Contents

Optimizing IP performance

IP performance optimization tasks at a glance

All IP performance optimization tasks are optional.

1. Configuring features for IP packets
   - Enabling an interface to forward directed broadcasts destined for the directly connected network
   - Setting the interface MTU for IPv4 packets
   - Enabling IPv4 local fragment reassembly

     This feature is applicable in IRF networks.
2. Configuring features for ICMP messages
   - Enabling sending ICMP error messages
   - Configuring rate limit for ICMP error messages
   - Disabling forwarding ICMP fragments
   - Specifying the source address for ICMP packets
3. Configuring features for TCP packets
   - Setting TCP MSS for an interface
   - Configuring TCP path MTU discovery
   - Enabling SYN Cookie
   - Setting the TCP buffer size
   - Setting TCP timers
   - Enabling the Timestamps option encapsulation in outgoing TCP packets

Enabling an interface to forward directed broadcasts destined for the directly connected network

About forwarding broadcasts destined for the directly connected network

A directed broadcast packet is destined for all hosts on a specific network. In the destination IP address of the directed broadcast, the network ID identifies the target network, and the host ID is made up of all ones.

If an interface is allowed to forward directed broadcasts destined for the directly connected network, hackers can exploit this vulnerability to attack the target network. In some scenarios, however, an interface must send such directed broadcast packets to support the following features:

- **UDP helper**—Converts the directed broadcasts to unicasts and forwards them to a specific server.
- **Wake on LAN**—Sends the directed broadcasts to wake up the hosts on the target network.

You can configure this function to enable the interface to forward directed broadcast packets that are destined for directly connected network.

Procedure

1. Enter system view.

   system-view

2. Enter interface view.

   interface *interface-type interface-number*

3. Enable the interface to forward directed broadcasts destined for the directly connected network.

   ip forward-broadcast [**acl** *acl-number*]

   By default, an interface cannot forward directed broadcasts destined for the directly connected network.

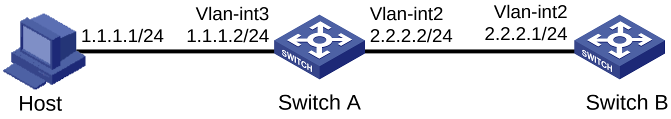
Example: Enabling an interface to forward directed broadcasts destined for the directly connected network

Network configuration

As shown in Figure 1, the default gateway of the host is the IP address 1.1.1.2/24 of VLAN-interface 3 of Switch A.

Switch B can receive directed broadcasts from the host to IP address 2.2.2.255.

Figure 1 Network diagram



Procedure

1. Configure Switch A:

   # Specify IP addresses for VLAN-interface 3 and VLAN-interface 2.

   ```
   <SwitchA> system-view
   [SwitchA] interface vlan-interface 3
   [SwitchA-Vlan-interface3] ip address 1.1.1.2 24
   [SwitchA-Vlan-interface3] quit
   [SwitchA] interface vlan-interface 2
   [SwitchA-Vlan-interface2] ip address 2.2.2.2 24
   ```

   # Enable VLAN-interface 2 to forward directed broadcasts directed for the directly connected network.

   ```
   [SwitchA-Vlan-interface2] ip forward-broadcast
   ```

2. Configure Switch B:

   # Configure a static route to the host.

   ```
   <SwitchB> system-view
   [SwitchB] ip route-static 1.1.1.1 24 2.2.2.2
   ```

   # Specify an IP address for VLAN-interface 2.

   ```
   [SwitchB] interface vlan-interface 2
   [SwitchB-Vlan-interface2] ip address 2.2.2.1 24
   ```

Verifying the configuration

After the configurations are completed, if you ping the subnet-directed broadcast address 2.2.2.255 on the host, VLAN-interface 2 of Switch B can receive the ping packets. If you delete the **`ip forward-broadcast`** configuration on any switch, the interface cannot receive the ping packets.

Setting the interface MTU for IPv4 packets

About setting the interface MTU for IPv4 packets

The interface MTU for IPv4 packets defines the largest size of an IPv4 packet that an interface can transmit without fragmentation. When a packet exceeds the MTU of the sending interface, the device processes the packet in one of the following ways:

- If the packet disallows fragmentation, the device discards it.
- If the packet allows fragmentation, the device fragments it and forwards the fragments.

Fragmentation and reassembling consume system resources, so set the MTU based on the network environment to avoid fragmentation.

Procedure

1. Enter system view.

   `system-view`

2. Enter interface view.

   `interface` *`interface-type interface-number`*

3. Set the interface MTU for IPv4 packets.

   `ip mtu` *`mtu-size`*

   By default, the interface MTU is not set.

Enabling IPv4 local fragment reassembly

About IPv4 local fragment reassembly

Use this feature on a multichassis IRF fabric to improve fragment reassembly efficiency. This feature enables a subordinate to reassemble the IPv4 fragments of a packet if all the fragments arrive at it. If this feature is disabled, all IPv4 fragments are delivered to the master device for reassembly. The feature applies only to fragments destined for the same subordinate.

Procedure

1. Enter system view.

   `system-view`

2. Enable IPv4 local fragment reassembly.

   `ip reassemble local enable`

   By default, IPv4 local fragment reassembly is disabled.

Enabling sending ICMP error messages

About sending ICMP error messages

ICMP messages are used by network layer and transport layer protocols to communicate updates and errors with other devices, facilitating network management.

Sending excessive ICMP messages increases network traffic. The device performance degrades if it receives a lot of malicious ICMP messages that cause it to respond with ICMP error messages. To prevent such problems, the sending of ICMP error messages is disabled by default. You can enable sending ICMP error messages of different types as needed.

ICMP error messages include redirect messages, time exceeded messages, and destination unreachable messages.

Enabling sending ICMP redirect messages

About ICMP redirect messages

A host that has only one default route sends all packets to the default gateway. The default gateway sends an ICMP redirect message to inform the host of a correct next hop by following these rules:

- The receiving and sending interfaces are the same.
- The selected route is not created or modified by any ICMP redirect messages.
- The selected route is not destined for 0.0.0.0.
- There is no source route option in the received packet.

ICMP redirect messages simplify host management and enable hosts to gradually optimize their routing table.

Procedure

1. Enter system view.

```
system-view
```

2. Enable sending ICMP redirect messages.

```
ip redirects enable
```

By default, the sending of ICMP redirect messages is disabled.

Enabling sending ICMP time exceeded messages

About ICMP time exceeded messages

A device sends ICMP time exceeded messages by following these rules:

- The device sends the source an ICMP TTL exceeded in transit message when the following conditions are met:
  - The received packet is not destined for the device.
  - The TTL field of the packet is 1.
- When the device receives the first fragment of an IP datagram destined for it, it starts a timer. If the timer expires before all the fragments of the datagram are received, the device sends an ICMP fragment reassembly time exceeded message to the source.

Restrictions and guidelines

If the ICMP time exceeded message sending is disabled, the device does not send ICMP TTL exceeded in transit messages. However, it can still send ICMP fragment reassembly time exceeded messages.

Procedure

1. Enter system view.

```
system-view
```

2. Enable sending ICMP time exceeded messages.

```
ip ttl-expires enable
```

By default, the sending of ICMP time exceeded messages is disabled.

Enable sending ICMP destination unreachable messages

About ICMP destination unreachable messages

A device sends ICMP destination unreachable messages by following these rules:

- The device sends the source an ICMP network unreachable message when the following conditions are met:
  - The packet does not match any route.
  - No default route exists in the routing table.
- The device sends the source an ICMP protocol unreachable message when the following conditions are met:
  - The packet is destined for the device.
  - The transport layer protocol of the packet is not supported by the device.
- The device sends the source an ICMP port unreachable message when the following conditions are met:
  - The UDP packet is destined for the device.
  - The packet's port number does not match the corresponding process.
- The device sends the source an ICMP source route failed message when the following conditions are met:
  - The source uses Strict Source Routing to send packets.
  - The intermediate device finds that the next hop specified by the source is not directly connected.
- The device sends the source an ICMP fragmentation needed and DF set message when the following conditions are met:
  - The MTU of the sending interface is smaller than the packet.
  - The packet has DF set.

Restrictions and guidelines

If a DHCP-enabled device receives an ICMP echo reply without sending any ICMP echo requests, the device does not send any ICMP protocol unreachable messages to the source. For more information about DHCP, see *Layer 3—IP Services Configuration Guide*.

Procedure

1. Enter system view.

```
system-view
```

2. Enable sending ICMP destination unreachable messages.

```
ip unreachables enable
```

By default, the sending of ICMP destination unreachable messages is disabled.

Configuring rate limit for ICMP error messages

About the token bucket algorithm

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

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

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

Procedure

1. Enter system view.

   `system-view`

2. Set the interval for tokens to arrive in the bucket and the bucket size for ICMP error messages.

   `ip icmp error-interval` *`interval`* [*`bucketsize`*]

   By default, a token is placed in the bucket at intervals of 100 milliseconds and the bucket allows a maximum of 10 tokens.

   To disable the ICMP rate limit, set the interval to 0 milliseconds.

Disabling forwarding ICMP fragments

Restrictions and guidelines

Disabling forwarding ICMP fragments can protect your device from ICMP fragment attacks.

Procedure

1. Enter system view.

   `system-view`

2. Disable forwarding ICMP fragments.

   `ip icmp fragment discarding`

   By default, forwarding ICMP fragments is enabled.

Specifying the source address for ICMP packets

About specifying source address for ICMP packets

Specifying the source IP address for outgoing ping echo requests and ICMP error messages helps users to locate the sending device easily. As a best practice, specify the IP address of the loopback interface as the source IP address.

Restrictions and guidelines

If you specify an IP address in the **`ping`** command, ping echo requests use the specified address as the source IP address rather than the IP address specified by the **`ip icmp source`** command.

Procedure

1. Enter system view.

   `system-view`

2. Specify the source address for outgoing ICMP packets.

   `ip icmp source` [**`vpn-instance`** *`vpn-instance-name`*] *`ip-address`*

   By default, no source address is specified for outgoing ICMP packets. The device uses the IP address of the sending interface as the source IP address for outgoing ICMP packets.

Setting TCP MSS for an interface

About TCP MSS

The maximum segment size (MSS) option informs the receiver of the largest segment that the sender can accept. Each end announces its MSS during TCP connection establishment. If the size of a TCP segment is smaller than the MSS of the receiver, TCP sends the TCP segment without fragmentation. If not, it fragments the segment according to the receiver's MSS.

Restrictions and guidelines

- If you set the TCP MSS on an interface, the size of each TCP segment received or sent on the interface cannot exceed the MSS value.
- This configuration takes effect only for TCP connections established after the configuration rather than the TCP connections that already exist.
- This configuration is effective only for IP packets. If MPLS is enabled on the interface, do not set the TCP MSS on the interface.

Procedure

1. Enter system view.

   ```
   system-view
   ```

2. Enter interface view.

   ```
   interface interface-type interface-number
   ```

3. Set the TCP MSS for the interface.

   ```
   tcp mss value
   ```

   By default, the TCP MSS is not set.

Configuring TCP path MTU discovery

About TCP path MTU discovery

TCP path MTU discovery (in RFC 1191) discovers the path MTU between the source and destination ends of a TCP connection. The device uses the path MTU to calculate the MSS to avoid IP fragmentation. The path MTU uses an aging mechanism to ensure that the source device can increase the path MTU when the minimum link MTU on the path increases.

TCP path MTU discovery works as follows:

1. A TCP source device sends a packet with the Don't Fragment (DF) bit set.
2. A router discards the packet that exceeds the MTU of the outgoing interface and returns an ICMP error message. The error message contains the MTU of the outgoing interface.
3. Upon receiving the ICMP message, the TCP source device calculates the current path MTU of the TCP connection.
4. The TCP source device sends subsequent TCP segments that each are smaller than the MSS (MSS = path MTU – IP header length – TCP header length).

If the TCP source device still receives ICMP error messages when the MSS is smaller than 32 bytes, the TCP source device will fragment packets.

An ICMP error message received from a router that does not support RFC 1191 has the MTU of the outgoing interface set to 0. Upon receiving the ICMP message, the TCP source device selects the path MTU smaller than the current path MTU from the MTU table as described in RFC 1191. Based on the selected path MTU, the TCP source device calculates the TCP MSS. The MTU table contains MTUs of 68, 296, 508, 1006, 1280, 1492, 2002, 4352, 8166, 17914, 32000, and 65535 bytes. Because the minimum TCP MSS specified by the system is 32 bytes, the actual minimum MTU is 72 bytes.

The aging mechanism of the path MTU is as follows:

- When the TCP source device receives an ICMP error message, it reduces the path MTU and starts an aging timer for the path MTU.
- After the aging timer expires, the source device uses a larger MSS in the MTU table, as described in RFC 1191.

- If no ICMP error message is received within two minutes, the source device increases the MSS again until the MSS negotiated during TCP three-way handshake is reached.

Prerequisites

Make sure all devices on a TCP connection are enabled to send ICMP error messages by using the **`ip unreachables enable`** command.

Procedure

1. Enter system view.

   `system-view`

2. Enable TCP path MTU discovery.

   `tcp path-mtu-discovery` [**`aging`** *`age-time`* | **`no-aging`**]

   By default, TCP path MTU discovery is disabled.

Enabling SYN Cookie

About SYN Cookie

A TCP connection is established through a three-way handshake. An attacker can exploit this mechanism to mount SYN Flood attacks. The attacker sends a large number of SYN packets, but does not respond to the SYN ACK packets from the server. As a result, the server establishes a large number of TCP semi-connections and can no longer handle normal services.

SYN Cookie can protect the server from SYN Flood attacks. When the server receives a SYN packet, it responds with a SYN ACK packet without establishing a TCP semi-connection. The server establishes a TCP connection and enters ESTABLISHED state only when it receives an ACK packet from the client.

Procedure

1. Enter system view.

   `system-view`

2. Enable SYN Cookie.

   `tcp syn-cookie enable`

   By default, SYN Cookie is disabled.

Setting the TCP buffer size

1. Enter system view.

   `system-view`

2. Set the size of TCP receive/send buffer.

   `tcp window` *`window-size`*

   The default buffer size is 63 KB.

Setting TCP timers

About TCP timers

You can set the following TCP timers:

- **SYN wait timer**—TCP starts the SYN wait timer after sending a SYN packet. Within the SYN wait timer if no response is received or the upper limit on TCP connection tries is reached, TCP fails to establish the connection.

- **FIN wait timer**—TCP starts the FIN wait timer when TCP changes the connection state to FIN_WAIT_2. If no FIN packet is received within the timer interval, TCP terminates the connection. If a FIN packet is received, TCP changes the connection state to TIME_WAIT. If a non-FIN packet is received, TCP restarts the timer, and tears down the connection when the timer expires.

Procedure

1. Enter system view.

   `system-view`

2. Set the TCP SYN wait timer.

   `tcp timer syn-timeout` *`time-value`*

   By default, the TCP SYN wait timer is 75 seconds.

3. Set the TCP FIN wait timer.

   `tcp timer fin-timeout` *`time-value`*

   By default, the TCP FIN wait timer is 675 seconds.

Enabling the Timestamps option encapsulation in outgoing TCP packets

About enabling the Timestamps option encapsulation in outgoing TCP packets

Devices at each end of the TCP connection can calculate the RTT value by using the TCP Timestamps option carried in TCP packets. For security purpose in some networks, you can disable this feature at one end of the TCP connection to prevent intermediate devices from obtaining the Timestamps option information.

Procedure

1. Enter system view.

   `system-view`

2. Enable the device to encapsulate the TCP Timestamps option in outgoing TCP packets.

   `tcp timestamps enable`

   By default, the TCP timestamps option is encapsulated in outgoing TCP packets.

Display and maintenance commands for IP performance optimization

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

Task	Command
Display ICMP statistics.	**`display icmp statistics`** [**`slot`** *`slot-number`*]
Display IP packet statistics.	**`display ip statistics`** [**`slot`** *`slot-number`*]

Task	Command
Display brief information about RawIP connections.	**display rawip** [**slot** *slot-number*]
Display detailed information about RawIP connections.	**display rawip verbose** [**slot** *slot-number* [**pcb** *pcb-index*]]
Display brief information about TCP connections.	**display tcp** [**slot** *slot-number*]
Display TCP traffic statistics.	**display tcp statistics** [**slot** *slot-number*]
Display detailed information about TCP connections.	**display tcp verbose** [**slot** *slot-number* [**pcb** *pcb-index*]]
Display brief information about UDP connections.	**display udp** [**slot** *slot-number*]
Display UDP traffic statistics.	**display udp statistics** [**slot** *slot-number*]
Display detailed information about UDP connections.	**display udp verbose** [**slot** *slot-number* [**pcb** *pcb-index*]]
Clear IP packet statistics.	**reset ip statistics** [**slot** *slot-number*]
Clear TCP traffic statistics.	**reset tcp statistics**
Clear UDP traffic statistics.	**reset udp statistics**