The following example configures the 6RD parameter for the DHCP client on the DHCP server.

```
Qtech#configure terminal
Enter configuration commands, one per line.  Exit with CNTL/Z.
Qtech(config)#ip dhcp pool 6rd
Qtech(dhcp-config)#option 6rd ipv4masklen 16 ipv6prefixlen 32 ipv6prefix 2002:DA8::
br-addr 1.1.1.1
```

2.2 Configuring IPv6 Tunnels

2.2.1 Manually Configuring IPv6 Tunnels

This section describes how to configure tunnels manually.

To configure a tunnel manually, configure an IPv6 address on the tunnel interface and manually configure the IPv4 addresses of the source and destination of the tunnel. Then, configure the hosts or devices at the two ends of the tunnel to ensure that they support the dual stacks (the IPv6 and IPv4 protocol stacks).

Caution Do not configure tunnels manually with the same runnel source and tunnel destination.

Brief steps

```
config terminal
interface tunnel tunnel-num
tunnel mode ipv6ip
ipv6 enable
tunnel source {ip-address | type num}
tunnel destination ip-address
end
```

To configure an IPv6 tunnel manually, use the following commands in global configuration mode.

Command	Function
configure terminal	Enters global configuration mode.
interface tunnel *tunnel-num*	Specifies a tunnel interface number to create a tunnel interface and enters interface configuration mode.
tunnel mode ipv6ip	Sets the tunnel type to manually configured tunnel.
ipv6 enable	Enables the IPv6 function on the interface. You can also configure the IPv6 address to directly enable the IPv6 function on the interface.
tunnel source {*ip-address* | *type num*}	Specifies the IPv4 source address or referenced source interface number of the tunnel. Note: If you specify an interface, the IPv4 address must have been configured on the interface.
tunnel destination *ip address*	Specifies the destination address of the tunnel.
end	Returns to privileged EXEC mode.
copy running-config startup-config	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" section to check the operation of the tunnel.

2.2.2 Configuring GRE Tunnels

This section describes how to configure GRE tunnels.

To configure a GRE tunnel, you need to manually configure the tunnel source IP address and tunnel destination IP address on the tunnel interface. The corresponding configurations must also be done on the peer host or device.

Caution Do not configure a GRE tunnel with the same tunnel source IP address and tunnel destination IP address on the device.

Use the following commands to configure a GRE tunnel.

Command	Function
Qtech# **configure terminal**	Enters global configuration mode.
Qtech(config)# **interface tunnel** *tunnel-num*	Specifies the tunnel interface number to create a tunnel interface and enters interface configuration mode.
Qtech(config-if-Tunnel id)# **tunnel mode gre** {**ip** | **ipv6**}	Sets the tunnel type to GRE tunnel, and specifies the carrier protocol as IPv4 or IPv6.
Qtech(config-if-Tunnel id)# **tunnel source** {*ipv4-address* | *ipv6-address* | *interface-type interface-num* }	Specifies the IPv4 address of the tunnel source or source interface number referenced. If the interface is specified, the IPv4 address must have been configured on the interface.
Qtech(config-if-Tunnel id)# **tunnel destination** *{ipv4-address* | *ipv6-address}*	Specifies the destination address of the tunnel. If the carrier protocol is IPv6, the IP address must be configured as the IPv6 address of the peer device.
Qtech(config-if-Tunnel id)# **end**	Returns to privileged EXEC mode.
Qtech# **copy running-config startup-config**	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" section to check the operation of the tunnel.

Caution Because GRE features differ from device to device, the aforementioned commands may not be available on certain products.

2.2.3 Configuring 6to4 Tunnels

This section describes how to configure a 6to4 tunnel.

The destination address of a 6to4 tunnel is determined by the IPv4 address which is extracted from a 6to4 IPv6 address. The devices at the two ends of the 6to4 tunnel must support the dual stacks, namely, the IPv4 and IPv6 protocol stacks.

Caution A device supports only one 6to4 tunnel. The encapsulation source address (IPv4 address) used by the 6to4 tunnel must be a globally routable address. Otherwise, the 6to4 tunnel will not work normally.

Brief steps

```
config terminal
interface tunnel tunnel-num
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source {ip-address | type num}
exit
ipv6 route 2002::/16 tunnel tunnel-number
end
```

To configure a 6to4 tunnel, use the following commands.

Command	Function
configure terminal	Enters global configuration mode.
interface tunnel *tunnel-num*	Specifies a tunnel interface number to create a tunnel interface and enters interface configuration mode.
tunnel mode ipv6ip 6to4	Sets the tunnel type to 6to4 tunnel.

Command	Function
ipv6 enable	Enables the IPv6 function of the interface. You can also configure the IPv6 address to directly enable the IPv6 function of the interface.
tunnel source {*ip-address* | *type num*	Specifies the encapsulation source address or referenced source interface number of the tunnel. Note: The IPv4 address must have been configured on the referenced interface. The used IPv4 address must be a globally routable address.
Exit	Returns to global configuration mode.
ipv6 route *2002::/16* **tunnel** *tunnel-number*	Configures a static route for the IPv6 6to4 prefix 2002::/16 and associates the output interface with the tunnel interface, i.e., the tunnel interface specified above.
End	Returns to privileged EXEC mode.
copy running-config startup-config	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" to check the operation of the tunnel.

2.2.4 Configuring ISATAP Tunnels

This section describes how to configure ISATAP tunnels.

On an ISATAP tunnel interface, the configuration of an ISATAP IPv6 address and the advertisement configuration of a prefix are the same as that of a common IPv6 interface. However, the address configured for an ISATAP tunnel interface must be a revised EUI-64 address. The reason is that the last 32 bits of the interface identifier in the IPv6 address are composed of the IPv4 address of the interface referenced by the tunnel source address. See the above sections for the information about ISATAP address formats.


Caution

A device supports multiple ISATAP tunnels. However, the source of each ISATAP tunnel must be different.
Otherwise, there is no way to know which ISATAP tunnel a received ISATAP tunnel message belongs to.

Brief steps

```
config terminal
interface tunnel tunnel-num
tunnel mode ipv6ip isatap
ipv6 address ipv6-prefix/prefix-length eui-64
tunnel source interface-type num
no ipv6 nd suppress-ra
end
```

To configure an ISATAP tunnel, use the following commands.

Command	Function
configure terminal	Enters global configuration mode.
interface tunnel *tunnel-num*	Specifies a tunnel interface number to create a tunnel interface and enters interface configuration mode.
tunnel mode ipv6ip isatap	Sets the tunnel type to ISATAP tunnel.
ipv6 address ipv6-prefix/prefix-length eui-64	Configures the IPv6 ISATAP address. Be sure to use the **eui-64** keyword. In this way, the ISATAP address will be automatically generated. The address configured on an ISATAP interface must be an ISATAP address.
tunnel source type *num*	Specifies the source interface number referenced by the tunnel. On the referenced interface, the IPv4 address must have been configured.
no ipv6 nd suppress-ra	Sending router advertisement messages on an interface is disabled by default. You can use the command to enable the function, allowing the ISATAP host to be automatically configured.

Command	Function
End	Returns to privileged EXEC mode.
copy running-config startup-config	Saves the configuration information.

See the "Verifying and Monitoring IPv6 Tunnel Configuration" to check the operation of the tunnel.

2.2.5 Configuring the Tunnel to Support IPv6 Multicast

Currently, on the IPv6 network, both IPv6 unicast and multicast services need to be able to traverse the IPv4 network.

It is easy to configure IPv6 tunnel multicast. The tunnel interface can be configured in the same way as other common interfaces such as an SVI interface.

Caution

Multicast is supported only in the manually configured IPv6 tunnel. For tunnels of other types, multicast can be configured, but multicast data cannot be received or forwarded after the configuration.
For the manually configured IPv6 tunnel, if the tunnel is created based on an IPv6 tunnel of any type, multicast can be configured, but multicast data cannot be received or forwarded after the configuration.
When IPv6 multicast data traverses the IPv4 network, the MTU restrictions are the same as those for IPv6 unicast data.

2.3 Verifying and Monitoring IPv6 Tunnel Configuration

This section describes how to verify the configuration and operation of an IPv6 tunnel.

Brief steps

```
enable
show interface tunnel number
show ipv6 interface tunnel mumber
ping protocol destination
show ip route
show ipv6 route
```

To verify the configuration and operation of a tunnel, use the following commands.

Command	Function
show interface tunnel *tunnel-num*	Displays the information of a specified tunnel interface.
show ipv6 interface tunnel *tunnel-num*	Displays the IPv6 information of the tunnel interface.
ping protocol destination	Checks the basic connectivity of a network.
show ip route	Displays the IPv4 routing table.
show ipv6 route	Displays the IPv6 router table.

8) Display the information of a tunnel interface.

```
Qtech# show interface tunnel 1
Tunnel 1 is up, line protocol is Up
Hardware is Tunnel, Encapsulation TUNNEL
Tunnel source 192.168.5.215  ,  destination 192.168.5.204
Tunnel protocol/transport IPv6/IP
Tunnel TTL is 9
Tunnel source do conformance check set
Tunnel source do ingress filter set
Tunnel destination do safety check not set
Tunnel disable receive packet not set
```

9) Display the IPv6 information of a tunnel interface.

```
Qtech# show ipv6 interface tunnel 1
interface Tunnel 1 is Up, ifindex: 6354
address(es):
Mac Address: N/A
INET6: fe80::3d9a:1601 , subnet is fe80::/64
Joined group address(es):
```

```
ff02::2
ff01::1
ff02::1
ff02::1:ff9a:1601
INET6: 3ffe:4:0:1::1 , subnet is 3ffe:4:0:1::/64
Joined group address(es):
ff02::2
ff01::1
ff02::1
ff02::1:ff00:1
MTU is 1480 bytes
ICMP error messages limited to one every 100 milliseconds
ICMP redirects are enabled
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 milliseconds
ND retransmit interval is 1000 milliseconds
ND advertised retransmit interval is 0 milliseconds
ND router advertisements are sent every 200 seconds<240--160>
ND router advertisements live for 1800 seconds
```

2.4 IPv6 Tunnel Configuration Examples

2.4.1 Example of Configuring IPv6 Tunnels Manually

Figure 14

As shown in the above figure, IPv6 networks N1 and N2 are isolated by the IPv4 network. Now, the two networks are interconnected by configuring a tunnel manually. For example, the H-A3 host in N1 can access the H-B3 host in N2.

In the figure, RT-A and RT-B are routers that support the IPv4 and IPv6 protocol stacks. Tunnel configuration is performed on the ABRs (RT-A and RT-B) in N1 and N2. Note that the tunnel must be configured manually in pairs, that is, on RT-A and RT-B.

The following presents the tunnel configuration on routers:

Prerequisite: Assume that the routes of IPv4 are connected. In the following content, no more route configuration condition about IPv4 is listed.

RT-A:

#Connect the interfaces of the IPv4 network.

```
interface FastEthernet 2/1
no switchport
ip address 192.1.1.1 255.255.255.0
```

#Connect the interfaces of the IPv6 network.

```
interface FastEthernet 2/2
no switchport
ipv6 address 2001::1/64
```

```
no ipv6 nd suppress-ra  (optional)
```

#Configure the manual tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip
ipv6 enable
tunnel source FastEthernet 2/1
tunnel destination 211.1.1.1
```

#Configure the route to the tunnel.

```
ipv6 route 2005::/64 tunnel 1
```

RT-B:

#Connect the interfaces of the IPv4 network.

```
interface FastEthernet 2/1
no switchport
ip address 211.1.1.1 255.255.255.0
```

Connect the interfaces of the IPv6 network.

```
interface FastEthernet 2/2
no switchport
ipv6 address 2005::1/64
no ipv6 nd suppress-ra  (optional)
```

#Configure the manual tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip
ipv6 enable
tunnel source FastEthernet 2/1
tunnel destination 192.1.1.1
```

#Configure the route to the tunnel.

```
ipv6 route 2001::/64 tunnel 1
```

2.4.2 Example of Manually Configuring IPv6 Tunnels to Support Multicast

Assume that the network topology is shown in Figure 4. On the basis of the previous example, the additional support to PIM SMv6 multicast is required. Detailed configurations related to multicast are shown below:

- RT-A

Globally enable multicast.

```
ipv6 multicast-routing
```

Enable PIM SMv6 on the interface.

```
interface Tunnel 1
IPv6 pim sparse-mode
```

- RT-B

Globally enable multicast.

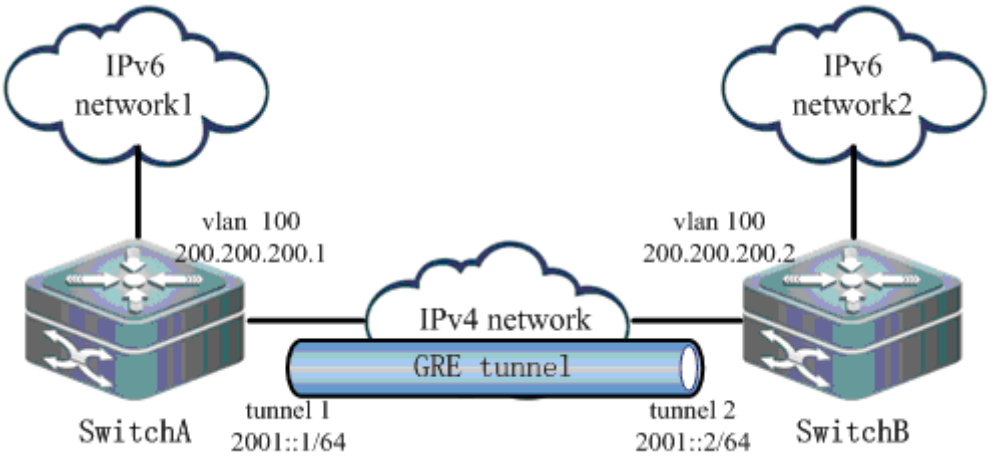
```
ipv6 multicast-routing
```

Enable PIM SMv6 on the interface.

```
interface Tunnel 1
IPv6 pim sparse-mode
```

2.4.3 Example of Configuring IPv6 over IPv4 GRE Tunnels

Figure 15



As shown in Figure 15, two IPv6 networks, IPv6 network1 and IPv6 network2, need to be connected via a public IPv4 network to realize intercommunication. Layer-3 devices Switch A and Switch B supports both IPv4 and IPv6 stacks, and are interconnected via the IPv4 network. An IPv6 over IPv4 GRE tunnel needs to be created over this IPv4 network.

Assuming that IPv6 network1 is 2002::/64 and IPv6 network2 is 2003::/64, the configurations on Switch A and Switch B are shown below:

10) Configure Switch A.

Configure interface vlan 100.

```
interface vlan 100
ip address 200.200.200.1 255.255.255.0
```

Configure interface Tunnel 1.

```
interface Tunnel 1
 ipv6 address 2001::1/64
 tunnel mode gre ip
 tunnel source vlan 100
 tunnel destination 200.200.200.2
```

Configure the route to pass through the tunnel interface to reach IPv6 network2.

```
ipv6 route 2003::/64 tunnel 1 2001::2
```

11) Configure Switch B.

Configure interface vlan 100.

```
interface vlan 100
 ip address 200.200.200.2 255.255.255.0
```

Configure interface Tunnel 1.

```
interface Tunnel1
 ipv6 address 2001::2/64
 tunnel mode gre ip
 tunnel source vlan 100
 tunnel destination 200.200.200.1
```

Configure the route to pass through the tunnel interface to reach IPv6 network1.

```
ipv6 route 2002::/64 tunnel 1 2001::1
```

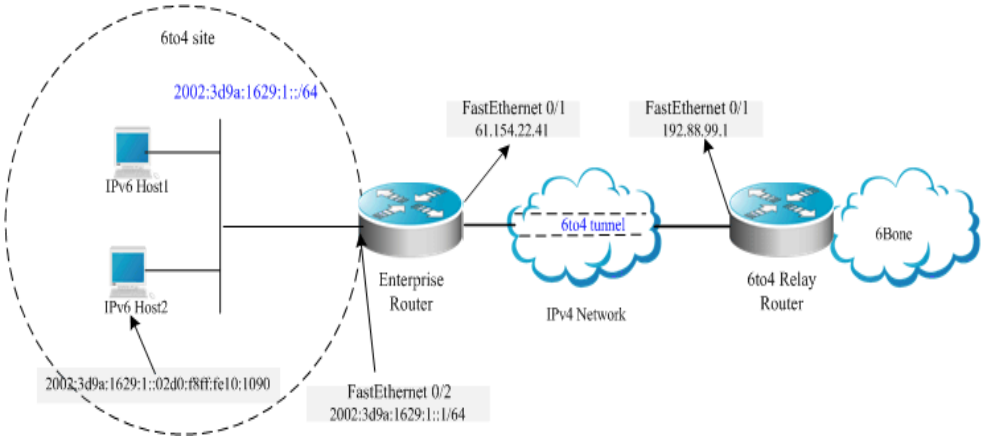
12) View the operation of the tunnel, taking Switch A as an example.

```
show interface tunnel 1
Index(dec):3 (hex):3
Tunnel 1 is UP  , line protocol is UP
Hardware is  Tunnel
Interface address is: no ip address
  MTU 1496 bytes, BW 9 Kbit
  Encapsulation protocol is Tunnel, loopback not set
  Keepalive set (10 sec), retries 3
  Carrier delay is 2 sec
  RXload is 1 ,Txload is 1
```

```
Tunnel source 200.200.200.1 (VLAN 100), destination 200.200.200.2
Tunnel TOS 0x14,  Tunnel TTL 255
Tunnel protocol/transport GRE/IP
 Key disabled, Sequencing disabled
 Checksumming of packets disabled
Path MTU Discovery, ager 10 mins, min MTU 92, MTU 0, expires never
Queueing strategy: FIFO
  Output queue 0/40, 0 drops;
  Input queue 0/75, 0 drops
5 minutes input rate 0 bits/sec, 0 packets/sec
5 minutes output rate 0 bits/sec, 0 packets/sec
  0 packets input, 0 bytes, 0 no buffer, 0 dropped
  Received 0 broadcasts, 0 runts, 0 giants
  0 input errors, 0 CRC, 0 frame, 0 overrun, 0 abort
  0 packets output, 0 bytes, 0 underruns , 0 dropped
  0 output errors, 0 collisions, 0 interface resets
```

2.4.4 Example of Configuring 6to4 Tunnels

Figure 16



As shown in the above figure, an IPv6 network (6to4 site) uses a 6to4 tunnel to access the IPv6 backbone network (6bone) via a 6to4 relay router.

As described above, the 6to4 tunnel technology is used to interconnect isolated IPv6 networks and the IPv6 backbone network can be accessed via the 6to4 relay router very easily. The 6to4 tunnel is an automatic tunnel and the IPv4 address embedded in the IPv6 address will be used to look for the other end of the automatic tunnel. Therefore, you need not configure the destination end for the 6to4 tunnel. Additionally, unlike a manual tunnel, the 6to4 tunnel need not be configured in pairs.

61.154.22.41 is 3d9a:1629 in hexadecimal notation.

192.88.99.1 is c058:6301 in hexadecimal notation.

Caution When configuring a 6to4 tunnel on an ABR, be sure to use a globally routable IPv4 address. Otherwise, the 6to4 tunnel will not work normally.

The following is the configuration of the two routers in the figure (Assume that IPv4 routes are connected. Ignore the configuration of IPv4 routes.):

Enterprise router:

Connect the interfaces of the IPv4 network.

```
interface FastEthernet 0/1
no switchport
ip address 61.154.22.41 255.255.255.128
```

Connect the interfaces of the IPv6 network.
```
interface FastEthernet 0/2
no switchport
ipv6 address 2002:3d9a:1629:1::1/64
no ipv6 nd suppress-ra
```

Configure the 6to4 tunnel interface.
```
interface Tunnel 1
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source FastEthernet 0/1
```

Configure the route to the tunnel.
```
ipv6 route 2002::/16 Tunnel 1
```

Configure the route to the 6to4 relay router to access 6bone.
```
ipv6 route ::/0 2002:c058:6301::1
```

ISP 6to4 relay router:

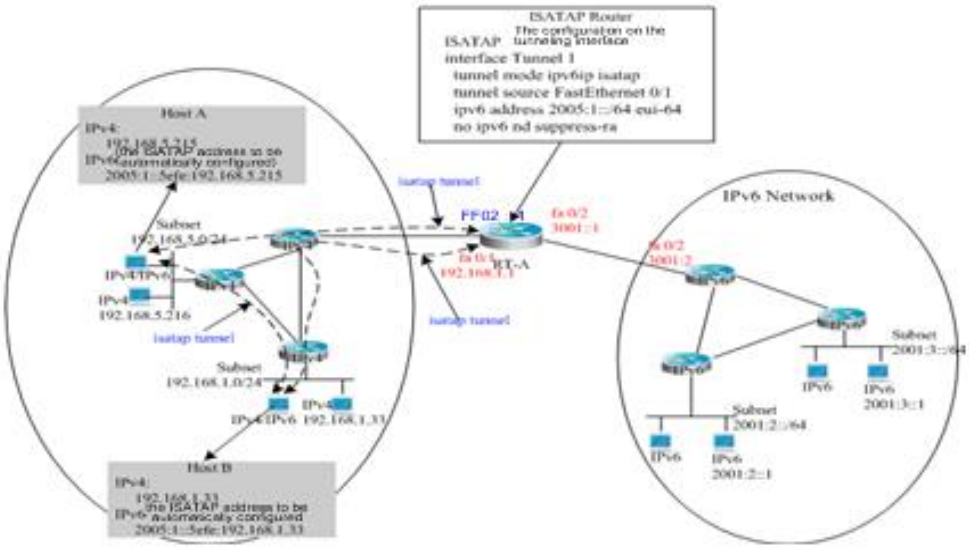
Connect the interface of the IPv4 network.
```
interface FastEthernet 0/1
no switchport
ip address 192.88.99.1 255.255.255.0
```

Configure the 6to4 tunnel interface.
```
interface Tunnel 1
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source FastEthernet 0/1
```

Configure the route to the tunnel.
```
ipv6 route 2002::/16 Tunnel 1
```

2.4.5 Example of Configuring ISATAP Tunnels

Figure 17



The above figure is one typical topology using an ISATAP tunnel. The ISATAP tunnel is used to communicate between isolated IPv4/IPv6 dual-stack hosts inside the IPv4 site. The ISATAP router has the two following functions inside the ISATAP site:

- Receive a router solicitation message from the ISATAP host inside the site and then respond with a router advertisement message for the ISATAP host inside the site to be automatically configured.

- Be responsible for the message forwarding function of the ISATAP host inside the site and the IPv6 host outside the site.

In the above figure, when Host A and Host B send the router solicitation message to the ISATAP router, the ISATAP router will respond with a router advertisement message. After receiving the message, the hosts will automatically perform configuration and generate their own ISATAP addresses respectively. Then, the IPv6 communication between Host A and Host B will be done via the ISATAP tunnel. When Host A or Host B need to communicate with the IPv6 host outside the site, Host A sends the message to the ISATAP router RT-A via the ISATAP tunnel and then RT-A forwards the message to the IPv6 network.

In the above figure, the ISATAP router (RT-A) is configured as follows:

Connect the interfaces of the IPv4 network.

```
interface FastEthernet 0/1
no switchport
ip address 192.168.1.1 255.255.255.0
```

Configure the ISATAP tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2005:1::/64 eui-64
no ipv6 nd suppress-ra
```

Connect the interfaces of the IPv6 network.

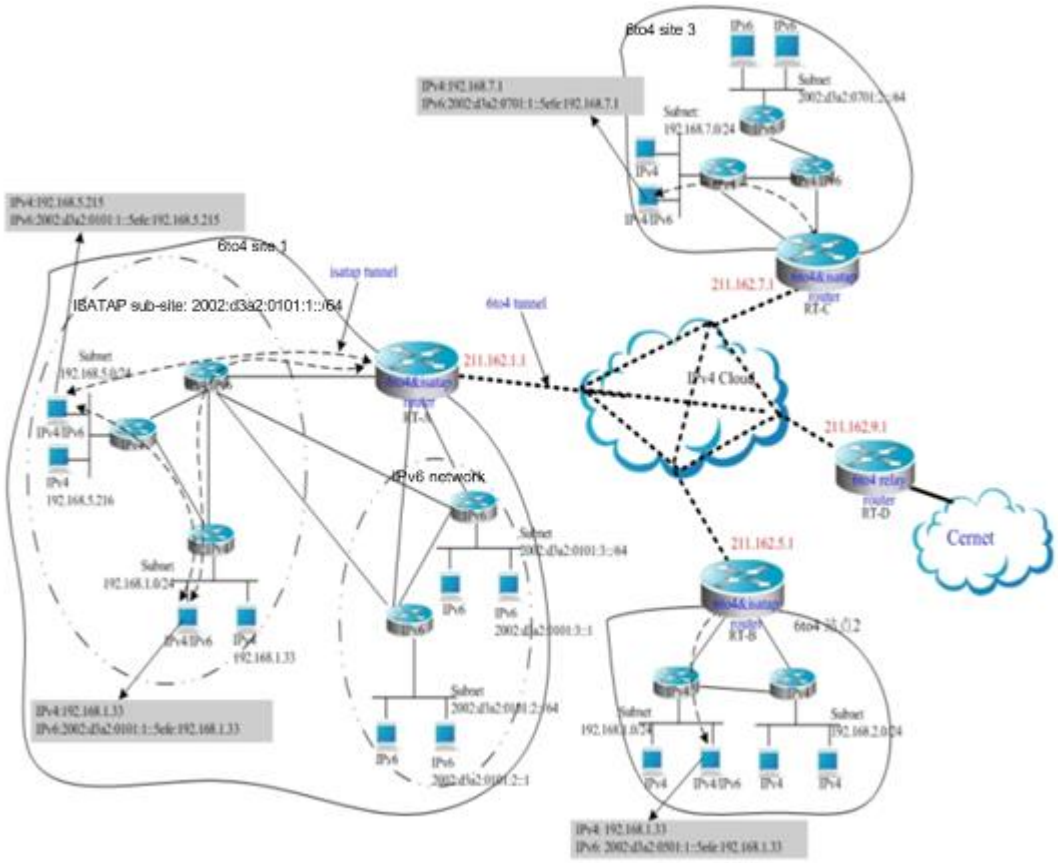
```
interface FastEthernet  0/2
no switchport
ipv6 address 3001::1/64
```

Configure the route to the IPv6 network.

```
ipv6 route 2001::/64 3001::2
```

2.4.6 Example of Configuring ISATAP and 6to4 Tunnels

Figure 18



Note

The above figure shows a hybrid application of a 6to4 tunnel and an ISATAP tunnel. By using the 6to4 tunnel technology, various 6to4 sites are interconnected and the 6to4 sites access the Cernet network via the **6to4 relay router**. At the same time, by using the ISATAP tunnel technology inside the 6to4 sites, the IPv6 hosts isolated by IPv4 inside the sites perform IPv6 communication via the ISATAP tunnel.

Caution

In the above figure, the used global IP addresses including the address of the 6to4 relay router are only for convenience. When actually planning topologies, you should use a true global IP address and the address of the 6to4 relay. At present, many organizations provide the addresses of open and free 6to4 relay routers.

The configurations of ABRs at the 6to4 sites shown in the above figure are described respectively below. Note that only main related configurations are listed here.

RT-A:

Connect the interfaces of the Internet.

```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.1.1 255.255.255.0
```

Connect the interfaces of the IPv4 network inside the site.

```
interface FastEthernet  0/1.
no switchport
ip address 192.168.0.1 255.255.255.0
```

Configure the ISATAP tunnel interface.
```
interface Tunnel 1
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2002:d3a2:0101:1::/64 eui-64
no ipv6 nd suppress-ra
```

Connect interface 1 of the IPv6 network.
```
interface FastEthernet 0/2
no switchport
2002:d3a2:0101:10::1/64
```

Connect interface 2 of the IPv6 network.
```
interface FastEthernet  0/2
no switchport
2002:d3a2:0101:20::1/64
```

Configure the 6to4 tunnel interface.
```
interface Tunnel 2
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source GigabitEthernet 0/1
```

Configure the route to the 6to4 tunnel.
```
ipv6 route 2002::/16 Tunnel 2
```

Configure the route to the 6to4 relay router RT-D to access the Cernet network.
```
ipv6 route ::/0 2002:d3a2::0901::1
```

RT-B:

Connect the interfaces of the Internet.
```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.5.1 255.255.255.0
```

Connect interface 1 of the IPv4 network inside the site.
```
interface FastEthernet  0/1
no switchport
ip address 192.168.10.1 255.255.255.0
```

Connect interface 2 of the IPv4 network inside the site.
```
interface FastEthernet  0/2
no switchport
ip address 192.168.20.1 255.255.255.0
```

Configure the ISATAP tunnel interface.
```
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2002:d3a2:0501:1::/64 eui-64
no ipv6 nd suppress-ra
```

Configure the 6to4 tunnel interface.
```
interface Tunnel 2
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source GigabitEthernet 0/1
```

Configure the route to the 6to4 tunnel.
```
ipv6 route 2002::/16 Tunnel 2
```

Configure the route to the 6to4 relay router RT-D to access the Cernet network.
```
ipv6 route ::/0 2002:d3a2::0901::1
```

RT-C:

Connect the interfaces of the Internet.
```
interface GigabitEthernet 0/1
no switchport
```

```
ip address 211.162.7.1 255.255.255.0
```

Connect the interfaces of the IPv4 network inside the site.

```
interface FastEthernet 0/1
no switchport
ip address 192.168.0.1 255.255.255.0
```

Configure the ISATAP tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip isatap
tunnel source FastEthernet 0/1
ipv6 address 2002:d3a2:0701:1::/64 eui-64
no ipv6 nd suppress-ra
```

Connect the interfaces of the IPv6 network.

```
interface FastEthernet 0/2
no switchport
2002:d3a2:0701:10::1/64
```

Configure the 6to4 tunnel interface.

```
interface Tunnel 2
tunnel mode ipv6ip 6to4
ipv6 enable
tunnel source GigabitEthernet 0/1
```

Configure the route to the 6to4 tunnel.

```
ipv6 route 2002::/16 Tunnel 2
```

#Configure the route to the 6to4 relay router RT-D to access the Cernet network.

```
ipv6 route ::/0 2002:d3a2::0901::1
```

RT-D (6to4 relay):

Connect the interfaces of the Internet.

```
interface GigabitEthernet 0/1
no switchport
ip address 211.162.9.1 255.255.255.0
```

Connect the interfaces of the IPv6 network.

```
interface FastEthernet 0/1
no switchport
2001::1/64
no ipv6 nd suppress-ra
```

Configure the 6to4 tunnel interface.

```
interface Tunnel 1
tunnel mode ipv6ip 6to4
ipv6 address 2002:d3a2::0901::1/64
tunnel source GigabitEthernet 0/1
```

#Configure the route to the 6to4 tunnel.

```
ipv6 route 2002::/16 Tunnel 1
```

3 CONFIGURING NAT-PT

3.1 Overview

With the rapid development of the Internet, IPv4 becomes inadequate for the Internet, and IPv6 deployment has been on the agenda. To implement an IPv6 network, you are advised make the most of the existing network environment to build the next generation of the Internet to achieve smooth evolution and avoid excessive investment waste. As the Internet is currently based on IPv4 and unlikely to completely evolve to the IPv6 network within a short time, IPv6 and IPv4 networks will co-exist for quite some time.

That brings a challenge to maintain services and functions of the existing network and achieve transparent transmission between IPv4 and IPv6 networks at a low cost. Network Address Translation-Protocol Translation (NAT-PT) emerges in response to the communication between directly connected IPv6 and IPv4 networks.

Free communication between IPv6 and IPv4 networks can be ensured by using NAT-PT. Communication can be initiated by a host on either network. You can use NAT-PT to translate protocols and semantics without transforming or upgrading the host.

3.1.1 Basic Concepts

NAT for the IPv4 network is adopted and improved for IPv4-IPv6 address translation. It aims to establish and maintain address mappings.

PT is responsible for IPv4-IPv6 protocol translation. It builds new packets by replacing the IPv6 header with the IPv4 header or vice versa. Only certain types of the Internet Control Message Protocol version 4 (ICMPv4) and ICMPv6 packets can be translated because of their protocols, for example, translation between ICMPv4 and ICMPv6 request/response packets and translation between ICMPv4 and ICMPv6 destination-unreachable packets.

As a mode of dynamic address translation, Port Address Translation (PAT) is used for IPv6-IPv4 dynamic translation. Multiple IPv6 addresses can be mapped to the same IPv4 address differentiated by different ports to avoid IPv4 address depletion.

3.1.2 Working Principle

NAT-PT works on the border router between IPv6 and IPv4 networks. The NAT-PT module translates IP header addresses between IPv6 and IPv4 networks, and translates semantics of groups according to different protocols to achieve transparent transmission between IPv4 and IPv6 networks. NAT-PT can replace addresses statically or use an address pool containing global addresses. When a session passes through the border router, the border router takes an address from the address pool and assigns the address to the correct IPv4/IPv6 host. At the same time, to track the session to be translated, the session must pass though the same NAT-PT border router.

3.1.3 Protocols and Standards

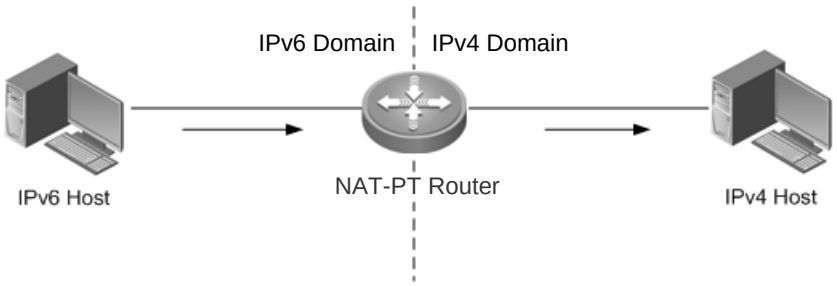
RFC2765: Stateless IP/ICMP Translation Algorithm (SIIT).

RFC2766: Network Address Translation - Protocol Translation (NAT-PT).

3.1.4 Applications

Scenario 1

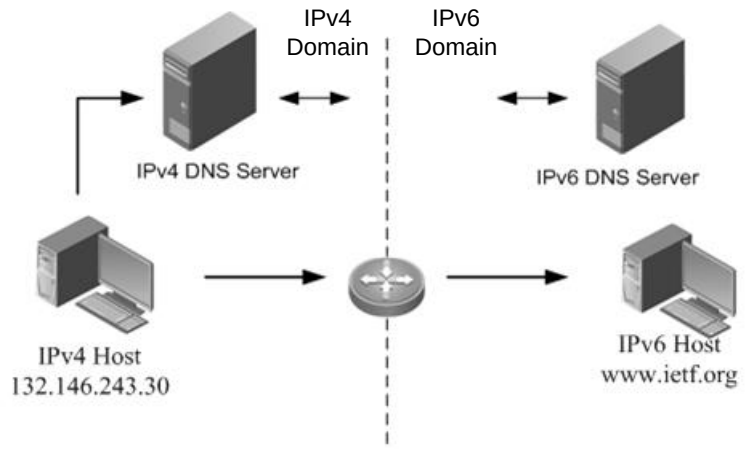
Figure 3–1 IP Address-based Access



In this scenario, NAT-PT aims at interworking between the IPv6 host and the IPv4 host to implement transparent transmission between IPv6 and IPv4 networks at a low cost without changing the topology or incurring extra expenses during packet transmission.

Scenario 2

Domain Name-based Access



In this scenario, as the IPv4 DNS packets and IPv6 DNS packets have different formats, the IPV6 DNS server cannot identify DNS requests from the IPv4 host and thereby cannot perform DNS resolution. NAT-PT is used to enable domain name-based access to the destination host, so that the IPv6 DNS server can respond to DNS requests from the IPv4 host and that domain name-based transparent transmission can be implemented between the IPv4 host and the IPv6 host.

3.2 Configuring NAT-PT

Defaults

Feature	Default Settings
NAT-PT	Disabled.
Maximum NAT-PT entry	100K
NAT-PT timeout	300 seconds
dns-timeout	300 seconds
finrst-timeout	60 seconds
icmp-timeout	90 seconds
syn-timeout	60 seconds
tcp-timeout	24 hours
udp-timeout	300 seconds

3.2.1 Configuring Static Source Address-based NAT-PT Mapping

Command	Description
Qtech>**enable**	Enters privileged EXEC mode.

Command	Description
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 nat prefix***ipv6-prefix/prefix-length*	Assigns an IPv6 prefix as the global NAT-PT prefix.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Qtech(config-if)#**exit**	Returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Qtech(config-if)#**exit**	Returns to global configuration mode.
Qtech(config)#**ipv6nat v6v4 source** *ipv6-address ipv4-address*	Configures the static IPv6-IPv4 address mapping.
Qtech(config)#**ipv6 nat v4v6 source** *ipv4-address ipv6-address*	Configures the static IPv4-IPv6 address mapping.

Example: Configuring static source address-based NAT-PT mappings

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface fastethernet 0/1
Qtech(config-if)#ipv6 nat
RouterB(config-if)#ipv6 address 2001::1/64
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v4v6 source 8.0.0.2 2001:DA8:1::5
RouterB(config)#ipv6 nat v6v4 source 2001::18.0.0.5
```

3.2.2 Configuring Dynamic Source Address-based NAT-PT Mappings

Command	Description
Qtech>**enable**	Enters privileged EXEC mode.
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 nat prefix***ipv6-prefix/prefix-length*	Assigns an IPv6 prefix as the global NAT-PT prefix.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Qtech(config-if)#**exit**	Returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**ipv6 nat**	Enables NAT-PT for the interface.
Qtech(config-if)#**exi**t	Returns to global configuration mode.
Qtech(config)#**ipv6 nat v6v4 source list***access-list-name***pool***v4pool-name* or Qtech(config)#**ipv6 nat v4v6 source list** *access-list-name***pool***v6pool-name*	Enables the dynamic IPv6-IPv4 or IPv4-IPv6 address mapping. Takes an address from *v4pool-name* (name of the IPv6 address pool) for translation to match the IPv6 Access Control List (ACL) entry **access-list-name**; or takes an address from *v6pool-name* (name of the IPv4 address pool) for translation to match the IPv4 ACL entry **access-list-name**.
Qtech(config)#**ipv6 nat v6v4 pool** *v4pool-name start-ipv4 end-ipv4***prefix-length** *prefix-length* or Qtech(config)#**ipv6 nat v4v6 pool***v6pool-namestart-ipv6 end-ipv6***prefix-length** *prefix-length*	Configures the IPv4 address pool with a specified prefix length or configures the IPv6 address pool with a specified prefix length.

Qtech(config)#**ipv6 access-list** *access-list-name* or Qtech(config)#**ip access-list**{ **standard** | **extended** } { *id* | *access-list-name* }	Configures IPv6 or IPv4 ACL Permit entries.

Example: Configuring dynamic source address-based NAT-PT mappings

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface fastethernet 0/1
Qtech(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v6v4 source list v6_acl pool v4pool
RouterB(config)#ipv6 nat v6v4 pool v4pool 20.0.0.1 20.0.0.30 prefix-length 24
RouterB(config)#ipv6 nat v4v6 source 8.0.0.1 2001:DA8:1::5
RouterB(config)#ipv6 access-list v6_acl
RouterB(config-ipv6-acl)#permit ipv6 2001:DA8:2::/64 any
```

3.3 Monitoring

Command	Description
Qtech#**clear ipv6 nat statistics**	Clears all NAT-PT statistics.
Qtech#**clear ipv6 nat translations ***	Clears all records of NAT-PT.
Qtech#**clear ipv6 nat translations icmp**	Clears all records of NAT-PT ICMP.
Qtech#**clear ipv6 nat translations tcp**	Clears all records of NAT-PT Transmission Control Protocol (TCP).
Qtech#**clear ipv6 nat translations udp**	Clears all records of NAT-PT User Datagram Protocol (UDP).
Qtech#**debug ipv6 nat**	Displays NAT-PT debugging information.
Qtech#**show ipv6 nat memory**	Displays NAT-PT memory information.
Qtech#**show ipv6 nat pool**	Displays NAT-PT address pools.
Qtech#**show ipv6 nat rule**	Displays translation rules.
Qtech#**show ipv6 nat statistics**	Displays NAT-PT statistics.
Qtech#**show ipv6 nat translations**	Displays records of NAT-PT.

3.4 Configuration Example

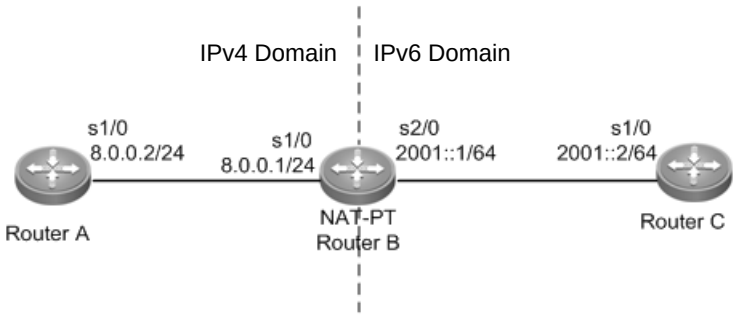
3. 4. 1 Static NAT-PT

Networking Requirements

Router C whose IPv6 address is 2001::2/64 wants to interwork with Router A whose IPv4 address is 8.0.0.2/24. For this purpose, you need to deploy Router B as an NAT-PT device between the IPv4 domain and the IPv6 domain and configure static IPv4 and IPv6 packet mappings on Router B to implement interworking between the IPv4 domain and the IPv6 domain.

Network Topology

Static NAT-PT Configuration



Configuration Tips

The following describes the Configuration Tips of configuration for static NAT-PT mappings:

- Configure an IPv4 address.
- Configure an IPv6 address.
- Configure the NAT-PT prefix and enable NAT-PT.
- Configure static IPv4 and IPv6 packet mappings.
- Configure the static route for IPv4 or IPv6.

Caution When configuring the static source address-based mapping on Router C, you must avoid confliction between the mapped source IPv4 address and other addresses in the IPv4 domain to ensure a reachable route from Router A to the NAT-PT device (Router B).

Steps

- Configure the IP address of Router A in the IPv4 domain.

```
RouterA>enable
RouterA#configure terminal
RouterA(config)#interface s1/0
RouterA(config-if)#ip address 8.0.0.2 255.255.255.0
```

- Configure the IP address of Router C in the IPv6 domain.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#interface s1/0
RotuerC(config-if)#ipv6 enable
RouterC(config-if)#ipv6 address 2001::2/64
```

- Configure Router B.

```
RouterB>enable
RouterB#configure terminal
RouterB(config)#ipv6 nat prefix 2001:DA8:1::/96
RouterB(config)#interface s1/0
RouterB(config-if)#ip address 8.0.0.1 255.255.255.0
RouterB(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#interface s2/0
RouterB(config-if)#ipv6 enable
RouterB(config-if)#ipv6 nat
RouterB(config-if)#ipv6 address 2001::1/64
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v4v6 source 8.0.0.2 2001:DA8:1::5
RouterB(config)#ipv6 nat v6v4 source 2001::2 8.0.0.5
```

- On Router C, configure a static route destined for the network segment matched to the NAT-PT prefix.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#ipv6 route 2001:DA8:1::/96 2001::1
```

Verification

Qtech#**show ipv6 nat translations**

Prot IPv4 source IPv6 source

IPv4 destination IPv6 destination

--- 8.0.0.52001::2

--- ---

--- --- ---

8.0.0.2 2001:DA8:1::1

icmp 8.0.0.2,47 2001:DA8:1::5,47

8.0.0.5,47 2001::2,47

Run the **ping 8.0.0.5** command on Router A to receive response packets.

Qtech#**ping 8.0.0.5**

Sending 5, 100-byte ICMP Echoes to 8.0.0.5, timeout is 2 seconds:

< press Ctrl+C to break >

!!!!!

Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms

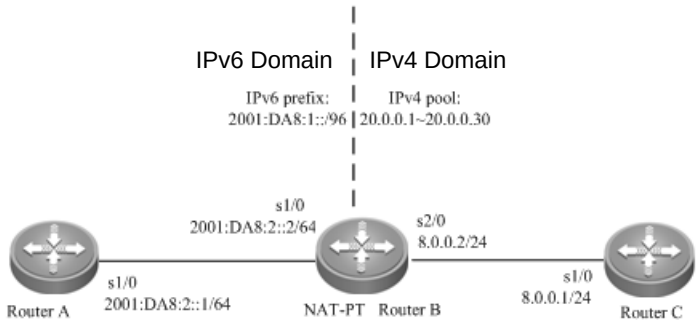
3.4.2 Dynamic NAT-PT

Networking Requirements

Router A whose IPv6 address is 2001:DA8:2::1 wants to access Router C whose IPv4 address is 8.0.0.1, but Router C is forbidden to access Router A. For this purpose, you need to deploy Router B between the IPv6 domain and the IPv4 domain for dynamic NAT-PT.

Network Topology

Dynamic NAT-PT configuration



Configuration Tips

The following describes the Configuration Tips of configuration for dynamic NAT-PT:

- Configure an IPv4 address.
- Configure an IPv6 address.
- Configure the NAT-PT prefix and enable NAT-PT.
- Configure dynamic address mapping in the IPv6 domain and static address mapping in the IPv4 domain.
- Configure the IPv4 address pool with a specified length.
- Configure IPv6 ACL Permit entries.

Steps

- Configure the IP address of Router C in the IPv4 domain.

```
RouterA>enable
RouterA#configure terminal
RouterA(config)#interface s1/0
RouterA(config-if)#ip address 8.0.0.1 255.255.255.0
```

- Configure Router B.

```
RouterB>enable
RouterB#configure terminal
RouterB(config)#ipv6 nat prefix 2001:DA8:1::/96
RouterB(config)#interface s1/0
RouterB(config-if)#ipv6 enable
RouterB(config-if)#ipv6 address 2001:DA8:2::2/64
RouterB(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#interface s2/0
RouterB(config-if)#ipv6 address 8.0.0.2 255.255.255.0
RouterB(config-if)#ipv6 nat
RouterB(config-if)#exit
RouterB(config)#ipv6 nat v6v4 source list v6_acl pool v4pool
RouterB(config)#ipv6 nat v6v4 pool v4pool 20.0.0.1 20.0.0.30 prefix-length 24
```

```
RouterB(config)#ipv6 nat v4v6 source 8.0.0.1 2001:DA8:1::5
RouterB(config)#ipv6 access-list v6_acl
RouterB(config-ipv6-acl)#permit ipv6 2001:DA8:2::/64 any
```

- Configure the IP address of Router A in the IPv6 domain.

```
RouterC>enable
RouterC#configer terminal
RouterC(config)#interface s1/0
RotuerC(config-if)#ipv6 enable
RouterC(config-if)#ipv6 address 2001:DA8:2::1/64
```

- On Router A, configure the static route destined for the network segment matched to the NAT-PT prefix.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#ipv6 route 2001:DA8:1::/96 2001:DA8:2::2
```

- On Router C, configure the static route destined for the network segment 20.0.0.0/24.

```
RouterC>enable
RouterC#configure terminal
RouterC(config)#ip route 20.0.0.0 255.255.255.08.0.0.2
```

Verification

```
Qtech#show ipv6 nat translations
Prot  IPv4 source              IPv6 source
      IPv4 destination         IPv6 destination
---   8.0.0.12001:DA8:1::5
      ---                      ---
icmp  20.0.0.6,1024         2001:DA8:2::1,1024
8.0.0.1,10242001:DA8:1::5,1024
udp20.0.0.6,532001:DA8:2::1,53
8.0.0.1,13642001:DA8:1::5,1364
```

Run the **ping 2001:DA8:1::5** command on Router A to receive response packets from Router C.

```
Qtech#ping 2001:DA8:1::5
Sending 5, 100-byte ICMP Echoes to 2001:DA8:1::5, timeout is 2 seconds:
< press Ctrl+C to break >
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 1/2/10 ms
```

4 CONFIGURING STATEFUL NAT64

4.1 Understanding Stateful NAT64

4.1.1 Overview

With the fast development of the Internet, IPv4 can no longer meet Internet requirements. Under this circumstance, IPv6 is about to be deployed. To support an IPv6 network, you must make full use of existing network resources to construct a next-generation Internet, thereby implementing smooth transition and avoiding excessive investment. The current Internet is based on IPv4 and cannot be transformed to the IPv6 network in a short time. Therefore, the IPv4 and IPv6 networks will coexist in a rather long time.

The coexistence, however, causes the following problems: how to keep current network services and functions at minimum cost; and how to implement transparent transmission between the IPv6 network and the IPv4 network. Network Address Translation 64 (NAT64, also called the IPv6-to-IPv4 address mapping), includes Stateful NAT64 and Stateless NAT64. Stateful NAT64 is mainly used when IPv6 network users initiate access requests to hosts/servers on the IPv4 network.

4.1.2 Basic Concepts

Stateful NAT64: Stateful IPv6-to-IPv4 network address translation protocol

Port Address Translation (PAT)

Network-Specific Prefix (NSP): Mainly used to check IPv6 destination addresses and IPv6 network addresses mapping to IPv4 host addresses.

Well-Known Prefix (WKP): Network prefix used by Stateful NAT64. It is used by default with the value of 64:ff9b::/96.

4.1.3 Working Principle

Stateful NAT64 provides a translation mechanism between IPv6 packets and IPv4 packets. This mechanism uses a Stateful NAT6 IPv6 prefix to implement translation from IPv4 host addresses to IPv6 addresses and takes NAT to implement translation from IPv6 host addresses to IPv4 addresses. Moreover, Statefull NAT64 performs protocol translation. NAT64 implements intercommunication between the pure IPv6 network and the IPv4 network.

4.1.4 Protocol Specification

RFC6052: IPv6 Addressing of IPv4/IPv6 Translators

RFC6144: Framework for IPv4/IPv6 Translation

RFC6145: IP/ICMP Translation Algorithm

RFC6146: Stateful NAT64: Network Address and Protocol Translation from IPv6 Clients to IPv4 Servers

4.1.5 Typical Application

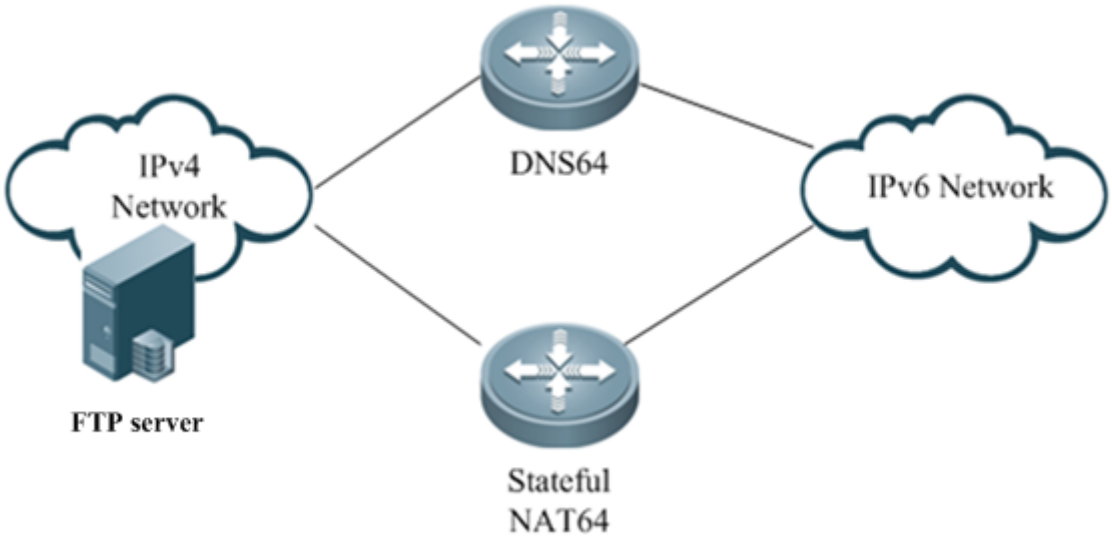


Figure 1-1 IPv6 Network Initiating a Session to IPv4 Network

4.2 Configuring Stateful NAT64

Default Configuration

Feature	Default setting
Stateful NAT64	Disabled
Default WKP	64:ff9b::/96

4.2.1 Configuring Static NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateful** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateful NAT64 prefix.
Qtech(config)#**nat64 v6v4 static** *ipv6-address ipv4-address*	Configures NAT64 to map an IPv6 address to an IPv4 address in static mode.
Qtech(config)#**end**	Exits global configuration mode and retunes to privileged mod.

The following example configures static Stateful NAT64.

```
Qtech#configure terminal
```

```
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:1::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 1/2/0
Qtech(config-if)#ip address 209.165.201.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96
Qtech(config)#nat64 v6v4 static 2001:db8:1::fffe 209.165.201.2
Qtech(config)#end
```

4.2.2 Configuring Dynamic NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.
Qtech**#configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	Enables unicast routing, which is enabled by default.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)**#exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)**#ipv6 access-list** *access-list-name*	Configures an IPv6 ACL permit entry and enters IPv6 ACL mode.
Qtech(config-ipv6-acl)**#permit ipv6** *ipv6-address* **any**	Filters IPv6 addresses based on ACL.
Qtech(config-ipv6-acl)**#exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateful** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateful NAT64 prefix.
Qtech(config)#**nat64 v4 pool** *pool-name start-ip-address end-ip-address*	Configures a NAT64 IPv4 address pool.
Qtech(config)**#nat64 v6v4 list** *access-list-name* **pool** *pool-name*	Enables dynamic **NAT64.** *access-list-name* specifies an IPv6 ACL. The IPv6 addresses match the ACL are mapped to the addresses in *pool-name* address pool. *pool-name* specifies the name of an IPv4 address pool.
Qtech(config)**#end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures dynamic Stateful NAT64.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:1::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#ip address 209.165.201.24 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list nat64-acl
```

```
Qtech(config-ipv6-acl)#permit ipv6 2001:db8:2::/96 any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:1::/96
Qtech(config)#nat64 v4 pool v4pool 209.165.201.1 209.165.201.254
Qtech(config)#nat64 v6v4 list nat64-acl pool v4pool
Qtech(config)#end
```

4.2.3 Configuring Dynamic PAT-Based NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.
Qtech#**configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**ipv6 access-list** *access-list-name*	Configures an IPv6 ACL permit entry and enters IPv6 ACL mode.
Qtech(config-ipv6-acl)#**permit ipv6** *ipv6-address* **any**	Filters IPv6 addresses based on ACL.
Qtech(config-ipv6-acl)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateful** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateful NAT64 prefix.
Qtech(config)#**nat64 v4 pool** *pool-name start-ip-address end-ip-address*	Configures a NAT64 IPv4 address pool.
Qtech(config)#**nat64 v6v4 list** *access-list-name* **pool** *pool-name* **overload**	Enables dynamic NAT64. *access-list-name* specifies an IPv6 ACL. The IPv6 addresses match the ACL are mapped to the addresses in *pool-name* address pool. *pool-name* specifies the name of an IPv4 address pool.
Qtech(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures dynamic Stateful NAT64.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:1::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#ip address 209.165.201.24 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list nat64-acl
Qtech(config-ipv6-acl)#permit ipv6 2001:db8:2::/96 any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96
Qtech(config)#nat64 v4 pool v4pool 209.165.201.1 209.165.201.254
Qtech(config)#nat64 v6v4 list nat64-acl pool v4pool overload
Qtech(config)#end
```

4.2.4 Configuring VRF-Based Stateful NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.
Qtech**#configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Qtech(config)#**vrf definition** *vrf-name*	Creates a multi-protocol VRF.
Qtech(config-vrf)#**address-family ipv4**	Enables an IPv4 address family.
Qtech(config-vrf-af)#**exit-address-family**	Exits a VRF address family.
Qtech(config-vrf)#**address-family ipv6**	Enables an IPv6 address family.
Qtech(config-vrf-af)#**exit-address-family**	Exits the VRF address family.
Qtech(config-vrf)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**vrf forwarding** *vrf-name*	Binds the multi-protocol VRF on the interface.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config)#**vrf forwarding** *vrf-name*	Binds the multi-protocol VRF on the interface. The IPv4 and IPv6 protocol families of the multi-protocol VRF need to be enabled.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**ipv6 access-list** *access-list-name*	Configures an IPv6 ACL permit entry and enters IPv6 ACL mode.
Qtech(config-ipv6-acl)#**permit ipv6** *ipv6-address* **any**	Filters IPv6 addresses based on ACL.
Qtech(config-ipv6-acl)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateful** *ipv6-address/prefix-length* [**vrf** *vrf-name*]	Allocates an IPv6 prefix to *vrf-name* as a Stateful NAT64 prefix.
Qtech(config)#**nat64 v4 pool** *pool-name start-ip-address end-ip-address* [**vrf** *vrf-name*]	Configures a NAT64 IPv4 address pool for *vrf-name*.
Qtech(config)#**nat64 v6v4 list** *access-list-name* **pool** *pool-name* [**vrf** *vrf-name*]	Enables dynamic NAT64. *access-list-name* specifies an IPv6 ACL. The IPv6 addresses match the ACL are mapped to the addresses in *pool-name* address pool. *pool-name* specifies the name of an IPv4 address pool. *vrf-name* specifies the name of the VRF.
Qtech(config)#**ipv6 route vrf** *vrf-name ipv6-prefix/prefix-length interface next-hop*	Adds an IPv6 route. *vrf-name* specifies the name of the VRF; *ipv6-prefix* specifies an IPv6 prefix; *prefix-length* specifies the length of the IPv6 prefix; *interface* specifies an outbound interface; *next-hop* specifies a next-hop address.
Qtech(config)#**ip route vrf** *vrf-name network mask interface next-hop*	Adds an IPv4 route. *vrf-name* specifies the name of VRF; *network* specifies a destination network segment; *mask* specifies a mask; *interface* specifies an outbound interface; *next-hop* specifies a next-hop address.
Qtech(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures VRF-based dynamic Stateful NAT64.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#vrf definition vrf-name1
Qtech(config-vrf)#address-family ipv4
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#address-family ipv6
```

```
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#interface gigabitethernet 0/0/0
Qtech(config-if)#vrf forwarding vrf-name1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:1::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#vrf forwarding vrf-name1
Qtech(config-if)#ip address 209.165.201.24 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list nat64-acl
Qtech(config-ipv6-acl)#permit ipv6 2001:db8:2::/96 any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96 vrf vrf-name1
Qtech(config)#nat64 v4 pool v4pool 209.165.201.1 209.165.201.254 vrf vrf-name1
Qtech(config)#nat64 v6v4 list nat64-acl pool v4pool vrf vrf-name1
Qtech(config)#ip route vrf vrf-name1 209.165.201.0 255.255.255.0 gigabitethernet 0/0/1
Qtech(config)#ipv6 route vrf vrf-name1 2001:db8:0:1::D1A5:C918/120 gigabitethernet
0/0/0
Qtech(config)#end
```

4.3 Monitoring and Maintaining Stateful NAT64

Command	Function
clear nat64 stateful statistics	Clears statistics about Stateful NAT64.
debug nat64 stateful {alg | control | event | memory | packet | pool | rule | translations}	Enables NAT64 debugging. Use the **no** form of this command to disable NAT64 debugging.
show nat64 stateful debug-buf	Displays the debugging buffer.
show nat64 mappings dynamic	Displays NAT64 dynamic mapping information.
show nat64 mappings static	Displays NAT64 static mapping information.
show nat64 prefix stateful [interfaces]	Displays all configured IPv6 prefixes of Stateful NAT64.
show nat64 services	Displays related NAT64 application layer gateway (ALG) information.
show nat64 stateful statistics	Displays statistics about Stateful NAT64.
show nat64 translations	Displays translation records of NAT64.

4.4 Configuration Examples

4.4.1 Static NAT64 Configuration Example

Networking Requirements

Host B on the IPv6 network can initiate a session to Host A on the IPv4 network and record the session entry. To meet the requirements, deploy a NAT64 device between the IPv6 network and the IPv4 network.

Networking Topology

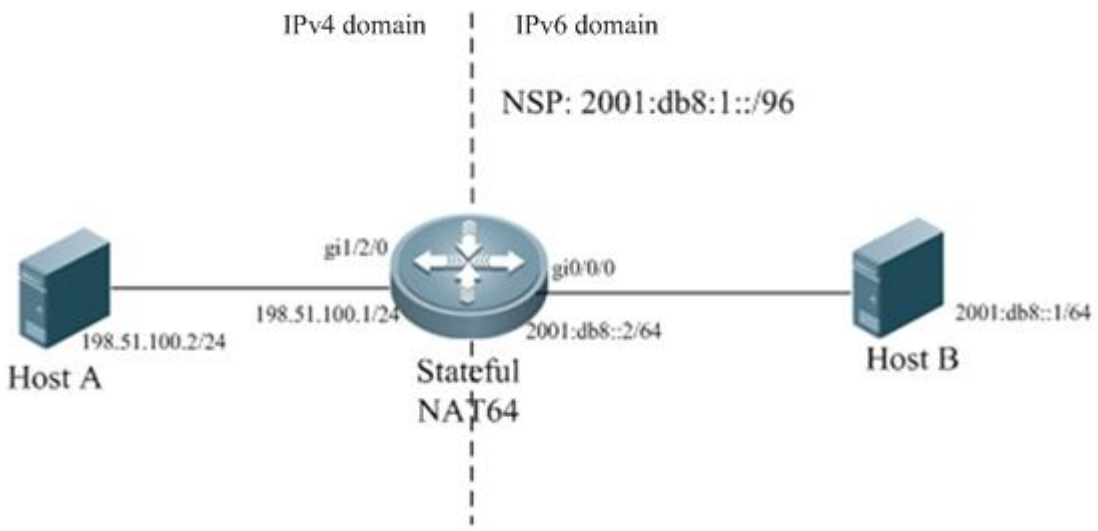


Figure 1-2 Topology of Static Stateful NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure a global NAT64 prefix.
- Configures static IPv6-to-IPv4 address translation.

Configuration Steps

13) Perform the following configurations on the Stateful NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 1/2/0
Qtech(config-if)#ip address 198.51.100.2 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:1::/96
Qtech(config)#nat64 v6v4 static 2001:db8::1 198.51.100.4
Qtech(config)#end
```

14) On Host B

Configure the IPv6 address 2001:db8::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

15) On Host A

Configure the IP address 198.51.100.2/24 on Host A.

Verification

Run the **ping 2001:db8:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Qtech#show nat64 translations
```

```
Prot   IPv4 source              IPv6 source
       IPv4 destination          IPv6 destination
icmp   198.51.100.4,47          2001:db8::1 ,47
       198.51.100.1 ,47          2001:db8:1::c633:6401,47
```

4.4.2 Dynamic NAT64 Configuration Example

Networking Requirements

Host B, Host C or another host on the IPv6 network can initiate a session to Host A on the IPv4 network and record the session entry. To meet the requirements, deploy a NAT64 device between the IPv6 network and the IPv4 network. Dynamic NAT64 takes effect as long as the number of IPv6-domain hosts that initiate a session to Host A does not exceed the number of IPv4 addresses in the address pool.

Networking Topology

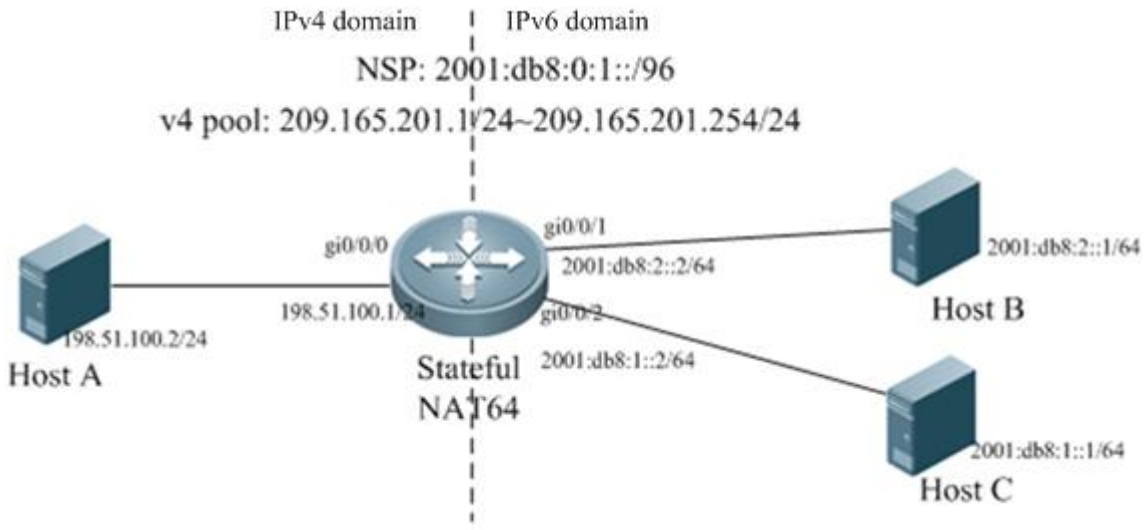


Figure 1-3 Topology of Dynamic Stateful NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure an ACL entry.
- Configure a global NAT64 prefix.
- Configures an IPv4 address pool.
- Configure a dynamic IPv6-to-IPv4 address translation list.

Configuration Steps

16) Perform the following configurations on the Stateful NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:2::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/2
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001db8:1::2/64
```

```
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list v6_list1
Qtech(config-ipv6-acl)#permit ipv6 any any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96
Qtech(config)#nat64 v4 pool v4_pool 209.165.201.1 209.165.201.254
Qtech(config)#nat64 v6v4 list v6_list1 pool v4_pool
Qtech(config)#end
```

17) On Host B

Configure the IPv6 address 2001:db8:2::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

18) On Host C

Configure the IPv6 address 2001:db8:1::1/64 on Host C and configure a static route to the prefix 2001:db8:0:1::/96.

19) On Host A

Configure the IP address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 209.165.201.0/24.

Verification

Run the **ping 2001:db8:0:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 2001:db8:0:1::c633:6401** command on Host C.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Qtech#show nat64 translations
Prot   IPv4 source              IPv6 source
       IPv4 destination          IPv6 destination
---     ----                    -----
icmp   209.165.201.1,47          2001:db8:2::1,47
       198.51.100.1 ,47         2001:db8:0:1::c633:6401,47
icmp   209.165.201.2,47          2001:db8:1::1,47
        198.51.100.1,47         2001:db8:0:1::c633:6401,47
```

4.4.3 Configuration Example of Dynamic PAT-Based NAT64

Networking Requirements

When the number of IPv6-domain hosts that initiate a session to Host A does not exceed the number of IPv4 addresses in the address pool, Host B or Host C on the IPv6 network can initiate a session to Host A on the IPv4 network and record the session entry. Otherwise, deploy a dynamic PAT-based NAT64 device between the IPv6 network and the IPv4 network, so that Host B or Host C can continue to access Host A.

Networking Topology

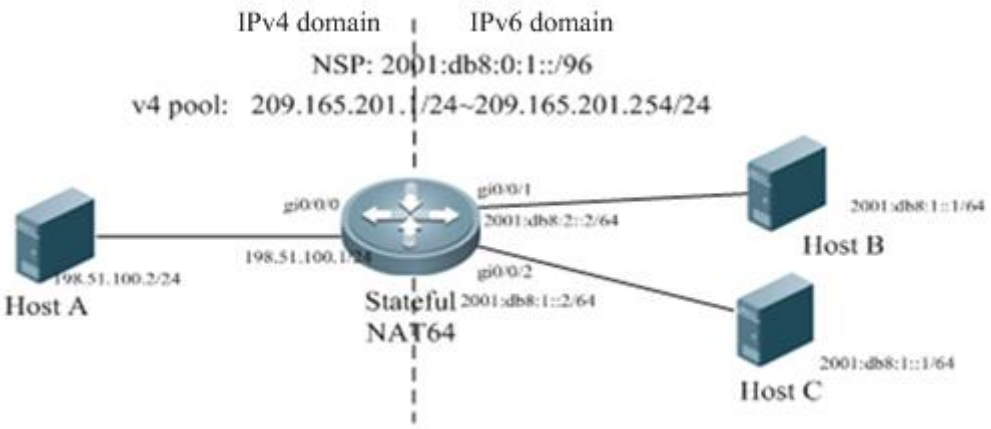


Figure 1-4 Topology of Dynamic PAT-Based NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure an ACL entry.
- Configure a global NAT64 prefix.
- Configures an IPv4 address pool.
- Configure a dynamic PAT-based IPv6-to-IPv4 address translation list.

Configuration Steps

20) Perform the following configurations on the Stateful NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:2::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/2
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:1::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list v6_list1
Qtech(config-ipv6-acl)#permit ipv6 any any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96
Qtech(config)#nat64 v4 pool v4_pool 209.165.201.1 209.165.201.254
Qtech(config)#nat64 v6v4 list v6_list1 pool v4_pool overload
Qtech(config)#end
```

21) On Host B

Configure the IPv6 address 2001:db8:2::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

22) On Host C

Configure the IPv6 address 2001:db8:1::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

23) On Host A

Configure the IPv address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 209.165.201.0/24.

Verification

Run the **ping 2001:db8:0:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 2001:db8:0:1::c633:6401** command on Host C.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Qtech#show nat64 translations
Prot   IPv4 source              IPv6 source
       IPv4 destination          IPv6 destination
---     ----                      -----
icmp   209.165.201.1,47         2001:db8:2::1 ,47
        198.51.100.1,47        2001:db8:0:1::c633:6401,47
icmp   209.165.201.1,1029         2001:db8:1::1, 1029
        198.51.100.1, 1029      2001:db8:0:1::c633:6401, 1029
```

4.4.4 Configuration Example of VRF-Based Stateful NAT64

Networking Requirements

Only hosts on the IPv6 network can initiate a session to Host A on the IPv4 network. Meanwhile, the router device can be divided into independently logical routers. To meet the requirements, deploy a VRF-based Stateful NAT64 device on the boundary between the IPv6 network and the IPv4 network, so that logical translation devices can communicate with each other.

Networking Topology

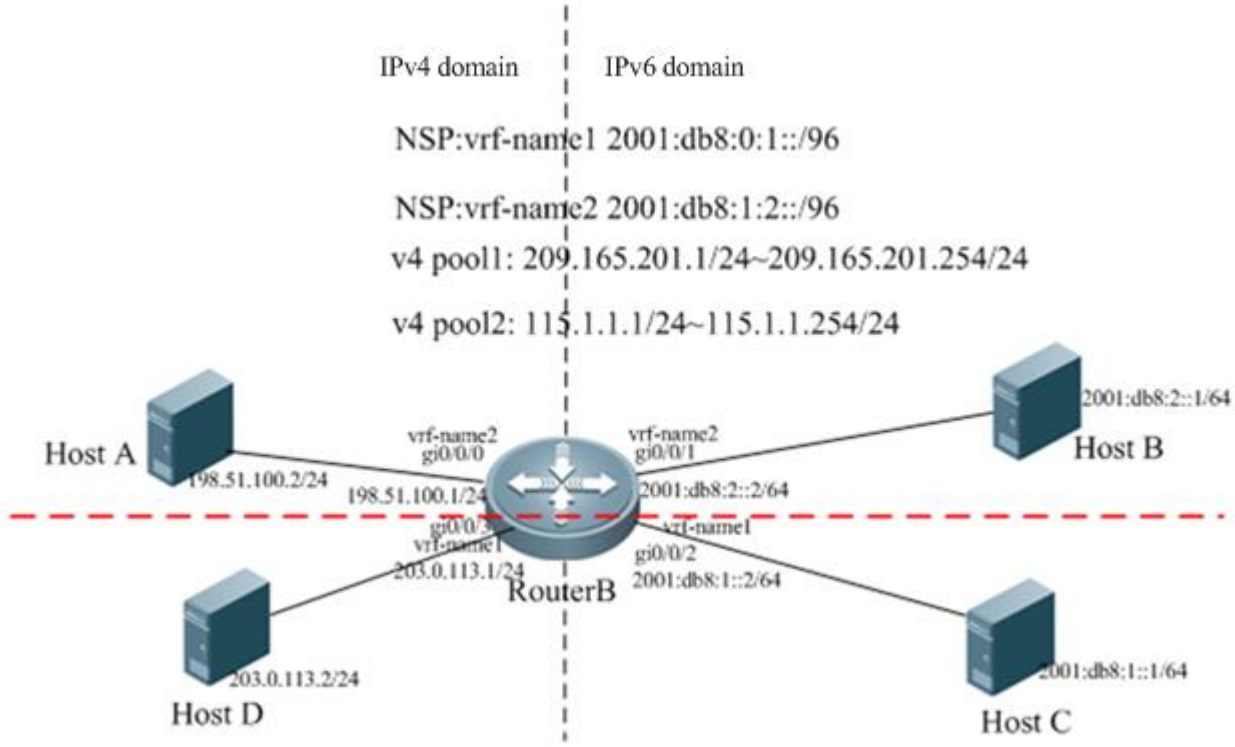


Figure 1-5 Topology of VRF-Based Stateful NAT64

Configuration Tips

- Create VRFs.
- Enable VRFs on interfaces.
  - Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
  - Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
  - Configure a routing protocol.
  - Configure an ACL.
- Configure global NAT64 prefixes for different VRFs.
  - Configure IPv4 address pools for different VRFs.
  - Configure dynamic IPv6-to-IPv4 translation access lists for different VRFs.

Configuration Steps

24) Perform the following configurations on Router B, which serves as the Stateful NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#vrf definition vrf-name1
Qtech(config-vrf)#address-family ipv4
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#address-family ipv6
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#interface gigabitethernet 0/0/0
Qtech(config-if)#vrf forwarding vrf-name1
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/1
```

```
Qtech(config-if)#vrf forwarding vrf-name1
Qtech(config)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:2::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#vrf definition vrf-name2
Qtech(config-vrf)#address-family ipv4
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#address-family ipv6
Qtech(config-vrf-af)#exit-address-family
Qtech(config-vrf)#interface gigabitethernet 0/0/3
Qtech(config-if)#vrf forwarding vrf-name2
Qtech(config-if)#ip address 203.0.113.2 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config-vrf)#interface gigabitethernet 0/0/2
Qtech(config-if)#vrf forwarding vrf-name2
Qtech(config-if)#ipv6 address 2001:db8:1::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list nat64-acl1
Qtech(config-ipv6-acl)#permit ipv6 2001:db8:2::/64 any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96 vrf vrf-name1
Qtech(config)#nat64 v4 pool v4pool1 209.165.201.1 209.165.201.254
Qtech(config)#nat64 v4 pool v4pool2 115.1.1.1 115.1.1.20
Qtech(config)#nat64 v6v4 list nat64-acl1 pool v4pool1 vrf vrf-name1
Qtech(config)#ipv6 access-list nat64-acl2
Qtech(config-ipv6-acl)#permit ipv6 4001::/64 any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:1:2::/96 vrf vrf-name2
Qtech(config)#nat64 v6v4 list nat64-acl2 pool v4pool2 vrf vrf-name2
Qtech(config)#ip route vrf vrf-name1 209.165.201.0 255.255.255.0 gi0/0/0
Qtech(config)#ip route vrf vrf-name1 209.165.201.0 255.255.255.0 gi0/0/3
Qtech(config)#ipv6 route vrf vrf-name1 2001:db8:0:1::/96 gi0/0/2
Qtech(config)#ipv6 route vrf vrf-name1 2001:db8:0:1::/96 gi0/0/1
Qtech(config)#ipv6 route vrf vrf-name2 2001:db8:1:2::/96 gi0/0/1
Qtech(config)#ipv6 route vrf vrf-name2 2001:db8:1:2::/96 gi0/0/2
Qtech(config)#ip route vrf vrf-name2 115.1.1.0 255.255.255.0 gi0/0/0
Qtech(config)#ip route vrf vrf-name2 115.1.1.0 255.255.255.0 gi0/0/3
Qtech(config)#end
```

25) On Host B

Configure the IPv6 address 2001:db8:2::1/64 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

26) On Host C

Configure the IPv6 address 2001:db8:1::1/64 on Host C and configure static routes to the prefixes 2001:db8:0:1::/96 and 2001:db8:1:2::/96.

27) On Host A

Configure the IP address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 209.165.201.0/24.

28) On Host D

Configure the IPv6 address 203.0.113.2/24 on Host D and configure a static route to the destination network segment 209.165.201.0/24.

Verification

Run the **ping 2001:db8:0:1::c633:6401** command on Host B.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
```

```
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 2001:db8:0:1::c633:6401** command on Host C.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Qtech#show nat64 translations
Prot   IPv4 source              IPv6 source
       IPv4 destination         IPv6 destination
---     ----                     -----
icmp   209.165.201.1,47         2001:db8:2::1 ,47
        198.51.100.1,47        2001:db8:0:1::c633:6401,47
icmp   209.165.201.2,1027       2001:db8:1::1 ,1027
       198.51.100.1 ,1027      2001:db8:0:1::c633:6401,1027
```

4.4.5 Configuration Example of ALG-Based Stateful NAT64

Networking Requirements

The various typical applications in address translation scenario on the pure IPv6 network or IPv4 network need not only address translation but also application-layer information transformation. For example, when IPv6 users on the IPv6 network initiate access requests to the IPv4 FTP server, the FTP ALG function needs to be added to the NAT64 device to meet application requirements. NAT64 can only translate addresses.

☑ When some applications (such as DNS, VoIP, and multimedia applications) perform address family traversal, the corresponding protocols must ensure that the traversal function works well. NAT64 is only a transition technology and cannot meet all application requirements.

Networking Topology

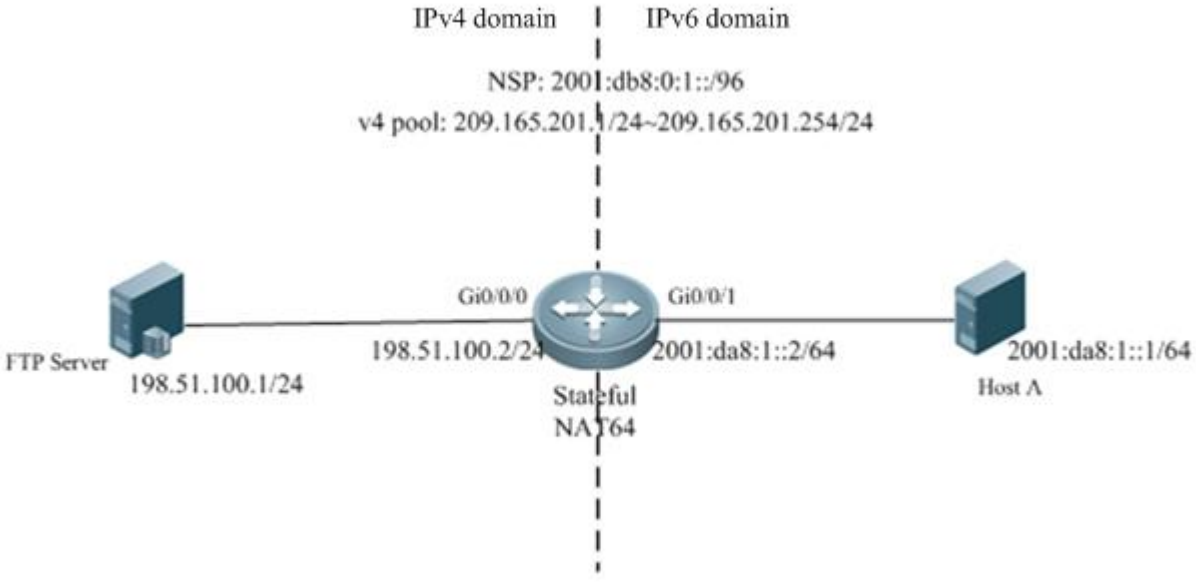


Figure 1-6 Topology of FTP ALG-Based Stateful NAT64

Configuration Tips

- Configure the IPv6 address of an IPv6 network interface and enable NAT64 on the interface.
- Configure the IPv4 address of an IPv4 network interface and enable NAT64 on the interface.
- Configure an ACL entry.
- Configure a global NAT64 prefix.
- Configures an IPv4 address pool.

- Configure a dynamic IPv6-to-IPv4 address translation list.
- Enable FTP ALG (enabled by default).

Configuration Steps

29) Perform the following configurations on the Stateful NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:da8:1::2/64
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/0
Qtech(config-if)#ipv4 address 198.51.100.2 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#ipv6 access-list v6_list1
Qtech(config-ipv6-acl)#permit ipv6 any any
Qtech(config-ipv6-acl)#exit
Qtech(config)#nat64 prefix stateful 2001:db8:0:1::/96
Qtech(config)#nat64 v4 pool v4_pool 209.165.201.1 209.165.201.254
Qtech(config)#nat64 v6v4 list v6_list1 pool v4_pool
Qtech(config)#end
```

30) On Host A in the IPv6 domain

Configure the IP address 2001:da8:1::1/64 for Host A and configure a corresponding static route.

31) On the FTP server

Configure the IPv4 address as 198.51.100.1/24 and configure a static route to the destination network segment 209.165.201.0/24.

Verification

32) Run the **ping 2001:db8:0:1::c633:6401** command on Host A.

```
Ping statistics for 2001:db8:0:1::c633:6401:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

33) Enter FTP mode.

Running the **put**, **get**, and **dir** commands can upload a file to the FTP server or download a file from it.

5 CONFIGURING STATELESS NAT64

5.1 Understanding Stateless NAT64

5.1.1 Overview

With the fast development of the Internet, IPv4 can no longer meet Internet requirements. Under this circumstance, IPv6 is about to be deployed. To support an IPv6 network, you must make full use of existing network resources to construct a next-generation Internet, thereby implementing smooth transition and avoiding excessive investment. The current Internet is based on IPv4 and cannot be transited to the IPv6 network in a short time. Therefore, the IPv4 and IPv6 networks will coexist in a rather long time.

The coexistence however, causes the following problems: how to keep current network services and functions at minimum cost; and how to implement transparent transmission between the IPv6 network and the IPv4 network. Network Address Translation 64 (NAT64, also called the IPv6-to-IPv4 network address translation protocol) includes Stateful NAT64 and Stateless NAT64. Stateless NAT64 is mainly used when IPv4 network users initiate access requests to hosts on the IPv6 network.

5.1.2 Basic Concepts

Stateless Network Address Translation 64 (Stateless NAT64): Stateless IPv6-to-IPv4 network address translation protocol. Stateless NAT64 provides a translation mechanism, which implements translation between IPv4 addresses and IPv6 addresses. The translation involves parsing the entire IPv6 header, obtaining related information and translating it into an IPv4 header or a completely converse translation process. Stateless NAT64 can translate the IP addresses of only some types of ICMPv4 and ICMPv6 packets due to the protocol features, such as translation between ICMPv4 request/response packets and ICMPv6 request/response packets and translation between unreachable ICMPv4 packets and unreachable ICMPv6 packets. Address translation for each packet relies on interface configurations. Stateless NAT64 does not maintain data flow statuses.

IPv4-translatable IPv6 address: IPv6 address after Stateless NAT64 that is allocated to an IPv6 host

- ☑ Stateless NAT64 can be used only when there are IPv4-translatable IPv6 addresses.
- ☑ Stateless NAT64 does not support multicast.
- ☑ Stateless NAT64 cannot be used by the application without a corresponding ALG.
- ☑ Stateless NAT64 cannot translate an IPv4 option, an IPv6 routing header in an IPv6 extension header, a hop-by-hop extension header, or a destination option header.

5.1.3 Working Principle

Stateless NAT64 works on the boundary device between the IPv6 network and the IPv4 network. The Stateless NAT64 module translates IP headers between the two networks and performs semantic translation for packets according to different protocols so as to implement transparent transmission between the two networks.

5.1.4 Protocol Specification

RFC6052: IPv6 Addressing of IPv4/IPv6 Translators

RFC6144: Framework for IPv4/IPv6 Translation

RFC6145: IP/ICMP Translation Algorithm

5.1.5 Typical Application

Application Scenario 1

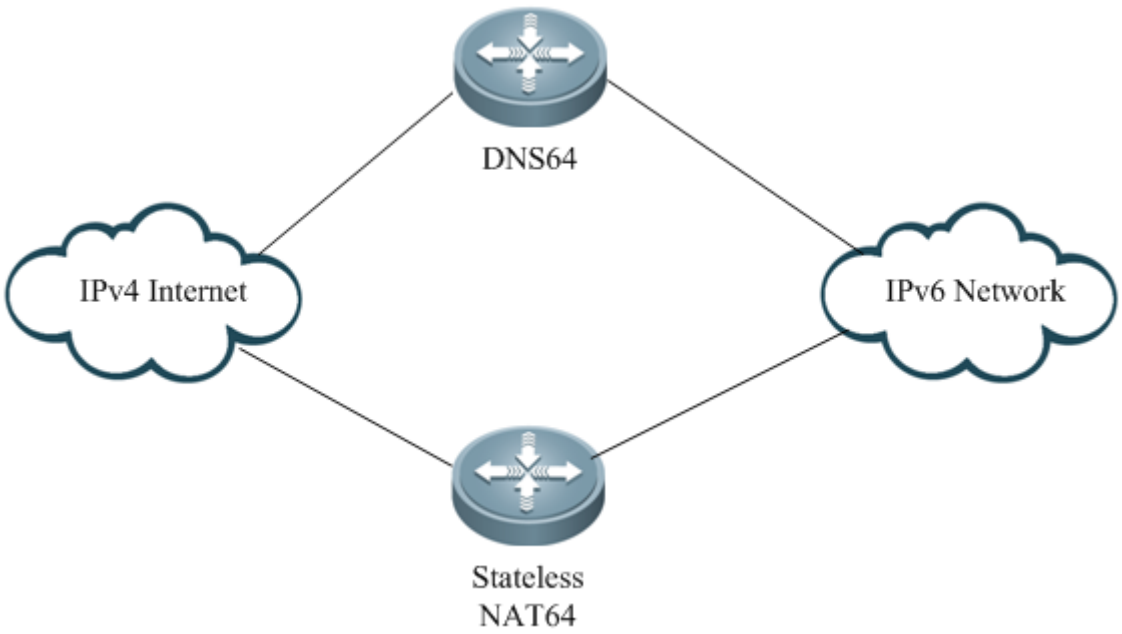


Figure 2-1 Interaction Between the IPv4 Internet and the IPv6 Network
This application scenario supports IPv4 network users to access IPv6 network resources.

Users of the IPv4 Internet can access IPv6 network resources in a specific scope.

This scenario has the following functions:

34) A new IPv6 content provider can provide resources for users both on the IPv6 network and the IPv4 Internet.

35) An IPv4 content provider that is transited to the pure IPv6 network can still provide resources for original IPv4 users and keep the connections with them.

This scenario has the following access modes:

- IP-based access
- Domain name-based access

Application Scenario 2

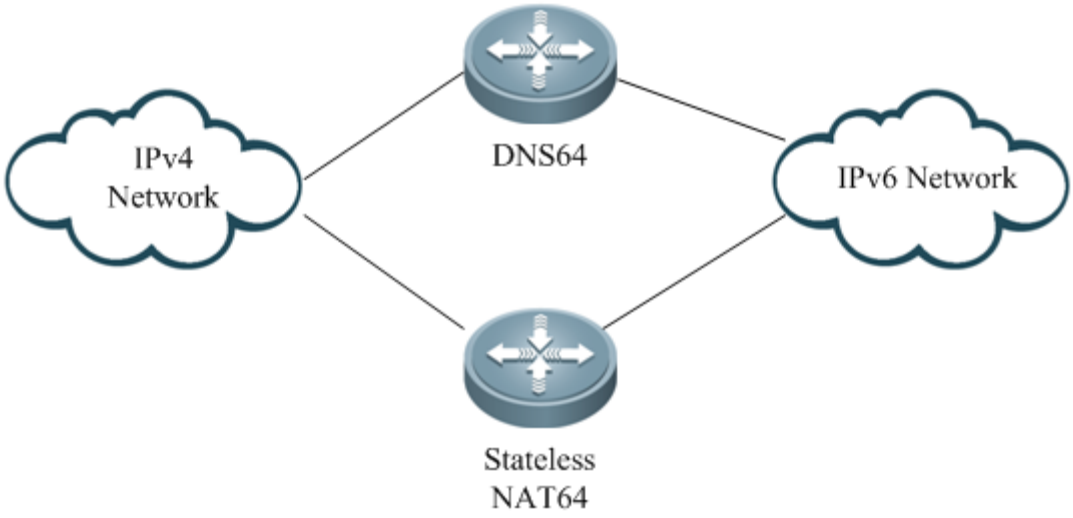


Figure 2-2 Interaction Between the IPv4 Network and the IPv6 Network
This application scenario mainly allows IPv4 network users to initiate access requests.

This scenario supports interaction between the IPv4 network and the IPv6 network, and mainly allows IPv4 network users to initiate access requests. The internal address can be a public address or a private address. The host addresses on the IPv6 network must be IPv4-translatable IPv6 addresses.

5.2 Configuring Stateless NAT64

5.2.1 Configuring Stateless NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.
Qtech**#configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateless** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateless NAT64 prefix.
Qtech(config)#**nat64 route** *ipv4-prefix/length interface-name interface-number*	Configures a route which starts from an IPv4 network segment and is destined for a specific IPv6 interface. Running this command can configure a route with a specific prefix and the default route with the prefix 0.0.0.0/0 in a VRF.
Qtech(config)#**ipv6 route** *ipv6-prefix/length interface-name interface-number nexthop-address*	Configures an IPv6 route, which is used to transmit the packets whose routes are changed to the destination IPv6 address.
Qtech(config)#**end**	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures Stateless NAT64.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/1
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateless 2001:db8:0:1::/96
Qtech(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0
Qtech(config)#ipv6 route 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8::2
Qtech(config)#ip route 0.0.0.0 0.0.0.0 198.51.100.2
Qtech(config)#end
```

5.2.2 Configuring Multi-Prefix Stateless NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.

Qtech**#configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**nat64 prefix stateless v6v4** *ipv6-address/prefix-length*	Configures the IPv6 prefix for Statelesss NAT64 (IPv6-to-IPv4 address translation) on an interface. The prefix must work with the global Stateless NAT64 prefix (v4v6). Otherwise, addresses cannot be translated.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateless v4v6** *ipv6-address/prefix-length*	Allocates an IPv6 prefix as a global Stateless NAT64 prefix. The prefix must work with the Stateless NAT64 prefix (v6v4). Otherwise, addresses cannot be translated.
Qtech(config)#**nat64 route** *ipv4-prefix/length interface-name interface-number*	Configures a route which starts from an IPv4 network segment and is destined for a specific IPv6 interface. Running this command can configure a route with a specific prefix and the default route with the prefix 0.0.0.0/0 in a VRF.
Qtech(config)#**ipv6 route** *ipv6-prefix/length interface-name interface-number*	Configures an IPv6 route, which is used to transmit the packets whose routes are changed to the destination IPv6 address.
Qtech(config)#end	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures multi-prefix Stateless NAT64.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#nat64 prefix stateless v6v4 2001:db8:0:1::/96
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/1
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateless v4v6 2001:db8:2::1/96
Qtech(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0
Qtech(config)#ipv6 route 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8::2
Qtech(config)#ip route 0.0.0.0 0.0.0.0 198.51.100.2
Qtech(config)#end
```

5.2.3 Configuring VRF-Based Stateless NAT64

Command	Function
Qtech>**enable**	Enters privileged EXEC mode.
Qtech**#configure terminal**	Enters global configuration mode.
Qtech(config)#**ipv6 unicast-routing**	(Optional) Enables unicast routing, which is enabled by default.

Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv6 network interface and enters interface configuration mode.
Qtech(config-if)#**vrf forwarding vrf** *vrf-name*	Enables a VRF on an interface. The IPv4 and IPv6 protocol families of the VRF also need to be enabled.
Qtech(config-if)#**ipv6 enable**	Enables IPv6.
Qtech(config-if)# **ipv6 address** *ipv6-address/prefix-length*	Configures the IPv6 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**interface** *interface-name interface-number*	Specifies an IPv4 network interface and enters interface configuration mode.
Qtech(config-if)#**vrf forwarding vrf** *vrf-name*	Enables a VRF on an interface. The IPv4 and IPv6 protocol families of the VRF also need to be enabled.
Qtech(config-if)#**ip address** *ip-address mask*	Configures the IPv4 address of the interface.
Qtech(config-if)#**nat64 enable**	Enables NAT64 on the interface.
Qtech(config-if)#**exit**	Exits interface configuration mode and returns to global configuration mode.
Qtech(config)#**nat64 prefix stateless** *ipv6-address/prefix-length* **vrf** *vrf-name*	Configures Stateless NAT64 *ipv6-address/prefix-length* under *vrf-name*.
Qtech(config)#**nat64 route** *ipv4-prefix/length interface-name interface-number* **vrf** *vrf-name*	Configures a route which starts from an IPv4 network segment and is destined for a specific IPv6 interface. Running this command can configure a route and the default route with the prefix 0.0.0.0/0 in a VRF.
Qtech(config)#**ipv6 route** *ipv6-prefix/length interface-name interface-number*	Configures an IPv6 route, which is used to transmit the packets whose routes are changed to the destination IPv6 address.
Qtech(config)#end	Exits global configuration mode and returns to privileged EXEC mode.

The following example configures VRF-based Stateless NAT64.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/0
Qtech(config-if)# vrf forwarding 1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8::1/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/1
Qtech(config-if)# vrf forwarding 1
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateless 2001:db8:0:1::/96 vrf 1
Qtech(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0 vrf 1
Qtech(config)#ipv6 route vrf 1 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8::2
Qtech(config)#ip route vrf 1 0.0.0.0 0.0.0.0 198.51.100.2
Qtech(config)#end
```

5.3 Monitoring and Maintaining Stateless NAT64

Command	Function
clear nat64 stateless statistics	Clears statistics about Stateless NAT64.
debug nat64 stateless { **control** | **packet**]	Enables Stateless NAT64 debugging. Use the **no** form of this command to disable Stateless NAT64 debugging.
show nat64 stateless debug-buf	Displays the debugging buffer.
show nat64 prefix stateless [**interfaces**]	Displays all configured IPv6 prefixes of Stateless NAT64.
show nat64 stateless statistics	Displays statistics about Stateless NAT64.

5.4 Configuration Examples

5.4.1 Stateless NAT64 Configuration Example

Networking Requirements

Host A with the address 198.51.100.2/24 in the IPv4 domain can access Host B with the address 2001:db8:0:1::cb00:7101 in the IPv6 domain. To meet this requirement, deploy a NAT64 device (Router B) between the IPv4 domain and the IPv6 domain and configure a global IPv6 prefix for NAT64 on Router B to implement intercommunication between the two domains.

Networking Topology

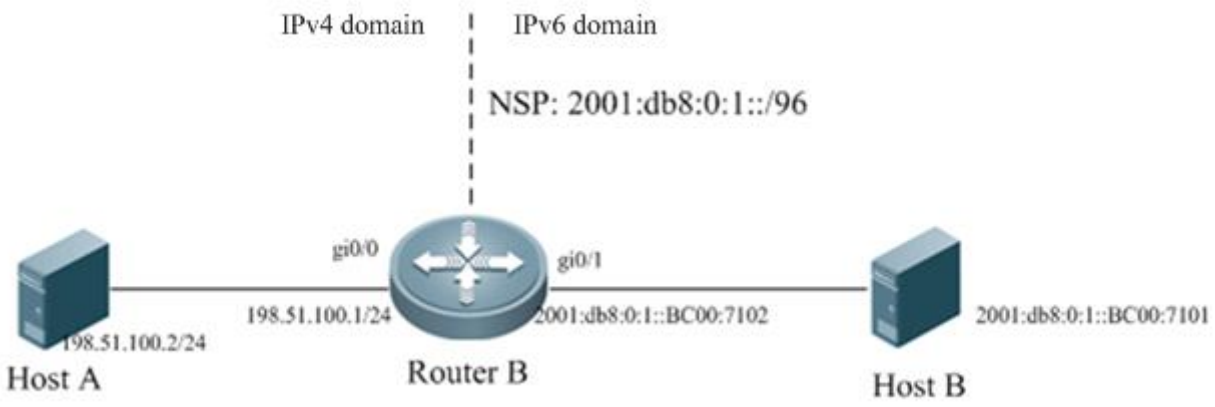


Figure 2-3 Topology of Stateless NAT64

Configuration Tips

Perform the following configurations on the Stateless NAT64 device:

- Configure the IPv6 address of an IPv6 network interface.
- Configure the IPv4 address of an IPv4 network interface.
- Configure a global NAT64 prefix and enable NAT64 on the interface.
- Configure a route which starts from the IPv4 network segment and is destined for the interface for NAT64.
- Configure a static route that is used to transmit translated packets to the IPv6 address.

Configuration Steps

36) Perform the following configurations on Router B, which serves as the Stateless NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)#interface gigabitethernet 0/1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:0:1::cb00:7102/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateless 2001:db8:0:1::/96
Qtech(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/1
Qtech(config)#ipv6 route 2001:db8:0:1::/96 gigabitethernet 0/0 2001:db8:0:1::cb00:7101
Qtech(config)#ip route 0.0.0.0 0.0.0.0 198.51.100.2
Qtech(config)#end
```

37) On Host B

Configure the IPv6 address 2001:db8:0:1::cb00:7101/128 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

38) On Host A

Configure the IP address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 203.0.113.0/24.

Verification

Run the **ping 203.0.113.1** command on Host A.

```
Ping statistics for 203.0.113.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Qtech#sh nat64 stateless statistics
NAT64 Stateless Global stats:
    Created Packets translation (IPv4 -> IPv6): 0.
    Created Packets translation (IPv6 -> IPv4): 0.
    Packets droped in IPv4: 0.
    Packets droped in IPv6: 0.
NAT64 Stateless Interface stats:
    Gi0/1:
       Created Packets translation (IPv4 -> IPv6): 0.
       Created Packets translation (IPv6 -> IPv4): 0.
    Gi0/0:
       Created Packets translation (IPv4 -> IPv6): 0.
       Created Packets translation (IPv6 -> IPv4): 0.
```

5.4.2 Configuration Example of Multi-Prefix Stateless NAT64

Networking Requirements

Host A with the address 198.51.100.2/24 in the IPv4 domain can access Host B and Host C in different network segments of the IPv6 network. To meet this requirement, configure IPv6 prefixes on the IPv6 interfaces of different network segments on the Stateless NAT64 device (Router B).

Networking Topology

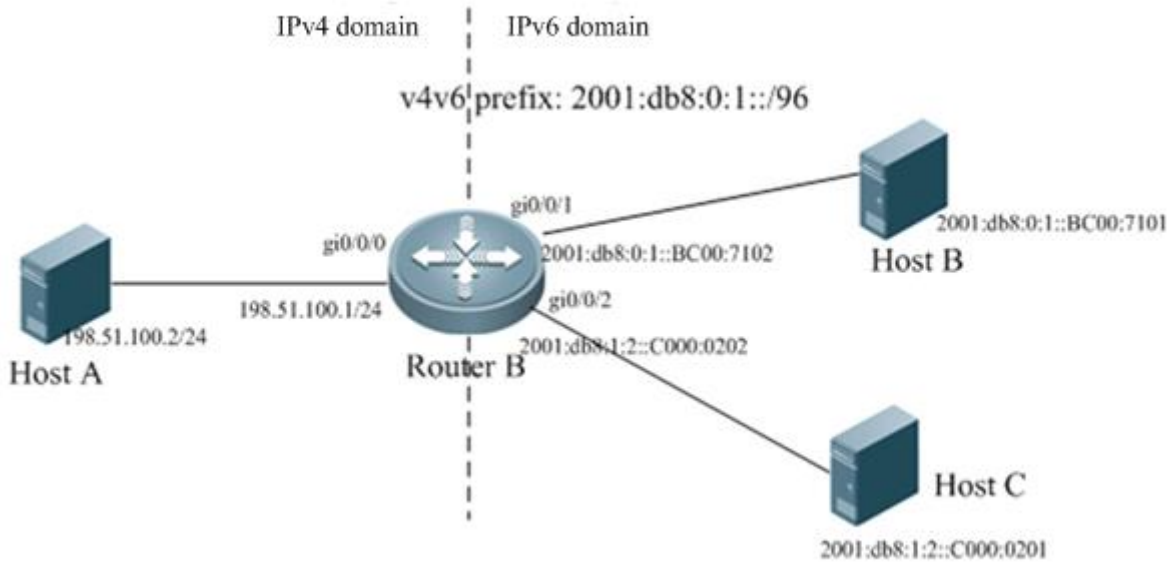


Figure 2-4 Multi-Prefix Stateless NAT64
Host A accesses Host B and Host C in two IPv6 network segments.

Configuration Tips

Perform the following configurations on the Stateless NAT64 device:

- Configure the IPv6 address of an IPv6 network interface, enable NAT64, and configure the Stateless NAT64 prefix (v6v4).
- Configure the IPv4 address of an IPv4 network interface and enable NAT64.
- Configure the Stateless NAT64 prefix (v4v6) in global mode.
- Configure a route which starts from an IPv4 network segment and is destined for the IPv6 interface for NAT64.
- Configure a static route that is used to transmit translated packets to the IPv6 address.

Configuration Steps

39) Perform the following configurations on Router B, which serves as the Stateless NAT64 device.

```
Qtech#configure terminal
Enter configuration commands, one per line.  End with CNTL/Z.
Qtech(config)# interface gigabitethernet 0/0/0
Qtech(config-if)#ip address 198.51.100.1 255.255.255.0
Qtech(config-if)#nat64 enable
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/1
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:0:1::cb00:7102/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#nat64 prefix stateless v6v4 2011:db8:0:1::/96
Qtech(config-if)#exit
Qtech(config)#interface gigabitethernet 0/0/2
Qtech(config-if)#ipv6 enable
Qtech(config-if)#ipv6 address 2001:db8:1:2::C000:0202/96
Qtech(config-if)#nat64 enable
Qtech(config-if)#nat64 prefix stateless v6v4 2011:db8:1:2::/96
Qtech(config-if)#exit
Qtech(config)#nat64 prefix stateless v4v6 2011:db8:2::1/96
Qtech(config)#nat64 route 203.0.113.0/24 gigabitethernet 0/0/1
Qtech(config)#ipv6 route 2011:db8:0:1::/96 gigabitethernet 0/0/1
2001:db8:0:1::cb00:7101
Qtech(config)#nat64 route 0.0.0.0/0 gigabitethernet 0/0/2
Qtech(config)#ipv6 route 2011:db8:1:2::/96 gigabitethernet 0/0/2
2001:db8:1:2::C000:0201
Qtech(config)#end
```

40) On Host B

Configure the IPv6 address 2001:db8:0:1::cb00:7101/128 on Host B and configure a static route to the prefix 2001:db8:0:1::/96.

41) On Host C

Configure the IPv6 address 2001:db8:0:1::c000:0201/128 on Host C and configure a static route to the prefix 2001:db8:1:2::/96.

42) On Host A

Configure the IPv6 address 198.51.100.2/24 on Host A and configure a static route to the destination network segment 203.0.113.0/24.

Verification

Run the **ping 203.0.113.1** command on Host A.

```
Ping statistics for 203.0.113.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
```

Run the **ping 192.0.2.1** command on Host A.

```
Ping statistics for 192.0.2.1:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Qtech#sh nat64 stateless statistics
NAT64 Stateless Global stats:
    Created Packets translation (IPv4 -> IPv6): 16.
    Created Packets translation (IPv6 -> IPv4): 31.
    Packets droped in IPv4: 0.
    Packets droped in IPv6: 0.

NAT64 Stateless Interface stats:
    Gi0/0/0:
       Created Packets translation (IPv4 -> IPv6): 16.
       Created Packets translation (IPv6 -> IPv4): 0.
    Gi0/0/1:
       Created Packets translation (IPv4 -> IPv6): 0.
       Created Packets translation (IPv6 -> IPv4): 19.
    Gi0/0/2:
       Created Packets translation (IPv4 -> IPv6): 0.
       Created Packets translation (IPv6 -> IPv4): 12.
```