Link Layer Protocol

QSR-1920, QSR-2920, QSR-3920

Оглавление

1. HDLC 7
1.1. Overview 7
1.2. HDLC Function Configuration 7
1.2.1. Configure HDLC Basic Functions 8
1.2.2. Configure HDLC Link Keepalive 8
1.2.3. Configure HDLC Peer Address 9
1.2.4. HDLC Monitoring and Maintaining 10
1.3. HDLC Typical Configuration Example 10
1.3.1. Configure HDLC Link Protocol 10
2. PPP 13
2.1. Overview 13
2.2. PPP Function Configuration 13
2.2.1. Configure PPP Basic Functions 14
2.2.2. Configure PPP Link Keepalive 15
2.2.3. Configure PPP Authentication 16
2.2.4. Configure PPP Negotiation Parameters 23
2.2.5. Configure PPP AAA Function 28
2.2.6. Configure PPP Multi-link 32
2.2.7. PPP Monitoring and Maintaining 35
2.3. PPP Typical Configuration Example 35
2.3.1. Encapsulate PPP Protocol 35
2.3.2. Configure CHAP Uni-directional Authentication of Using User Name 37
2.3.3. Configure CHAP Uni-directional Authentication of Not Using User Name 41
2.3.4. Configure PAP Uni-directional Authentication 45
2.3.5. Configure PPP Compression Instance 48
2.3.6. Configure PPP AAA Authentication 51
2.3.7. Configure MPPP Binding 55
3. FRAME RELAY 59
3.1. Overview 59
3.1.1. Work Principle of Frame Relay 59
3.1.2. Frame Format of Frame Relay 60
3.1.3. Congestion Control of Frame Relay 61
3.1.4. LMI Protocol of Frame Relay 62
3.1.5. Address Mapping of Frame Relay 63
3.1.6. Multi-link Frame Relay 63
3.2. Frame Relay Function Configuration 64

3.2.1. Configure Basic Functions of Frame Relay 64
3.2.2. Configure Frame Relay DLCI 66
3.2.3. Configure Address Mapping of Frame Relay 67
3.2.4. Monitoring and Maintaining of Frame Relay 69
3.3. Typical Configuration Example of Frame Relay 69
3.3.1. Configure Private Line Interconnection of Frame Relay Devices 69
3.3.2. Configure Network Interconnection of Frame Relay Devices 71
3.3.3. Configure Point-to-Multipoint Sub Interface of Frame Relay 72
3.3.4. Configure Point-to-Point Sub Interface of Frame Relay 75
4. VIRTUAL ETHERNET 77
4.1. Overview 77
4.2. Virtual Ethernet Function Configuration 77
4.2.1. Configure Basic Functions of Virtual Ethernet 77
4.2.2. Configure MAC Address of Virtual Ethernet Interface 78
4.2.3. Configure Keepalive Function of Virtual Ethernet Interface 79
4.2.4. Configure Keepalive Receiving Timeout of Virtual Ethernet Interface 80
4.2.5. Monitoring and Maintaining of Virtual Ethernet 80
4.3. Typical Configuration Example of Virtual Ethernet 81
4.3.1. Configure Private Line Interconnection of Virtual Ethernet Devices 81
4.3.2. Configure Virtual Ethernet Bridge 84
5. BRIDGING 88
5.1. Overview 88
5.2. Bridge Function Configuration 88
5.2.1. Configure Bridging Basic Functions 89
5.2.2. Configure Bridging Parameters 89
5.2.3. Configure BVI Interface 91
5.2.4. Bridge Monitoring and Maintaining 92
5.3. Bridge Typical Configuration Example 92
5.3.1. Configure Common Bridging 92
5.3.2. Configure Virtual Interface of Bridging Group 94
6. PPPOE 96
6.1. Overview 96
6.1.1. PPPoE Discovery Stage 96
6.1.2. PPPoE Session Stage 97
6.2. PPPoE Function Configuration 97
6.2.1. Configure DDR Basic Functions 97
6.2.2. Configure PPPoE Dialing Interface 100

6.2.3. Configure PPPoE Dialing Mode 100
6.2.4. Configure PPPoE Server 101
6.2.5. PPPoE Monitoring and Maintaining 102
6.3. PPPoE Typical Configuration Example 103
6.3.1. Configure PPPoE Conventional Dialing 103
6.3.2. Configure PPPoE Auto Dialing 105
6.3.3. Configure PPPoE Auto Dialing Based on Time Period 108
6.3.4. Configure PPPoE IPv6 Protocol to Trigger Dialing 111
6.3.5. Configure PPPoE to Assign to IPv6 Address via DHCPv6 Protocol 113
7. DDR 117
7.1. Overview 117
7.1.1. Overview of DDR 117
7.1.2. DDR Terms 117
7.2. DDR Function Configuration 117
7.2.1. Configure DDR Dialing Control List 118
7.2.2. Configure DDR Basic Functions 119
7.2.3. Configure DDR Interface Parameters 120
7.2.4. DDR Monitoring and Maintaining 122
8. NDIS-DIAL 123
8.1. Overview 123
8.2. NDIS-DIAL Function Configuration 123
8.2.1. Configure NDIS-DIAL Basic Function 123
8.2.2. Configure 4G to Associate with Track 125
8.2.3. Configure 4G to Associate with BFD 126
8.2.4. NDIS-DIAL Monitoring and Maintaining 128
9. QINQ TERMINATION 129
9.1. Overview 129
9.2. QinQ Termination Function Configuration 129
9.2.1. Daughter Interface Encapsulating QinQ Termination 129
9.2.2. Configure IP Priority Copy 130
9.2.3. Configure VLAN TPID 131
9.3. Typical Configuration Example of QinQ Termination 133
9.3.1. Configure QinQ Termination 133
10. ETHERNET LINK 137
10.1. Overview 137
10.2. Ethernet Link Function Configuration 137
10.2.1. Sub Interface Encapsulating VLAN 137

10.2.2. Configure Small Packet Software Filling 138
11. ETHERNET LINK AGGREGATION 139
11.1. Overview 139
11.2. Basic Concepts 139
11.2.1. Link Aggregation Mode 140
11.3. Ethernet Link Aggregation Function Configuration 142
11.3.1. Configure Basic Functions of Ethernet Link Aggregation 143
11.3.2. Configure MAC Address of Ethernet Link Aggregation Interface 145
11.3.3. Configure Load Mode of Ethernet Link Aggregation Group 146
11.3.4. Configure System Priority of Ethernet Link Aggregation 146
11.3.5. Configure LACP Priority of Ethernet Link Aggregation Member Interface 147
11.3.6. Configure LACP Timeout Period of Ethernet Link Aggregation Member Interface 147
11.3.7. Configure the Independent Port Mode of Ethernet Link Aggregation Group 148
11.3.8. Configure Ethernet Link Aggregation Group to Link with BFD 148
11.3.9. Configure Bandwidth Weight Load of Ethernet Link Aggregation Group Member 149
11.3.10. Configure Ethernet Link Aggregation Group Member qos-group 150
11.3.11. Configure Ethernet Link Aggregation Group Member Interface to Force Congestion Packet to Fall Back 151
11.3.12. Ethernet Link Aggregation Monitoring and Maintaining 152
11.4. Typical Configuration Example of Ethernet Link Aggregation 153
11.4.1. Configure Static Ethernet Link Aggregation Group 153
11.4.2. Configure Dynamic Ethernet Link Aggregation Group 156
11.4.3. Configure Sub Interface Link Aggregation Group 159
11.4.4. Configure QoS on Ethernet Link Aggregation 163
11.4.5. Configure Ethernet Link Aggregation Group to Link with BFD 168
11.4.6. Link Aggregation Combines with qos-group id to Realize Service Diversion and Link Detection 171
12. SNA 177
12.1. Overview of SNA 177
12.2. DLSw Configuration 177
12.2.1. Configure DLSw Local Peer Entity 177
12.2.2. Configure DLSw Remote Peer Entity 178
12.2.3. Disable DLSw 179
12.2.4. Configure DLSw SSP Protocol Capability 179
12.2.5. Configure DLSw MAC Address Time Domain Control 179
12.3. SDLC Configuration 180
12.3.1. Configure SDLC Basic Functions 180

12.3.2. Configure SDLC Local Physical Address 180
12.3.3. Configure WMAC Address of the Specified Interface 181
12.3.4. Configure SDLC Peer Physical Address 182
12.3.5. Configure SDLC Interface DLSw 182
12.3.6. Configure XID Value of the SDLC Interface 183
12.3.7. Configure Delay Response Time of SDLC Interface 184
12.3.8. Configure the Window Size of SDLC Interface 184
12.3.9. Configure the Times of SDLC Interface Re-sending Frames 185
12.3.10. Configure the Interval of SDLC Interface Polling Frames 185
12.3.11. Configure the Time of SDLC Interface waiting for Polling Frames 186
12.3.12. Configure Link Site Role of SDLC Interface 186
12.3.13. Configure Source and Destination snap Value of Remote SDLC Link 187
12.3.14. Configure Max. Size of Received and Sent Frames of SDLC Interface 187
12.3.15. Configure the Time of SDLC Interface Waiting for Response Information 188
13. ОБЩАЯ ИНФОРМАЦИЯ 189
13.1. Замечания и предложения 189
13.2. Гарантия и сервис 189
13.3. Техническая поддержка 189

1. HDLC

1.1. Overview

HDLC (High-level Data Link Control) is one bit-oriented data link layer protocol. In the early computer communications, the data link layer protocol is character-oriented, that is, the data transmitted on the link should comprise the characters in the defined character set (such as ASCII code). Moreover, the control information transmitted on the link also should comprise several specified control characters in one character set. However, with the development of the computer communication, the character-oriented link control procedure gradually exposes its weakness and we need to design one new link control protocol. In 1974, IBM put forward the systems network architecture (SNA). The data link layer procedure of SNA adopts the bit-oriented regulation SDLC (Synchronous Data Link Control). Later, the American National Standards Association ANSI SDLC was modified to ADCCP (advanced data communication control procedure) as the national standard of the United States, while ISO called the SDLC after the modification as HDLC, and it was regarded as the international standard ISO 3309 (data communication - high level data link control procedure - frame structure). The corresponding national standard is GB 7496 (frame structure of advanced data link control procedure of information processing system). SDLC is put forward earlier, but in fact, it is one sub set of HDLC.

The HDLC protocol includes the following points:

- Defines the frame structure transmitted on the link is defined;
- Defines the basic configuration and data transmission mode of the link, and provides the corresponding command response subsets and extended function options according to different data transmission modes;
- Defines a set of protocol control procedures for establishing and closing connections, data transmission, flow control, error control, etc.

In addition, the general application rarely needs to use the full set of HDLC. When using HDLC of a manufacturer, it is necessary to find out what subset the manufacturer chooses.

In the current implementation and practical application of HDLC protocol, most manufacturers only encapsulate the upper layer data according to the frame format of HDLC standard, without any confirmation mechanism, retransmission mechanism, traffic control, etc. all the error correction processing is processed by the upper layer protocol.

1.2. HDLC Function Configuration

Table 1-1HDLC function configuration list

Configuration Task	
Configure the HDLC basic functions	Encapsulate the HDLC protocol
Configure the HDLC link keepalive	Configure the detection period of the HDLC link keepalive
Configure the HDLC peer address	Configure the HDLC peer IP address
	Configure HDLC to update the peer IP address regularly

1.2.1. Configure HDLC Basic Functions

Configuration Condition

None

Encapsulate HDLC Protocol

Table 1-2 Encapsulate the HDLC protocol

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Encapsulate the HDLC protocol	**encapsulation hdlc**	Mandatory By default, the WAN card interface is encapsulated with the link-layer protocol PPP.

1.2.2. Configure HDLC Link Keepalive

Link keepalive is called keepalive, and it periodically sends the keepalive packets to detect the link status.

Configuration Condition

Before configuring the HDLC link keepalive, first complete the following task:

- The interface encapsulates the HDLC protocol.

Configure Detection Period of HDLC Link Keepalive

Table 1-3 Configure the detection period of the HDLC link keepalive

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the detection period of the HDLC link keepalive	**keepalive** [*keepalive-seconds*]	Mandatory By default, the detection period of the HDLC link keepalive is 10s.

Note:

- The periods of the link keepalive at the two sides of HDLC should be consistent. Otherwise, it may result in the check abnormality and the link is regarded as interrupted.

1.2.3. Configure HDLC Peer Address

HDLC is the point-to-point interface and the two parties should know the peer IP address so that they can communicate.

HDLC sends the address request to the peer and responds the peer address request to perform the address negotiation. After negotiating successfully, the two parties get the peer IP address, but if one does not have the address negotiation function, we need to set the peer IP address via the command.

After getting the peer IP address via the address negotiation successfully and if the peer modifies the IP address, but does not actively re-negotiate, the save peer IP address of the local end is till old and it will affect the normal communication. Therefore, we need to regularly update the peer IP address.

Configuration Condition

Before configuring the HDLC peer address, first complete the following task:

- The interface encapsulates the HDLC protocol.Configure HDLC Peer IP Address

Table 1-4 Configure the peer IP address of HDLC

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the peer IP address	**peer ip addr** *peer-ip-address*	Mandatory By default, the peer IP address is not configured, but is got from the address negotiation.

Configure HDLC to Update Peer IP Address Regularly

Table 1-5Configure HDLC to update the peer IP address regularly

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure HDLC to update the peer IP address regularly	**peer ip addr update** [**interval** *interval-number*]	Optional By default, update the peer IP address every 30s.

1.2.4. HDLC Monitoring and Maintaining

None

1.3. HDLC Typical Configuration Example

1.3.1. Configure HDLC Link Protocol

Network Requirement

- Device1 and Device2 are connected via WAN port and the interface encapsulation type is HDLC. The WAN interface of Device1 can communicate with the WAN interface of Device2.

Network Topology

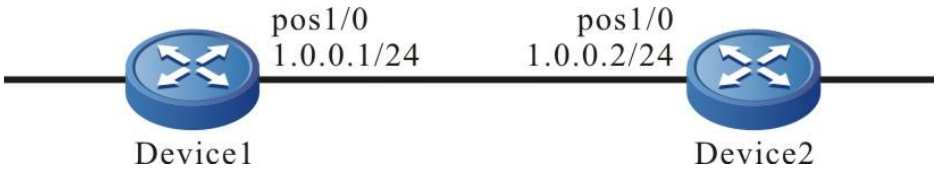

Figure 1-1Networking of configuring the HDLC link protocol

Configuration Steps

Step 1: Configure the clock mode of the POS interface. Device1 configure the internal clock and Device2 configures the external clock (the external clock is the default clock mode and does not need to configure manually).

#Configure Device1.

```
Device1#configure terminal
Device1(config)#interface pos1/0
Device1(config-if-pos1/0)#clock source internal
```

#Configure Device2.

```
Device2#configure terminal
```

```
Device2(config)#interface pos1/0
```

Step 2: Configure the link protocol of Device1 and Device2 WAN interface. pos1/0 of Device1 configures the HDLC protocol and pos1/0 of Device2 configures the HDLC protocol.

#Configure Device1.

```
Device1(config-if-pos1/0)#encapsulation  hdlc
```

#Configure Device2.

```
Device2(config-if-pos1/0)#encapsulation  hdlc
```

Step 3: Configure the IP address of the interface.

#Configure Device1.

```
Device1(config-if-pos1/0)#ip address  1.0.0.1 255.255.255.0
Device1(config-if-pos1/0)#exit
Device1(config)#exit
```

#Configure Device2.

```
Device2(config-if-pos1/0)#ip address  1.0.0.2 255.255.255.0
Device2(config-if-pos1/0)#exit
Device2(config)#exit
```

Step 4: Check the result.

#View the serial1/0 interface status of Device1.

```
Device1#show interface pos1/0
pos1/0:
    line protocol is up
    Flags: (0xc0080f1) POINT-TO-POINT MULTICAST RUNNING
    Type: HDLC
    Internet address: 1.0.0.1/24
    Destination Internet address: 1.0.0.2
    Metric: 0, MTU: 1500, BW: 155000 Kbps, DLY: 20000 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Last clearing of "show interface" counters never
    input peak rate 89 bits/sec,  0 hour 1 minute 0 second ago
    output peak rate 95 bits/sec,  0 hour 1 minute 0 second ago
    4 minutes 50 seconds input rate 23 bits/sec, 0 packet/sec
    4 minutes 50 seconds output rate 23 bits/sec, 0 packet/sec
    7 packets received; 7 packets sent
```

```
0 multicast packets received
0 multicast packets sent
0 input errors; 0 output errors
0 collisions; 0 dropped
 hdlc version: 3.5
 rxFrames: 271, rxChars 3296
 txFrames: 272, txChars 3330
 rxNoOctet 0, rxAbtErrs 0, rxCrcErrs 48
 rxOverrun 0, rxLenErrs 57, txUnderrun 0
 DCD: UP
```

#Ping the address of the peer interface pos1/0 on Device1 and the ping can be connected.

```
Device1#ping 1.0.0.2
```

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.

2. PPP

2.1. Overview

PPP (Point to Point Protocol) is one data link layer protocol of transmitting the network layer data on the point-to-point line. PPP includes LCP, NCP and authentication protocol (PAP and CHAP). It can support synchronous and asynchronous lines. PPP is suitable for serial systems with different characteristics and can transmit a variety of network layer protocol data. It is a common method for connecting various types of hosts, bridges and routers.

PPP mainly consists of the following three parts:

- The method of encapsulating multiple network layer protocol packets.
- Link control protocol (LCP) for establishing, configuring and testing data link connections.
- A set of network control protocols (NCPs) for establishing and configuring different network layer protocols.

2.2. PPP Function Configuration

Table 2-1 PPP function configuration list

Configuration Task	
Configure the PPP basic functions	Encapsulate the PPP protocol
Configure the PPP link keepalive	Configure the detection period of the PPP link keepalive
Configure the PPP authentication	Configure the PPP PAP authentication
	Configure the PPP CHAP authentication
	Configure the PPP MS-CHAP authentication
	Configure PPP MS-CHAP-V2 authentication
	Configure refusing authentication
Configure the PPP negotiation parameters	Configure the PPP LCP re-negotiation interval
	Configure the forecasting of the LCP connection status

Configuration Task	
	Configure the ID binding function of the LCP system
	Configure the PPP IP address negotiation
	Configure the PPP IP address pool
	Configure the PPP WINS DNS address negotiation
	Configure PPP negotiation address control field compression
Configure the PPP AAA function of PPP	Configure the PPP AAA authentication
	Configure the PPP AAA authorization
	Configure the PPP AAA statistics
Configure the PPP multi-link	Configure the PPP multi-link basic functions
	Configure the PPP multi-link fragment inserting
	Configure the PPP multi-link end-point ID
	Configure the PPP multi-link fragment delay
	Configure the PPP multi-link loading mode

2.2.1. Configure PPP Basic Functions

Configuration Condition

None

Encapsulate PPP Protocol

Table 2-2 Encapsulate the PPP protocol

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Encapsulate the PPP protocol	**encapsulation ppp**	Mandatory By default, the encapsulated link-layer protocol of the POS interface is PPP.

2.2.2. Configure PPP Link Keepalive

Link keepalive periodically sends the keepalive packets to detect the link status.

Configuration Condition

Before configuring the PPP authentication, first complete the following task:

- Encapsulate the PPP protocol on the interface

Configure Detection Period of PPP Link Keepalive

Table 2-3Configure the detection period of the PPP link keepalive

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the detection period of the PPP link keepalive	**keepalive** [*keepalive-seconds*]	Optional By default, the detection period of the PPP link keepalive is 10s.

2.2.3. Configure PPP Authentication

The configuration of the PPP authentication is optional. For the security of the PPP link communication, we need to configure the PPP authentication, which includes three authentication modes, that is, PAP, CHAP, MS-CHAP and MS-CHAP-V2

To configure the PPP authentication, we need to configure one side of PPP as the PPP authenticating end and the other side as the authenticated end.

Configuration Condition

Before configuring the PPP authentication, first complete the following task:

- Encapsulate the PPP protocol on the interface

Configure PPP PAP Authentication

To configure the PPP PAP authentication, we need to configure one end as the authenticating end and the other is configured as the authenticated end. The configuration of the authenticating end is as follows:

Table 2-4 Configure the PAP authenticating end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP authentication mode as the PAP authentication	**ppp authentication pap**	Mandatory By default, do not configure the authentication mode.
Exit the interface configuration mode	**exit**	-
Configure the access user name	**local-user** *user-name* **class network**	Mandatory By default, do not configure the access user name.
Configure the password of the access user	**password 0** *password*	Configure the password of the access user

Step	Command	Description
Configure the PPP service used by the access user	**serverice-type ppp**	Mandatory By default, the service of the access user is not configured.

The configuration of the authenticated end is as follows:

Table 2-5Configure the authenticated end of PAP

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the authenticated user name and password of the PPP PAP	**ppp pap sent-username** *username* **password** [*encryption-type*] *password*	Mandatory By default, do not configure the user name or password.

Configure PPP CHAP Authentication

To configure the PPP CHAP authentication, we need to configure one end as the authenticating end and the other end as the authenticated end.

The configuration of the authenticating end is as follows:

Table 2-6 Configure the authenticating end of CHAP

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP authentication mode as CHAP authentication	**ppp authentication chap**	Mandatory By default, do not configure the authentication mode
Exit the interface configuration mode	**exit**	-
Configure the access user name	**local-user** *user-name* **class network**	Mandatory By default, do not configure the access user name.
Configure the password of the access user	**password 0** *password*	Configure the password of the access user
Configure the PPP service used by the access user	**serverice-type ppp**	Mandatory By default, the service of the access user is not configured.

The configuration of the authenticated end is as follows:

Table 2-7 Configure the authenticated end of CHAP

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the authenticated user name of PPP CHAP	**ppp chap hostname** *host-name*	Mandatory By default, do not configure the user name.
Configure the authenticated password of PPP CHAP	**ppp chap password** [*encryption-type*] *password*	Mandatory By default, do not configure the password.

Configure PPP MS-CHAP Authentication

To configure PPP MS-CHAP authentication, we need to configure one end as the authenticating end and the other as the authenticated end.

The configuration of the authenticating end is as follows:

Table 2-8 Configure MS-CHAP authenticating end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP authentication mode as MS-CHAP	**ppp authentication ms-chap**	Mandatory By default, do not configure the authentication mode.

Step	Command	Description
Exit the interface configuration mode	**exit**	-
Configure the access user name	**local-user** *user-name* **class network**	Mandatory By default, do not configure the access user name.
Configure the password of the access user	**password 0** *password*	Configure the password of the access user
Configure the PPP service used by the access user	**serverice-type ppp**	Mandatory By default, the service of the access user is not configured.

The configuration of the authenticated end is as follows:

Table 2-9 Configure the MS-CHAP authenticated end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the authenticated user name of PPP CHAP	**ppp chap hostname** *host-name*	Mandatory By default, do not configure the user name.
Configure the authenticated password of PPP CHAP	**ppp chap password** [*encryption-type*] *password*	Mandatory By default, do not configure the password.

Configure PPP MS-CHAP-V2 authentication

To configure PPP MS-CHAP-V2 authentication, we need to configure one end as the authenticating end and the other as the authenticated end.

The configuration of the authenticating end is as follows:

Table 2-10 Configure MS-CHAP-V2 authenticating end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP authentication mode as MS-CHAP-V2	**ppp authentication ms-chap-v2**	Mandatory By default, do not configure the authentication mode.
Exit the interface configuration mode	**exit**	-
Configure the access user name	**local-user** *user-name* **class network**	Mandatory By default, do not configure the access user name.
Configure the password of the access user	**password 0** *password*	Configure the password of the access user
Configure the PPP service used by the access user	**serverice-type ppp**	Mandatory By default, the service of the access user is not configured.

The configuration of the authenticated end is as follows:

Table 2-11 Configure the MS-CHAP-V2 authenticated end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the authenticated user name of PPP CHAP	**ppp chap hostname** *host-name*	Mandatory By default, do not configure the user name.
Configure the authenticated password of PPP CHAP	**ppp chap password** [*encryption-type*] *password*	Mandatory By default, do not configure the password.

Configure Refusing authentication

For the network security, adopt multiple authentication modes during the PPP communication, but sometimes, to control the authentication mode better, we can refuse some authentication mode.

Table 2-12Configure refusing authentication

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure refusing authentication	**ppp { chap | ms-chap | pap | ms-chap-v2} refuse**	Mandatory By default, do not configure refusing authentication.

2.2.4. Configure PPP Negotiation Parameters

The PPP negotiation parameters are mainly used to set the negotiation parameters, such as LCP, IPCP and authentication.

Configuration Condition

Before configuring the PPP negotiation, first complete the following task:

- Encapsulate the PPP protocol on the interface

Configure PPP LCP Re-negotiation Interval

When the PPP LCP negotiation fails, re-negotiate and the interval of the two negotiations can be configured as follows:

Table 2-13 Configure the PPP LCP re-negotiation interval

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP LCP re-negotiation interval	**ppp timeout retry** *timeout*	Mandatory By default, the PPP LCP re-negotiation interval is 5s.

Configure LCP Connection Status Forecasting

PPP can detect the link status via the link keepalive packet and data packet. Usually, adopt the link keepalive packet to detect, but sometimes, to save the bandwidth, we can detect the link status via the data packet. The configuration is as follows:

Table 2-14 Configure the LCP connection status forecasting

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the packet type of detecting the link status	**ppp lcp predictive**	Mandatory By default, enable the LCP connection status forecasting.

Configure Binding Function of LCP System ID

The function is to send the system ID of the dialing end to the server. When configuring the binding function of the LCP system ID, we need to configure the dialing end as the system ID accept mode and the server as the system ID request mode.

The configuration of the dialing end is as follows:

Table 2-15 Configure the dialing end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the system ID	**ppp sn { ***serial-number* **| default }**	Mandatory By default, the dialing end does not configure the system ID.
Configure the accept mode	**ppp lcp sn accept**	Mandatory By default, the dialing end is not configured as the accept mode.

The configuration of the server is as follows:

Table 2-16 Configure the server

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the request mode	**ppp lcp sn request**	Mandatory By default, the server is not configured as the request mode.

Configure PPP IP Address Negotiation

The IP address of the PPP interface can be configured manually and also can be got by negotiating with the peer. When configuring the IP address negotiation, one end is configured to request the IP address from the peer and the other end is configured to distribute the IP address for the peer. The distributing mode can be specifying the specific IP address and also can be distributed via the IP address pool.

To get the IP address from the peer, we need to perform the following configuration:

Table 2-17Configure PPP to request IP address from the peer

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure PPP to request IP address from the peer	**ip address negotiated**	Mandatory By default, do not request IP address from the peer.

To distribute the IP address for the peer, we need to perform the following configuration:

Table 2-18Configure PPP to distribute the IP address for the peer

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure PPP to distribute the IP address for the peer	**peer default ip address** { *ip-address* | **pool** *pool-name* }	Mandatory Specify the IP address or distribute the IP address for the peer via the IP address pool. By default, do not distribute the IP address for the peer.

Configure PPP IP Address Pool

PPP performs the IP address negotiation. When one end is configured to distribute the IP address for the peer via the IP address pool, we need to configure the corresponding address pool as follows:

Table 2-19 Configure the PPP IP address pool

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the PPP IP address pool	**ip local pool** *pool-name begin-ip-address end-ip-address*	Mandatory By default, do not configure the IP address pool.

Configure PPP WINS DNS Address Negotiation

When configuring the PPP WINS DNS negotiation, one end needs to be configured as requesting WINS DNS address and the other end is configured as distributing the WINS DNS address.

To configure the WINS DNS negotiation, we need to perform the following configuration:

Table 2-20 Configure PPP to request the WINS DNS address

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure requesting the DNS address	**ppp ipcp dns request**	Mandatory By default, do not configure requesting the DNS address.
Configure requesting the WINS address	**ppp ipcp wins request**	Mandatory By default, do not configure requesting the WINS address.

Table 2-21 Configure PPP to distribute the WINS DNS address

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the DNS address distributed for the peer	**ppp ipcp dns** *primary-address* [*second-address*]	Mandatory By default, do not configure the DNS address.
Configure the WINS address distributed for the peer	**ppp ipcp wins** *primary-address* [*second-address*]	Mandatory

Step	Command	Description
		By default, do not configure the WINS address.

Configure PPP Negotiation Address Control Field Compression

Table 2-22 Configure PPP negotiation address control field compression

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP address field compression	**ppp ac**	Mandatory By default, do not configure the PPP address field compression.
Configure the PPP protocol field compression	**ppp pc**	Mandatory By default, do not configure the PPP protocol field compression.

2.2.5. Configure PPP AAA Function

AAA includes authenticating, authorizing and accounting. PPP AAA authentication is used to send the user name and password information of the PPP dialing end to the AAA server for authentication. PPP AAA authorizing is mainly used by the AAA server to perform the authorizing and distributing the IP address function for the PPP dialing end. AAA accounting is mainly used for the accounting usage of the PPP line.

Configuration Condition

Before configuring the PPP AAA function, first complete the following task:

- Encapsulate the PPP protocol on the interface
- The authenticating end, authorizing end and accounting end need to enable the AAA function

Configure PPP AAA Authentication

To configure the PPP AAA authentication, we need to configure one end as the authenticating end and the authenticating mode as the AAA authentication, and the other end as the authenticated end.

The configuration of the authenticating end is as follows:

Table 2-23 Configure the PPP AAA authenticating end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the PPP AAA authentication server group name (take the RADIUS server as an example)	**aaa server group radius** *group-name*	Mandatory By default, do not configure the RADIUS server group name.
Configure the related parameters of the AAA authentication server (take the RADIUS server as an example)	**server {***ip-address*|*ipv6 ip-address***} [acc-port** *acc-port-num* **] [auth-port** *auth-port-num* **] [priority** *priority* **] { key [0 | 7]** *key* **}**	Mandatory By default, do not configure the RADIUS server.
Exit the server configuration mode	**exit**	
Enter the ISP domain view	**domain** *isp-name*	isp-name is the specified ISP domain name.
Configure the PPP authentication method list in the ISP domain	**aaa authentication ppp { none | local | radius-group** group-name **| tacacs-group** group-name **}**	Optional By default, the default authentication method list in the ISP domain is local.
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the PPP authentication mode and the AAA isp-name	**ppp authentication** { **chap** | **pap** | **ms-chap | ms-chap-v2** } [isp-name]	Mandatory The AAA isp-name should be configured. By default, do not configure the PPP authentication mode and AAA isp-name.

For the configuration of the authenticated end, refer to the section 1.2.3. The authentication modes of the authenticating end and the authenticated end need to be consistent.

Configure PPP AAA Authorizing

To configure the PPP AAA authorizing, we need to configure one end as the authorizing end and the other end as the authorized end.

Table 2-24 Configure the PPP AAA authorized end

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the PPP AAA authentication server group name (take the RADIUS server as an example)	**aaa server group radius** *group-name*	Mandatory By default, do not configure the RADIUS server group name.
Configure the related parameters of the AAA authentication server (take the RADIUS server as an example)	**server {***ip-address*|*ipv6 ip-address***} [acc-port** *acc-port-num* **] [auth-port** *auth-port-num* **] [priority** *priority* **] { key [0 | 7]** *key* **}**	Mandatory By default, do not configure the RADIUS server.
Exit the server configuration mode	**exit**	
Enter the ISP domain view	**domain** *isp-name*	isp-name is the specified ISP domain name.
Configure the PPP authorizing method list in the ISP domain	**aaa authorization ppp** { **if-authenticated** | **local** | **none** | **radius-group** *group-name* | **tacacs-group** *group-name* }	Optional By default, configure the authorizing method in the ISP domain as none.
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the AAA isp-name used by PPP authorizing	**ppp authentication** { **chap** | **pap** | **ms-chap | ms-chap-v2** } [isp-name]	Mandatory AAA isp-name is mandatory. By default, do not configure the PPP authentication mode and AAA isp-name.

For the configuration of the authorized end, refer to the section 1.2.4.

Configure PPP AAA Accounting

Table 2-25 Configure the AAA accounting

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the PPP AAA authentication server group name (take the RADIUS server as an example)	**aaa server group radius** *group-name*	Mandatory By default, do not configure the RADIUS server group name.
Configure the related parameters of the AAA authentication server (take the RADIUS server as an example)	**server {***ip-address|ipv6 ip-address***} [acc-port** *acc-port-num* **] [auth-port** *auth-port-num* **] [priority** *priority* **] { key [0 | 7]** *key* **}**	Mandatory By default, do not configure the RADIUS server.
Exit the server configuration mode	**exit**	
Enter the ISP domain view	**domain** *isp-name*	isp-name is the specified ISP domain name.
Configure the ppp accounting method list in the ISP domain	**aaa accounting ppp** { **none** | { **start-stop** | **stop-only** | **wait-start** [**broadcast**] { **radius-group** *group-name* | **tacacs-group** *group-name* } } }	Optional By default, configure the accounting method list in the ISP domain as none.
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the isp-name used by PPP AAA accounting	**ppp authentication** { **chap** | **pap** | **ms-chap** | **ms-chap-v2**} [isp-name]	Mandatory AAA isp-name is mandatory. By default, do not configure the PPP authentication mode or AAA isp-name.

2.2.6. Configure PPP Multi-link

PPP multi-link binds multiple physical links, so as to improve the throughput and reduce the transmission delay between the systems. If necessary, it fragments the packet and transmits on multiple physical links. When receiving, re-assembly the fragments and ensure receiving and sending in order.

Configuration Condition

None

Configure PPP Multi-link Basic Functions

Table 2-26 Configure the PPP multi-link basic functions

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Create the multi-link interface	**interface multilink** *multilink-unit*	Mandatory
Exit the interface mode	**exit**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Bind the physical interface and the multi-link interface	**mutlink-group** *multilink-unit*	Mandatory

Configure PPP Multi-link Fragment Inserting

Table 2-27 Configure the PPP multi-link fragment inserting

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface multilink** *multilink-unit*	-
Configure the PPP multi-link fragment inserting	**ppp multilink interleave**	Mandatory By default, the multi-link interface does not configure the fragment inserting.

Configure PPP Multi-link Endpoint ID

Table 2-28 Configure the PPP multi-link endpoint ID

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface multilink** *multilink-unit*	-
Configure the PPP multi-link endpoint ID	**ppp multilink endpoint string** *endpoint-name*	Mandatory By default, the multi-link interface does not configure the endpoint ID. By default, use the host name as the endpoint ID.

Configure PPP Multi-link Fragment Delay

Table 2-29 Configure the PPP multi-link fragment delay

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface multilink** *multilink-unit*	-
Configure the maximum delay of the multi-link fragment	**ppp multilink fragment-delay** *timeout*	Mandatory By default, the maximum fragment delay of the multi-link interface is 30ms.

Configure PPP Multilink to Receive Packets by Order

Table 2-30 Configure PPP multilink to receive packets by order

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface multilink** *multilink-unit*	-
Configure PPP multilink to receive packets by order	**ppp multilink input-order**	Mandatory By default, multilink interface receives the packets by order.

Configure PPP Multilink Loading Mode

Table 2-31 Configure the PPP multilink loading mode

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface multilink** *multilink-unit*	-
Configure the PPP multilink loading mode	**ppp multilink load-sharing** { **destination-ip** | **poll** | **source-destination-ip** | **source-ip** | **weight** }	Mandatory By default, the multilink loading mode is loading by the bandwidth weight of the member port.

2.2.7. PPP Monitoring and Maintaining

Table 2-32 monitoring and maintaining

Command	Description
show ppp information [*interface-name*]	Display the configuration information of the PPP interface and the status machine information

2.3. PPP Typical Configuration Example

2.3.1. Encapsulate PPP Protocol

Network Requirements

- Device1 and Device2 are connected via the WAN interface; in the example, use the CE1 interface.
- Two interfaces encapsulate the PPP protocol.

Network Topology

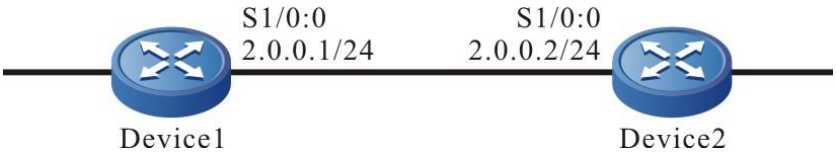

Figure 2-1 PPP networking

Configuration Steps

Step 1: Configure the un-framed in the controller of CE1 and configure the clock mode. Device1 configures the internal clock and Device2 configures the external clock (the external clock is the default clock mode and does not need to be configured).

#Configure Device1.

```
Device1#configure terminal
Device1(config)#controller e1 1/0
Device1(config-controller)#unframed
Device1(config-controller)#exit
```

Configure Device1 to use the internal clock.

```
Device1(config-controller)#clock source internal
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#controller e1 1/0
Device2(config-controller)#unframed
Device2(config-controller)#exit
```

Configure Device2 to use the defualt line clock.

Step 2: Configure the encapsulation type of the interface.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#encapsulation ppp
Device1(config-if-serial1/0:0)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#encapsulation ppp
Device2(config-if-serial1/0:0)#exit
```

Step 3: Configure the IP address of the interface.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ip address 2.0.0.1 255.255.255.0
Device1(config-if-serial1/0:0)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ip address 2.0.0.2 255.255.255.0
Device2(config-if-serial1/0:0)#exit
```

Step 4: Check the result.

#View whether the protocol is up and whether the peer IP address is got via the **show interface serial 1/0:0** command.

```
Device1#show  interface  serial1/0:0 Serial1/0:0:
   line protocol is up
   Flags: (0x80080f1) POINT-TO-POINT MULTICAST RUNNING
   Type: PPP
   Internet address: 2.0.0.1/24
   Destination Internet address: 2.0.0.2
   Metric: 0, MTU: 1500, BW: 2048 Kbps, DLY: 20000 usec, VRF: global
   Reliability 255/255, Txload 1/255, Rxload 1/255
   Last clearing of "show interface" counters never
```

```
input peak rate 64 bits/sec,  0 hour 2 minutes 27 seconds ago
output peak rate 64 bits/sec,  0 hour 2 minutes 27 seconds ago
2 minutes 40 seconds input rate 0 bit/sec, 0 packet/sec
2 minutes 40 seconds output rate 0 bit/sec, 0 packet/sec
29 packets received; 30 packets sent
0 multicast packets received
0 multicast packets sent
2 input errors; 0 output errors
0 collisions; 0 dropped
 LCP:OPENED
 IPCP:OPENED
 encap-type: simply PPP
   rxFrames 95, rxChars 1414
   txFrames 66, txChars 1266
   rxNoOctet 0, rxAbtErrs 0, rxCrcErrs 0
   rxOverrun 0, rxLenErrs 0, txUnderrun 0
   DCD=up
   rate=2048000 bps
```

On Device1, run the **show interface serial1/0**: **0** command and we can see “line protocol is up” and LCP and IPCP of PPP are opened. It indicates that the PPP negotiation of the link is successful. According to the information “Destination Internet address: ”, we can confirm that the peer IP address is got.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

2.3.2. Configure CHAP Uni-directional Authentication of Using User Name

Network Requirements

- Device1 and Device2 are connected via WAN interface; in the example, use the CE1 interface.
- Two interfaces encapsulate the PPP protocol.
- It is required that Device1 uses the CHAP mode to authenticate Device2 uni-directionally.
- Authenticate the peer via the CHAP mode when configuring the user name.

Network Topology

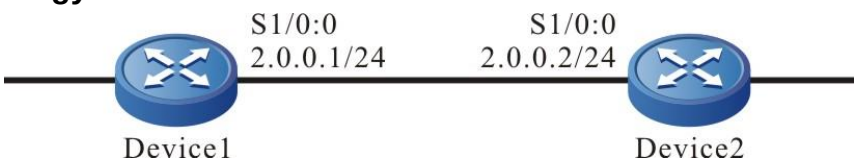

Figure 2-2 Networking of the CHAP uni-directional authentication using user name

Configuration Steps

Step 1: Configure unframed in the controller of CE1 and configure the clock mode.

(Omitted)

Step 2: Configure the encapsulation type and IP address of the interface.

(Omitted)

Step 3: Configure Device1 as the CHAP authenticating end.

#Configure Device1.

#Create the local user name and password for the peer CHAP authentication.

```
Device1#configure terminal
Device1(config)#local-user admin2 class network
Device1(config-user-network-admin2)#password 0 admin
Device1(config-user-network-admin2)#service-type ppp
Device1(config-user-network-admin2)#exit
```

#Configure Device1 as the CHAP authenticating end.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp authentication chap
Device1(config-if-serial1/0:0)#exit
```

#Configure the user name of Device1 used for the CHAP authentication.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp chap hostname admin1
Device1(config-if-serial1/0:0)#exit
```

Step 4: Configure Device2 as the CHAP authenticated end.

#Configure Device2.

#Create the local user name and password for the peer CHAP authentication.

```
Device2#configure terminal
```

```
Device2(config)#local-user admin1 class network
Device2(config-user-network-admin1)#password 0 admin
Device2(config-user-network-admin1)#service-type ppp
Device2(config-user-network-admin1)#exit
```

#Configure the user name of Device2 used for the CHAP authentication.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ppp chap hostname admin2
Device2(config-if-serial1/0:0)#exit
```

Step 5: Check the result.

#View the PPP information on Device1.

```
Device1#show ppp information serial 1/0:0
Serial1/0:0
    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                0x0
        local magic number       0x5d9935ca
        protocol field compression    OFF
        addr/ctrl field compression   OFF

        lcp echo interval        10
        lcp predictive           ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address      ON
        set peer address         ON
        local IP address         2.0.0.1
        remote IP address        2.0.0.2
        peer IP address          0.0.0.0
        vj compression protocol      OFF
        vj compression passive       OFF
        RTP compression protocol     OFF
```

```
        RTP compression passive       OFF
    PAP Stats
        client PAP state            CLOSED
        server PAP state             CLOSED
    CHAP Stats
        client CHAP state           CLOSED
        server CHAP state            OPEN
    Ifindex 1384710688, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
    PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
    Phy up, no shutdown, Reneg
```

#View the PPP information on Device2.

```
Device2#show ppp information serial 1/0:0
serial 1/0:0
    LCP Stats
        LCP phase                  NETWORK
        LCP state                 OPENED
        mru                     1500
        mtu                     1500
        async map                  0x0
        local magic number           0x3c16b24f
        protocol field compression     OFF
        addr/ctrl field compression    OFF
        lcp echo interval           10
        lcp predictive              ON
    IPCP Stats
        IPCP state                OPENED
        accept zero address          ON
        set peer address            ON
        local IP address           2.0.0.2
        remote IP address            2.0.0.1
        peer IP address            0.0.0.0
        vj compression protocol       OFF
        vj compression passive        OFF
        RTP compression protocol      OFF
        RTP compression passive       OFF
    PAP Stats
```

```
        client PAP state          CLOSED
        server PAP state          CLOSED
    CHAP Stats
        client CHAP state         OPEN
        server CHAP state         CLOSED
    Ifindex 1384219168, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
    PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
    Phy up, no shutdown, Reneg
```

After CHAP authentication succeeded, the PPP protocol is up. Run the **show ppp information** command and we can see that LCP and CHAP are in the OPEN state. After the NCP negotiation is complete, IPCP is in the OPENED state and can get the peer address.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

2.3.3. Configure CHAP Uni-directional Authentication of Not Using User Name

Network Requirements

- Device1 and Device2 are connected via WAN interface; in the example, use the CE1 interface.
- Two interfaces encapsulate the PPP protocol.
- It is required that Device1 uses the CHAP mode to authenticate Device2 uni-directionally.
- Authenticate the peer via the CHAP mode uni-directionally when sending the empty user name.

Network Topology

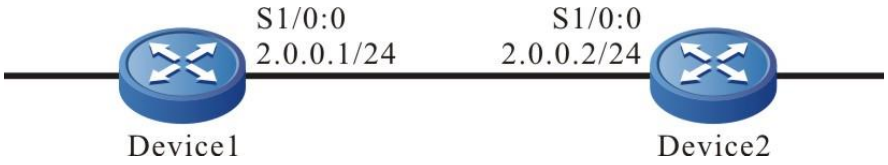

Figure 2-3 Networking of the CHAP uni-directional authentication not using user name

Configuration Steps

Step 1: Configure unframed in the controller of CE1 and configure the clock mode. (Omitted)

Step 2: Configure the encapsulation type and IP address of the interface. (Omitted)

Step 3: Configure Device1 as the CHAP authenticating end.

#Configure Device1.

#Configure Device1 as the CHAP authenticating end.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp authentication chap
Device1(config-if-serial1/0:0)#exit
```

#Configure the password of the CHAP authenticating end.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp chap password admin
Device1(config-if-serial1/0:0)#exit
```

#Configure sending empty user name during the CHAP authentication.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#no ppp chap send-hostname
Device1(config-if-serial1/0:0)#exit
```

Step 4: Configure Device2 as the CHAP authenticated end.

#Configure Device2.

#Configure the password of the CHAP authenticated party.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ppp chap password admin
Device2(config-if-serial1/0:0)#exit
```

#Configure sending empty user during CHAP authenticated.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#no ppp chap send-hostname
Device2(config-if-serial1/0:0)#exit
```

Step 5: Check the result.

#View the PPP information on Device1.

```
Device1#show  ppp  information serial 1/0:0
serial1/0:0
    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                0x0
        local magic number       0x3ec37fb4
        protocol field compression    OFF
        addr/ctrl field compression   OFF
        lcp echo interval        10
        lcp predictive           ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address      ON
        set peer address         ON
        local IP address         2.0.0.1
        remote IP address        2.0.0.2
        peer IP address          0.0.0.0
        vj compression protocol      OFF
        vj compression passive       OFF
        RTP compression protocol     OFF
        RTP compression passive      OFF
    PAP Stats
        client PAP state         CLOSED
        server PAP state         CLOSED
    CHAP Stats
        client CHAP state        CLOSED
        server CHAP state        OPEN
    Ifindex 1384710688, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
    PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
    Phy up, no shutdown, Reneg
```

#View the PPP information on Device2.

```
Device2#show  ppp  information serial 1/0:0
serial1/0:0
```

```
LCP Stats
    LCP phase                NETWORK
    LCP state                OPENED
    mru                      1500
    mtu                      1500
    async map                0x0
    local magic number       0x790ea080
    protocol field compression   OFF
    addr/ctrl field compression  OFF
    lcp echo interval        10
    lcp predictive           ON
IPCP Stats
    IPCP state               OPENED
    accept zero address      ON
    set peer address         ON
    local IP address         2.0.0.2
    remote IP address        2.0.0.1
    peer IP address          0.0.0.0
    vj compression protocol      OFF
    vj compression passive       OFF
    RTP compression protocol     OFF
    RTP compression passive      OFF
PAP Stats
    client PAP state         CLOSED
    server PAP state         CLOSED
CHAP Stats
    client CHAP state        OPEN
    server CHAP state        CLOSED
Ifindex 1384219168, IfType 0, DnsVrf -1
L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
Phy up, no shutdown, Reneg
```

After CHAP authentication succeeded, the PPP protocol is up. Run the **show ppp information** command and we can see that LCP and CHAP are in the OPEN state. After the NCP negotiation is complete, IPCP is in the OPENED state and can get the peer address. Device1 and Device2 can ping each other.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2
```

```
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

2.3.4. Configure PAP Uni-directional Authentication

Network Requirements

- Device1 and Device2 are connected via WAN interface; in the example, use the CE1 interface.
- Two interfaces encapsulate the PPP protocol.
- It is required that Device1 adopts the PAP mode to authenticate Device2 uni-directionally.

Network Topology

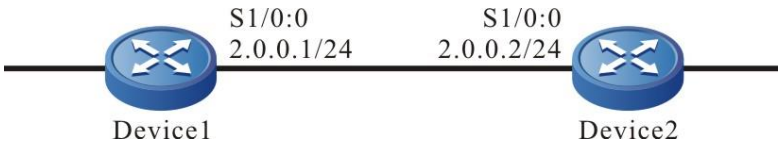

Figure 2-4 Networking of PPP PAP authentication

Configuration Steps

Step 1: Configure unframed in the controller of CE1 and configure the clock mode (Omitted)

Step 2: Configure the encapsulation type and IP address of the interface. (Omitted)

Step 3: Create the local user name and password for Device2.

Configure Device1.

#Create the local user name and password for the peer end.

```
Device1#configure terminal
Device1(config)#local-user admin class network
Device1(config-user-network-admin)#password 0 admin
Device1(config-user-network-admin)#service-type ppp
Device1(config-user-network-admin)#exit
```

Step 4: Configure the PAP authentication, configure Device1 as the PAP authenticating end and Device2 as the authenticated end.

#Configure Device1.

#Configure Device1 as the PAP authenticating end.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp authentication pap
```

```
Device1(config-if-serial1/0:0)#exit
```

#Configure Device2.

#Configure the user name and password sent by Device2 during the PAP authentication negotiation.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ppp pap sent-username admin password admin
Device2(config-if-serial1/0:0)#exit
```

Step 5: Check the result.

#View the PPP information on Device1.

```
Device1#show ppp information serial 1/0:0
Serial1/0:0

    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                0x0
        local magic number       0x2fb89dda
        protocol field compression   OFF
        addr/ctrl field compression  OFF
        lcp echo interval        10
        lcp predictive           ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address      ON
        set peer address         ON
        local IP address         2.0.0.1
        remote IP address        2.0.0.2
        peer IP address          0.0.0.0
        vj compression protocol  OFF
        vj compression passive   OFF
        RTP compression protocol OFF
```

```
        RTP compression passive       OFF
    PAP Stats
        client PAP state            CLOSED
        server PAP state             OPEN
    CHAP Stats
        client CHAP state           CLOSED
        server CHAP state            CLOSED
    Ifindex 1384710688, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
    PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
    Phy up, no shutdown, Reneg
#View the PPP information on Device2.
Device2#show  ppp  information  serial 1/0:0
Serial1/0:0


    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                  0x0
        local magic number           0x37c081ce
        protocol field compression     OFF
        addr/ctrl field compression    OFF
        lcp echo interval          10
        lcp predictive             ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address          ON
        set peer address           ON
        local IP address           2.0.0.2
        remote IP address            2.0.0.1
        peer IP address            0.0.0.0
        vj compression protocol       OFF
        vj compression passive        OFF
        RTP compression protocol      OFF
```

```
        RTP compression passive       OFF
    PAP Stats
        client PAP state          OPEN
        server PAP state           CLOSED
    CHAP Stats
        client CHAP state          CLOSED
        server CHAP state           CLOSED
    Ifindex 1384219168, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
    PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
    Phy up, no shutdown, Reneg
```

After PAP authentication succeeded, the PPP protocol is up. Run the **show ppp information** command and we can see that LCP is in the opened state. If it is PAP authenticating end, server pap state is OPEN; if it is the PAP authenticated end, client pap state is OPEN. After the NCP negotiation is complete, the IPCP is in the OPEN state and can get the peer address.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

2.3.5. Configure PPP Compression Instance

Network Requirement

- Device1 and Device2 are connected via WAN interface; in the example, use the CE1 interface.
- Two interfaces encapsulate the PPP protocol.
- Perform the PC and AC compression for the data packet transmitted on the interface to reduce the consumption of the physical bandwidth.

Network Topology

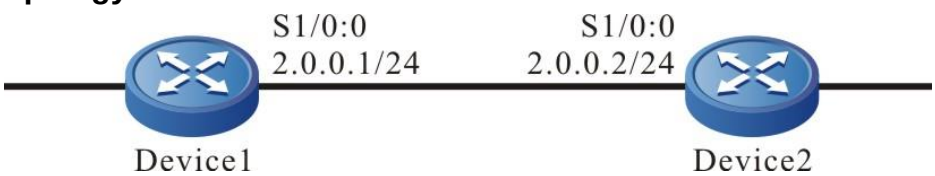

Figure 2-5 Networking of the PPP compression protocol

Configuration Steps

Step 1: Configure unframed in the controller of CE1 and configure the clock mode (Omitted)

Step 2: Configure the encapsulation type and IP address of the device interface. (Omitted)

Step 3: Configure the PC and AC compression protocol.

#Configure Device1.

#Configure Protocol-Field-Compression (PC) compression on the interface.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp pc
Device1(config-if-serial1/0:0)#exit
```

#Configure Address-and-Control-Field-Compression (AC) compression on the interface.

```
Device1#configure terminal
Device1(config)#int serial1/0:0
Device1(config-if-serial1/0:0)#ppp ac
Device1(config-if-serial1/0:0)#exit
```

#Configure Device2.

#Configure Protocol-Field-Compression (PC) compression on the interface.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ppp pc
Device2(config-if-serial1/0:0)#exit
```

#Configure Address-and-Control-Field-Compression (AC) compression on the interface.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ppp ac
Device2(config-if-serial1/0:0)#exit
```

Step 4: Let PPP re-negotiate manually.

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#shutdown
Device2(config-if-serial1/0:0)#no shutdown
Device2(config-if-serial1/0:0)#exit
```

After configuring the PC and AC compression of PPP, it can take effect only after re-negotiating. PC, AC compression can be used separately. For two direct-connected interfaces, we can configure compression at one end and do not configure compression at the other end.

Step 5: Check the result.

#View the PPP information on Device1 and the ping can be connected.

```
Device1#show  ppp information serial1/0:0
Serial1/0:0


    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                0x0
        local magic number       0x5ccafde1
        protocol field compression    ON
        addr/ctrl field compression   ON
        lcp echo interval        10
        lcp predictive           ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address      ON
        set peer address         ON
        local IP address         2.0.0.1
        remote IP address        2.0.0.2
        peer IP address          0.0.0.0
        vj compression protocol      OFF
        vj compression passive       OFF
        RTP compression protocol     OFF
        RTP compression passive      OFF
    PAP Stats
        client PAP state         CLOSED
        server PAP state         CLOSED
    CHAP Stats
        client CHAP state        CLOSED
        server CHAP state        CLOSED
```

```
Ifindex 1384710688, IfType 0, DnsVrf -1
L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
Phy up, no shutdown, Reneg
```

After the PPP compression protocol takes effect, run the show ppp information command and we can see that the status of the corresponding compression protocol is ON.

Warning:

- If the protocol is UP and after configuring the PPP compression, it can take effect only after PPP re-negotiates.

2.3.6. Configure PPP AAA Authentication

Network Requirements

- Device1 and Device2 are connected with the WAN interface; the interface encapsulates the PPP protocol;
- The route between Device1 and the AAA server is reachable;
- Device1 performs the PPP authentication via the AAA server and is authorized by the server to distribute the IP address for the peer;
- Define one user on the AAA server; the user name is admin and the password is admin;
- Create one key admin on the AAA server;
- Create one address pool with address range 2.0.0.2/24-2.0.0.32/24 on the AAA server.

Network Topology

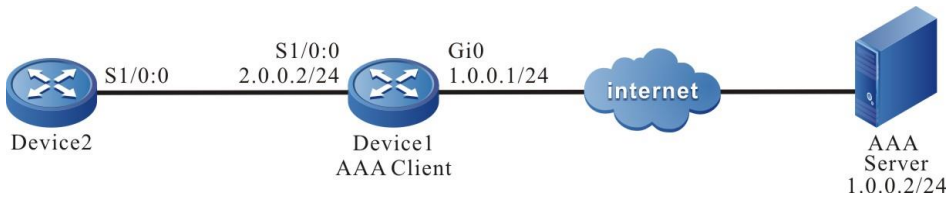

Figure 2-6 Networking of PPP using AAA authentication

Configuration Steps

Step 1: Configure unframed in the controller of CE1 and configure the clock mode (Omitted)

Step 2: Configure the AAA server. (Omitted)

Step 3: Configure the encapsulation type and IP address of Device1. (Omitted)

Step 4: Enable the AAA RADIUS server function on Device1.

#Configure Device1.

#Configure RADIUS server.

```
Device1#configure terminal
Device1(config)#aaa server group radius rg
```

#Configure the AAA RADIUS server, authentication/authorizing port as 1812, accounting port as 1813, and shared key as admin.

```
Device1(config-sg-radius-rg)#server 1.0.0.2 auth-port 1812 acct-port 1813 key 0 admin
Device1(config-sg-radius-rg)#exit
```

#Configure AAA isp-name.

```
Device1(config)# domain test
```

#On isp-name, configure RADIUS authentication authorizing and accounting.

```
Device1(config-isp-test)#aaa authentication ppp radius-group rg
Device1(config-isp-test)#aaa authorization ppp radius-group rg
Device1(config-isp-test)#aaa accounting ppp start-stop radius-group rg
Device1(config-isp-test)#exit
```

Step 5: Enable the AAA CHAP authentication on the interface.

#Configure Device1.

#Configure Device1 to enable the AAA CHAP authentication.

```
Device1#configure terminal
Device1(config)#interface serial1/0:0
Device1(config-if-serial1/0:0)#ppp authentication chap test
Device1(config-if-serial1/0:0)#exit
```

#Configure Device2.

#Configure the user name and password used by Device2 for the AAA CHAP authentication.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)# ppp chap password 0 admin
Device2(config-if-serial1/0:0)# ppp chap hostname admin
Device2(config-if-serial1/0:0)# exit
```

Step6: Configure the auto negotiation of Device2 address.

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial1/0:0
Device2(config-if-serial1/0:0)#ip address negotiated
Device2(config-if-serial1/0:0)#exit
```

The IP address of Device2 gets one random address from the address pool of the AAA server via the auto negotiation.

Step 7: Check the result.

#View the PPP information on Device1.

```
Device1#show  ppp  information serial 1/0:0
Serial1/0:0


    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                0x0
        local magic number       0x773daa3d
        protocol field compression    OFF
        addr/ctrl field compression   OFF
        lcp echo interval        10
        lcp predictive           ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address      ON
        set peer address         ON
        local IP address         2.0.0.10
        remote IP address        2.0.0.2
        peer IP address          0.0.0.0
        vj compression protocol      OFF
        vj compression passive       OFF
        RTP compression protocol     OFF
        RTP compression passive      OFF
    PAP Stats
        client PAP state         CLOSED
        server PAP state         CLOSED
    CHAP Stats
        client CHAP state        CLOSED
        server CHAP state        OPEN
    Ifindex 1384710688, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
```

PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0

Phy up, no shutdown, Reneg

#View the PPP information on Device2.

```
Device2#show  ppp  information serial 1/0:0
Serial1/0:0


    LCP Stats
        LCP phase                NETWORK
        LCP state                OPENED
        mru                      1500
        mtu                      1500
        async map                0x0
        local magic number       0x7bfc782e
        protocol field compression    OFF
        addr/ctrl field compression   OFF
        lcp echo interval        10
        lcp predictive           ON
    IPCP Stats
        IPCP state               OPENED
        accept zero address      ON
        set peer address         ON
        local IP address         2.0.0.2
        remote IP address        2.0.0.10
        peer IP address          0.0.0.0
        vj compression protocol      OFF
        vj compression passive       OFF
        RTP compression protocol     OFF
        RTP compression passive      OFF
    PAP Stats
        client PAP state         CLOSED
        server PAP state         CLOSED
    CHAP Stats
        client CHAP state        OPEN
        server CHAP state        CLOSED
    Ifindex 1384219168, IfType 0, DnsVrf -1
    L2TP info: ses Manded No, recv ifindex 0x0, peer authen name
```

```
PPPoE info: ses ID 0, ses Inx 0, recv ifindex 0
Phy up, no shutdown, Reneg
```

After CHAP authentication succeeded, the PPP protocol is UP; run the show ppp information command and we can see that the LCP and CHAP are in the OPEN state; after the NCP negotiation is complete, IPCP is in the OPENED state and can get the peer address; Device1 and Device2 can ping each other.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

Note:

- AAA authentication and authorization type can adopt radius and tacacs.
- The IP address authorized to Device2 is one random address, but should be in the distributing range of the AAA server address pool.

2.3.7. Configure MPPP Binding

Network Requirements

- Bind two pairs of interconnected WAN interfaces via MPPP to increase the physical bandwidth.
- Device1 and Deviced2 are interconnected via two WAN interfaces serial1/0: 0 and serial1/1: 0; adopt the multilink-group binding mode.
- Two pairs of interconnected interfaces all encapsulate the PPP protocol.

Network Topology

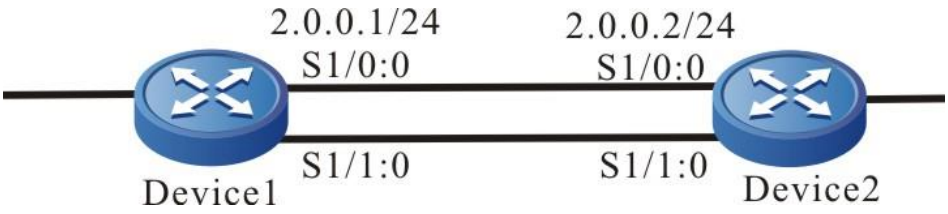

Figure 2-7 MPPP networking

Configuration Steps

Step 1: Configure unframed in the controller of CE1 and configure the clock mode (Omitted)

Step 2: Create the MPPP interface multilink 1.

#Configure Device1.

#Create one MPPP interface multilink 1 on Device1.

```
Device1#configure terminal
Device1(config)#interface multilink 1
Device1(config-if-multilink1)#exit
```

#Create one MPPP interface multilink 1 on Device2.

```
Device2#configure terminal
Device2(config)#interface multilink 1
Device2(config-if-multilink1)#exit
```

Step 3: Configure the IP address of multilink 1.

#Configure Device1.

#Configure the IP address of multilink 1 on Device1.

```
Device1#configure terminal
Device1(config)#interface multilink 1
Device1(config-if-multilink1)#ip address 2.0.0.1 255.255.255.0
Device1(config-if-multilink1)#exit
```

#Configure Device2.

#Configure the IP address of multilink 1 on Device2.

```
Device2#configure terminal
Device2(config)#interface multilink 1
Device2(config-if-multilink1)#ip address 2.0.0.2 255.255.255.0
Device2(config-if-multilink1)#exit
```

Step 4: Add the physical interface to the multilink-group1 interface group.

#Configure Device1.

#Add physical interface serial1/0: 0 to the multilink-group 1 interface group.

```
Device1#configure terminal
Device1(config)#int serial1/0:0
Device1(config-if-serial1/0:0)#multilink-group 1
Device1(config-if-serial1/0:0)#exit
```

#Add physical interface serial1/1: 0 to the multilink-group 1 interface group.

```
Device1#configure terminal
Device1(config)#int serial1/1:0
Device1(config-if-serial1/1:0)#multilink-group 1
Device1(config-if-serial1/1:0)#exit
```

#Configure Device2.

#Add physical interface serial1/0: 0 to the multilink-group 1 interface group.

```
Device2#configure terminal
Device2(config)#int serial1/0:0
Device2(config-if-serial1/0:0)#multilink-group 1
Device2(config-if-serial1/0:0)#exit
```

#Add physical interface serial1/1: 0 to the multilink-group 1 interface group.

```
Device2#configure terminal
Device2(config)#int serial1/1:0
Device2(config-if-serial1/1:0)#multilink-group 1
Device2(config-if-serial1/1:0)#exit
```

Step 5: Check the result.

#View whether the multilink 1 interface protocol is UP on Device1.

```
Device1# show interface multilink 1
multilink1:
    line protocol is up
    Flags: (0xc0080f1) POINT-TO-POINT MULTICAST RUNNING
    Type: MULTILINK
    Internet address: 2.0.0.1/24
    Destination Internet address: 2.0.0.2
    Metric: 0, MTU: 1500, BW: 4096 Kbps, DLY: 100000 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Last clearing of "show interface" counters is 0 hour 0 minute 2 seconds ago
    input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    3828 packets received; 3908 packets sent
    5288792 bytes received; 5429358 bytes sent
    1 multicast packets received
    3 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped

  multilink1 have 2 members
        serial1/0:0       Active
        serial1/1:0       Active
  Total Bandwidth Of Active Physical Interface: 4096 Kbps
```

After the mppp interface is created successfully and negotiated to up, execute the **show interface multilink 1** command and we can get multilink 1 interface and member interface information..

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

3. FRAME RELAY

3.1. Overview

3.1.1. Work Principle of Frame Relay

The rate of the early transmission network is low and the transmission error rate is high. To ensure the reliability, the packet switching mode like X.25 adopts the complicated error control and traffic control. The modern communication network mainly adopts the digital transmission technology and the features are high-speed and reliable; meanwhile, the intelligence of the user terminal is improved greatly. In the case, the X.25 processing mode is not only unnecessary, but also reduces the using efficiency of the high-speed digital transmission line. Under the background, the frame relay technology emerges.

For the general packet switching network, the data link layer has the complete error control, but for the frame relay network, the nodes of the network does not have the network layer and the data link layer only has the limited error control function. The communication host is responsible for the complete error control function. Besides, the nodes of the frame relay all only send data frames, but do not send the confirm frames. The communication host is responsible for the confirming function. The data link layer of the frame relay also does not have the traffic control capability and the traffic control is completed by the higher layer. Because of these reasons, the time of the frame relay processing frames is short, reducing much compared with the X.25 network. It improves the network throughput greatly.

The network service model of the frame relay is as follows:

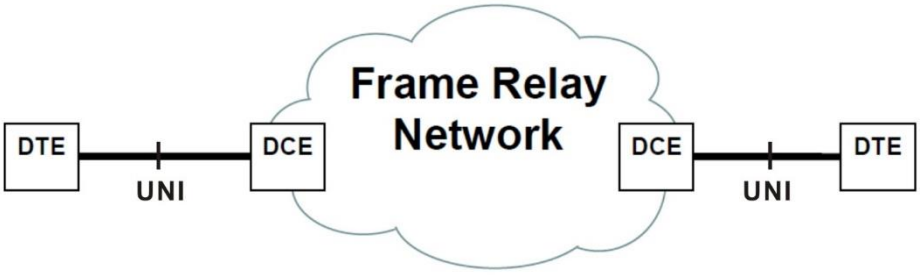

Figure 3–1 Network service model of the frame relay

The network service of the frame relay comprises the user access circuit of the frame relay (UNI in the figure) and frame relay network (Frame Relay Network in the figure). The user access circuit of the frame relay is also called user network interface (User-to-Network Interface). It is responsible for accessing the user to the frame relay network. Frame relay network comprises a series of interconnected frame relay switches. The frame relay switch is responsible for transmitting the user data to the corresponding destination. UNI has two interfaces and the frame relay device connected to the interface at the user side is called frame relay DTE (Data Terminal Equipment). The frame relay DTE is connected to the user and it connects the user to the frame relay network; the frame relay device connected to the interface at the network side is called frame relay DCE (Data Circuit-terminating Equipment) and the other side of the frame relay DCE connects to the frame relay switch.

The frame relay network provides the connection-oriented virtual circuit service. The virtual circuit includes PVC (Permanent Virtual Circuit) and SVC (Switched Virtual Circuit). PVC is set manually and SVC is distributed by the protocol automatically. Currently, we mainly use PVC and SVC is nearly not used. There is one or multiple virtual circuits in one UNI. From the aspect of the user, one PVC is one virtual channel or logical connection connected to two users. Each virtual circuit is bi-directional and each direction has one specified CIR (Committed Information Rate). To

distinguish different PVCs, two endpoints of each PVC have one DLCI (Data Link Connection Identifier).

The main advantages of the frame relay are as follows:

- Reduce the network interconnection cost. With the virtual circuit, multiple logical connections are multiplexed to one physical connection, which can reduce the access cost.
- Increase the interoperability. Because of using the international standards and not hard to realize the simplified link protocol, the manufacturers all can realize easily.
- The network complexity is reduced, but the performance is improved. Compared with X.25, The time of the frame relay processing frames is reduced greatly and can make use of the high-speed digital transmission line more efficiently, improving the network performance and response time obviously.
- Protocol independence: The frame relay can carry various network protocols easily (such as IP, IPv6 and MPLS) and can serve as the public backbone network.

According to the features of the frame relay, the frame relay can be applicable to the transmission of large files (such as the high-resolution image), the multiplexing of multiple low-speed lines, and the interconnection of the LANs.

3.1.2. Frame Format of Frame Relay

The frame format of the frame relay is as follows:

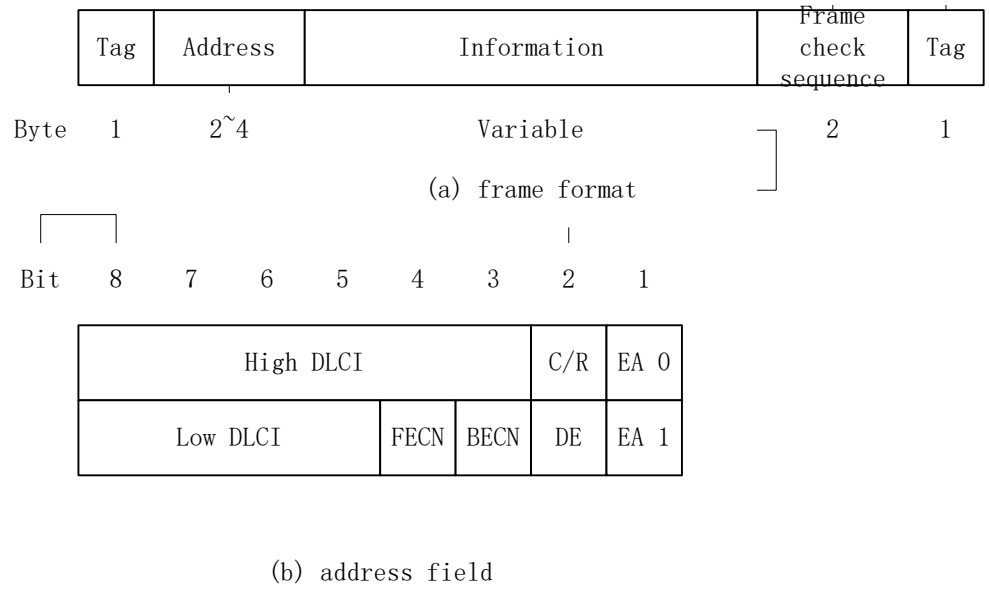

Figure 3–2 The frame format of the frame relay

The meanings of the fields are as follows:

- Tag field: It is 0x7E, indicating the start and end of one frame;
- Information field: It is the data with variable length;
- Frame check sequence field: It contains two-byte cyclic redundant check;
- Address field: Usually, it is two-byte (it also can be expended to 3 or 4 bytes; currently, it is not used any more);

The address field comprises the following several parts:

- DLCI (Data Link Connection Identifier): Used to identify the virtual circuit, call control or management information. DLCI only has the local meaning, that is to say, in one frame relay virtual connection and on the UNIs connected to the two ends, the used two DLCIs are different and also can be the same;
- Command/response C/R: It is related with the high-layer application. The frame relay itself does not use.
- Extended address (EA): The address field can be extended to 3 or 4 bytes. When EA is 0, it indicates that the next byte is also the address field; when EA is 1, it indicates that the address field ends here.
- FECN (Forward Explicit Congestion Notification): If it is 1, it indicates that the frame that is transmitted at the same direction as the frame may generate delay because of the network congestion.
- BECN (Backward Explicit Congestion Notification): If it is 1, it indicates that the frame that is transmitted at the reverse direction may generate delay because of the network congestion.
- DE (Discard Eligibility): If it is 1, it indicates that when there is congestion in the network, to maintain the network service level, the frame should be dropped first, compared with the frame with DE 0.

3.1.3. Congestion Control of Frame Relay

There are three kinds of congestion control methods used by the frame relay:

- Drop policy: When the congestion is serious enough, the network drops the frame;
- Congestion avoidance: When there is light congestion, we can adopt the congestion avoidance;
- Congestion recovery: When there is already congestion, the congestion recovery process can prevent the complete breakdown of the network.

To perform the congestion control, the frame relay adopts one concept, that is, CIR (Committed Information Rate) and the unit is bit/s. CIR is the information transmission rate agreed to support for one specified frame relay connection network. For PVC, the CIR of each connection should be confirmed when the connection is set up; for SVC, CIR should be negotiated when the call is set up. When congestion happens, which frame to be dropped depends on the DE field of the frame. If the data rate is smaller than CIR, all frames transmitted on the connection are set as DE=0, indicating that when the congestion happens to the network, try not to drop the frame with DE=0. Usually, the transmission can be ensured here. If the data rate is larger than CIR only within a short interval, the network can set the frame as DE=1 and transmit in the possible case, that is, not be sure to drop, depending on the congestion degree of the network. If the data rate exceeds CIR for a long time and as a result, the data quantity entering the network exceeds the set threshold of the network, drop the frames transmitted on the connection at once.

Frame relay also can adopt the explicit signaling avoidance congestion, that is, FECN and BECN. If the frames generate the congestion when passing the frame relay network, the frame relay switch sets its FECN bit as 1. After the frame reaches the destination, the receiver of the frame can see that the frame experiences the congestion when passing the frame relay network. And then, the information is transmitted to the high-layer protocol, which decides the traffic control policy. Similarly, if the frame generates the congestion when passing the frame relay network, the frame relay switch finds one frame (its destination address is the sender of the congestion frame), and then sets the BECN bit of the frame as 1. After the sender receives the frame, it knows that there is congestion at its sending direction. And then, the information is transmitted to the high-layer protocol, which decides the traffic control policy.

3.1.4. LMI Protocol of Frame Relay

LMI (Local Management Interface) is used to manage the PVC status. Currently, there are three kinds, that is, ANSI standard T1.617 appendix D, ITU-T standard Q.933 appendix A and the standards made by Cisco and another several companies. Here, we mainly describes ITU-T standard Q.933 appendix A and the other two standard LMIs are similar.

LMI protocol procedure includes:

- Add PVC notification
- Delete PVC detection
- Configured PVC status notification, that is, available (activated) or unavailable (inactivated):
  - Inactivated indicates that the PVC is configured, but is not available;
  - Activated indicates that the PVC is available.
- Validate the link integrality.

The above protocol procedure is realized by transmitting the messages of the LMI protocol. The message is transmitted on the virtual circuit with DLCI=0. There are two kinds of messages types, that is, status message and status request message. The status request message is transmitted by DTE, used to request the PVC status from DCE or validate the link integrality; the status message is response of DCE for the status request message, used to report the PVC status to DTE or validate the link integrality. The main process is as follows:

- DTE sends the status request message to DCE every T391 time. The message includes two kinds, that is, link integrality validating message and full-status request message. The link integrality validating message is to validate the integrality of the link between DTE and DCE. Besides validating the link integrality, the full-status request packet requests all PVC status. Every time sending N391 link integrality validating messages, send one full-status request message. The process of DTE regularly sending the status request message also can be called polling or link keepalive detection.
- After receiving the status request message, DCE answers by the status message. When the PVC status in the network changes or there is added/deleted PVC in the network, no matter whether DTE is the full-status request message, DCE should send the full-status message to DTE so that DTE can get to know the PVC change in time. The process of the DEC responding is also called polling validating.
- After receiving the status message, DTE analyzes to get to know the link integrality and PVC status and update the previous records.
- If DTE does not receive the status message within the T381 time or the received status message is wrong, it is regarded that one error event happens, record the error, and add the error times by 1. If error times of DTE exceeds N392 during N393 events, DTE regards that the physical channel is unavailable and all virtual circuits are unavailable. N393 is the monitor event counter and N392 is the error threshold.
- If DCE does not receive the status request message within the T392 time or the received status message is wrong, it is regarded that one error event happens, record the error, and add the error times by 1. If error times of DCE exceeds N392 during N393 events, DCE regards that the physical channel is unavailable and all virtual circuits are unavailable. N393 is the monitor event counter and N392 is the error threshold.
- When DTE or DCE detects that there is no error for N392 successive events, it is regarded that the physical channel recovers.

3.1.5. Address Mapping of Frame Relay

Frame relay virtual circuit is connection-oriented and the local different DLCIs are connected to different peer devices. The address mapping of the frame relay set sup the relation of the local DLCI and the peer user so that the local user service data can be transmitted to the peer via the corresponding PVC and the peer user is identified by the protocol address of the device. Therefore, the address mapping is the mapping of the local DLCI and the protocol address of the peer device.

Address mapping can be configured statically and also can be set up dynamically. The static configuration is to associate the local DLCI with the protocol address of the peer device manually. It is used when the network is simple and there are a few peer devices. Dynamic setup is to adopt the InARP (Inverse Address Resolution Protocol) to associate the local DLCI with the protocol address of the peer device. It is used when the peer device supports InARP and the network is complicated.

The InARP process is: Every time discovering one available PVC (the premise is that the local interface is configured with the protocol address), InARP sends the InARP request packet to the peer on the PVC and the request packet contains the local protocol address. After receiving the request, the peer device gets the local protocol address, generates the address mapping, and sends the InARP response packet to answer. In this way, the address mapping also can be generated at the local.

3.1.6. Multi-link Frame Relay

Multi-link frame relay aggregates one or multiple physical links to simulate one physical link to provide the frame relay service. Compared with the traditional frame relay, multi-link frame relay can improve the line bandwidth and also can increase the reliability. The simulated physical link for providing the frame relay service is called binding or binding interface, while the aggregated actual physical link is called bound link or bound link interface.

The binding mainly completes the following functions:

- Add the bound link to the binding;
- Remove the bound link from the binding;
- Receive the frame from the frame relay data link layer and transmit on the binding interface;
- Fragment the frame;
- Execute the scheduling policy and distribute the frame to the appropriate bound link for transmission;
- Re-assembly the fragmented frame received from the bound link, and transmit to the frame relay data link layer.

Besides providing the sending and receiving of the data frame, the bound link has the most important task of running the link integrality protocol of the multi-link frame relay. One binding interface of the multi-link frame relay protocol can contain multiple bound links. Therefore, the local device and peer device both should have the operations of adding the bound link to the binding interface, deleting the bound link from the binding interface, and maintaining the normal communication of multiple bound link connections. With the above three operations, we can set up the link connection of the multi-link frame relay and also can delete the setup link connection, so as to ensure that the multi-link frame relay between the local device and the peer device can transmit data normally. This is the work of the link integrality protocol. The link integrality protocol runs by exchanging messages between the bound link and the peer. The three basic operation processes are as follows:

- The process of setting up the bound link: When the local end adds one bound link to the binding, send one ADD_LINK message to the peer. If the peer agrees to set up the link after receiving the message, send the ADD_LINK_ACK message to answer. If not agreeing, send the ADD_LINK_REJ message to answer. If the two parties both send ADD_LINK message and receive the ADD_LINK_ACK message from the peer, the bound link is set up.
- The process of removing the bound link: When the local end removes one bound link from the binding, send one REMOVE_LINK message to the peer. After receiving the message, the peer informs the binding and sends the REMOVE_LINK_ACK message to answer. In this way, the bound link is removed.
- The process of maintaining the link: After the bound link is set up, the two parties periodically send the HELLO message to the peer. After receiving the message, send the HELLO_ACK message to answer as soon as possible. If receiving the HELLO_ACK message of the peer within the defined timeout, it is regarded that the link is available; if not receiving the HELLO_ACK message of the peer within the defined timeout, re-send the HELLO message. When the re-sending times reaches the threshold, it is regarded that the bound link is unavailable. Therefore, inform the binding to remove the bound link.

3.2. Frame Relay Function Configuration

Table 3-1 Frame relay function configuration list

Configuration Task	
Configure the basic functions of the frame relay	Encapsulate the frame relay protocol
	Configure the interface type of the frame relay
	Configure the LMI protocol type of the frame relay
Configure the frame relay DLCI	Configure the frame relay DLCI
Configure the frame relay address mapping	Configure the static address mapping of the frame relay
	Configure the dynamic address mapping of the frame relay
Configure the multi-link frame relay	Configure the basic functions of the multi-link frame relay

3.2.1. Configure Basic Functions of Frame Relay

To configure the basic functions of the frame relay, we need to complete the following three tasks:

- Encapsulate the frame relay protocol;
- Configure the interface type of the frame relay;

- Configure the LMI protocol type of the frame relay.

Configuration Condition

None

Encapsulate Frame Relay Protocol

Table 3-2 Encapsulate the frame relay protocol

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Encapsulate the frame relay protocol	**encapsulation frame-relay** [**cisco**]	Mandatory By default, the multi-link frame relay interface encapsulates the frame relay protocol.

Configure Interface Type of Frame Relay

After the interface is encapsulated with the frame relay protocol, it is necessary to configure the interface type of the frame relay. The interface type of the frame relay depends on the location of the frame relay device. If the device is at the user end, the interface type need to be configured as DTE; if the device is at the network side, the interface type needs to be configured as DCE. To configure the interface type of the frame relay as DCE, it is necessary to execute the **frame-relay switching** command in the global configuration mode to enable the frame relay switching function first.

Configuring the interface type of the frame relay as DTE is as follows:

Table 3-3 Configure the interface type of the frame relay as DTE

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the interface type of the frame relay as DTE	**frame-relay intf-type dte**	Optional By default, the interface type of the frame relay is DTE.

Configuring the interface type of the frame relay as DCE is as follows:

Table 3-4 Configure the interface type of the frame relay as DCE

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enable the frame relay switching function	**frame-relay switching**	Mandatory By default, disable the frame relay switching function.
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the interface type of the frame relay as DCE	**frame-relay intf-type dce**	Mandatory By default, the interface type of the frame relay is DTE.

Configure LMI Protocol Type of Frame Relay

The LMI protocol is used to manage the PVC status. When configuring the LMI protocol type, ensure that it is consistent with the LMI protocol type of the peer frame relay device. Otherwise, the LMI protocols of the two sides will fail and as a result, PVC is unavailable and cannot provide services for the user.

Table 3-5 Configure the LMI protocol type of the frame relay

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the LMI protocol type of the frame relay	**frame-relay lmi-type** { **ansi** | **lmi** | **q933a** }	Mandatory By default, the LMI protocol type of the frame relay is **ansi**

3.2.2. Configure Frame Relay DLCI

The frame relay DLCI is distributed by the service provider of the frame relay. Therefore, before configuration, it is necessary to consult the service provider, so as to ensure that the configured DLCI is consistent with the one distributed by the service provider. Otherwise, the corresponding PVC is unavailable and cannot provide services for the user.

Configuration Condition

Before configuring the frame relay DLCI, first complete the following task:

- The interface encapsulates the frame relay protocol.

Configure Frame Relay DLCI

Table 3-6 Configure the frame relay DLCI

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the frame relay DLCI	**frame-relay interface-dlci** *dlci-number*	Mandatory By default, the frame relay interface does not configure DLCI.

3.2.3. Configure Address Mapping of Frame Relay

The address mapping of the frame relay is the mapping of the local DLCI and the protocol address of the peer device, including static address mapping and dynamic address mapping. If the protocol address of the peer device is known, we can configure the static address mapping; if the protocol address of the peer device is unknown, we can get the protocol address of the peer device via the InARP function to set up the dynamic address mapping.

Configuration Condition

Before configuring the address mapping of the frame relay, first complete the following task:

- The interface encapsulates the frame relay protocol.

Configure Static Address Mapping of Frame Relay

Configure the static address mapping of the frame relay as follows:

Table 3-7 Configure the static address mapping of the frame relay

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the IPv4 static address mapping of the frame relay	**frame-relay map ip** *ip-address dlci-number* [[**cisco | ietf**] / [**broadcast**] / [**compress** [**passive**] | **nocompress** | **rtp header-compress** [**passive**] | **tcp header-compress** [**passive**]]]	Mandatory By default, do not configure the static address mapping of the frame relay.

Note:

- If the specified DLCI does not exist, create DLCI automatically.
- For the point-to-point sub interface of the frame relay, there is no address mapping, so we cannot configure the static address mapping at the point-to-point sub interface.
- When configuring the routing protocol, such as OCPF, on the frame relay interface,, use the **broadcast** parameter so that the broadcast or multicast packets of the routing protocol can be sent to the peer.

Configure Dynamic Address Mapping of Frame Relay

Table 3-8 Configure the static address mapping of the frame relay

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the static address mapping of the frame relay	**frame-relay inverse-arp** [**interval** *interval-time* | **update**]	Optional By default, the interface enables the InARP function.

3.2.4. Monitoring and Maintaining of Frame Relay

Table 3-9 Monitoring and maintaining of the frame relay

Command	Description
clear frame-relay multilink interface *interface-name*	Clear the statistics information of the specified multi-link frame relay interface
show frame-relay inarp [**interface** *interface-name*]	Display the statistics information of the frame relay InARP
show frame-relay lmi [**interface** *interface-name*]	Display the information of the frame relay LMI protocol
show frame-relay map	Display the information of the frame relay address mapping
show frame-relay multilink [*interface-name* [**detailed**] | **detailed**]	Display the information of the multi-link frame relay
show frame-relay pvc [*dlci-number* | **interface** *interface-name*]	Display the information of the frame relay PVC

3.3. Typical Configuration Example of Frame Relay

3.3.1. Configure Private Line Interconnection of Frame Relay Devices

Network Requirements

- Device1 and Devices are connected via the WAN interface. It is required that the interface encapsulation type is frame relay. Device1 can communicate with Device2.
- Device1 serves as the frame relay DCE and Device2 serves as the frame relay DTE.

Network Topology

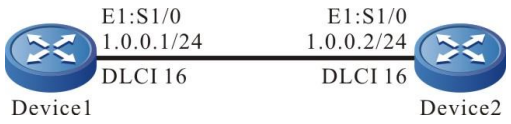

Figure 3–3 Networking of configuring private line interconnection of the frame relay devices

Configuration Steps

Step 1: Configure the clock mode of the E1 interface. Device1 configures the internal clock and Device2 configures the external clock (the external clock is the default clock mode and does not need to be configured manually).

#Configure Device1.

```
Device1#configure terminal
Device1(config)#interface serial 1/0
```

```
Device1(config-if-serial1/0)#clock source internal
Device1(config-if-serial1/0)#exit
```

Step 2: Configure the encapsulation type of the interface as the frame relay and configure the IP address.

#Configure Device1 as the frame relay DCE.

```
Device1(config)#frame-relay switching
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#encapsulation frame-relay
Device1(config-if-serial1/0)#frame-relay intf-type dce
Device1(config-if-serial1/0)#ip address 1.0.0.1 255.255.255.0
```

Configure Device2 as the frame relay DTE.

```
Device2#configure terminal
Device2(config)#interface serial 1/0
Device2(config-if-serial1/0)#encapsulation frame-relay
Device2(config-if-serial1/0)#ip address 1.0.0.2 255.255.255.0
```

Step 3: Configure DLCI and configure the static address mapping.

#Configure the DLCI number of Device1 as 16.

```
Device1(config-if-serial1/0)#frame-relay interface-dlci 16
Device1(config-fr-dlci)#exit
```

Configure the static address mapping.

```
Device1(config-if-serial1/0)#frame-relay map ip 1.0.0.2 16
Device1(config-if-serial1/0)#exit
```

Configure the DLCI number of Device2 as 16.

```
Device2(config-if-serial1/0)#frame-relay interface-dlci 16
Device2(config-fr-dlci)#exit
```

Configure the static address mapping.

```
Device2(config-if-serial1/0)#frame-relay map ip 1.0.0.1 16
Device2(config-if-serial1/0)#exit
```

Step 4: Check the result.

#View the frame relay mapping on Device1.

```
Device1#show frame-relay map
serial1/0 (up): ip 1.0.0.2, dlci 16, static,
```

```
IETF, status ACTIVE
```

We can see that serial1/0 can be mapped to the peer address and the status is ACTIVE.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 1.0.0.2

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
```

Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.

Note:

- To send the multicast and broadcast packets on the frame relay network, add the **broadcast** parameter when configuring the static mapping.

3.3.2. Configure Network Interconnection of Frame Relay Devices

Network Requirement

- Device1 and Device2 are connected to the frame relay network; Device1 can communicate with Device2.

Network Topology

Figure 3–4 Networking of configuring the network interconnection of the frame relay devices

Configuration Steps

Step 1: Configure the encapsulation type of the interface as the frame relay and configure the IP address.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#encapsulation frame-relay
Device1(config-if-serial1/0)#ip address 1.0.0.1 255.255.255.0
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial 1/0
Device2(config-if-serial1/0)#encapsulation frame-relay
Device2(config-if-serial1/0)#ip address 1.0.0.2 255.255.255.0
```

Step 2: Configure DLCI and configure the static address mapping.

Configure the DLCI number of Device1 as 20.

```
Device1(config-if-serial1/0)#frame-relay interface-dlci 20
Device1(config-fr-dlci)#exit
```

Configure the static address mapping.

```
Device1(config-if-serial1/0)#frame-relay map ip 1.0.0.2 20
Device1(config-if-serial1/0)#exit
```

Configure the DLCI number of Device2 as 30.

```
Device2(config-if-serial1/0)#frame-relay interface-dlci 30
Device2(config-fr-dlci)#exit
```

Configure the static address mapping.

```
Device2(config-if-serial1/0)#frame-relay map ip 1.0.0.1 30
Device2(config-if-serial1/0)#exit
```

Step 3: Check the result.

#View the frame relay mapping on Device1.

```
Device1#show frame-relay map
serial1/0 (up): ip 1.0.0.2, dlci 30, static,
     IETF, status ACTIVE
```

We can see that serial1/0 can be mapped to the peer address and the status is ACTIVE.

#Ping the peer address on Device1 and the ping can be connected.

```
Device1#ping 1.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

3.3.3. Configure Point-to-Multipoint Sub Interface of Frame Relay

Network Requirement

- Device1, Device2, and Device3 are connected to the frame relay network; Device1 uses the point-to-multipoint sub interface to communicate with Device2 and Device3.

Network Topology

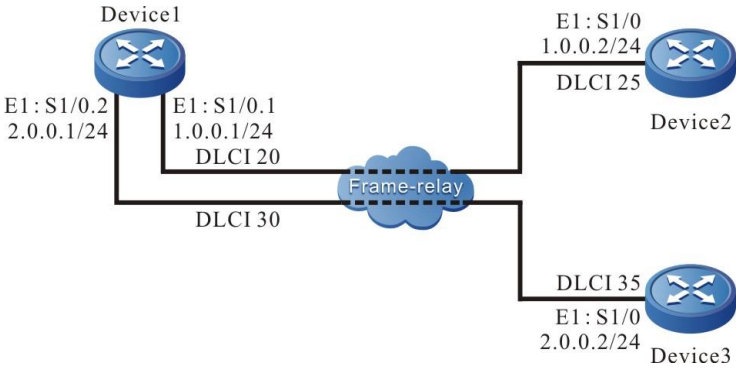

Figure 3–5 Networking of configuring the point-to-multipoint sub interface of frame relay

Configuration Steps

Step 1: Configure the interface to encapsulate the frame relay protocol and configure the IP address.

#Configure the point-to-multipoint sub interface of the frame relay on Device1.

```
Device1#configure terminal
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#encapsulation frame-relay
Device1(config-if-serial1/0)#exit
Device1(config)#interface serial 1/0.1 multipoint
Device1(config-if-serial1/0.1)#ip address 1.0.0.1 255.255.255.0
Device1(config-if-serial1/0.1)#exit
Device1(config)#interface serial 1/0.2 multipoint
Device1(config-if-serial1/0.2)#ip address 2.0.0.1 255.255.255.0
Device1(config-if-serial1/0.2)#exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial 1/0
Device2(config-if-serial1/0)#encapsulation frame-relay
Device2(config-if-serial1/0)#ip address 1.0.0.2 255.255.255.0
```

#Configure Device3.

```
Device3#configure terminal
Device3(config)#interface serial 1/0
Device3(config-if-serial1/0)#encapsulation frame-relay
Device3(config-if-serial1/0)#ip address 2.0.0.2 255.255.255.0
```

Step 2: Configure DLCI and configure the static address mapping.

#Configure Device1.

```
Device1(config)#interface serial 1/0.1 multipoint
Device1(config-if-serial1/0.1)#frame-relay interface-dlci 20
Device1(config-fr-dlci)#exit
Device1(config-if-serial1/0.1)#frame-relay map ip 1.0.0.2 20
Device1(config-if-serial1/0.1)#exit
Device1(config)#interface serial 1/0.2 multipoint
Device1(config-if-serial1/0.2)#frame-relay interface-dlci 30
Device1(config-fr-dlci)#exit
Device1(config-if-serial1/0.2)#frame-relay map ip 2.0.0.2 30
Device1(config-if-serial1/0.2)#exit
```

#Configure Device2.

```
Device2(config-if-serial1/0)#frame-relay interface-dlci 25
Device2(config-fr-dlci)#exit
Device2(config-if-serial1/0)#frame-relay map ip 1.0.0.1 25
Device2(config-if-serial1/0)#exit
```

#Configure Device3.

```
Device3(config-if-serial1/0)#frame-relay interface-dlci 35
Device3(config-fr-dlci)#exit
Device3(config-if-serial1/0)#frame-relay map ip 2.0.0.1 35
Device3(config-if-serial1/0)#exit
```

Step 3: Check the result.

#View the frame relay mapping on Device1.

```
Device1#show frame-relay map
serial1/0.1 (up): ip 1.0.0.2, dlci 20, static,
      IETF, status ACTIVE
serial1/0.2 (up): ip 2.0.0.2, dlci 30, static,
      IETF, status ACTIVE
```

We can see that both serial1/0.1 and serial1/0.2 can be mapped to the peer address and the status is ACTIVE.

Ping the address of the peer Device2 on Device1 and the ping can be connected.

```
Device1#ping 1.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
```

```
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

#Ping the address of the peer Device3 on Device1 and the ping can be connected.

```
Device1#ping 2.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 2.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

Note:

- By default, the sub interface type of the frame relay is point-to-multipoint.

3.3.4. Configure Point-to-Point Sub Interface of Frame Relay

Network Requirement

- Device1 and Device2 are connected to the frame relay network; Device1 uses the point-to-point sub interface to communicate with Device2.

Network Topology

Figure 3–6 Networking of configuring the point-to-point sub interface of frame relay

Configuration Steps

Step 1: Configure the point-to-point sub interface of frame relay and configure the IP address.

#Configure Device1.

```
Device1#configure terminal
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#encapsulation frame-relay
Device1(config-if-serial1/0)#exit
Device1(config)#interface serial 1/0.1 point-to-point
Device1(config-if-serial1/0.1)#ip address 1.0.0.1 255.255.255.0
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface serial 1/0
Device2(config-if-serial1/0)#encapsulation frame-relay
Device2(config-if-serial1/0)#exit
Device2(config)#interface serial 1/0.1 point-to-point
Device2(config-if-serial1/0.1)#ip address 1.0.0.2 255.255.255.0
```

Step 2: Configure DLCI.

#Configure Device1.

```
Device1(config-if-serial1/0.1)#frame-relay interface-dlci 20
Device1(config-fr-dlci)#exit
Device1(config-if-serial1/0.1)#exit
```

#Configure Device2.

```
Device2(config-if-serial1/0.1)#frame-relay interface-dlci 30
Device2(config-fr-dlci)#exit
Device2(config-if-serial1/0.1)#exit
```

Step 3: Check the result.

#View the frame relay mapping on Device1.

```
Device1#show frame-relay map
serial1/0.1 (up): point-to-point, dlci 20, static,
      broadcast,
      IETF, status ACTIVE
serial1/0.1 (up): point-to-point6, dlci 20, static,
      broadcast,
      IETF, status ACTIVE
```

We can see that serial1/0.1 type is point-to-point and the status is ACTIVE.

Ping the address of the peer Device2 on Device1.

```
Device1#ping 1.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

Note:

- The corresponding PVC of the frame relay point-to-point sub interface DLCI can send the multicast and broadcast packets.

4. VIRTUAL ETHERNET

4.1. Overview

Virtual Ethernet protocol works at the data link layer. Its basic principle is to forward the Ethernet frame transparently on the WAN line. The WAN interface encapsulated with the protocol is bound with one virtual Ethernet interface. The Ethernet frame is bridged between the virtual Ethernet interface and WAN interface, that is, the data sent by the virtual Ethernet interface is directly sent to the WAN interface for transmitting, while the data received by the WAN interface is directly sent to the virtual Ethernet interface for processing. In this way, the network layer protocol above the data link layer regards that the data is sent and received via the virtual Ethernet interface. However, for the user, configuring and using the virtual Ethernet interface is the same as operating one real Ethernet interface. WAN interface is transparent for them.

4.2. Virtual Ethernet Function Configuration

Table 4-1 Virtual Ethernet function configuration list

Configuration Task	
Configure the basic functions of the virtual Ethernet	Configure the basic functions of the virtual Ethernet
Configure the MAC address of the virtual Ethernet interface	Configure the MAC address of the virtual Ethernet interface
Configure the keepalive function of virtual Ethernet	Configure the keepalive function of virtual Ethernet
Configure the keepalive timeout of virtual Ethernet	Configure the keepalive timeout of virtual Ethernet

4.2.1. Configure Basic Functions of Virtual Ethernet

Configuration Condition

None

Configure Basic Functions of Virtual Ethernet

Configuring the basic functions of the virtual Ethernet includes two steps: one is to create one virtual Ethernet interface; the other is to encapsulate the virtual Ethernet protocol on the physical interface. When encapsulating the virtual Ethernet, specify the corresponding virtual Ethernet interface and the corresponding physical interface becomes the member interface of the virtual Ethernet interface.

Table 4-2 Configure the basic functions of the virtual Ethernet

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Create one virtual Ethernet interface	**interface virtualethernet** *virtualethernet-unit*	Mandatory By default, do not create virtual Ethernet interface.
Exit the virtual Ethernet interface mode	**exit**	-
Enter the interface configuration mode	**interface** *interface-name*	The interface should be the WAN physical interface.
Encapsulate the virtual Ethernet protocol	**encapsulation virtualethernet virtualethernet** *virtualethernet-unit*	Mandatory By default, the interface is not encapsulated with the virtual Ethernet protocol.

Note:

- After creating one virtual Ethernet interface, we can configure like the common Ethernet interface.
- When the WAN port encapsulates virtual Ethernet, it can only encapsulate the virtual Ethernet main interface, not the virtual Ethernet sub interface.

4.2.2. Configure MAC Address of Virtual Ethernet Interface

Configuration Condition

Before configuring the MAC address of the virtual Ethernet interface, first complete the following task:

- Create one virtual Ethernet interface.

Configure MAC Address of Virtual Ethernet Interface

Table 4-3 Configure the MAC address of the virtual Ethernet interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the virtual Ethernet interface configuration mode	**interface virtualethernet** *virtualethernet-unit*	-
Configure the MAC address of the virtual Ethernet interface	**veth-macaddr** *mac-address*	Optional By default, generate the MAC address automatically after the virtual Ethernet interface is created.

4.2.3. Configure Keepalive Function of Virtual Ethernet Interface

Configuration Condition

Before configuring the keepalive function of the virtual Ethernet interface, first complete the following task:

- Create one virtual Ethernet interface

Configure the Keepalive Function of Virtual Ethernet Interface

Table 4-4 Configure the keepalive function of the virtual Ethernet interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the virtual Ethernet interface configuration mode	**interface virtualethernet** *virtualethernet-unit*	-
Configure the keepalive function of the virtual Ethernet interface	**snmp manage**	Optional By default, the keepalive function is not enabled after the virtual Ethernet interface is created.

4.2.4. Configure Keepalive Receiving Timeout of Virtual Ethernet Interface

Configuration Condition

Before configuring the keepalive timeout of the virtual Ethernet interface, first complete the following task:

- Create one virtual Ethernet interface

Configure Keepalive Receiving Timeout of Virtual Ethernet Interface

Table 4-5 Configure the keepalive timeout of the virtual Ethernet interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the virtual Ethernet interface configuration mode	**interface virtualethernet** *virtualethernet-unit*	-
Configure the keepalive timeout of the virtual Ethernet interface	**snmp-manage keepalive** *timeouts*	Optional By default, the keepalive timeout is 20s after the virtual Ethernet interface is created.

4.2.5. Monitoring and Maintaining of Virtual Ethernet

Table 4-6 Monitoring and maintaining of the virtual Ethernet bridge

Command	Description
show interface virtualethernet *virtualethernet-unit*	Display the information of the specified virtual Ethernet interface
show virtualethernet { detail | *veth-name* **detail}**	Display the details of the virtual Ethernet interface and encapsulated physical interface
show virtualethernet [*veth-name*]	Display the binding information of virtual Ethernet interface and encapsulated physical interface
debug virtualethernet	The output switch of the virtual Ethernet event
debug virtualethernet error	The output switch of the virtual Ethernet error

Command	Description
debug virtualethernet rpc	The output switch of the virtual Ethernet rpc information
debug virtualethernet keepalive	The virtual Ethernet keepalive switch
debug virtualethernet event	The event output information switch of virtual Ethernet adaptation layer
debug virtualethernet process	The processing flow output information switch of virtual Ethernet adaptation layer

4.3. Typical Configuration Example of Virtual Ethernet

4.3.1. Configure Private Line Interconnection of Virtual Ethernet Devices

Network Requirement

- Device1 and device2 are connected through WAN interface, and the type of interface encapsulation is virtual Ethernet. Device1 can communicate with device2.

Network Topology

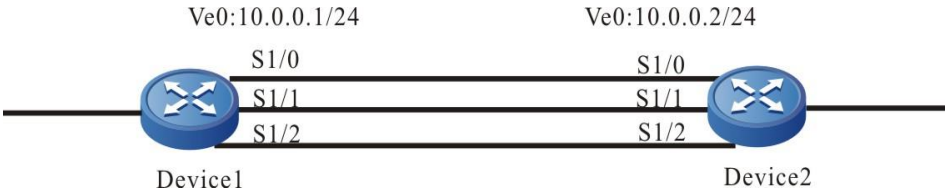

Figure 4-1 Networking for configuring virtual Ethernet

Configuration Steps

Step 1: Configure E1 interface clock source.

#Configure Device1 to use the internal clock.

```
Device1#configure terminal
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#clock source internal
Device1(config-if-serial1/0)#exit
Device1(config)#interface serial 1/1
Device1(config-if-serial1/1)#clock source internal
Device1(config-if-serial1/1)#exit
Device1(config)#interface serial 1/2
Device1(config-if-serial1/2)#clock source internal
Device1(config-if-serial1/2)#exit
```

Device2 uses default line clock.

Step 2: Configure the virtual Ethernet interface.

#Configure Device1.

```
Device1(config)#interface virtualethernet 0
Device1(config-if-virtualethernet0)#ip address 10.0.0.1 255.255.255.0
Device1(config-if-virtualethernet0)exit
```

#Configure Device2.

```
Device2#configure terminal
Device2(config)#interface virtualethernet 0
Device2(config-if-virtualethernet0)#ip address 10.0.0.2 255.255.255.0
Device2(config-if-virtualethernet0)#exit
```

Step 3: Configure E1 interface to encapsulate link layer protocol as virtual Ethernet.

#Configure Device1.

```
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#encapsulation virtualethernet virtualethernet 0
Device1(config-if-serial1/0)#exit
Device1(config)#interface serial 1/1
Device1(config-if-serial1/1)#encapsulation virtualethernet virtualethernet 0
Device1(config-if-serial1/1)#exit
Device1(config)#interface serial 1/2
Device1(config-if-serial1/2)#encapsulation virtualethernet virtualethernet 0
Device1(config-if-serial1/2)#exit
```

#Configure Device2.

```
Device2(config)#interface serial  1/0
Device2(config-if-serial1/0)#encapsulation virtualethernet virtualethernet 0
Device2(config-if-serial1/0)#exit
Device2(config)#interface serial 1/1
Device2(config-if-serial1/1)#encapsulation virtualethernet virtualethernet 0
Device2(config-if-serial1/1)#exit
Device2(config)#interface serial 1/2
Device2(config-if-serial1/2)#encapsulation virtualethernet virtualethernet 0
Device2(config-if-serial1/2)#exit
```

Step 4: Check the result.

#View the virtual Ethernet interface status of Device1.

```
Device1#show interface virtualethernet 0
```

```
virtualethernet0:
    line protocol is up
    Flags: (0x1c008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 10.0.0.1/24
    Broadcast address: 10.0.0.255
    Metric: 0, MTU: 1500, BW: 6144 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 001f.ce7b.6dd6
    Last clearing of "show interface" counters is 1 week 1 day ago
    input peak rate 805 bits/sec,  5 days 17 hours ago
    output peak rate 528 bits/sec,  5 days 17 hours ago
    5 minutes input rate 50 bits/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 50 bits/sec, 0 packet/sec, bandwidth utilization -
    0 packets received; 0 packets sent
    0 bytes received; 0 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
    Bundle member list:
    virtualethernet0 have 3 member node:
     serial1/0  BandWidth 2048Kbps              Active
     serial1/1  BandWidth 2048Kbps              Active
     serial1/2  BandWidth 2048Kbps              Active
```

You can see that the members of the virtualethernet0 interface are Serial1/0, Serial1/1 and Serial1/2. To view the virtual Ethernet interface status on device2, please refer to the operation of device1.

#On Device1, ping virtualethernet0 interface address of peer Device2.

```
Device1#ping 10.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 10.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

4.3.2. Configure Virtual Ethernet Bridge

Network Requirement

- The WAN interfaces of Device1 are bound together by using virtual Ethernet to increase the physical bandwidth.
- Add the lower virtual Ethernet port and upper Ethernet port of Device1 into a bridge group, and the lower Ethernet port and upper Ethernet port of Device2 into a bridge group.
- PC1 communicates with PC2 through the bridge link between Device1 and Device2.

Network Topology

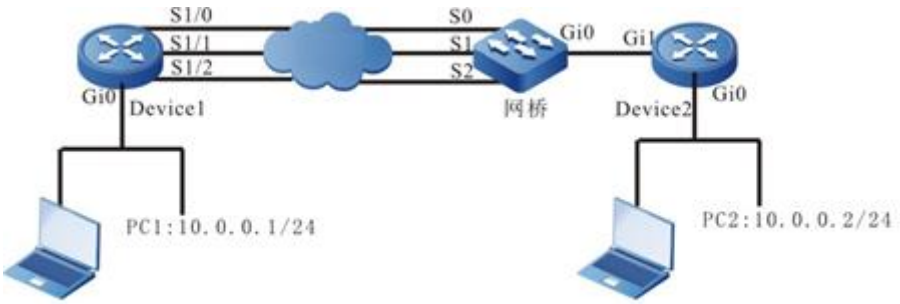

Figure 4-2 Networking of configuring virtual Ethernet

Configuration Steps

Step 1: Configure the IP address of the PC. (omitted)

Step 2: Configure the clock source of the E1 interface.

#Configure Device1 to use the internal clock.

```
Device1#configure terminal
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#clock source internal
Device1(config-if-serial1/0)#exit
Device1(config)#interface serial 1/1
Device1(config-if-serial1/1)#clock source internal
Device1(config-if-serial1/1)#exit
Device1(config)#interface serial 1/2
Device1(config-if-serial1/2)#clock source internal
Device1(config-if-serial1/2)#exit
```

Step 3: Configure the virtual Ethernet interface and enable keepalive.

#Configure Device1.

```
Device1(config)#interface virtualethernet 0
Device1(config-if-virtualethernet0)#snmp manage
Device1(config-if-virtualethernet0)exit
```

Step 4: Configure the E1 interface to encapsulate the link layer protocol as virtual Ethernet.

#Configure Device1.

```
Device1(config)#interface serial 1/0
Device1(config-if-serial1/0)#encapsulation virtualethernet virtualethernet 0
Device1(config-if-serial1/0)#exit
Device1(config)#interface serial 1/1
Device1(config-if-serial1/1)#encapsulation virtualethernet virtualethernet 0
Device1(config-if-serial1/1)#exit
Device1(config)#interface serial 1/2
Device1(config-if-serial1/2)#encapsulation virtualethernet virtualethernet 0
Device1(config-if-serial1/2)#exit
```

Step 5: Configure the bridge and add the interface to the bridge group.

#Configure Device1.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#bridge-group 1
Device1(config-if-gigabitethernet0)#exit
Device1(config)#interface virtualethernet 0
Device1(config-if-virtualethernet0)#bridge-group 1
Device1(config-if-virtualethernet0)#exit
```

#Configure Device2.

```
Device2(config)#interface gigabitethernet 0
Device2(config-if-gigabitethernet0)#bridge-group 1
Device2(config-if-gigabitethernet0)#exit
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#bridge-group 1
Device2(config-if-gigabitethernet1)#exit
```

Step 6: Check the result.

#View the virtual Ethernet interface status of Device1.

```
Device1#show interface virtualethernet 0
virtualethernet0:
    line protocol is up
    Flags: (0x1c008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 10.0.0.1/24
```

```
Broadcast address: 10.0.0.255
Metric: 0, MTU: 1500, BW: 6144 Kbps, DLY: 100 usec, VRF: global
Reliability 255/255, Txload 1/255, Rxload 1/255
Ethernet address is 001f.ce7b.6dd6
Last clearing of "show interface" counters is 1 week 1 day ago
input peak rate 805 bits/sec,  5 days 17 hours ago
output peak rate 528 bits/sec,  5 days 17 hours ago
5 minutes input rate 50 bits/sec, 0 packet/sec, bandwidth utilization -
5 minutes output rate 50 bits/sec, 0 packet/sec, bandwidth utilization -
0 packets received; 0 packets sent
0 bytes received; 0 bytes sent
0 multicast packets received
0 multicast packets sent
0 input errors; 0 output errors
0 collisions; 0 dropped
Unknown protocol 0
Bundle member list:
virtualethernet0 have 3 member node:
 serial1/0  BandWidth 2048Kbps              Active
 serial1/1  BandWidth 2048Kbps              Active
 serial1/2  BandWidth 2048Kbps              Active
```

You can see that the members of the virtualethernet0 interface are serial1/0, serial1/1 and serial1/2. To view the virtual Ethernet interface status on device2, please refer to the operation of Device1.

#On PC1, ping PC2 address.

```
Device1#ping 10.0.0.2


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 10.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/0/0 ms.
```

Note:

- When the lower device is QTECH bridge, it is necessary to enable the keepalive function when configuring the virtual Ethernet. When the lower device is QTECH router, it is not necessary to enable the keepalive function.
- If there are multiple interfaces in the same bridge group, it is recommended to configure the port isolation function (set interface insulation) under the interface to prevent loop.

- If there are a large number of unknown unicast packets, it is recommended to configure the storm suppression function (**bridge-group 1 storm-control unicast pps**) under the uplink interface to filter the unknown unicast packets.
- If you need to configure QTECH bridge, please consult QTECH technical personnel.

5. BRIDGING

5.1. Overview

Bridging realizes Ethernet L2 functions, forwarding the Ethernet frames transparently. One bridging group is equivalent to one Ethernet L2 switch. The interface configured to the bridging group is equivalent to one port of the switch. When one interface receives the frame, the bridging group creates the MAC address table according to the source MAC address of the frame and searches the MAC address table to forward according to the destination MAC address of the frame. If the destination MAC address is broadcast or multicast, or unknown unicast (that is, do not find the corresponding MAC address entry of the unicast frame), forward to all interfaces in the bridging group (also called flood). Bridging maintains its own MAC address entries and will age and update.

When multiple Ethernets are interconnected via WAN, we can bridge between Ethernet interface and WAN interface, so as to perform the L2 communication directly between Ethernets. This is transparent for Ethernets. It seems that they are directly connected.

Bridging itself does not have standard protocols, realizing the address learning and forwarding functions complying with Ethernet standard IEEE 802.1D.

Bridging also supports the IRB function. IRB is short for Integrated Routing and Bridging. It can run one specified protocol between the routing interface and bridging group or between different bridging groups. The local or un-routable data can be bridged between bridging interfaces of one bridging group. The routable data can be routed between other routing interfaces or bridging groups. To set up the relation between the bridging group and route, bring in Bridge-Group Virtual Interface (BVI). BVI interface is one virtual Ethernet interface. It does not support bridging, indicating the routing interface of the bridging group. It has all network later attributes and the attributes are applied to the corresponding bridging groups. BVI interface unit number corresponds to the bridging group number.

5.2. Bridge Function Configuration

Table 5-1 Bridge function configuration list

Configuration Task	
Configure the basic functions of the bridging	Add one interface to the bridging group
Configure the bridging parameters	Configure the age time of the MAC address entry of the bridging group
	Configure the max. MAC addresses of the bridging group
	Configure the function of updating the source MAC address of the bridging interface
Configure the BVI interface	Create one BVI interface
	Configure the MAC address of the BVI interface

5.2.1. Configure Bridging Basic Functions

Add the interface to the bridging group and then the interface enables the bridging function.

Configuration Condition

None

Add One Interface to Bridging Group

Table 5-2 Add one interface to the bridging group

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Add one interface to the bridging group	**bridge-group** *group-number*	Mandatory By default, the interface is not added to the bridging group.

5.2.2. Configure Bridging Parameters

Bridging parameters include age time of the MAC address of the bridging group, the maximum addresses of the birdging group, and the updating function of the source MAC address of the bridging interface.

Configuration Condition

Before configuring the bridging parameters, first complete the following task:

- Add the interface to the bridging group

Configure Age Time of MAC Address Entry of Bridging Group

The age time of the MAC address entry is used to control the MAC address table to be deleted automatically because of not being updated. It should be set according to the actual network environment.

Table 5-3 Configure the age time of the MAC address entry of the bridging group

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the age time of the MAC address entry of the bridging group	**bridge** *group-number* **aging-time** *aging-time-value*	Optional By default, the age time of the MAC address entry of the bridging group is 300s.

Configure Max. MAC Addresses of Bridging Group

The upper limit of MAC addresses is used to limit the number of forwarding table entries of bridge group, which can be set according to the actual network environment.

Table 5-4 Configure the upper limit of the MAC addresses of the bridging group

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the upper limit of the MAC addresses of the bridging group	**bridge** *group-number* **limit-mac** *limit-mac-value*	Optional By default, the upper limit of the MAC addresses of the bridging group is 16k.

Configure Updating Function of Source MAC Address of Bridging Interface

After the bridging interface receives the frame, create the MAC address table according to the source MAC address of the frame. After creating the entry successfully and if the other interface in the bridging group receives the frame with the same source MAC address, we need to update the interface to which the MAC address entry belongs. This is the updating function of the source MAC address of the bridging. Sometimes, to ensure the network security, we need to disable the source MAC address function to prevent the frequent change of the belonging interface of some MAC address entry from causing the network oscillation.

Table 5-5 Configure the function of updating the source MAC address of the bridging interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the function of updating the source MAC address of the bridging interface	**bridge-group** *group-number* **address update**	For the physical Ethernet interface and its sub interface, VLAN interface, it is optional. By default, the function of updating the source MAC addresses of these interfaces is enabled. For the other types of the bridging interfaces, it is mandatory. By default, the function of updating the source MAC addresses of these interfaces is disabled.

5.2.3. Configure BVI Interface

BVI interface is one virtual Ethernet interface. It is used to perform the route forwarding in the bridging group. Its interface unit number corresponds to the bridge group number. If the destination MAC address of the frame received from any interface in the bridging group is the MAC address of the BVI interface, submit the frame to the BVI interface for processing. When the BVI interface is created, it is also added to the bridging group. However, the BVI interface just has the routing function, but does not have the bridging function.

Configuration Condition

None

Create BVI Interface

Table 5-6 Create one BVI interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Create one BVI interface	**interface bvi** *bvi-unit*	Mandatory By default, do not create BVI interface.

Note:

- After creating one BVI interface, we can configure the network layer parameters to support the routing function like the common Ethernet interface.

Configure MAC Address of BVI Interface

The MAC address of the BVI interface can be modified manually as needed.

Table 5-7 Configure the MAC address of the BVI interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the BVI interface configuration mode	**interface bvi** *bvi-unit*	Mandatory
Configure the MAC address of the BVI interface	**bvi-macaddr** *mac-address*	Optional By default, the BVI interface automatically generates the MAC address.

5.2.4. Bridge Monitoring and Maintaining

Table 5-8 Bridge monitoring and maintaining

Command	Description
show bridge group control-table summary	Display the bridge group table entry information
show bridge dev control-table summary	Display the bridge device information

5.3. Bridge Typical Configuration Example

5.3.1. Configure Common Bridging

Network Requirements

- Two Ethernet ports of the device are connected to two LANs.
- Configure the bridging to realize the packet L2 transparent forwarding between two Ethernet ports.

Network Topology

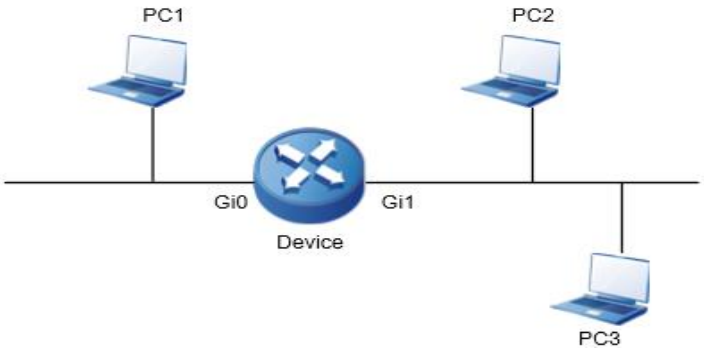

Figure 5-1 Networking of configuring the common bridge

Configuration Steps

Step 1: Configure the bridge and add the interface to bridge group 1.

Add two Ethernet interfaces to one bridge group.

```
Device(config)#interface gigabitethernet 0
Device(config-if-gigabitethernet0)#bridge-group 1
Device(config-if-gigabitethernet0)#exit
Device(config)#interface gigabitethernet 1
Device(config-if-gigabitethernet1)#bridge-group 1
Device(config-if-gigabitethernet1)#exit
Device(config)#exit
```

#View the interface added to the bridge group.

```
Device#show bridge dev control-table summary
3 Dev Entries found
index ifIndex devId fdbDevId groupId fdbGrpId M|S|C flags broadcastPps unicastPps type sync ifname
----- ---------- ----- -------- ------- -------- -------- ---- ---------- ---------- ---- ---- ----------
1 671088641 1 1 1 1 0|FF|FF 0 0 0 3 0 bvi1
2 1284243458 2 2 1 1 0|00|00 0 0 0 1 0 gigabitethernet0
3 1284243489 3 3 1 1 0|00|00 0 0 0 1 0 gigabitethernet1
```

Step 2: Check the result.

#Three PCs can perform L2 intercommunication.

Note:

- To configure bridge on the HDLC link and PPP link, it is necessary to configure the IP address.

5.3.2. Configure Virtual Interface of Bridging Group

Network Requirement

- gigabitethernet0 and gigabitethernet1 of Device1 are connected to two LANs with segment 200.0.0.0/24; gigabitethernet2 is connected to WAN.
- On gigabitethernet0 and gigabitethernet1 of Device1, configure the bridging to realize the L2 transparent forwarding of the packet between two LANs.
- On Device1, configure the BVI interface to realize the L3 communication between LAN and WAN. Meanwhile, the PCs in two LANs 200.0.0.0/24 can manage Device1 via the BVI interface.

Network Topology

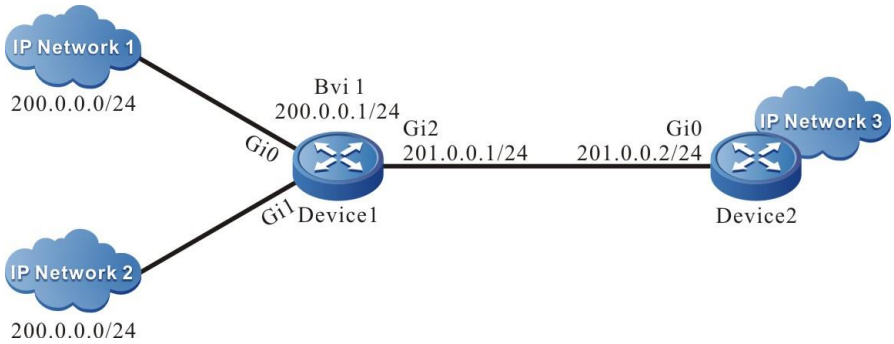

Figure 5-2 Networking of configuring the virtual interface of bridging group

Configuration Steps

Step 1: Configure the IP address of the interface and configure the route.

(Omitted)

Step 2: Configure the common bridging.

#Add the two Ethernet interfaces connected to LAN to one bridging group.

```
Device1#configure terminal
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#bridge-group 1
Device1(config-if-gigabitethernet0)#exit
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#bridge-group 1
Device1(config-if-gigabitethernet1)#exit
```

Step 3: Configure the corresponding BVI interface of the bridging group 1.

```
Device1(config)#interface bvi 1
Device1(config-if-bvi1)#ip address 200.0.0.1 255.255.255.0
```

#View the interface added to the bridge group.

```
Device1#show bridge dev control-table summary
3 Dev Entries found
```

```
index ifIndex    devId fdbDevId groupId fdbGrpId M|S|C   flags broadcastPps
unicastPps  type sync ifname
----- ---------- ----- -------- ------- -------- -------- ---- ------------
------------ ---- ---- ----------------------------
1    671088641   1    1      1      1      0|FF|FF  0    0        0        3    0    bvi1
2    1284243458  2    2      1      1      0|00|00  0    0        0        1    0
gigabitethernet0
3    1284243489  3    3      1      1      0|00|00  0    0        0        1    0
gigabitethernet1
```

Step 4: Check the result.

#IP Network1 and IP Network2 can communicate with each other at layer 2.

#The users in IP Network1 and IP Network2 can manage Device 1 via the address 200.0.0.1.

#IP Network1 and IP Network2 can configure the gateway as 200.0.0.1 to communicate with WAN IP Network3 at layer 3.

6. PPPOE

6.1. Overview

With the development of the broadband network technology, the mainstream broadband access technologies, such as xDSL, CableModem and Ethernet are widely applied. Meanwhile, it brings some puzzles for the network carriers. No matter which access technology to be used, how to manage the users efficiently and how to get larger returns from the network investment are very important for them. Therefore, the problems about the charges for the various broadband access technologies become more sensitive. In the traditional Ethernet model, there is no the concept of user charges. Either the user sets/gets the IP address to access the network, or the user cannot access the network. IETF engineers adhere to the operation idea of narrowband dial-up for accessing Internet (use the NAS devices to terminate the PPP packets of the user) to work out the protocol of transmitting the PPP packets on the Ethernet

PPPoE (Point To Point Protocol Over Ethernet) is one protocol of realizing the PPP connection on Ethernet. It provides one standard of connecting multiple hosts in the broadcast network (such as Ethernet) to the remote access concentrator (broadband access server). It adopts the client/server mode. On typical PPPoE application is that PC uses the PPPoE dialing software to set up the point-to-point connection via Ethernet and access concentrator.

PPPoE has two stages, that is, PPPoE Discovery Stage and PPPoE Session Stage. No matter which stage of data packets will eventually be encapsulated into Ethernet frames for transmission.

6.1.1. PPPoE Discovery Stage

At the different stages of PPPoE, the data contents in the payload domain of the PPPoE frame are different. During the discovery stage of PPPoE, fill in some tag in the domain; while during the PPPoE session stage, fill in the PPP packets in the domain. The discovery stage is divided to four steps:

Step 1: The user host first sends one PADI (PPPoE Active Discovery Initiation) packet. The user host sends the packet via the broadcast mode, so the destination address of the Ethernet frame of the packet is filled with all 1, while the source address is filled with the MAC address of the user host. The broadcast packet may be received by multiple access concentrators. The PADI packet should contain one correct service name tag requested by the user host.

Step 2: The access concentrator sends the PADO (PPPoE Active Discovery Offer) packet to answer the PADI packet sent by the user host. The source address of the Ethernet frame of the packet is filled with the MAC address of the access concentrator, while the destination address is filled with the MAC address of the user host got from PADI. The PADO packet should contain one access concentrator name tag, the confirm tag for the service name tag in the PADI and some confirm tags for other tags. If the service access concentrator applied by the user host does not support, the access concentrator does not answer the PADO packet.

Step 3: The user host sends the unicast PADR (PPPoE Active Discovery Request) packet to the access concentrator. The user host selects one access concentrator according to the service name tag of the received PADO packet, and then sends the PADR packet to it. After the user host receives the PADO packet, it gets to know the MAC address of the access concentrator, so the source address of the Ethernet frame of the PADR packet is filled with the MAC address of the user host and the destination address is filled with the MAC address of the access concentrator. The PADR packet should contain one service name tag, indicating the the service requested by the user host and other tag types.

Step 4: When receiving the PADR packet, the access concentrator prepares to start one PPP session, while the access concentrator distributes one unique session ID for the session and

carries the session ID in the PADS (PPPoE Active Discovery Session-confirmation) packet sent to the host. If the access concentrator does not meet the service applied by the user, the PADS packet carries the error tag of one service name.

With the above four steps, the user host and access concentrator get the peer MAC address and the unique session ID, so as to enter the PPPoE session stage. Here, the unique point-to-point session is set up between the host and the access concentrator.

6.1.2. PPPoE Session Stage

Once PPPoE enters the session stage, PPP packet is filled in the payload of the PPPoE frame and transmitted to the peer. Here, all Ethernet packets sent by the both are all the peer MAC address. During the session stage, any party of the host or access concentrator can send the PADT (PPPoE Active Discovery Terminate) packet to inform the peer party to end the session.

6.2. PPPoE Function Configuration

Table 6-1 PPPoE function configuration list

Configuration Task	
Configure DDR basic functions	Enable the DDR function
	Configure the dialing set to associate with the dialing interface
Configure the PPPoE dialing interface	Configure the dialing interface
Configure the PPPoE dialing mode	Configure the dialing mode
Configure the PPPoE server	Configure the server

6.2.1. Configure DDR Basic Functions

Confguration Conditions

None

Create Dialing Control List

By configuring the dial-up control list, various packets flowing through the dialing interface can be filtered.

Table 6-1 Create dialing control list

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the dialing control list	**dialer-list** *dialer-list-number* **protocol** { **ip** | **ipv6** } { **deny** | **list** *access-list* **| permit** }	Mandatory By default, the dialing control list is not configured Dialing control list can not only directly configure the filtering conditions of packets, but also introduce the rules of ACL access control list.

Enable the DDR Function

Enable the DDR function on the dialer interface.

Table 6-2 Enable the DDR function

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Enable the DDR function on the interface	**dialer in-band**	Mandatory By default, the interface does not enable DDR.

Configure Dialing Control List to Associate with Dialing Interface

Associate a dialing interface with the dialing access control list to control access. Use the following command to configure the interface.

Table 6-3 Configure the dialing control list to associate with the dialing interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the dialing control list to associate with the dialing interface	**dialer-group** *group-number*	Mandatory By default, the dialing control list is not configured to be associated with the dialing interface Before configuring this command, the interface must enable DDR function. This command does not need to be configured for auto dialing.

Configure Dialing Set to Associate with Dialing Interface

For dialing interfaces, to specify which dialing set to be used to connect to a specific destination subnet, use the following command to configure.

Table 6-4 Configure the dialing set to associate with the dialing interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the dialing set to associate with the dialing interface	**dialer pool** *pool-number*	Mandatory By default, do not configure the dialing set to associate with the dialing interface. The command can only be used in the dialer interface.

6.2.2. Configure PPPoE Dialing Interface

Configuration Conditions

Before configuring the PPPoE dialing interface, first complete the following tasks:

- Configure the dialer interface
- Configure the dialing pool of the dialer interface (refer to the DDR configuration)

Configure Dialing Interface

Before configuring the PPPoE session, we need to configure one dialer interface at first and configure the dialing pool on the dialer interface. Each PPPoE session corresponds to one dialing pool uniquely, and each dialing pool corresponds to one dialer interface uniquely. This is equivalent to set up one PPPoE session via one dialer interface.

To configure the dialing interface, we need to configure in the interface.

Table 6-2 Configure the dialing interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the dialing interface	**pppoe-client dial-pool-number** *dial-pool-number* [**ac-name** *ac-name*]	Mandatory By default, do not configure the dialing interface.

6.2.3. Configure PPPoE Dialing Mode

Configuration Condition

Before configuring the PPPoE dialing mode, first complete the following task:

- Configure the PPPoE dialing interface

Configure Dialing Mode

The work mechanism of the PPPoE auto dialing mode is described as follows:

- Auto dialing: After the device configuration is complete, initiate the PPPoE call at once to set up the PPPoE session.

To configure the dialing mode, we need to configure in the interface.

Table 6-3 Configure the dialing mode

Operation	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the dialing mode	**pppoe-client auto-dial** { **always** | **time-range** *time-range-name*}	Mandatory By default, do not configure the dialing mode.

6.2.4. Configure PPPoE Server

Configuration Condition

- It can be configured at the Ethernet main interface and Ethernet sub interface.

Configure PPPOE Server

Enable the PPPoE server function. To monitor the PPPoE negotiation packet sent by the client on the line, set up the session connection with the client. Before enabling the server, first configure VPDN, encapsulate the PPPOE protocol, and configure the virtual template. To prevent the client attack, the server can limit the number of the connections of one MAC.

Table 6-4 Configure enabling the server mode

Operation	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the enable mode of the PPPOE server	**pppoe enable**	Mandatory By default, do not configure the enable mode of the server.

Table 6-5 Configure using the PPPOE protocol on VPDN

Operation	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the VPDN mode	vpdn-group *vpdn-name*	-
Enter the VPDN access mode	accept-dialin	-
Encapsulate the PPPOE mode	protocol pppoe	Mandatory By default, do not encapsulate the PPPOE protocol.

Table 6-6 Configure the maximum number of the connections of one MAC

Operation	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the VPDN mode	vpdn-group *vpdn-name*	-
Configure the maximum connections of one MAC	pppoe limit per-mac *num*	By default, one MAC permits setting up 10 connections.

6.2.5. PPPoE Monitoring and Maintaining

Table 6-7 PPPoE monitoring and maintaining

Command	Description
debug pppoe { **receive | send | events | data-disp** }	View the packet content and event during the PPPoE negotiation
show pppoe list	View the interface configured with PPPoE
show pppoe session { **count | information** }	View the PPPoE session information, including session quantity and session details

6.3. PPPoE Typical Configuration Example

6.3.1. Configure PPPoE Conventional Dialing

Network Requirement

- Device1 serves as the PPPoE client and Device2 serves as the PPPoE server. Use the PPPoE conventional dialing mode to set up the PPPoE session between Device1 and Device2, and the server assigns the IP address for the client

Network Topology

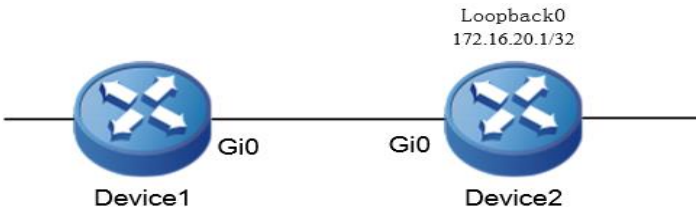

Figure 6-1 Networking of configuring the PPPoE conventional dialing

Configuration Steps

Step 1: Configure the PPPoE client.

#Configure Device1.

Configure the trigger dialing type.

```
Device1#configure terminal
Device1(config)#dialer-list 1 protocol ip permit
```

Create the dialer0 interface and configure the dialing information.

```
Device1#configure terminal
Device1(config)#interface dialer0
Device1(config-if-dialer0)#ip address negotiated
Device1(config-if-dialer0)#dialer in-band
Device1(config-if-dialer0)#dialer pool 1
Device1(config-if-dialer0)#dialer-group 1
Device1(config-if-dialer0)#exit
```

Add the interface gigabitethernet0 to dialing pool 1.

```
Device1(config)#interface gigabitethernet0
Device1(config-if-gigabitethernet0)#pppoe-client dial-pool-number 1
Device1(config-if-gigabitethernet0)#exit
```

Configure one default route with one egress interface dialer0.

```
Device1(config)#ip route 0.0.0.0 0.0.0.0 dialer0
```

Step 2: Configure the PPPoE server.

#Create the loopback port Loopback0 and configure the address.

```
Device2#configure terminal
Device2(config)#interface loopback0
Device2(config-if-loopback0)#ip address  172.16.20.1 255.255.255.255
Device2(config-if-loopback0)#exit
```

#Configure the address pool information, and create the virtual template virtual-template0.

```
Device2(config)#ip local pool pppoe-pool  172.16.20.10 172.16.20.100
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#ip unnumbered loopback0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#peer default ip address  pool pppoe-pool
Device2(config-if-virtual-template0)#exit
```

#Create virtual VPDN group QTECH and use the virtual template virtual-template0.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group QTECH
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol pppoe
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#pppoe enable
Device2(config-if-gigabitethernet0)#exit
```

Step 3: On Device1, ping 1.1.1.1 to trigger PPPoE dialing and check the result.

#View the dialer0 interface information of Device1.

```
Device1#show int dialer 0
dialer0:
    line protocol is up
    Flags: (0xc0080f1) POINT-TO-POINT MULTICAST RUNNING
    Type: PPP
    Internet address: 172.16.20.10/32
    Destination Internet address: 172.16.20.1
    Metric: 0, MTU: 1500, BW: 56 Kbps, DLY: 20000 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Last clearing of "show interface" counters never
    output peak rate 46 bits/sec,  1 week 3 days ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
```

```
5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
4 packets received; 5 packets sent
288 bytes received; 340 bytes sent
4 multicast packets received
5 multicast packets sent
0 input errors; 0 output errors
0 collisions; 0 dropped
```

We can see that the dialer interface protocol of Device1 is up and can get the IP address.

#View the PPPoE session information of Device1.

```
Device1#show pppoe session information

PPPoE Session Information:(Max Sessions=1024
UID  SID  RemMAC       LocMAC       O-Intf              O-VA            state C/S
ActiveTime Local-IP      Peer-IP
 ----- ----- -------------- --------------  -------------------- ---------------
--- ----- --- ---------- ---------------- ----------------
1    1    001f.cecf.6c5a  001f.ce92.e729 gigabitethernet0     virtual-access3001 UP   S
00:02:44   172.16.20.10      172.16.20.1

There are 1 PPPoE sessions up
```

We can see that the PPPoE session is set up between Device1 and Device2.

#We can ping the Loopback0 interface address of Device2 on Device1.

```
Device1#ping 172.16.20.1

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 172.16.20.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 6/8/12 ms.
```

6.3.2. Configure PPPoE Auto Dialing

Network Requirement

- Device1 serves as the PPPoE client and Device2 serves as the PPPoE server. Use the PPPoE auto dialing mode to set up the PPPoE session between Device1 and Device2. The server assgns the IP address for the client.

Network Topology

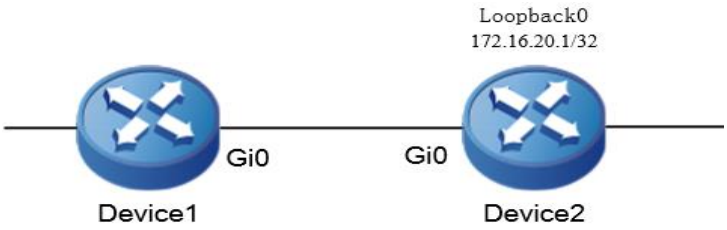

Figure 6-2 Networking of configuring the PPPoE auto dialing

Configuration Steps

Step 1: Configure the PPPoE client.

#Create dialer0 interface and configure the dialing information.

```
Device1#configure terminal
Device1(config)#interface dialer0
Device1(config-if-dialer0)#ip address negotiated
Device1(config-if-dialer0)#dialer in-band
Device1(config-if-dialer0)#dialer pool 1
Device1(config-if-dialer0)#exit
```

Add the interface gigabitethernet0 to dialing pool 1 and configure the auto dialing.

```
Device1(config)#interface gigabitethernet0
Device1(config-if-gigabitethernet0)#pppoe-client dial-pool-number 1
Device1(config-if-gigabitethernet0)#pppoe-client auto-dial  always
Device1(config-if-gigabitethernet0)#exit
```

Step 2: Configure the PPPoE server.

#Create the loopback port Loopback0 and configure the address.

```
Device2#configure terminal
Device2(config)#interface loopback0
Device2(config-if-loopback0)#ip address 172.16.20.1 255.255.255.255
Device2(config-if-loopback0)#exit
```

#Configure the address pool information, and create the virtual template virtual-template0.

```
Device2(config)#ip local pool  pppoe-pool  172.16.20.10 172.16.20.100
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#ip unnumbered loopback0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#peer default ip address  pool pppoe-pool
Device2(config-if-virtual-template0)#exit
```

#Create the virtual VPDN group QTECH and use the virtual template virtual-template0.

```
Device2(config)#vpdn enable
```

```
Device2(config)#vpdn-group QTECH
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol pppoe
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#pppoe enable
Device2(config-if-gigabitethernet0)#exit
```

Step 3: Check the result.

#View the dialer0 interface information of Device1.

```
Device1#show int dialer 0
dialer0:
    line protocol is up
    Flags: (0xc0080f1) POINT-TO-POINT MULTICAST RUNNING
    Type: PPP
    Internet address: 172.16.20.10/32
    Destination Internet address: 172.16.20.1
    Metric: 0, MTU: 1500, BW: 56 Kbps, DLY: 20000 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Last clearing of "show interface" counters never
    output peak rate 46 bits/sec,  1 week 3 days ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    4 packets received; 5 packets sent
    288 bytes received; 340 bytes sent
    4 multicast packets received
    5 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
```

We can see that the dialer0 interface protocol of Device1 is up and can get the IP address.

#View the PPPoE session information of Device1.

```
Device1#show pppoe session information

PPPoE Session Information:(Max Sessions=1024
UID  SID  RemMAC       LocMAC       O-Intf          O-VA          state C/S
ActiveTime Local-IP      Peer-IP
```

```
----- ----- ------------- -------------  ------------------- --------------
--- ----- --- ---------- ---------------- ----------------
1    1     001f.cecf.6c5a  001f.ce92.e729 gigabitethernet0     virtual-access3001 UP   S
00:02:44   172.16.20.10       172.16.20.1


There are 1 PPPoE sessions up
```

We can see that the PPPoE session is set up between Device1 and Device2 successfully.

#We can ping the Loopback0 interface address of Device2 on Device1.

```
Device1#ping 172.16.20.1


Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 172.16.20.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 6/8/12 ms.
```

6.3.3. Configure PPPoE Auto Dialing Based on Time Period

Network Requirement

- Device1 serves as the PPPoE client and Device2 serves as the PPPoE server. Use the PPPoE auto dialing mode based on the time period to permit the client to perform auto dialing every 8:00-18:00 o'clock to set up the PPPoE session with Device2. In the other time period, the client is not permitted to perform auto dialing and set the PPPoE session.

Network Topology

Figure 6-3 Networking of configuring the PPPoE auto dialing

Configuration Steps

Step 1: Configure the PPPoE client.

#Create the dialer0 interface, and configure the dialing information.

```
Device1#configure terminal
Device1(config)#interface dialer0
Device1(config-if-dialer0)#ip address negotiated
Device1(config-if-dialer0)#dialer in-band
Device1(config-if-dialer0)#dialer pool 1
Device1(config-if-dialer0)#exit
```

#Configure the auto dialing period as 8:00 to 18:00.

```
Device1(config)# time-range time
```

```
Device1(config-if-dialer0)# periodic daily 08:00 to 18:00
Device1(config-if-dialer0)#exit
```

Add the interface gigabitethernet0 to the dialing pool 1, and configure the auto dialing.

```
Device1(config)#interface gigabitethernet0
Device1(config-if-gigabitethernet0)#pppoe-client dial-pool-number 1
Device1(config-if-gigabitethernet0)#pppoe-client auto-dial time-range time
Device1(config-if-gigabitethernet0)#exit
```

Step 2: Configure the PPPOE server.

#Create loopback port Loopback0, and configure the address pool.

```
Device2#configure terminal
Device2(config)#interface loopback0
Device2(config-if-loopback0)#ip address  172.16.20.1 255.255.255.255
Device2(config-if-loopback0)#exit
```

#Configure the address pool information and create virtual template virtual-template0.

```
Device2(config)#ip local pool pppoe-pool  172.16.20.10 172.16.20.100
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#ip unnumbered loopback0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#peer default ip address  pool pppoe-pool
Device2(config-if-virtual-template0)#exit
```

#Create virtual VPDN group QTECH, and use the virtual template virtual-template0.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group QTECH
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol pppoe
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#pppoe enable
Device2(config-if-gigabitethernet0)#exit
```

Step 3: Check the result.

View the status of the time domain on Device1.

```
Device1#show time-range time
```

```
Timerange name:time  (STATE:active)
 10  periodic daily 08:00:00 to 18:00:00 (active)
```

#View the dialer0 interface information of Device1.

```
Device1#show int dialer 0
dialer0:
   line protocol is up
   Flags: (0xc0080f1) POINT-TO-POINT MULTICAST RUNNING
   Type: PPP
  Internet address: 172.16.20.10/32
   Destination Internet address: 172.16.20.1
   Metric: 0, MTU: 1500, BW: 56 Kbps, DLY: 20000 usec, VRF: global
   Reliability 255/255, Txload 1/255, Rxload 1/255
   Last clearing of "show interface" counters never
   output peak rate 46 bits/sec,  1 week 3 days ago
   5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
   5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
   4 packets received; 5 packets sent
   288 bytes received; 340 bytes sent
   4 multicast packets received
   5 multicast packets sent
   0 input errors; 0 output errors
   0 collisions; 0 dropped
```

We can see that the dialer0 interface protocol of Device1 is up and can get the IP address.

#View the PPPoE session information of Device1.

```
Device1#show pppoe session information

PPPoE Session Information:(Max Sessions=1024
UID   SID   RemMAC         LocMAC         O-Intf                O-VA            state C/S
ActiveTime Local-IP       Peer-IP
 ----- ----- -------------- --------------  -------------------- ---------------
--- ----- --- ---------- ---------------- ----------------
1     1     001f.cecf.6c5a  001f.ce92.e729 gigabitethernet0     virtual-access3001 UP   S
00:02:44   172.16.20.10       172.16.20.1

There are 1 PPPoE sessions up
```

We can see that the PPPoE session is set up between Device1 and Device2 successfully.

#We can ping the Loopback0 interface address of Device2 on Device1.

```
Device1#ping 172.16.20.1

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 172.16.20.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 6/8/12 ms.
```

6.3.4. Configure PPPoE IPv6 Protocol to Trigger Dialing

Network Requirement

- Device1 is the PPPoE client and Device2 is the PPPoE server. PPPoE IPv6 protocol is used to trigger the dialing mode to set up the PPPoE session between Device1 and Device2.

Network Topology

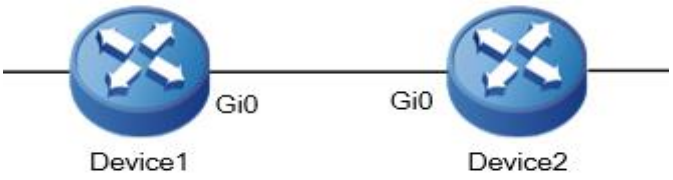

Figure 6-4 Networking for configuring the PPPoE IPv6 protocol to trigger dialing

Configuration Steps

Step 1: Configure the PPPoE client.

#Configure Device1, permitting the IPv6 protocol to trigger dialing.

```
Device1#configure terminal
Device1(config)#dialer-list 1 protocol ipv6 permit
```

#Create dialer0 interface, configure the dialing information, and configure the IP6 address.

```
Device1#configure terminal
Device1(config)#interface dialer0
Device1(config-if-dialer0)#dialer in-band
Device1(config-if-dialer0)#dialer pool 1
Device1(config-if-dialer0)#dialer-group 1
Device1(config-if-dialer0)#ipv6 enable
Device1(config-if-dialer0)#ipv6 address 172:80::2/64
Device1(config-if-dialer0)#exit
```

#Add gigabitethernet0 interface to dialing pool1.

```
Device1(config)#interface gigabitethernet0
Device1(config-if-gigabitethernet0)#pppoe-client dial-pool-number 1
Device1(config-if-gigabitethernet0)#exit
```

Configure the IP6 default route of one egress interface dialer0.

```
Device1(config)#ipv6 route ::/0 dialer0
```

Step 2: Configure the PPPoE server.

#Create virtual template virtual-template0, and configure the IPv6 address.

```
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#ipv6 enable
Device2(config-if-virtual-template0)#ipv6 address 172:80::1/64
Device2(config-if-virtual-template0)#exit
```

#Create virtual VPDN group group1, reference virtual-template0, and enable PPPoE server on the interface.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group group1
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol pppoe
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#pppoe enable
Device2(config-if-gigabitethernet0)#exit
```

Step 3: On Device1, ping 1:: 1 to trigger PPPoE dialing and check the result.

#View the dialer0 interface information of Device1.

```
Device1#show ipv6 interface dialer 0
dialer0 is up
  VRF: global
  IPv6 is enable, link-local address is fe80::201:7aff:fe7c:7179
  Global unicast address(es):
    172:80::2, subnet is 172:80::/64
  Joined group address(es):
    ff02::1:ff00:2
    ff02::1:ff00:0
    ff02::2
    ff02::1
    ff02::1:ff7c:7179
  ND control flags: 0x1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ICMP unreachables are enabled
```

```
ND DAD is enabled, number of DAD attempts: 1
ND reachable time is 30000 milliseconds
ND advertised reachable time is 0 (unspecified)
ND advertised retransmit interval is 0 (unspecified)
ND config flags is 0x0
ND MaxRtrAdvInterval is 600
ND MinRtrAdvInterval is 198
ND AdvDefaultLifetime is 1800
```

You can see that the dialer0 interface protocol of Device1 is up.

#View the PPPoE session information of Device1.

```
Device1#show pppoe session information

PPPoE Session Information:(Max Sessions=1024)
UID  SID  RemMAC        LocMAC        O-Intf              O-VA            state C/S
ActiveTime Local-IP      Peer-IP
 ----- ----- -------------- --------------  -------------------- ---------------
--- ----- --- ---------- --------------- ----------------
1    1    001f.ce22.39ed  001f.ce7f.9b51 gigabitethernet0  virtual-access3001 UP   C
00:01:31  0.0.0.0        0.0.0.0
There are 1 PPPoE sessions up
```

It can be observed that PPPoE session is successfully established between device1 and device2.

#On Device1, you can ping the Ipv6 address of the virtual-template0 interface on Device2.

```
Device1#ping 172:80::1

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 172:80::1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 1/1/2 ms.
```

6.3.5. Configure PPPoE to Assign to IPv6 Address via DHCPv6 Protocol

Network Requirement

- Device1 is the PPPoE client and Device2 is the PPPoE server.
- Device2 allocates IPv6 address to Device1 through DHCPv6 protocol to realize IPv6 communication between Device1 and Device2.

Network Topology

Figure 6-5 Networking of configuring PPPoE to assign the IPv6 address via DHCPv6 protocol

Configuration Steps

Step 1: Configure the PPPoE client.

#Configure Device1, permitting the IPv6 protocol to trigger dialing.

```
Device1#configure terminal
Device1(config)#dialer-list 1 protocol ipv6 permit
```

#Create dialer0 interface, configure the dialing information, and configure the DHCPv6 client.

```
Device1(config)#interface dialer0
Device1(config-if-dialer0)#dialer in-band
Device1(config-if-dialer0)#dialer pool 1
Device1(config-if-dialer0)#dialer-group 1
Device1(config-if-dialer0)#ipv6 enable
Device1(config-if-dialer0)#ipv6 dhcp client address
Device1(config-if-dialer0)#exit
```

#Add the interface gigabitethernet0 to dialing pool 1.

```
Device1(config)#interface gigabitethernet0
Device1(config-if-gigabitethernet0)#pppoe-client dial-pool-number 1
Device1(config-if-gigabitethernet0)#exit
```

Configure the IPv6 default route of one egress interface dialer0.

```
Device1(config)#ipv6 route ::/0 dialer0
```

Step 2: Configure the PPPoE server.

#Configure the DHCPv6 address pool named pool1.

```
Device2#configure terminal
Device2(config)#ipv6 dhcp pool pool1
Device2(dhcp6-config)#range 172:80::2 172:80::100 64
Device2(dhcp6-config)#exit
```

#Create virtual template virtual-template0, configure the IPv6 address, and configure the DHCPv6 server.

```
Device2(config)#interface virtual-template 0
Device2(config-if-virtual-template0)#encapsulation ppp
Device2(config-if-virtual-template0)#ipv6 enable
Device2(config-if-virtual-template0)#ipv6 address 172:80::1/64
```

```
Device2(config-if-virtual-template0)#ipv6 dhcp server
Device2(config-if-virtual-template0)#exit
```

#Create virtual VPDN group group1, reference virtual template virtual-template0, and enable the PPPoE server on the interface.

```
Device2(config)#vpdn enable
Device2(config)#vpdn-group group1
Device2(config-vpdn)#accept-dialin
Device2(config-vpdn-acc-in)#protocol pppoe
Device2(config-vpdn-acc-in)#virtual-template 0
Device2(config-vpdn)#exit
Device2(config)#interface gigabitethernet0
Device2(config-if-gigabitethernet0)#pppoe enable
Device2(config-if-gigabitethernet0)#exit
```

Syep 3: On Device1, ping any address 1::1 to trigger PPPoE dialing, and check the result.

#View the dialer0 interface information of Device1.

```
Device1#show ipv6 interface dialer 0
dialer0 is up
  VRF: global
  IPv6 is enable, link-local address is fe80::201:7aff:fe7c:7179
  Global unicast address(es):
    172:80::2, subnet is 172:80::/64 [DHCP/PRE]
  Joined group address(es):
    ff02::1:ff00:2
    ff02::1:ff00:0
    ff02::2
    ff02::1
    ff02::1:ff7c:7179
  ND control flags: 0x1
  MTU is 1500 bytes
  ICMP redirects are enabled
  ICMP unreachables are enabled
  ND DAD is enabled, number of DAD attempts: 1
  ND reachable time is 30000 milliseconds
  ND advertised reachable time is 0 (unspecified)
  ND advertised retransmit interval is 0 (unspecified)
  ND config flags is 0x0
  ND MaxRtrAdvInterval is 600
```

```
ND MinRtrAdvInterval is 198
ND AdvDefaultLifetime is 1800
```

We can observe that the dialer0 interface protocol of Device1 is up, which can be assigned with the IPv6 address from the server.

#View the PPPoE session information of Device1.

```
Device1#show pppoe session information

PPPoE Session Information:(Max Sessions=1024
UID  SID  RemMAC        LocMAC        O-Intf              O-VA            state C/S
ActiveTime Local-IP      Peer-IP
 ----- ----- -------------- --------------  ------------------- ---------------
--- ----- --- ---------- --------------- ----------------
1    1    001f.ce22.39ed  001f.ce7f.9b51 gigabitethernet0  virtual-access3001 UP   C
00:01:31  0.0.0.0        0.0.0.0
There are 1 PPPoE sessions up
```

It can be observed that PPPoE session is successfully established between Device1 and Device2.

#On Device1, you can ping the IPv6 address of the virtual template 0 interface on Device2.

```
Device1#ping 172:80::1

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 172:80::1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 1/1/2 ms.
```

Step 4: On Device2, view the DHCPv6 address pool information of the server and check the result.

#On Deice2, view the lease information dynamcially assigned in the DHCPv6 address pool.

```
Device2#show ipv6 dhcp pool pool1 lease
IPv6 Address                              Duid                                  Iaid     Type
Time Left(s)
---------------------------------------------- ----------------------------
---------------------- -------- ------- ------------
172:80::2                                 000200001613303030313761376337313739
20180001  Lease    2591895
```

You can observe the lease information of the IPv6 address dynamically allocated in DHCPv6 address pool of Device2.

7. DDR

7.1. Overview

7.1.1. Overview of DDR

DDR (Dial-On-Demand Routing) is the network technology adopted when the devices are interconnected via the public switching network. DDR means that the devices connected across the public switching network do not set up the connection in advance, but when there is data needed to be transmitted between them, set up the connection via the dialing mode. Enable the DDR dialing process to set up the connection and transmit the information. When the link is idle again, DDR automatically cuts off the connection. This does not affect the user communication and can reduce the communication cost of the user.

7.1.2. DDR Terms

Dial prototype: The dialing prototype separates the logical parts of DDR, including the parameters related with the network layer, encapsulation and dialing, from the physical interface responsible for receiving and sending the call. In the dialing prototype, bind and combine the physical interface and logical interface by each call so that the physical interface can dynamically select different parameters according to the incoming or outgoing call.

Physical interface: The actual existing physical interface, such as serial port and 3G interface. We can use the **dialer pool-member** command to associate the physical interface with the dialer pool. One physical interface can be associated with multiple dialer pools.

Dialer interface: It is the physical interface set for configuring the DDR parameters. The physical interface can be bound to the Dialer interface to inherit the configuration information.

Dialing interface: It is the general term for the dialing connection interface. It can be Dialer interface or the physical interface bound to the Dialer interface, or the physical interface directly configuring the DDR parameters.

Dialer pool: The dialer interface usually references one dialer pool. The dialer pool is a group of one or multiple interfaces associated with the dialing prototype.

7.2. DDR Function Configuration

Table 7-1 DDR function configuration list

Configuration Task	
Configure the DDR dialing control list	Create the dialing control list
Configure the DDR basic functions	Enable the DDR function
	Associate the dialing control list with the dialing interface
	Associate the dialing set with the dialing interface

Configuration Task	
Configure the DDR interface parameters	Configure the idle time of the link
	Configure the waiting time for re-dialing
	Configure the longest time of waiting for the carrier

7.2.1. Configure DDR Dialing Control List

Configuration Condition

None

Create Dialing Control List

We can filter the packets passing the dialing interface by configuring the dialing control list. According to whether the packet complies with the permit or deny condition of the dialing ACL control list, the packets are divided to two kinds:

1. For the packet complying with the permit condition of the dialing ACL control list or not complying with the deny condition of the dialing ACL control list, if the corresponding link is set up, DDR sends out the packet via the link and clears up the Idle timeout timer; if the link is not set up, send out new call.
2. For the packet not complying with the permit condition of the dialing ACL control list or complying with the deny condition of the dialing ACL control list, if the corresponding link is set up, DDR sends out the packet via the link, but does not clear up the idle timeout timer; if the corresponding link is not set up, do not send out call, but drop the packet.

Table 7-2 Create the dialing control list

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the dialing control list	**dialer-list** *dialer-list-number* **protocol** { **ip** | **ipv6** } { **deny** | **list** *access-list* **| permit** }	Mandatory By default, do not configure the dialing control list. The dialing control list can directly configure the filter condition of the packet and also can bring in the rules of the ACL.

7.2.2. Configure DDR Basic Functions

Configuration Condition

None

Enable DDR Function

DDR supports the data flow trigger dialing and auto dialing. The common used is the data flow trigger dialing.

Table 7-3 Enable the DDR function

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
The interface enables the DDR function	**dialer in-band**	Mandatory By default, the interface does not enable DDR.

Configure Dialing Control List to Associate with Dialing Interface

Associate one dialing interface with the dialing ACL to control the access. Use the following command to configure on the interface:

Table 7-4 Configure the dialing ACL to associate with the dialing interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the dialing ACL to associate with the dialing interface	**dialer-group** *group-number*	Mandatory By default, do not configure the dialing ACL to associate with the dialing interface. Before configuring the command, the interface should enable the DDR function.

Step	Command	Description
		Auto dialing does not need to configure the command.

Configure Dialing Pool to Associate with Dialing Interface

For the dialing interface, to specify which dialing pool to connect the specified destination subnet, use the following command to configure.

Table 7-5 Configure the dialing pool to associate with the dialing interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the dialing pool to associate with the dialing interface	**dialer pool** *pool-number*	Mandatory By default, do not configure the dialing pool to associate with the dialing interface. The command can only be used in the dialer interface; auto dialing does not need to configure the command.

7.2.3. Configure DDR Interface Parameters

Configuration Condition

None

Configure Link Idle Time

After one link is set up, the link idle time configured by the user takes effect. After the idle time of the link exceeds the specified time, DDR disconnects the link.

Table 7-6 Configure the idle time of the link

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the idle time of the link	**dialer idle-timeout** *seconds* [{ **either** | **inbound** }]	Optional By default, the idle time of the link is 120s.

Configure Wait Time for Re-dialing

To set the wait time of the DDR interface before re-dialing after the dialing ends or fails, use the command to configure on the interface. If the dialing link is unstable, we need to configure the wait time for re-dialing, so as to prevent the unnecessary overload.

Table 7-7 Configure the wait time for re-dialing

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the wait time for re-dialing	**dialer enable-timeout** *seconds*	Optional By default, the DDR re-dialing wait time is 15s.

Configure Maximum Time of Waiting for Carrier

To control the wait time permitted between dialing initiating and the connection setup, configure the maximum time of waiting for the carrier. If exceeding the time, the dialing is still not set up, DDR ends the dialing.

Table 7-8 Configure the maximum time of waiting for carrier

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface dialer** *dialer-id*	-
Configure the maximum time of waiting for carrier	**dialer wait-for-carrier-time** *seconds*	Optional By default, the maximum time of waiting for carrier is 60s.

7.2.4. DDR Monitoring and Maintaining

Table 7-9 DDR monitoring and maintaining

Command	Description
clear dialer [**interface dialer** *dialer-id*]	Clear up the DDR statistics information
show dialer [{ **interface dialer** *dialer-id* | **maps** | **statement** }]	Display the DDR information

8. NDIS-DIAL

8.1. Overview

NDIS-DIAL (Network Driver Interface Standard-Dial) indicates the dialing function especially applied on the 4G interface, mainly completing the dialing actions, such as trigger dialing, hang up link, and disconnect and re-dial, and providing the perfect link status detection mechanism (Track detection and BFD detection). When the link fails, automatically disconnect and re-dial, trying to restore the data communication automatically. Compared with the 3G DDR dialing, the NDIS-DIAL function is simpler and more reliable.

8.2. NDIS-DIAL Function Configuration

Table 8-1NDIS-DIAL function configuration list

Configuration Task	
Configure NDIS-DIAL basic function	Configure NDIS-DIAL to auto dial
	Configure NDIS-DIAL as dial-on-demand
	Configure NDIS-DIAL link idle time
	Configure NDIS-DIAL re-dial waiting time
Configure 4G to associate with Track	Configure the 4G interface to associate with Track
Configure 4G to associate with BFD	Configure the 4G interface to associate with BFD
	Configure the BFD detection parameters

8.2.1. Configure NDIS-DIAL Basic Function

Configuration Condition

No

Configure NDIS-DIAL to Auto Dialing

After enabling the auto dialing function, the interface will always try to dial, so try to ensure that the 4G interface is available.

Table 8-2 Configure NDIS-DIAL to auto dialing

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure NDIS-DIAL to auto dialing	**dialer mode auto**	Mandatory By default, the interface does not configure the NDIS-DIAL dialing mode.

Configure NDIS-DIAL to Dial on Demand

After enabling the dial-on-demand function on the 4G interface, trigger dialing when there is some data flow sent. You can select any IP data flow or the data flow meeting the ACL rule to configure the dial-on-demand function.

Table 8-3 Configure NDIS-DIAL to dial on demand

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure NDIS-DIAL to dial on demand	**dialer-group ip [any | access-list [***access-list-name* **|** *access-list-number***]]**	Mandatory By default, do not configure the NDIS-DIAL dial-on-demand mode on the interface.

Configure Idle Time of NDIS-DIAL Link

If the 4G interface has no data flow for a period of time, it will actively hang up the line when the idle time exceeds the configured link idle time. The link idle time is only effective in the dial-on-demand mode, and it is divided into 3 kinds: the 4G interface sends no idle time of the data link at the sending direction, the 4G interface has no idle time of the data link at the receiving direction, the 4G interface has no idle time of the data link at the sending and receiving directions.

Table 8-4 Configure the idle time of the NDIS-DIAL link

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the idle time of the NDIS-DIAL link	**dialer idle-timeout** *time* **[either | inbound]**	Optional In the dial-on-demand mode, the link idle time of the sending direction is 120s by default.

Note:

- It is suggested not to configure the link idle time in the auto dialing mode.

Configure NDIS-DIAL Redial Interval

Configure the interval before starting the next dialing after the 4G interface hangs up.

Table 8-5 Configure the NDIS-DIAL re-dial interval

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the NDIS-DIAL re-dial interval	**dialer interval-time** *time*	Optional By default, the re-dial interval is 10s.

Note:

- It is suggested that the re-dial interval is no less than 10s.

8.2.2. Configure 4G to Associate with Track

Configuration Condition

Before configuring 4G to associate with Track, first complete the following tasks:

- Create Track group
- Configure Track monitor object

- Configure the dialing mode of 4G interface

Configure 4G Interface to Associate with Track

Associate the 4G interface with the Track object. According to the reported status of the Track object, determine whether to disconnect the 4G interface and re-dial, and try to restore the data communication automatically.

Table 8-6 Configure 4G interface to associate with Track

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure 4G interface to associate with Track	**dialer track id** *id*	Mandatory By default, do not configure the 4G interface to associate with Track.

Note:

- For Track configuration, refer to Track configuration manual.

8.2.3. Configure 4G to Associate with BFD

Configuration Condition

Before configuring 4G to associate with BFD, first complete the following task:

- Configure the interface dialing mode

Configure 4G Interface to Associate with BFD

The BFD (Bidirectional Forwarding Detection) protocol is one standard unified detection mechanism, used to fast detect and monitor the path in the network or the connection status of the IP route forwarding. Associating the 4G interface with BFD can fast detect the link fault between the 4G interface and upper end.

Table 8-7 Configure 4G interface to associate with BFD

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the 4G interface to associate with BFD	**dialer bfd remote-ip** *ipaddress* [**local-ip** *ipaddress*]	Mandatory By default, do not configure the 4G interface to associate with BFD.

Note:

- Usually, it is not suggested to configure **local-ip**, and NDIS-DIAL will use the interface address as the source address.

Configure BFD Detection Parameter

The BFD detection parameter is used to control the frequency of the 4G interface sending and receiving the BFD detection packets.

Table 8-6 Configure the BFD detection parameter

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the interface configuration mode	**interface** *interface-name*	-
Configure the BFD detection parameter	**dialer bfd** { **min-receive-interval** *value* | **min-transmit-interval** *value* | **multiplier** *value* }	Optional By default, the value of **min-transmit-interval** is 10000, the value of **min-transmit-interval** is 10000, and the value of **multiplier** is 3.

Note:

- For the meaning of **min-receive-interval**, **min-transmit-interval**, and **multiplier**, refer to the BFD command manual.

8.2.4. NDIS-DIAL Monitoring and Maintaining

Table 8-8 NDIS-DIAL monitoring and maintaining

Command	Description
clear dialer mode [*interface-name*]	Clear the NDIS-DIAL information
show dialer mode [*interface-name*]	Display the NDIS-DIAL information
[no] debug ndis [global| *interface-name***[mode | packet]]**	Display the NDIS-DIAL debug information

9. QINQ TERMINATION

9.1. Overview

QinQ termination identifies the two layers of VLAN of the packet, resolves the two layers of VLAN, and then performs the subsequent forwarding. The daughter interface of QinQ termination is similar to the common VLAN daughter interface. The common VLAN daughter interface identifies and terminates the single layer of VLAN. The daughter interface of QinQ termination identifies and terminates the two layers of VLAN. The inner and outer layers of VLAN IDs of the QinQ packet that can be received by the QinQ termination daughter interface should be the configured value. When sending the packet, add two layers of VLAN to the packet, fill in the inner and outer VLAN ID fields with the configured values.

9.2. QinQ Termination Function Configuration

Table 9-1QinQ termination function configuration list

Configuration Task	
The daughter interface encapsulates the QinQ termination	Configure the daughter interface to encapsulate the QinQ termination
Configure the IP priority copy	Configure the IP priority copy function of the daughter interface
Configure the VLAN TPID (Tag Protocol Identifier)	Configure the TPID value of the outer VLAN
	Configure the TPID value of the inner VLAN

9.2.1. Daughter Interface Encapsulating QinQ Termination

After the Ethernet interface configures encapsulating QinQ termination, it can process the 802.1Q packet with two layers of VLAN TAG.

Configuration Condition

None

Daughter Interface Encapsulating QinQ Termination

Table 9-2 Daughter interface encapsulating QinQ termination

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the daughter interface configuration mode	**interface** *interface-name*	-
Encapsulate the QinQ termination	**encapsulation dot1q** *vid* **second-dot1q** *in-vid*	Mandatory By default, the Ethernet sub interface is not encapsulated with QinQ termination. Vid is the outer VLAN ID and the value range is 1-4094. in-vid is the inner VLAN ID and the value range is 1-4094.

9.2.2. Configure IP Priority Copy

The copy function of the IP priority is copy the TOS priority of the IP head to the priority of the outer VLAN, taking effect at the egress direction. Currently, it only supports the IPv4 packet.

Configuration Condition

Before configuring the IP priority, first complete the following task:

- Ethernet daughter interface is configured with the QinQ termination

Configure IP Priority Copy

Table 9-3 Configure the IP priority copy

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the daughter interface configuration mode	**interface** *interface-name*	-

Step	Command	Description
Configure the IP priority copy	**dot1q ip priority inherit**	Mandatory By default, the Ethernet daughter interface is not configured with the IP priority copy.

Note:

- When the IP priority copy function is used with QoS and if configuring QoS to change the IP priority, the IP priority copy function cannot change the VLAN priority and you can configure in the QoS to change the VLAN priority.

9.2.3. Configure VLAN TPID

Ethernet frame VLAN contains four fields, that is, TPID (Tag Protocol Identifier), User Priority, CFI, and VLAN ID.

For the TPID, adopt the 0x8100 defined by the IEEE802.1Q protocol, while the devices of some manufacturer sets the TPID value of the outer VLAN of the packet to 0x9100 or other value. To be compatible with the devices, it is necessary to provide the TPID configurable function. The user can configure the TPID value of the port by self. When the port forwards the packet, replace the IP ID value in the outer VLAN of the packet with the set value of the user, so as to intercommunicate with the device of the third-party manufacturer.

Because the location of the TPID field in the Ethernet packet is the same as the location of the protocol type field in the packet without VLAN, to avoid causing the confusion of packet receiving and forwarding in the network, do not permit the user to configure the TPID value to the common protocol type value listed in Table 5-4.

Table 9-4 Common protocol type of Ethernet link

ARP	0x0806
IP	0x0800
PUP	0x0200
MPLS	0x8847/0x8848
IS-IS	0x8000
802.1x	0x888E
RARP	0x0835

ARP	0x0806
IPv6	0x86DD
PPPoE	0x8863/0x8864
IPX/SPX	0x8137
LACP	0x8809
CLUSTER	0x88A7

Configuration Condition

None

Configure VLAN TPID

Table 9-5 Configure VLAN TPID

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the main interface configuration mode	**interface** *interface-name*	-
Configure the VLAN TPID	**dot1q tunneling ethertype** *tpid*	Mandatory Configure the TPID of the outer VLAN. By default, the TPID of the outer VLAN is 0x8100.
	dot1q tunneling second-ethertype *tpid*	Mandatory Configure the TPID of the inner VLAN. By default, the TPID of the inner VLAN is 0x8100.

9.3. Typical Configuration Example of QinQ Termination

9.3.1. Configure QinQ Termination

Network Requirements

- On Device1 and Device2, four daughter interfaces are interconnected via the carrier network; dot1q is 100, 200, 300, and 400 respectively.
- Device1 and Device2 use VLAN1000 via the public switch network; the device TPID is 0x9100.
- Make the 802.1p value of the outer VLAN of the packet be consistent with the IP priority of the packet.
- The Gi0.1-Gi0.4 daughter interfaces on Device1 bear service A, B, C, D respectively. On Device1, configure QoS, the A, B. C and D service traffic is totally shaped to 15M, and is scheduled via the CBWFQ policy serv in a unified manner; the D service traffic is shaped to 5M.

Network Topology

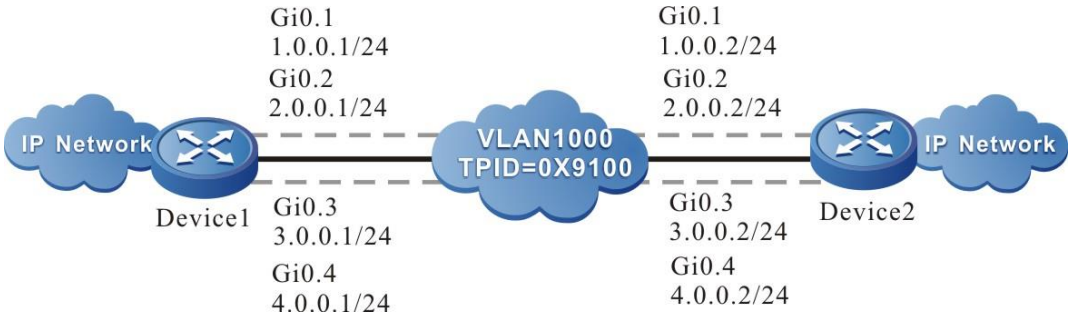

Figure 9-1 Networking of configuring QinQ termination

Configuration Steps

Step 1: Configure the IP address of the interface, configure the route and so on. On Device1, configure CBWFQ policy serv (omitted).

Step 2: Configure the QinQ termination interface on Device1 and Device2.

#The daughter interface of Device1 is the QinQ termination interface, the outer VLAN ID of the packet is 1000, and the inner VLAN ID is 100, 200, 300, 400 respectively.

```
Device1#configure terminal
Device1(config)#interface gigabitethernet 0.1
Device1(config-if-gigabitethernet0.1)#encapsulation dot1q 1000 second-dot1q 100
Device1(config-if-gigabitethernet0.1)#exit
Device1(config)#interface gigabitethernet 0.2
Device1(config-if-gigabitethernet0.2)#encapsulation dot1q 1000 second-dot1q 200
Device1(config-if-gigabitethernet0.2)#exit
Device1(config)#interface gigabitethernet 0.3
Device1(config-if-gigabitethernet0.3)#encapsulation dot1q 1000 second-dot1q 300
Device1(config-if-gigabitethernet0.3)#exit
Device1(config)#interface gigabitethernet 0.4
```

```
Device1(config-if-gigabitethernet0.4)#encapsulation dot1q 1000 second-dot1q 400
Device1(config-if-gigabitethernet0.4)#exit
```

#Configure the outer VLAN TAG TPID of the QinQ termination interface packet on the main interface of Device1 as 0x9100.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)#dot1q tunneling ethertype 0x9100
Device1(config-if-gigabitethernet0)#exit
```

#Configure the QinQ termination interface packet of Device1 to copy the IP priority to the outer VLAN TAG priority.

```
Device1(config)#interface group 1 range gigabitethernet 0.1 gigabitethernet 0.4
Device1(config-if-group1)#dot1q ip priority inherit
Device1(config-if-group1)#exit
```

The configuration of Device2 QinQ termination interface is the same as Device1.

Step 3: Configure QoS daughter interface re-direction on Device1 and apply QoS.

#Shape the D service traffic to 5M.

```
Device1(config)#interface gigabitethernet 0.4
Device1(config-if-gigabitethernet0.4)#traffic-shape 5000000 125000
Device1(config-if-gigabitethernet0.4)#exit
```

#Configure the QoS daughter interface to re-direct to the daughter interface; the traffic of B, C service Gi0.2 and Gi0.3 is re-directed to the QoS channel of A service Gi0.1.

```
Device1(config)#qos sub-interface redirect
Device1(config-qosredirect)#redirect gigabitethernet 0.2 gigabitethernet 0.1
Device1(config-qosredirect)#redirect gigabitethernet 0.3 gigabitethernet 0.1
Device1(config-qosredirect)#exit
```

#On Gi1.1, apply the traffic shape and the configured CBWFQ policy serv.

```
Device1(config)#interface gigabitethernet 0.1
Device1(config-if-gigabitethernet0.1)#traffic-shape 15000000 375000
Device1(config-if-gigabitethernet0.1)#service-policy output serv
Device1(config-if-gigabitethernet0.1)#exit
```

Step 4: Check the result.

#Gi0.1 of Device1 can ping the peer QinQ termination daughter interface.

```
Device1#ping 1.0.0.2
Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 1.0.0.2 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 16/18/20 ms.
```

The check method of other interface is the same.

#View the statistics information of the CBWFQ queue.

```
Device1#show policy-map interface gigabitethernet 0.1
 interface gigabitethernet0.1
  Service-policy output: serv

    Class-map: A-serv (match-all)
      26338 packets 11721879 bytes
      5 minute offered rate 8525000 bps
      match access-group A-serv
      Queueing
      queue limit 1024 packets
      (queue depth/total drops) 0/0
      (packets output/bytes output) 26336/11721098
      Bandwidth: 30% (4500 Kbps)

    Class-map: B-serv (match-all)
      13083 packets 6646164 bytes
      5 minute offered rate 6646160 bps
      match access-group B-serv
      Queueing
      queue limit 256 packets
      (queue depth/total drops) 0/0
      (packets output/bytes output) 0/0
      Priority: 30% (4500 Kbps)  , burst bytes 50000, b/w exceed drops: 0

    Class-map: C-serv (match-all)
      1383 packets 61064 bytes
      5 minute offered rate 2346160 bps
      match access-group C-serv
      Queueing
      queue limit 256 packets
      (queue depth/total drops) 0/0
      (packets output/bytes output) 0/0
      Priority: 20% (3000 Kbps)  , burst bytes 40000, b/w exceed drops: 0

    Class-map: class-default (match-any)
```

0 packets 0 bytes

5 minute offered rate 0 bps

match any

Queueing

queue limit 256 packets

(queue depth/total drops) 0/0

(packets output/bytes output) 0/0

Fair-queue 256: per-flow queue limit 64 Packets

10. ETHERNET LINK

10.1. Overview

Ethernet is one computer LAN networking technology. The IEEE 802.3 standard made by IEEE provides the Ethernet technology standard. It defines the contents of the connection of the physical layer, electrical signal and media access slayer protocol. Ethernet is currently the most common LAN technology.

10.2. Ethernet Link Function Configuration

Table 10-1 Ethernet link function configuration list

Configuration task	
Sub interface encapsulates VLAN	Configure the Ethernet sub interface to encapsulate VLAN
Configure the small packet software filling	Configure the small packet software filling function of the Ethernet sub interface

10.2.1. Sub Interface Encapsulating VLAN

After the Ethernet sub interface encapsulates VLAN, it can process the 802.1Q packet with VLAN TAG.

Configuration Condition

None

Sub Interface Encapsulating VLAN

Table 10-2 The sub interface encapsulating VLAN

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the sub interface configuration mode	**interface** *interface-name*	-
Encapsulate VLAN	**encapsulation dot1q** *vid*	Mandatory By default, the Ethernet sub interface does not encapsulate VLAN. The value range of vid is 1-4094.

10.2.2. Configure Small Packet Software Filling

After configuring the small packet software filling function, you can ensure that the length of the sent packet reaches the minimum length of the Ethernet packet 64 bytes.

Configuration Condition

None

Configure Small Packet Software Filling

Table 10-3 Configure the small packet software filling

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter the sub interface configuration mode	**interface** *interface-name*	-
Configure the small packet software filling	**padding-soft**	Mandatory By default, the Ethernet sub interface does not configure the small packet software filling function.

According to the above information, we can see the basic configuration information, the converted absolute value of the bandwidth guarantee, the statistics of each type of packets, 5-min traffic statistics, as well as the stacking of the current QoS queue packets, and simply judge whether the configuration is correct and effective.

11. ETHERNET LINK AGGREGATION

11.1. Overview

Link aggregation aggregates multiple physical Ethernet links between two devices to form a logical link to extend the link bandwidth. The physical link in the logical link are redundant and dynamically back up for each other, providing higher network connection reliability. The routing device only supports L3 Ethernet link aggregation interface type (Route Aggregation).

11.2. Basic Concepts

Aggregation Group and Member Interface

The combination formed by binding multiple physical interfaces together is called aggregation group. The logical interfaces corresponding to aggregation group become aggregation group interfaces, and these bound physical interfaces are called member interfaces of aggregation group.

Status of Member Interface

The member interfaces of the aggregation group have the following two states:

- Selected load state: The interface in this state can participate in user service traffic forwarding, and the member interface in this state is referred to as "selected load interface";
- Unselected load state: The interface in this state cannot participate in user traffic forwarding, and the member interface in this state is referred to as "unselected load interface".

The bandwidth, rate, and duplex mode of an aggregation group depend on the selected interfaces within the aggregation group. The bandwidth of the aggregation group interface is equal to the sum of the bandwidths of all selected interfaces, the rate of the aggregation group is the same as that of all selected interfaces, and the duplex mode of the aggregation group is the same as that of the selected interfaces.

Operation key

Operation key is the attribute configuration of member interface, which is composed of rate, duplex mode and management key (i.e. aggregation group number). In attribute configuration, the change of duplex mode or rate will cause the re-calculation of the operation key.

In the same aggregation group, if the duplex modes or rates of member interfaces are different, the generated operation keys must be different, but the selected member interfaces must have the same operation key.

LAC Protocol

LACP (Link Aggregation Control Protocol) is a protocol based on IEEE802.3ad standard. LACP communicates with the peer through LACPDU (Link Aggregation Control Protocol Data Unit).

LACP Priority

LACP priority is divided into system priority and port member priority

- LACP system priority: used to distinguish the LACP priorities of two-end devices;
- LACP member priority: used to determine the priority of the member interface of the local device.

System ID

System ID: the aggregation attribute of the device, which is composed of the system LACP priority and the system MAC address of the device. The higher the priority of LACP, the better the system ID of the device. Under the same LACP priority, the smaller the MAC address is, the better the system ID is.

Member ID

Member ID: Aggregation attribute of the interface.

In static aggregation mode, the member ID consists of LACP priority, interface rate, duplex mode and interface number. The higher the priority of LACP, the better the member ID. In the case of the same LACP priority, the higher the member interface speed is, the better the member ID is. The full duplex mode of member interface is better than the half duplex mode of member interface when the speed of member interface is the same. Under the same member duplex mode, the smaller the member interface number is, the better the member ID is.

In dynamic aggregation mode, the member ID consists of LACP priority and interface number. The higher LACP priority is, the better the member ID is. In the case of the same LACP priority, the smaller the member interface number is, the better the member ID is.

Root Port of Aggregation Group

All the protocols used in aggregation group receive and send protocol packets through the root port of aggregation group. The root port of the aggregation group is selected from the aggregation group member interface. The physical link state of the root port must be up.

11.2.1. Link Aggregation Mode

Link aggregation mode is divided into static aggregation mode and dynamic aggregation mode. The aggregation group types are divided into static aggregation group and dynamic aggregation group.

Static Aggregation Mode

In static aggregation mode, the LACP protocol of the device member interfaces at both ends is disabled. In the static aggregation group, the local device sets the selected state of the member interface according to the following principles:

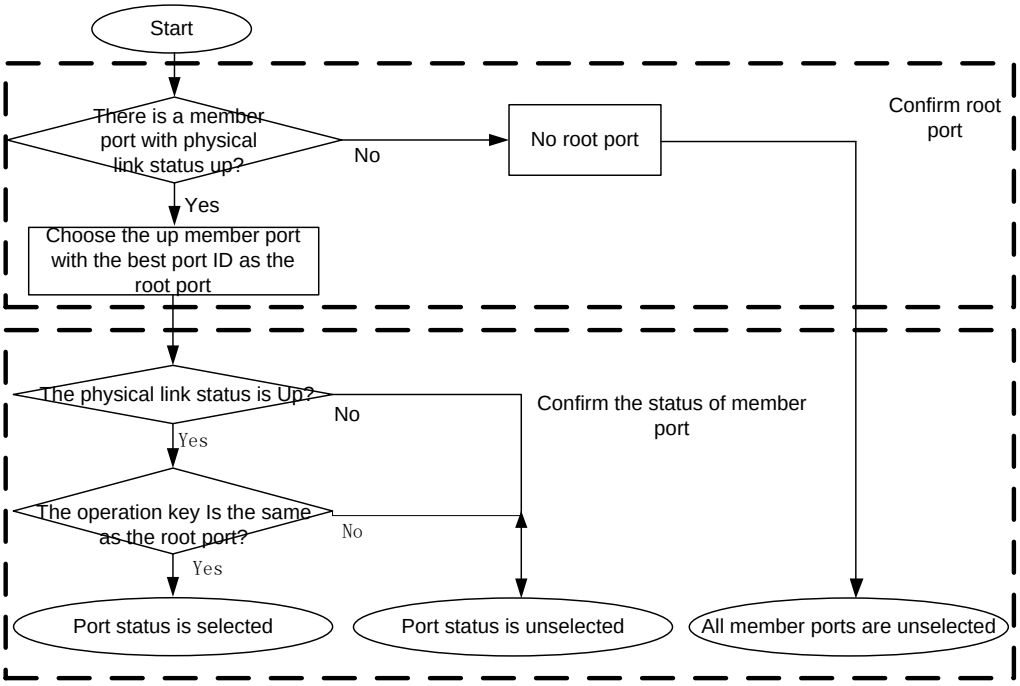

Figure 11-1 The principles of selecting the member interface in the static aggregation mode

Dynamic Aggregation Mode

In dynamic aggregation mode, the interface can join the dynamic aggregation group in two modes (Active or Passive).

- The duplex mode of the interface is the full-duplex:

Join the dynamic aggregation group in active mode, and the LACP protocol of the interface is enabled;

After joining dynamic aggregation group in passive mode, the LACP protocol is disabled. When the LACPDU packet is received from peer interface, the LACP protocol becomes enabled.

- The duplex mode of the interface is half duplex. No matter how the interface joins the dynamic aggregation group, the LACP protocol of the interface is closed.

In dynamic aggregation group, the system sets the selected state of member interface according to the following principles:

Determine the device with better system ID, and the device determines the state of the member interface at both ends. The device with better system ID can set the selected state of member interface according to the following principles:

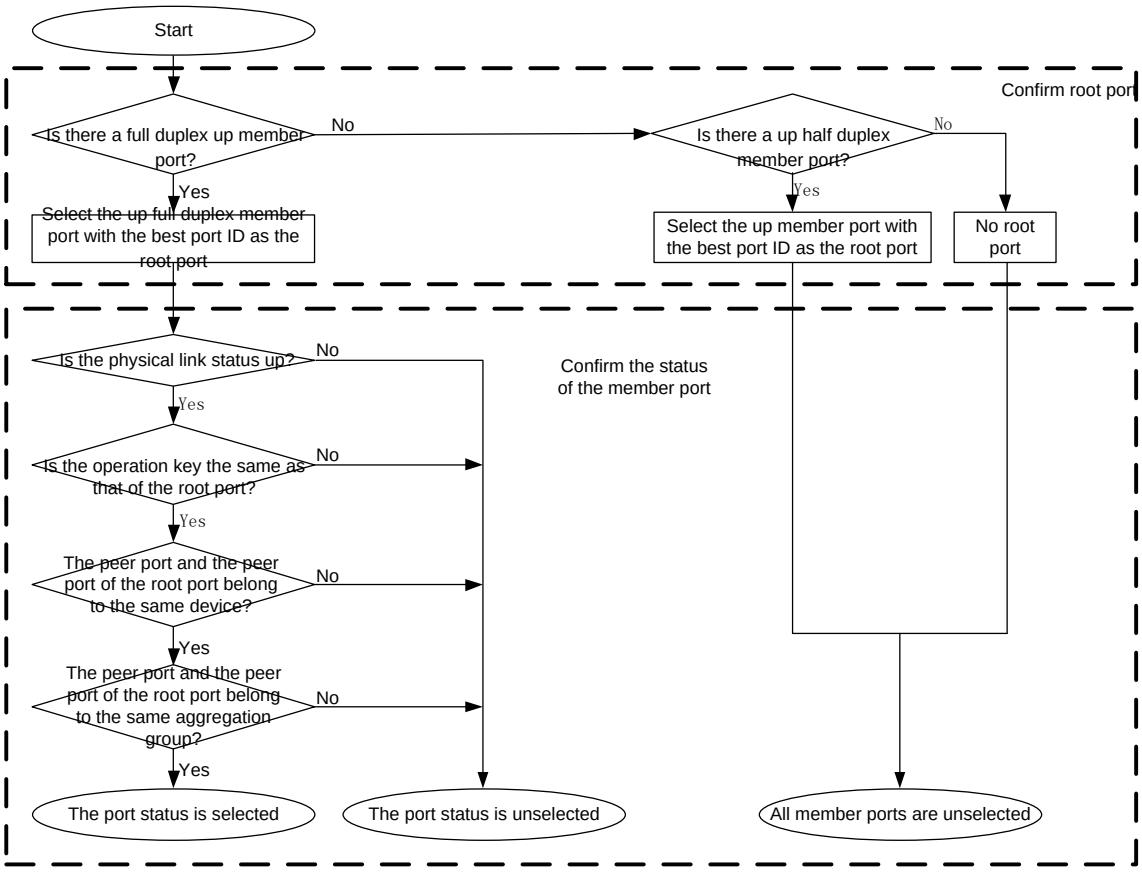

Figure 11-2 The principles of selecting the member interface in the dynamic aggregation mode

11.3. Ethernet Link Aggregation Function Configuration

Table 11-1 Ethernet link aggregation function configuration list

Configuration Task	
Configure the basic functions of the Ethernet link aggregation group	Configure the basic functions of the Ethernet link aggregation group
Configure the MAC address of the Ethernet link aggregation group interface	Configure the MAC address of the Ethernet link aggregation group interface
Configure the load mode of the Ethernet link aggregation group	Configure the load mode of the Ethernet link aggregation group
Configure the system priority of the Ethernet link aggregation	Configure the LACP protocol system priority of the Ethernet link aggregation group
Configure the member priority of the Ethernet link aggregation	Configure the LACP protocol member priority of the Ethernet link aggregation group

Configuration Task	
Configure the timeout period of the Ethernet link aggregation member	Configure the LACP protocol member timeout period of the Ethernet link aggregation group
Configure the independent port compatible mode of the Ethernet link aggregation group	Configure the independent port compatible mode of the Ethernet link aggregation group
Configure the Ethernet link aggregation group to associate with BFD	Configure the Ethernet link aggregation group to associate with BFD
Configure the member bandwidth weight load of the Ethernet link aggregation group	Configure the Ethernet link aggregation group member to forward the traffic according to the configured bandwidth width proportional load

11.3.1. Configure Basic Functions of Ethernet Link Aggregation

Configuration Condition

None

Configure Basic Functions of Ethernet Link Aggregation

There are two steps to configure the basic functions of Ethernet link aggregation: one is to create an Ethernet link aggregation group interface and specify the aggregation group mode; the other is to add the physical Ethernet interface to the specified corresponding Ethernet link aggregation group and become the member interface of the aggregation group.

Table 11-2 Configure the basic functions of the Ethernet link aggregation

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Create Ethernet link aggregation interface	**interface route-aggregation** *aggregation -unit*	Mandatory By default, do not create Ethernet link aggregation interface.
Specify the aggregation mode of the Ethernet link aggregation group	**route-aggregation mode { lacp | manual }**	Optional By default, the Ethernet link aggregation group is the static aggregation mode.
Specify the aggregation type of the Ethernet link aggregation group	**route-aggregation type { normal | sub-interface }**	Optional By default, the Ethernet link aggregation group is the common link aggregation type.
Create Ethernet link aggregation sub interface	**interface route-aggregation** *aggregation–unit. sub-unit*	Optional By default, do not create Ethernet link aggregation sub interface.
Exit Ethernet link aggregation interface mode	**exit**	-
Enter interface configuration mode	**interface** *interface-name*	The interface should be physical Ethernet interface.
Add to the specified Ethernet link aggregation group	**route-aggregation** **group** *aggregation-unit* { **manual** | **active** | **passive** }	Mandatory By default, the interface is not added to any aggregation group.

Note:

- After one Ethernet link aggregation interface is created, it can be configured like a normal Ethernet interface.
- Before the physical Ethernet interface joins the aggregation group, it is necessary to clear all L3 configurations on the interface, even if they are configured, they will not take effect.
- Before creating an Ethernet link aggregation sub interface, you need to first create an Ethernet link aggregation group interface. All kinds of configurations of the Ethernet link aggregation sub interface are the same as those of the common physical interface sub interface. When you clear the link aggregation group interface, you will clear the sub interface as well.
- A port can only be added to the static aggregation group in the manual mode, and can only be added the dynamic aggregation group in active/passive mode.
- The peer port of the port that joins dynamic aggregation group in passive mode should join dynamic aggregation group in active mode. Otherwise, the two ports are in unselected load state, and they cannot participate in user traffic forwarding.

11.3.2. Configure MAC Address of Ethernet Link Aggregation Interface

Configuration Condition

Before configuring the MAC address of the Ethernet link aggregation interface, first complete the following task:

- Create one Ethernet link aggregation interface

Configure MAC Address of Ethernet Link Aggregation Interface

Table 11-3 Configure the MAC address of the Ethernet link aggregation interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter Ethernet link aggregation interface configuration mode	**interface route-aggregation** *aggregation -unit*	-
Configure the MAC address of the Ethernet link aggregation interface	**ragg-macaddr** *mac-address*	Optional By default, generate the MAC address automatically after one Ethernet link aggregation interface is created.

11.3.3. Configure Load Mode of Ethernet Link Aggregation Group

Configuration Condition

Before configuring the load sharing mode of Ethernet link aggregation interface, the following tasks should be completed:

- Create one Ethernet link aggregation interface.

Configure Load Mode of Ethernet Link Aggregation Group

Table 11-4 Configure the load mode of the Ethernet link aggregation interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter Ethernet link aggregation interface configuration mode	**interface route-aggregation** *aggregation-unit*	-
Configure the load sharing mode of Ethernet link aggregation interface	**route-aggregation load-sharing {** *source-ip* | *source-destination-ip* | *destination-ip* | *per-packet* | *flowid}*	Optional By default, after the Ethernet link aggregation interface, load based on source IP mode.

11.3.4. Configure System Priority of Ethernet Link Aggregation

Configuration Condition

None

Configure Load Mode of Ethernet Link Aggregation Interface

Table 11-5 Configure the system priority mode of the Ethernet link aggregation group

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the system priority of the LACP protocol	**lacp system-priority** *system-priority-value*	Optional By default, the system priority of the Ethernet link aggregation is 32768.

11.3.5. Configure LACP Priority of Ethernet Link Aggregation Member Interface

Configuration Condition

None

Configure Load Mode of Ethernet Link Aggregation Interface

Table 11-6 Configure the priority mode of the link aggregation group member interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	The interface should be physical Ethernet interface.
Configure the LACP priority of the member interface	**lacp port-priority** *priority-value*	Optional By default, the priority of Ethernet link aggregation member interface is 32768.

11.3.6. Configure LACP Timeout Period of Ethernet Link Aggregation Member Interface

Configuration Condition

None

Configure LACP Timeout Period of Ethernet Link Aggregation Member Interface

Table 11-7 Configure LACP timeout period of link aggregation group member interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	The interface should be physical Ethernet interface.
Configure the LACP timeout period of the member interface	**lacp rate { fast | normal }**	Optional By default, the LACP of Ethernet link aggregation member is normal.

11.3.7. Configure the Independent Port Mode of Ethernet Link Aggregation Group

Configuration Condition

Before configuring the independent port mode of Ethernet link aggregation interface, first complete the following task:

- Create one Ethernet link aggregation interface

Configure the Independent Port Mode of Ethernet Link Aggregation Group

Table 11-8 Configure the independent port mode of Ethernet link aggregation group

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter Ethernet link aggregation interface configuration mode	**interface route-aggregation** *aggregation -unit*	-
Configure the independent port mode of Ethernet link aggregation group	**route-aggregation allow-individual-port**	Optional By default, do not set the independent port mode after one Ethernet link aggregation interface is created.

11.3.8. Configure Ethernet Link Aggregation Group to Link with BFD

Configuration Condition

Before configuring Ethernet link aggregation group to link with BFD, first complete the following task:

- Create one Ethernet link aggregation interface

Configure Ethernet Link Aggregation Group to Link with BFD

BFD (bidirectional forwarding detection) provides a fast method to detect the link aggregation group member line status between two devices. When link aggregation group starts BFD member link state detection, if there is a line fault between devices, BFD will quickly detect the fault and inform link aggregation group, trigger link aggregation group to select the load state of member interface, recalculate and switch to line forwarding flow, so as to achieve the purpose of fast switching of forwarding flow of aggregation group in the failed member interface.

Table 11-9 Configure Ethernet link aggregation group to link with BFD

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter Ethernet link aggregation interface configuration mode	**interface route-aggregation** *aggregation -unit*	-
Configure Ethernet link aggregation group to link with BFD	**route-aggregation bfd** [**compatible-mode**] **source-ip** *source-ip-address* **destination-ip** *destination-ip-address*	Optional By default, after one Ethernet link aggregation interface is created, do not configure the Ethernet link aggregation to link with BFD.

Note:

- If other BFD linkage sessions are configured in link aggregation group interface mode, it is not recommended to configure link aggregation and BFD linkage sessions compatible with protocol mode.
- The BFD linkage session of link aggregation group has been negotiated to the up state. If only the BFD configuration under the interface of link aggregation group at one end is deleted, the member of aggregation group will be loaded normally by none method. It is recommended to delete the BFD configuration under the interface of link aggregation group at both ends at the same time.

11.3.9. Configure Bandwidth Weight Load of Ethernet Link Aggregation Group Member

Configuration Condition

Before configuring the bandwidth weight load of Ethernet link aggregation group member, first complete the following task:

- Create one Ethernet link aggregation interface

Configure Bandwidth Weight Load of Ethernet Link Aggregation Group Member

Table 11-10 Configure the bandwidth weight load of Ethernet link aggregation group member

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter Ethernet link aggregation interface configuration mode	**interface route-aggregation** *aggregation -unit*	-
Configure the bandwidth weight load of Ethernet link aggregation group member	**route-aggregation bandwidth-weight-load**	Optional By default, Ethernet link aggregation group members forward traffic according to equal proportion weight load.
Enter aggregation group member interface configuration mode	**interface** *interface-name*	Optional
Configure the member interface bandwidth	**bandwidth** *bandwidth -value*	Optional By default, Ethernet link aggregation group calculates the load weight proportion according to the default bandwidth of the member interface.

11.3.10. Configure Ethernet Link Aggregation Group Member qos-group

Configure the qos-group id of link aggregation member port to match the qos-group id in the packet, and specify the matched packet to be forwarded from the member interface; if the unselect-drop parameter is added and when the member interface is not selected, the traffic of the matched qos-group id will be discarded.

Configuration Condition

Before configuring Ethernet link aggregation group member qos-group, first complete the following task:

- Create one Ethernet link aggregation interface

Configure Ethernet Link Aggregation Group Member qos-group

Table 11-11 Configure Ethernet link aggregation group member qos-group

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter aggregation group member interface configuration mode	**interface** *interface-name*	-
Configure the member interface to add to the link aggregation group	**route-aggregation group** *aggregation -unit*	-
Configure Ethernet link aggregation group member qos-group	**route-aggregation qos-group** *qosgroupid* **[unselect-drop]**	By default, do not configure Ethernet link aggregation group member qos-group.

11.3.11. Configure Ethernet Link Aggregation Group Member Interface to Force Congestion Packet to Fall Back

Configure the Ethernet link aggregation group member interface to force the congestion packet back to the aggregation group interface. When the member interface is congested, in order to prevent the member interface from discarding the congestion packet, you can use this command. The member interface will return the packet to the aggregation group interface, put the packet into the QoS queue of the aggregation group interface, and send it again through the aggregation group interface next time. When the member interface is congested, and if you don't want the packet to fall back, you can cancel configuring the Ethernet link aggregation group member interface to force the congestion packet to fall back to the aggregation group interface, and the member interface will directly discard the packet.

Configuration Condition

Before configuring the Ethernet link aggregation group member interface to force the congestion packet to fall back, first complete the following task:

- Create one Ethernet link aggregation interface

Configure Ethernet Link Aggregation Group Member Interface to Force Congestion Packet to Fall Back

Table 11-12 Configure the Ethernet link aggregation group member interface to force the congestion packet to fall back

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter Ethernet link aggregation interface configuration mode	**interface route-aggregation** *aggregation -unit*	-
Configure the Ethernet link aggregation group member interface to force the congestion packet to fall back	**route-aggregation force-return**	Optional By default, drop the packet when the Ethernet link aggregation group member interface is congested.

11.3.12. Ethernet Link Aggregation Monitoring and Maintaining

Table 11-13 Ethernet link aggregation group monitoring and maintaining

Command	Description
show interface route-aggregation *aggregation-unit*	Display the information of the specified Ethernet link aggregation interface
show route-aggregation [*aggregation-unit***]**	Display the aggregation interface status and member interface status information of the specified Ethernet link aggregation group interface
show route-aggregation brief	Display the brief information of all Ethernet link aggregation groups
show route-aggregation interface [*interface-name*]	Displays the details of the specified member interfaces of the Ethernet link aggregation group or all member interfaces of all created aggregation groups
show route-aggregation summary	Display the forwarding entry information of the Ethernet link aggregation interface

Command	Description
show route-aggregation member summary	Display the forwarding entry information of the Ethernet link aggregation member
show route-aggregation brief	Display the brief information of the Ethernet link aggregation interface

11.4. Typical Configuration Example of Ethernet Link Aggregation

11.4.1. Configure Static Ethernet Link Aggregation Group

Network Requirement

- Configure the static aggregation group between Device1 and Device2, so as to realize adding the bandwidth, load balance and service backup.

Network Topology

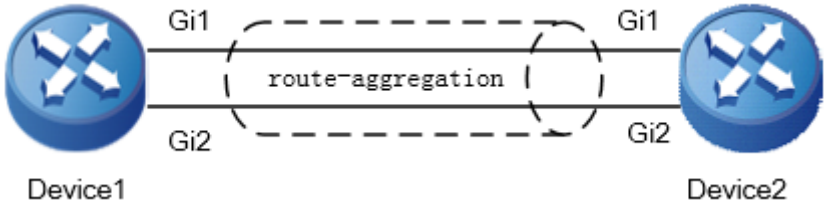

Figure 11-3 Networking of configuring the static aggregation group

Configuration Steps

Step 1: Create one static Ethernet link aggregation group.

#On Device1, create static aggregation group 0.

```
Device1#configure terminal
Device1(config)#interface route-aggregation 0
Device1(config-if-route-aggregation0)#ip address 192.168.1.1 255.255.255.0
Device1(config-if-route-aggregation0)#exit
```

#On Device 2, create static aggregation group 0.

```
Device2#configure terminal
Device2(config)#interface route-aggregation 0
Device2(config-if-route-aggregation0)#ip address 192.168.1.2 255.255.255.0
Device2(config-if-route-aggregation0)#exit
```

Step 2 : Add the interface to the aggregation group.

#On Device1, add interface gigabitethernet 1 and gigabitethernet 2 to aggregation group 0.

```
Device1(config)#interface gigabitethernet 1
```

```
Device1(config-if-gigabitethernet1)#route-aggregation group 0
Device1(config-if-gigabitethernet1)#exit
Device1(config)#interface gigabitethernet 2
Device1(config-if-gigabitethernet2)#route-aggregation group 0
Device1(config-if-gigabitethernet2)#exit
```

#On Device2, add interface gigabitethernet 1 and gigabitethernet 2 to aggregation group 0.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#route-aggregation group 0
Device2(config-if-gigabitethernet1)#exit
Device2(config)#interface gigabitethernet 2
Device2(config-if-gigabitethernet2)#route-aggregation group 0
Device2(config-if-gigabitethernet2)#exit
```

#After configuration, view the information of aggregation group 0 and its member ports on the device.

Take Device1 as an example :

```
Device1#show interface route-aggregation 0
route-aggregation0:
    line protocol is up
    Flags: (0xc008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 192.168.1.1/24
    Broadcast address: 192.168.1.255
    Metric: 0, MTU: 1500, BW: 2000000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 5002.cdd5.ce2a
    Last clearing of "show interface" counters never
    input peak rate 48 bits/sec,  0 hour 5 minutes 27 seconds ago
    output peak rate 67 bits/sec,  0 hour 10 minutes 7 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    2 packets received; 2 packets sent
    120 bytes received; 84 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
```

```
    0 collisions; 0 dropped
    Unknown protocol 0
Members(counts:2):
 gi1(up)
 gi2(up)

Device1#show route-aggregation member summary

 IfIndex   NodeId Agg type  Flag    State Sync location   Ifname
 --------- ---- ---  ----  -------- ----- ---- --------- --------------------
 50331698  0    0  0   1   UP   F   50  |333   gigabitethernet1
 50331699  0    0  0   1   UP   F   51  |334   gigabitethernet2
```

You can see that interface gigabitethernet1 and gigabitethernet2 on Device1 are successfully added to aggregation group 0 and are in the up state, the total bandwidth of the link of aggregation group 0 is 2000Mbps.

Note:

- For the check method of Device2, refer to Device1.

Step 3 : Configure the load balance mode of the aggregation group.

#On Device1, configure the load balance mode of aggregation group 0 as the dst-ip mode (by default, it is the scr-ip mode).

```
Device1(config)#interface route-aggregation 0
Device1(config-if-route-aggregation0)#route-aggregation load-sharing
destination-ip
Device1(config-if-route-aggregation0)#exit
```

#On Device2, configure the load balance mode of aggregation group 0 as the dst-ip mode (by default, it is the scr-ip mode).

```
Device2(config)#interface route-aggregation 0
Device2(config-if-route-aggregation0)#route-aggregation load-sharing
destination-ip
Device2(config-if-route-aggregation0)#exit
```

Step 4 : Check the result.

#On the device, view the summary information of Ethernet link aggregation group 0.

Take Device1 as an example :

```
Device1#show route-aggregation summary

Unit IfIndex  RootIndex LoadShare  LoadPolicy  Flag  State Sync Location  Ifname
---- -------  ------- ------   ------ -----  --- --- ------ ----------
```

```
0    16777226  50348947 DST-IP   equal   1    UP   F    0|0  route-aggregation0
```

You can see that the interface status of aggregation group 0 on Device1 is up, and the load balance mode is the dst-ip mode.

#On the device, view the summary information of the member ports of the aggregation group.

Take Device1 as an example:

```
Device1#show route-aggregation member summary

IfIndex   NodeId Agg type Flag    State Sync location   Ifname
--------- ---- --- -- --  ---- ----- ---- ---------- --------------------
50331698  0    0  0  1    UP   F    50  |333   gigabitethernet1
50331699  0    0  0  1    UP   F    51  |334   gigabitethernet2
```

You can see that interface gigabitethernet1 and gigabitethernet2 on Device1 both belong to the member ports of aggregation group 0.

Note:

- For the check method of Device2, refer to Device1.

#When Device1 and Device2 interact the service, the data can realize the load balance on the aggregation link. When one link of the aggregation group fails, the remaining links can back up the service.

11.4.2. Configure Dynamic Ethernet Link Aggregation Group

Network Requirement

- Configure the dynamic aggregation group between Device1 and Device2, so as to realize adding the bandwidth, load balance and service backup.

Network Topology

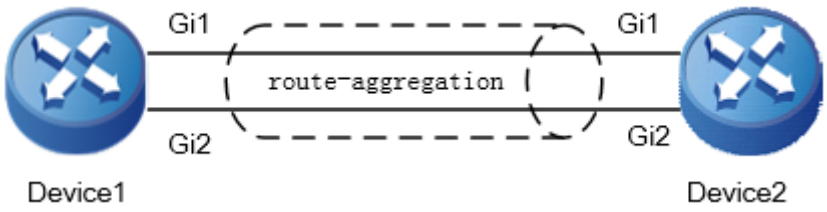

Figure 11-4 Networking of configuring the dynamic aggregation group

Configuration Steps

Step 1: Create Ethernet link aggregation group 0, and configure the work mode of the aggregation group as LACP.

#On Device1, create dynamic aggregation group 0.

```
Device1#configure terminal
Device1(config)#interface route-aggregation 0
Device1(config-if-route-aggregation0)# route-aggregation mode lacp
Device1(config-if-route-aggregation0)# ip address 192.168.1.1 255.255.255.0
```

```
Device1(config-if-route-aggregation0)#exit
```

#On Device2, create dynamic aggregation group 0.

```
Device2#configure terminal
Device2(config)#interface route-aggregation 0
Device2(config-if-route-aggregation0)# route-aggregation mode lacp
Device2(config-if-route-aggregation0)# ip address 192.168.1.2 255.255.255.0
Device2(config-if-route-aggregation0)#exit
```

#After configuration, view the information of aggregation group 0 on the Device.

Take Device1 as an example.

```
Device1#show route-aggregation 0
route-aggregation0    information:
  Mode:LACP
  Type:Normal-Agg
  loadShare:SRC-IP
  loadPolicy:equal
  State:DOWN
  Bfd:None
  Root Member:None
  Act Speed: None
  Act Duplex: None
  Members counts:0
```

You can see that the link aggregation group works in the LACP mode.

Note:

- For the check method of Device2, refer to Device1.

Step 2 : Add the interface to the aggregation group.

#On Device1, add interface gigabitethernet 1 and gigabitethernet 2 to aggregation group 0.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#route-aggregation group 0 active
Device1(config-if-gigabitethernet1)#exit
Device1(config)#interface gigabitethernet 2
Device1(config-if-gigabitethernet2)#route-aggregation group 0 active
Device1(config-if-gigabitethernet2)#exit
```

#On Device2, add interface gigabitethernet 1 and gigabitethernet 2 to aggregation group 0.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#route-aggregation group 0 active
Device2(config-if-gigabitethernet1)#exit
```

```
Device2(config)#interface gigabitethernet 2
Device2(config-if-gigabitethernet2)#route-aggregation group 0 active
Device2(config-if-gigabitethernet2)#exit
```

#After configuration, view the information of aggregation group 0 and its member ports on the device.

Take Device1 as an example :

```
Device1#show interface route-aggregation 0
route-aggregation0:
    line protocol is up
    Flags: (0xc008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 192.168.1.1/24
    Broadcast address: 192.168.1.255
    Metric: 0, MTU: 1500, BW: 2000000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 5002.cdd5.ce2a
    Last clearing of "show interface" counters never
    input peak rate 48 bits/sec,  0 hour 5 minutes 27 seconds ago
    output peak rate 67 bits/sec,  0 hour 10 minutes 7 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    2 packets received; 2 packets sent
    120 bytes received; 84 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
Members(counts:2):
 gi1(up)
 gi2(up)

Device1#show route-aggregation member summary

 IfIndex   NodeId Agg type Flag    State Sync location   Ifname
 ---------- ---- --- ----  ---- ----- ---- ---------- --------------------
 50331698  0    0   0  1    UP    F    50  |333   gigabitethernet1
```

```
50331699  0   0  0  1   UP   F   51 |334  gigabitethernet2
```

You can see that interface gigabitethernet1 and gigabitethernet2 on Device1 are successfully added to aggregation group 0 and are in the up state, the total bandwidth of the link of aggregation group 0 is 2000Mbps.

Note:

- For the check method of Device2, refer to Device1.

When Device1 and Device2 interact the service, the data can realize the load balance on the aggregation link. When one link of the aggregation group fails, the remaining links can back up the service.

11.4.3. Configure Sub Interface Link Aggregation Group

Network Requirement

- Device1 and Device2 create sub interface link aggregation group 1, and g1.1 and g2.1 are added to aggregation group 1. Device1 and Device3 create sub interface link aggregation group 2, and g1.2 and g2.2 are added to aggregation group 2 to realize bandwidth increasing, load balancing and service backup.

Network Topology

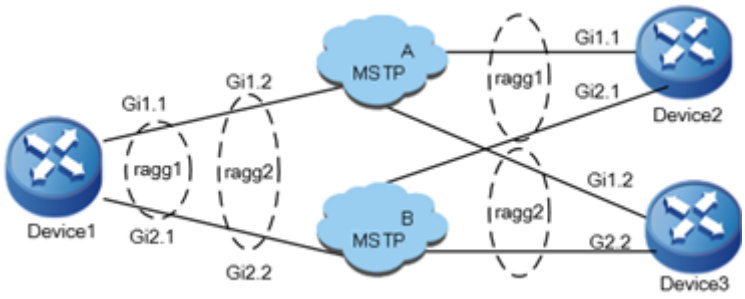

Figure 11-5 Networking of configuring sub interface link aggregation group

Configuration Steps

Step 1: Create sub interface Ethernet link aggregation group 1, 2, and configure the work mode of the aggregation group as LACP.

#On Device1, create sub interface link aggregation group 1, 2.

```
Device1#configure terminal
Device1(config)#interface route-aggregation 1
Device1(config-if-route-aggregation1)#route-aggregation mode lacp
Device1(config-if-route-aggregation1)#route-aggregation type sub-interface
Device1(config-if-route-aggregation1)#ip address 192.168.1.1 255.255.255.0
Device1(config-if-route-aggregation1)#exit
Device1#configure terminal
Device1(config)#interface route-aggregation 2
Device1(config-if-route-aggregation2)#route-aggregation mode lacp
Device1(config-if-route-aggregation2)#route-aggregation type sub-interface
```

```
Device1(config-if-route-aggregation2)#ip address 192.168.2.1 255.255.255.0
Device1(config-if-route-aggregation2)#exit
```

#On Device2, create sub interface link aggregation group 1.

```
Device2#configure terminal
Device2(config)#interface route-aggregation 1
Device2(config-if-route-aggregation1)#route-aggregation mode lacp
Device2(config-if-route-aggregation1)#route-aggregation type sub-interface
Device2(config-if-route-aggregation1)#ip address 192.168.1.2 255.255.255.0
Device2(config-if-route-aggregation1)#exit
```

#On Device3, create sub interface link aggregation group 2.

```
Device3#configure terminal
Device3(config)#interface route-aggregation 2
Device3(config-if-route-aggregation2)#route-aggregation mode lacp
Device3(config-if-route-aggregation2)#route-aggregation type sub-interface
Device3(config-if-route-aggregation2)#ip address 192.168.2.2 255.255.255.0
Device3(config-if-route-aggregation2)#exit
```

#After configuration, view the aggregation group information on the device.

Take aggregation group 1 of Device1 as an example:

```
Device1# show route-aggregation 1
 Mode:LACP
 Type:Sub-Interface-Agg
 loadShare:SRC-IP
 loadPolicy:equal
 State:DOWN
 Bfd:None
 Root Member:None
 Act Speed: None
 Act Duplex: None
 Members counts:0
```

You can see that the link aggregation group is the sub interface link aggregation mode.

Note:

- For the check method of Device2, Device3, refer to Device1.

Step 2: Add the interface to the aggregation group.

#On Device1, add interfaces gigabitethernet 1.1, gigabitethernet 2.1 to aggregatio group 1 respectively, and add interfaces gigabitethernet 1.2, gigabitethernet 2.2 to aggregation group 2.

```
Device1(config)#interface gigabitethernet 1.1
Device1(config-if-gigabitethernet1.1)#encapsulation dot1q 1
Device1(config-if-gigabitethernet1.1)#route-aggregation group 1 active
Device1(config-if-gigabitethernet1.1)#exit
Device1(config)#interface gigabitethernet 2.1
Device1(config-if-gigabitethernet2.1)#encapsulation dot1q 1
Device1(config-if-gigabitethernet2.1)#route-aggregation group 1 active
Device1(config-if-gigabitethernet2.1)#exit
Device1(config)#interface gigabitethernet 1.2
Device1(config-if-gigabitethernet1.2)#encapsulation dot1q 2
Device1(config-if-gigabitethernet1.2)#route-aggregation group 2 active
Device1(config-if-gigabitethernet1.2)#exit
Device1(config)#interface gigabitethernet 2.2
Device1(config-if-gigabitethernet2.2)#encapsulation dot1q 2
Device1(config-if-gigabitethernet2.2)#route-aggregation group 2 active
Device1(config-if-gigabitethernet2.2)#exit
```

#On Device2, add sub interfaces gigabitethernet 1.1 and gigabitethernet 2.1 to aggregation group 1.

```
Device2(config)#interface gigabitethernet 1.1
Device2(config-if-gigabitethernet1.1)#encapsulation dot1q 1
Device2(config-if-gigabitethernet1.1)#route-aggregation group 1 active
Device2(config-if-gigabitethernet1.1)#exit
Device2(config)#interface gigabitethernet 2.1
Device2(config-if-gigabitethernet2.1)#encapsulation dot1q 1
Device2(config-if-gigabitethernet2.1)#route-aggregation group 1 active
Device2(config-if-gigabitethernet2.1)#exit
```

#On Device3, add sub interfaces gigabitethernet 1.2 and gigabitethernet 2.2 to aggregation group 2.

```
Device3(config)#interface gigabitethernet 1.2
Device3(config-if-gigabitethernet1.2)#encapsulation dot1q 2
Device3(config-if-gigabitethernet1.2)#route-aggregation group 2 active
Device3(config-if-gigabitethernet1.2)#exit
Device3(config)#interface gigabitethernet 2.2
Device3(config-if-gigabitethernet2.2)#encapsulation dot1q 2
Device3(config-if-gigabitethernet2.2)#route-aggregation group 2 active
Device3(config-if-gigabitethernet2.2)#exit
```

#After configuration, view the information of aggregation group 1 and member ports on the device. Take aggregation group 1 of device1 as an example:

```
Device1#show interface route-aggregation 1
route-aggregation0:
    line protocol is up
    Flags: (0xc008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 192.168.1.1/24
    Broadcast address: 192.168.1.255
    Metric: 0, MTU: 1500, BW: 2000000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 5002.cdd5.ce2a
    Last clearing of "show interface" counters never
    input peak rate 48 bits/sec,  0 hour 5 minutes 27 seconds ago
    output peak rate 67 bits/sec,  0 hour 10 minutes 7 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    2 packets received; 2 packets sent
    120 bytes received; 84 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
Members(counts:2):
 gi1.1(up)
 gi2.1(up)
Device1#show interface route-aggregation 1
route-aggregation1   information:
  Mode:LACP
  Type:Sub-Interface-Agg
  loadShare:SRC-IP
  loadPolicy:equal
  State:UP
  Bfd:None
  Root Member:gi1.1
  Act Speed:1000M
  Act Duplex:FULL
  Members counts:2
```

```
Attached Members counts:2
Interface      Select    Fwd    Priority   Link   Speed  Duplex  VLAN     BFD
-------------- --------- ------ -------- ------ ------ ------- ------ ------
gi1.1    Selected  RX-TX   32768   UP-LINK  1000M  FULL   ENCAP  None
gi2.1    Selected  RX-TX   32768   UP-LINK  1000M  FULL   ENCAP  None
```

You can see that interface gigabitethernet1.1 and gigabitethernet2.1 on Device1 are successfully added to aggregation group 1 and are in the up state, the member ports are all in the selected state, the total bandwidth of the link of aggregation group 1 is 2000Mbps.

Note:

- For the check method of Device2 and Device3, refer to Device1.

When Device1 interact the service with Device2, Device3, the data can realize the load balance on the aggregation link. When one link of the aggregation group fails, the remaining links can back up the service.

11.4.4. Configure QoS on Ethernet Link Aggregation

Network Requirement

- Link aggregation group 1 is configured between Device1 and Device2 to increase bandwidth, load balance and service backup.
- The link bandwidth between Device1 and Device2 is 100M, and the bandwidth of each member port is 50M.
- Configure QoS on Device1, classify FTP, AAA and video traffic according to source IP, guarantee 60% low-delay bandwidth for FTP and AAA traffic to Device2, and 30% bandwidth for video traffic, but the maximum occupied bandwidth is not more than 40%.

Network Topology

Figure 11-6 Networking of configuring QoS on Ethernet link aggregation

Configuration Steps

Step 1: Create Ethernet link aggregation group 1, 2, and configure the work mode of the aggregation group as LACP.

#On Device1, create link aggregation group 1.

```
Device1#configure terminal
Device1(config)#interface route-aggregation 1
Device1(config-if-route-aggregation1)# route-aggregation mode lacp
```

```
Device1(config-if-route-aggregation1)#ip address 200.1.1.1 255.255.255.0
Device1(config-if-route-aggregation1)#exit
```

#On Device2, create link aggregation group 1.

```
Device2#configure terminal
Device2(config)#interface route-aggregation 1
Device2(config-if-route-aggregation1)# route-aggregation mode lacp
Device2(config-if-route-aggregation1)#ip address 200.1.1.2 255.255.255.0
Device2(config-if-route-aggregation1)#exit
```

Step 2: Add the interface to aggregation group.

#On Device1, add interfaces gigabitethernet 1 and gigabitethernet 2 to aggregation group 1 respectively.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#route-aggregation group 1 active
Device1(config-if-gigabitethernet1)#exit
Device1(config)#interface gigabitethernet 2
Device1(config-if-gigabitethernet2)#route-aggregation group 1 active
Device1(config-if-gigabitethernet2)#exit
```

#On Device2, add interfaces gigabitethernet 1 and gigabitethernet 2 to aggregation group 1 respectively.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#route-aggregation group 1 active
Device2(config-if-gigabitethernet1)#exit
Device2(config)#interface gigabitethernet 2
Device2(config-if-gigabitethernet2)#route-aggregation group 1 active
Device2(config-if-gigabitethernet2)#exit
```

#After configuration, view the information of aggregation 1 and member ports on the device.

Take Device1 as an example:

```
Device1#show interface route-aggregation 1
route-aggregation0:
    line protocol is up
    Flags: (0xc008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 192.168.1.1/24
    Broadcast address: 192.168.1.255
    Metric: 0, MTU: 1500, BW: 2000000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 5002.cdd5.ce2a
```

```
    Last clearing of "show interface" counters never
    input peak rate 48 bits/sec,  0 hour 5 minutes 27 seconds ago
    output peak rate 67 bits/sec,  0 hour 10 minutes 7 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    2 packets received; 2 packets sent
    120 bytes received; 84 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
  Members(counts:2):
   gi1(up)
   gi2(up)
```

You can see that the interfaces gigabitethernet1 and gigabitethernet2 on Device1 are added to aggregation group 1 and are in the up state.

#After configuration, detect the link via ping.

```
  Device1#ping 200.1.1.2

  Press key (ctrl + shift + 6) interrupt it.
  Sending 5, 76-byte ICMP Echos to 200.1.1.1 , timeout is 2 seconds:
  !!!!!
 Success rate is 100% (5/5). Round-trip min/avg/max = 0/34/170 ms.
```

Note:

- For the check method of Device2, refer to Device1.

Step 3: Configure the QoS policy and apply.

#On Device1, configure the extended ACL of ftp_aaa, match the traffic of FTP and AAA type, configure video extended ACL, and match the traffic of the video type.

```
Device1(config)#ip access-list extended ftp_aaa
Device1(config-ext-nacl)#permit ip host 100.1.1.1 any
Device1(config-ext-nacl)#permit ip host 100.1.2.1 any
Device1(config-ext-nacl)#exit
Device1(config)#ip access-list extended video
Device1(config-ext-nacl)#permit ip host 100.1.3.1 any
Device1(config-ext-nacl)#exit
```

#On Device1, configure CBWFQ, and match various kinds of data according to ACL.

```
Device1(config)#class-map match-all ftp_aaa
Device1(config-ext-nacl)#match access-group ftp_aaa
Device1(config-ext-nacl)#exit
Device1(config)#class-map match-all video
Device1(config-ext-nacl)#match access-group video
Device1(config-ext-nacl)#exit
```

#On Device1, configure the CBWFQ policy.

```
Device1(config)#policy-map qos
Device1(config-pmap)#class ftp_aaa
Device1(config-pmap-c)#priority percent 60
Device1(config-pmap-c)#exit
Device1(config-pmap)#class video
Device1(config-pmap-c)#bandwidth percent 30
Device1(config-pmap-c)#shape average percent 40
Device1(config-pmap-c)#exit
Device1(config-pmap)#exit
```

#On the link aggregation member port of Device1, configure GTS, bandwidth and apply the configured policy.

```
Device1(config)# interface gigabitethernet 1
Device1(config-if-gigabitethernet1)# bandwidth 50000
Device1(config-if-gigabitethernet1)# traffic-shape 50000000 1250000
Device1(config-if-gigabitethernet1)# service-policy output qos
Device1(config-if-gigabitethernet        )#exit
Device1(config)# interface gigabitethernet 2
Device1(config-if-gigabitethernet2)#bandwidth 50000
Device1(config-if-gigabitethernet2)#traffic-shape 50000000 1250000
Device1(config-if-gigabitethernet2)#service-policy output qos
Device1(config-if-gigabitethernet2)#exit
```

#View the input queue statistics.

```
Device1#show policy-map interface gigabitethernet 1
 interface gigabitethernet1
  Service-policy output: qos

   Class-map: ftp_aaa (match-all)
    85720 packets 11315040 bytes
    5 minute offered rate 12931624 bps
```

match access-group ftp_aaa
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 85724/11315568
Priority: 60% (30000 Kbps) , burst bytes 1000000, b/w exceed drops: 0

Class-map: video (match-all)
42871 packets 5658972 bytes
5 minute offered rate 6467400 bps
match access-group video
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 42883/5660556
Bandwidth: 30% (15000 Kbps)
Shaping
shape (average) cir 20000000, bc 4000000

Class-map: class-default (match-any)
0 packets 0 bytes
5 minute offered rate 0 bps
Queueing
queue limit 256 packets
(queue depth/total drops) 0/0
(packets output/bytes output) 0/0
match any

Note:

- CAR, GTS, QoS queue and GTS + QoS queue are independently configured for each member interface, and each member interface can achieve QoS effect independently. It is recommended to configure QoS on link aggregation member interface in the market.
- If the actual leased line bandwidth is inconsistent with the physical bandwidth of the interface, GTS needs to be configured on the interface. Otherwise, the packet may be discarded on the operator's line, and the QoS control on the device will not work. If the bandwidth of the actual leased line is inconsistent with the bandwidth configured by the interface, the bandwidth or qos max-bandwidth of the interface should be modified. If the Gigabit Ethernet port is connected to a 20M operator MSTP line, a 20M GTS needs to be configured in the out direction of the interface, and the bandwidth of the interface needs to be modified to 20M

- Because the device QoS calculation is processed according to the length of the link layer header plus the data part of the link layer, and the MSTP line provided by the operator is actually implemented by bundling one to multiple 2M channels, there is internal transmission overhead. According to the actual application experience, the overhead of each packet is generally about 20 bytes, so if connecting the MSTP line of the operator, It is also necessary to configure qos account output length add 20 to increase 20 bytes per packet when the device calculates the bandwidth, which is in line with the actual situation of the operator's MSTP line and ensures that the packet will not lose packets when it is transmitted on the operator's MSTP line

From the above information, you can see the basic configuration information, the absolute value converted from bandwidth guarantee, the packet statistics and 5-minute traffic statistics matched by each class, and the current QoS queue packet accumulation. You can simply judge whether the configuration is correct and effective. In the process of service interaction between Device1 and Device2, the data can be scheduled on the aggregation link. When a link in the aggregation group fails, the remaining links can perform service backup.

11.4.5. Configure Ethernet Link Aggregation Group to Link with BFD

Network Requirement

- Aggregation group is configured between Device1 and Device2, and BFD is configured on aggregation group to realize fast switching of forwarding traffic on fault member interface.

Network Topology

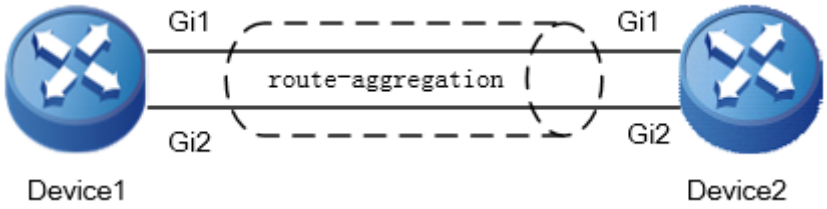

Figure 11-7 Networking of configuring aggregation grouo to link with BFD

Configuration Steps

Step 1: Create Ethernet link aggregation group 0, configure the work mode of the aggregation group as LACP, and configure the link aggregation BFD.

#On Device1, create dynamic aggregation group 0.

```
Device1#configure terminal
Device1(config)#interface route-aggregation 0
Device1(config-if-route-aggregation0)# route-aggregation mode lacp
Device1(config-if-route-aggregation0)#ip address 192.168.1.1 255.255.255.0
Device1(config-if-route-aggregation0)#bfd multiplier 3
Device1(config-if-route-aggregation0)#bfd min-transmit-interval 500
Device1(config-if-route-aggregation0)#bfd min-receive-interval 500
Device1(config-if-route-aggregation0)#route-aggregation bfd source-ip 192.168.1.1
destination-ip 192.168.1.2
Device1(config-if-route-aggregation0)#exit
```

#On Device2, create dynamic aggregation group 0.

```
Device2#configure terminal
Device2(config)#interface route-aggregation 0
Device2(config-if-route-aggregation0)#route-aggregation mode lacp
Device2(config-if-route-aggregation0)#ip address 192.168.1.2 255.255.255.0
Device2(config-if-route-aggregation0)#bfd multiplier 3
Device2(config-if-route-aggregation0)#bfd min-transmit-interval 500
Device2(config-if-route-aggregation0)#bfd min-receive-interval 500
Device2(config-if-route-aggregation0)#route-aggregation bfd source-ip 192.168.1.2
destination-ip 192.168.1.1
Device2(config-if-route-aggregation0)#exit
```

#After configuration, view the information of aggregation group 0 on the device.

Take Device1 as an example.

```
Device1#show route-aggregation 0
route-aggregation0    information:
 Mode:LACP
 Type:Normal-Agg
 loadShare:SRC-IP
 loadPolicy:equal
 State:DOWN
 Bfd:Set
 Root Member:None
 Act Speed: None
 Act Duplex: None
 Members counts:0
```

You can see that the link aggrehation group works in LACP mode, and the BFD status is Set.

Note:

- The BFD configuration method of the static aggregation group is consistent with that of dynamic aggregation group.
- For the check method of Device2, refer to Device1.

Step 2: Add the interface to the aggregation group.

#On Device1, add the interface gigabitethernet 1 and gigabitethernet 2 to aggregation group 0.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#route-aggregation group 0 active
Device1(config-if-gigabitethernet1)#exit
Device1(config)#interface gigabitethernet 2
Device1(config-if-gigabitethernet2)#route-aggregation group 0 active
Device1(config-if-gigabitethernet2)#exit
```

#On Device2, add the interface gigabitethernet 1 and gigabitethernet 2 to aggregation group 0.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#route-aggregation group 0 active
Device2(config-if-gigabitethernet1)#exit
Device2(config)#interface gigabitethernet 2
Device2(config-if-gigabitethernet2)#route-aggregation group 0 active
Device2(config-if-gigabitethernet2)#exit
```

#After configuration, view the information of aggregation group 0 and member port on the device. Take Device1 as an example:

```
Device1#show interface route-aggregation 0
route-aggregation0:
    line protocol is up
    Flags: (0xc008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 192.168.1.1/24
    Broadcast address: 192.168.1.255
    Metric: 0, MTU: 1500, BW: 2000000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 5002.cdd5.ce2a
    Last clearing of "show interface" counters never
    input peak rate 48 bits/sec,  0 hour 5 minutes 27 seconds ago
    output peak rate 67 bits/sec,  0 hour 10 minutes 7 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    2 packets received; 2 packets sent
    120 bytes received; 84 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
Members(counts:2):
 gi1(up)
 gi2(up)

Device1#show route-aggregation member summary
```

```
IfIndex   NodeId Agg Flag    State Sync location   Ifname
---------- ---- --- -------- ----- ---- ---------- --------------------
50331698  0    0  1     UP    F    50  |333   gigabitethernet1
50331699  0    0  1     UP    F    51  |334   gigabitethernet2
```

You can see that the interfaces gigabitethernet1 and gigabitethernet2 on Device1 are successfully added to aggregation group 0 and are in the up state.

#Check the BFD status of the aggregation group member port.

```
Device1##show bfd session lag-micro
LAG-Interface        Member-Interface        OurAddr        NeighAddr        LD/RD
State   Holddown
route-aggregation0  gigabitethernet1     192.168.1.1     192.168.1.2     71/80          UP
1500
route-aggregation0  gigabitethernet2     192.168.1.1     192.168.1.2     72/80
UP     1500
```

The BFD statys of the member port of the aggregation group is UP.

Note:

- For the check method of Device2, refer to Device1.

In the process of service interaction between Device1 and Device2, data can be load balanced on the aggregation link. BFD will quickly detect the fault and notify the link aggregation group, trigger the link aggregation group to recalculate the selected load status of the member port, and switch to the line forwarding traffic, so as to achieve the purpose of fast switching of forwarding traffic of aggregation group at the fault member interface.

11.4.6. Link Aggregation Combines with qos-group id to Realize Service Diversion and Link Detection

Network Requirement

- Link aggregation group 1 is configured between Device1 and Device2 to increase bandwidth, load balance and service backup.
- On Device1, the aggregation member interface combines qos-group id to specify the service FTP and AAA service to be sent from the line of operator A. when the line of operator A is abnormal, it is sent through the line of operator B.
- Combined with qos-group id, the aggregation member port uses SNMP manager packet to detect the connectivity of the member port line of the aggregation port. When the member port is in the unselected state after the line is interrupted, the SNMP traffic is discarded and the abnormal line is detected

Network Topology

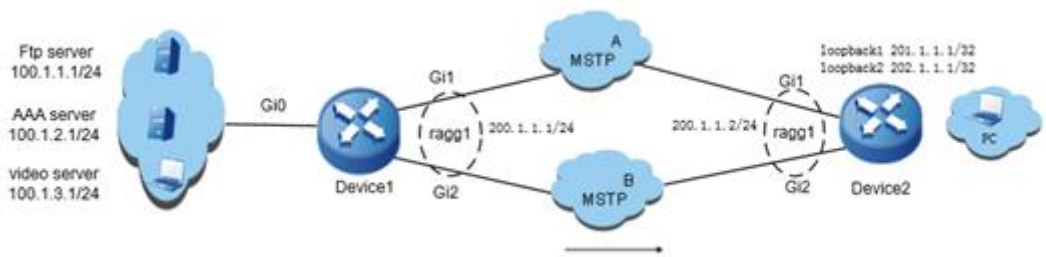

Figure 11-8 Link aggregation combines qos-group id to realize service division and link detection

Configuration Steps

Step 1: Configure the IP address of the interface (omitted).

Step 2: Create Ethernet link aggregation group 1, and configure the work mode of the aggregation group as LACP.

#On Device1, create link aggregation group 1.

```
Device1#configure terminal
Device1(config)#interface route-aggregation 1
Device1(config-if-route-aggregation1)# route-aggregation mode lacp
Device1(config-if-route-aggregation1)#ip address 200.1.1.1 255.255.255.0
Device1(config-if-route-aggregation1)#exit
```

#On Device2, create link aggregation group 1.

```
Device2#configure terminal
Device2(config)#interface route-aggregation 1
Device2(config-if-route-aggregation1)# route-aggregation mode lacp
Device2(config-if-route-aggregation1)#ip address 200.1.1.2 255.255.255.0
Device2(config-if-route-aggregation1)#exit
```

Step 3: Add the interface to aggregation group.

#On Device1, add interfaces gigabitethernet 1 and gigabitethernet 2 to aggregation group 1.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)#route-aggregation group 1 active
Device1(config-if-gigabitethernet1)#exit
Device1(config)#interface gigabitethernet 2
Device1(config-if-gigabitethernet2)#route-aggregation group 1 active
Device1(config-if-gigabitethernet2)#exit
```

#On Device2, add interfaces gigabitethernet 1 and gigabitethernet 2 to aggregation group 1.

```
Device2(config)#interface gigabitethernet 1
Device2(config-if-gigabitethernet1)#route-aggregation group 1 active
Device2(config-if-gigabitethernet1)#exit
Device2(config)#interface gigabitethernet 2
Device2(config-if-gigabitethernet2)#route-aggregation group 1 active
Device2(config-if-gigabitethernet2)#exit
```

#After configuration, view the information of aggregation group 1 and member ports on the device.

Take Device1 as an example:

```
Device1#show interface route-aggregation 1
```

```
route-aggregation1:
    line protocol is up
    Flags: (0xc008063) BROADCAST MULTICAST ARP RUNNING
    Type: ETHERNET_CSMACD
    Internet address: 200.1.1.1/24
    Broadcast address: 200.1.1.255
    Metric: 0, MTU: 1500, BW: 2000000 Kbps, DLY: 100 usec, VRF: global
    Reliability 255/255, Txload 1/255, Rxload 1/255
    Ethernet address is 5002.cdd5.ce2a
    Last clearing of "show interface" counters never
    input peak rate 48 bits/sec,  0 hour 5 minutes 27 seconds ago
    output peak rate 67 bits/sec,  0 hour 10 minutes 7 seconds ago
    5 minutes input rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    5 minutes output rate 0 bit/sec, 0 packet/sec, bandwidth utilization -
    2 packets received; 2 packets sent
    120 bytes received; 84 bytes sent
    0 multicast packets received
    0 multicast packets sent
    0 input errors; 0 output errors
    0 collisions; 0 dropped
    Unknown protocol 0
Members(counts:2):
 gi1(up)
 gi2(up)
```

You can see that interfaces gigabitethernet1 and gigabitethernet2 on Device1 are successfully added to aggregation group 1 and are in up state.

#After configuration, the line is detected by ping.

```
Device1#ping 200.1.1.2

Press key (ctrl + shift + 6) interrupt it.
Sending 5, 76-byte ICMP Echos to 200.1.1.1 , timeout is 2 seconds:
!!!!!
Success rate is 100% (5/5). Round-trip min/avg/max = 0/34/170 ms.
```

Note:

- For the check method of Device2, refer to Device1.

Step 4: Configure the route to ensure that FTP and AAA services can be connected, and the loopback address of SNMP manager and Device2 can be connected. Note that the out interface of SNMP manager to loopback1 is gigabitethernet1, and the outgoing interface to loopback2 is gigabitethernet2 (omitted).

Step 5: Configure QoS policy CBWFQ, and mark corresponding qos-group id for FTP, AAA and SNMP services.

#Configure extended ACL on Device1 to match various types of traffic.

```
Device1(config)#ip access-list extended ftp_aaa
Device1(config-ext-nacl)#permit ip host 100.1.1.1 any
Device1(config-ext-nacl)#permit ip host 100.1.2.1 any
Device1(config-ext-nacl)#exit
Device1(config)#ip access-list extended snmp_a
Device1(config-ext-nacl)#permit ip host 100.1.3.1 host 201.1.1.1
Device1(config-ext-nacl)#exit
Device1(config)#ip access-list extended snmp_b
Device1(config-ext-nacl)#permit ip host 100.1.3.1 host 202.1.1.1
Device1(config-ext-nacl)#exit
```

#Configure CBWFQ on Device1 to match all types of data according to ACL.

```
Device1(config)#class-map match-all ftp_aaa
Device1(config-ext-nacl)#match access-group ftp_aaa
Device1(config-ext-nacl)#exit
Device1(config)#class-map match-all snmp_a
Device1(config-ext-nacl)#match access-group snmp_a
Device1(config-ext-nacl)#exit
Device1(config)#class-map match-all snmp_b
Device1(config-ext-nacl)#match access-group snmp_b
Device1(config-ext-nacl)#exit
```

#Configure CBWFQ policy on Device1, mark qos-group id for traffic respectively, among which FTP and AAA packets are marked as 1, the packets from SNMP manager to loopback1 of Device2 are marked as 2, and those to loopback2 are marked as 3.

```
Device1(config)#policy-map qos
Device1(config-pmap)#class ftp_aaa
Device1(config-pmap-c)#set qos-group 1
Device1(config-pmap-c)#exit
```

```
Device1(config-pmap)#class snmp_a
Device1(config-pmap-c)#set qos-group 2
Device1(config-pmap-c)#exit
Device1(config-pmap)#class snmp_b
Device1(config-pmap-c)#set qos-group 3
Device1(config-pmap-c)#exit
```

#Configure the QoS policy of incoming direction on gigabitethernet0 of Device 1.

```
Device1(config)#interface gigabitethernet 0
Device1(config-if-gigabitethernet0)# service-policy input qos
Device1(config-if-gigabitethernet0)#exit
```

#View the incoming queue statistics.

```
Device1#show policy-map interface gigabitethernet 0
 interface gigabitethernet0
  Service-policy output: qos

    Class-map: ftp_aaa (match-all)
      85720 packets 11315040 bytes
      5 minute offered rate 12931624 bps
      match access-group ftp_aaa
      QoS Set
        qos-group 1
          Packets marked 156708
    Class-map: snmp_a (match-all)
      42871 packets 5658972 bytes
      5 minute offered rate 6467400 bps
      match access-group snmp_a
      QoS Set
        qos-group 2
          Packets marked 20300
    Class-map: snmp_b (match-all)
      42871 packets 5658972 bytes
      5 minute offered rate 6467400 bps
      match access-group snmp_b
      QoS Set
        qos-group 3
          Packets marked 20300
```

```
Class-map: class-default (match-any)
  0 packets 0 bytes
  5 minute offered rate 0 bps
  match any
```

Step 6: The member port of the aggregation group matches qos-group id.

#The member port of Device1 gigabitethernet1 matches FTP and AAA traffic. Specify the traffic to be sent from the line of operator A.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)# route-aggregation qos-group 1
Device1(config-if-gigabitethernet1)#exit
```

#The member port gigabitethernet1 matches the packet from SNMP to loopback1 of Device2, which is used to detect carrier line A. the member port gigabitethernet2 matches the packet from SNMP to loopback2 of Device2, which is used to detect carrier line B.

```
Device1(config)#interface gigabitethernet 1
Device1(config-if-gigabitethernet1)# route-aggregation qos-group 2 unselect-drop
Device1(config-if-gigabitethernet1)#exit
Device1(config)#interface gigabitethernet 2
Device1(config-if-gigabitethernet2)# route-aggregation qos-group 3 unselect-drop
Device1(config-if-gigabitethernet2)#exit
```

Step 7: Check the result.

#FTP and AAA traffic is sent from the member port of Device1 gigabitethernet1 by default. When the line is blocked, it can be switched to another line to send. You can view the effect by viewing the member port statistics on Device1.

#On the SNMP manager, ping the address of loopback1 of the branch node Device2 to detect the line of operator A, and ping the address of loopback2 to detect the line of operator B. If ping is connected, it means that the line is normal; if not, it means that the corresponding line is abnormal.

Note:

- For details of the function of configuring qos-group id for aggregation group, please refer to Configuration Manual > Link Layer Protocol > Ethernet Link Aggregation > Configure qos-group for members of Ethernet link aggregation group.
- Link detection packet and service packet cannot be marked with the same qos-group id.

12. SNA

12.1. Overview of SNA

SNA (Systems Network Architecture) is a network architecture developed by IBM, which is widely used in the host environment of IBM. Generally speaking, SNA is the main networking protocol of IBM's mainframe (ES/9000, S/390, etc.) and medium-sized computer (AS/400). SNA's history is as early as 1974. SNA was first announced by IBM to connect its 3270 series products. The related link layer protocols include SDLC, LLC2, QLLC and DLSw. By configuring the above protocols in the router, the conversion from SNA to TCP/IP can be realized, and the router access of SNA devices can be completed.

12.2. DLSw Configuration

DLSw standard describes a switch-to-switch protocol (SSP) for establishing peer connection, locating resources, transmitting data, controlling flow, and correcting error between routers. Through the local response of data link connection, the data link exchange standard stops the transmission of link layer response and keepalive information in WAN.

To set up connection between two terminal systems, first complete:

1. Establish peer connection
2. Exchange performance
3. Establish link

12.2.1. Configure DLSw Local Peer Entity

Configuration Condition

None

Configure DLSw Peer Entity

Table 12-1 Configure DLSw local peer entity

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure DLSw local peer entity	**dlsw local-peer** {[**init-pacing-window** *size*] | **peer-id** *ip_address* {[**cost** *cost*] | [**keepalive** *seconds*] | [**promiscuous**]}}	*size:* The setting value of the local initial window size, the valid range is 1-2000, and the default value is 20. *ip_address:* The IP address of the local router *cost:* The cost from the local router to the remote router. The valid range is 1-5, and the default value is 3. The larger the value is, the more

Step	Command	Description
		expensive it is to reach the remote router. *seconds:* The time interval value of sending the keepalive packet. The value range is 1-1200, the unit is seconds, and the default value is 30.

12.2.2. Configure DLSw Remote Peer Entity

Configuration Condition

None

Configure DLSw Remote Peer Entity

Table 12-2 Configure DLSw remote peer entity

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure DLSw remote peer entity	**dlsw remote-peer 0 tcp** *ip-address1* [**backup-peer** *ip-address2* **| cost** *cost* **| keepalive** *seconds* **| lf** *lf-value* **| passive**]	*ip-address1:* specifies remote backup entity *ip-address2:* Specifies the backed up remote entity *cost:* the cost value from the local router to the remote router. The valid range is 1-5, and the default value is 3. The larger the value is, the more expensive it is to reach the remote router. *seconds:* the time interval value of sending keepalive packets. The value range is 0-1200, the unit is seconds, and the default value is 30. *lf-value:* the maximum frame length used by the local router on the line notified to the remote router.

Note:

- In dlsw remote-peer list_number tcp ip_address1 backup-peer ip_address2, take the remote router specified by ip_address1 as the backup entity of the remote router specified by ip_address2, that is, the router specified by ip_address1 is backup peer, while the router specified by ip_address2 is primary peer. Besides, before configuring

backup peer, you should configure primary peer first; while before deleting primary peer, you should first delete its corresponding backup peer. One primary peer only permits one backup peer.

12.2.3. Disable DLSw

Configuration Condition

None

Disable DLSw

Table 12-3 Disable DLSw

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Disable DLSw	**dlsw disable**	By default, DLSw is enabled.

12.2.4. Configure DLSw SSP Protocol Capability

Configuration Condition

None

Configure DLSw SSP Protocol Capability

Table 12-4 Configure the DLSw SSP protocol capability

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure the DLSw SSP protocol capability	**dlsw local-peer** {**dlsw icanreach** {*mac-address* **mac-addr | mac-exclusive | saps** *sap*} **| icannotreach saps** *sap*}	*mac-addr:* The MAC address that the local router can reach *sap:* The service access point supported or not supported by the local router

12.2.5. Configure DLSw MAC Address Time Domain Control

Configuration Condition

None

Configure DLSw MAC Address Time Domain Control

Table 12-5 Configure DLSw MAC address time domain control

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Configure DLSw MAC address time domain control	**dlsw mac-address** *mac-addr* **time-range** *time-range*	*mac-addr*: The MAC address of the DLSW link *time-range*: The time domain

12.3. SDLC Configuration

Synchronous data link control protocol (SDLC) is the first bit-oriented synchronous link layer protocol developed by IBM for SNA environment. SDLC defines primary and secondary sites. The primary site controls other workstations (secondary sites), which poll the secondary sites in a predetermined order. If data is sent from secondary site, it can only be transmitted when it is polled. The primary site also needs to establish, terminate and manage the link in the working process.

12.3.1. Configure SDLC Basic Functions

Configuration Condition

None

Encapsulate SDLC Protocol

Table 12-6 Encpasupate the SDLC protocol

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Encapsulate the SDLC protocol	**encapsulation sdlc**	By default, the link layer protocol encapsulated by E1, CE1 and asynchronous serial interface is PPP.

12.3.2. Configure SDLC Local Physical Address

By configuring the physical address, the router can establish a connection on the data link layer with the device attached below.

Configuration Condition

None

Configure SDLC Local Physical Address

Table 12-7 Configure SDLC local physical address

Step	Command	Description
Enter the global configuration mode	configure terminal	-
Enter interface configuration mode	interface interface-name	-
Configure SDLC local physical address	sdlc address sdlc-address [xid-passthru | xid-poll [echo]]	sdlc-address: The physical address of the device attached to the corresponding interface of the router. Its valid range is a hexadecimal number between 01 and FE.

12.3.3. Configure WMAC Address of the Specified Interface

Configuration Condition

None

Configure VMAC Address of the Specified Interface

Table 12-8 Configure the VMAC address of the specified interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the VMAC address of the specified interface	**sdlc vmac** *mac-address* [*sdlc-address*]	*mac-address:* The MAC address of the interface/the MAC address of the physical device attached to the interface. The format is:XXXX.XXXX.XX00. X is any hexadecimal digit in 0-F. When the partner is configured on the peer router, this address is configured instead of changing the last two digits with the sdlc address.

Step	Command	Description
		*sdlc-addr*ess: The physical address of the device attached to the corresponding interface of the router. Its valid range is a hexadecimal number between 01 and FE.

12.3.4. Configure SDLC Peer Physical Address

By configuring the physical address, the router can establish a connection on the data link layer with the device attached below.

Configuration Condition

None

Configure SDLC Peer Physical Address

Table 12-9 Configure SDLC peer physical address

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure SDLC peer physical address	**sdlc partner** *mac-address sdlc-address*	*mac-address* The MAC address of the peer physical device *sdlc-address* The physical address of the local physical device

Note:

- When executing the **sdlc partner mac-address sdlc-address** command, it is necessary to ensure that the physical address of the lower physical device has been configured, and the peer MAC address configured on the local router must correspond to the peer MAC address.

12.3.5. Configure SDLC Interface DLSw

Configuration Condition

None

Configure SDLC Interface DLSw

Table 12-10 Configure SDLC interface DLSw

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure SDLC interface DLSw	**sdlc dlsw** *sdlc-addr*	*sdlc-addr:* Realize DLSw on the device specified by sdlc-addr

12.3.6. Configure XID Value of the SDLC Interface

Configuration Condition

None

Configure XID Value of the SDLC Interface

Table 12-11 Configure the XID value of the SDLC interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the XID value of the SDLC interface	**sdlc xid sdlc-address** *xid*	*sdlc-address:* The physical address of the low physical device to be configured with the XID value *xid:* XID value, the format is XXXXXXXX, X is one any hex value of 0-F.

Note:

- The **sdlc xid sdlc_address xid** command is valid only when the lower-end device type is PU2.0. The XID value will not be valid if the xid-passthru and xid-poll have been configured in the sdlc address command. In addition, before configuring the XID value, you must first configure the physical address of the corresponding lower-end device. Otherwise, you will not be able to configure the corresponding XID value. When configuring the XID value, the user must ensure that the XID value is consistent with the

configuration in the upper device. Otherwise, the SNA connection may not be established.

12.3.7. Configure Delay Response Time of SDLC Interface

Configuration Condition

None

Configure Delay Response Time of SDLC Interface

Table 12-12 Configure the delay response time of the SDLC interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the delay response time of the SDLC interface	**sdlc delay-response**	By default, it is not defined.

12.3.8. Configure the Window Size of SDLC Interface

Configuration Condition

None

Configure Window Size of SDLC Interface

Table 12-13 Configure the window size of the SDLC interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the window size of the SDLC interface	**sdlc k** *window-size*	*window-size:* The local sending window size. The minimum is 1 frame and the maximum is 7 frames. The default value is set to the maximum value.

12.3.9. Configure the Times of SDLC Interface Re-sending Frames

Configuration Condition

None

Configure the Times of SDLC Interface Re-sending Frames

Table 12-14 Configure the times of tge SDLC interface re-sending frames

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the times of tge SDLC interface re-sending frames	**sdlc n2** *retry-count*	*retry-count:* The re-sending times; If the re-sending times exceeds the value, the SDLC sitze will terminate the session with other sites. The valid range is 1-255 and the default value is 5.

12.3.10. Configure the Interval of SDLC Interface Polling Frames

Configuration Condition

None

Configure the Interval of SDLC Interface Polling Frames

Table 12-15 Configure the interval of the SDLC interface polling the frames

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the interval of the SDLC interface polling the frames	**sdlc poll-pause-timer** *time*	*time:* The time that the software waits to send polling frame to a single serial port in the unit of 100ms. The valid value range is 1-100, and the default value is 5.

12.3.11. Configure the Time of SDLC Interface waiting for Polling Frames

Configuration Condition

None

Configure the Time of SDLC Interface waiting for Polling Frames

Table 12-16 Configure the time of the SDLC interface waiting for polling frames

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the time of the SDLC interface waiting for polling frames	**sdlc poll-wait-timeout** *time*	*time:* The time for the software to wait for the polling frame from the master station before the link with the master station times out in the unit of 100ms. The valid value range is 1-640, and the default value is 100.

12.3.12. Configure Link Site Role of SDLC Interface

Configuration Condition

None

Configure Link Site Role of SDLC Interface

Table 12-17 Configure the link site role of the SDLC interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the link site role of the SDLC interfac	**sdlc role** {**primary | secondary**}	**Primary:** Set the local sdlc interface as the primary site

Step	Command	Description
		Secondary: Set the local sdlc interface as the secondary site

12.3.13. Configure Source and Destination snap Value of Remote SDLC Link

Configuration Condition

None

Configure Source and Destination snap Value of Remote SDLC Link

Table 12-18 Configure the source and destination snap value of the SDLC link

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the source and destination snap value of the SDLC link	**sdlc sap** *sdlc_address S-sap-P D-sap-P*	*sdlc_address:* Set the physical address of the lower physical device *S-sap-P:* Set the local snap value of the remote device of the sdlc link *D-sap-P:* Set the destination snap value of the remote device of the sdlc link

12.3.14. Configure Max. Size of Received and Sent Frames of SDLC Interface

Configuration Condition

None

Configure Max. Size of Received and Sent Frames of SDLC Interface

Table 12-19 Configure the maximum size of the receibed and sent frames of the SDLC interface

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the maximum size of the receibed and sent frames of the SDLC interface	**sdlc sdlc-largest-frame** *sdlc-addr size*	*sldc-addr:* The address of the SDLC site to communicate with the router *size:* The max. length of the frame that can be sent and received; the valid range is 0-4294967295, and the default value is 65535.

12.3.15. Configure the Time of SDLC Interface Waiting for Response Information

Configuration Condition

None

Configure the Time of SDLC Interface Waiting for Response Information

Table 12-20 Configure the time of the SDLC interface waiting for the response information

Step	Command	Description
Enter the global configuration mode	**configure terminal**	-
Enter interface configuration mode	**interface** *interface-name*	-
Configure the time of the SDLC interface waiting for the response information	**sdlc T1** *seconds*	*seconds:* The waiting time in the unit of second, the valid range is 2-30, and the default value is 4.

13. ОБЩАЯ ИНФОРМАЦИЯ

13.1. Замечания и предложения

Мы всегда стремимся улучшить нашу документацию и помочь вам работать лучше, поэтому мы хотим услышать вас. Мы всегда рады обратной связи, в особенности:

- ошибки в содержании, непонятные или противоречащие места в тексте;
- идеи по улучшению документации, чтобы находить информацию быстрее;
- неработающие ссылки и замечания к навигации по документу.

Если вы хотите написать нам по поводу данного документа, то используйте, пожалуйста, форму обратной связи на qtech.ru.

13.2. Гарантия и сервис

Процедура и необходимые действия по вопросам гарантии описаны на сайте QTECH в разделе «Поддержка» −> «Гарантийное обслуживание».

Ознакомиться с информацией по вопросам тестирования оборудования можно на сайте QTECH в разделе «Поддержка» −> «Взять оборудование на тест».

Вы можете написать напрямую в службу сервиса по электронной почте sc@qtech.ru.

13.3. Техническая поддержка

Если вам необходимо содействие в вопросах, касающихся нашего оборудования, то можете воспользоваться нашей автоматизированной системой запросов технического сервис-центра helpdesk.qtech.ru.

Телефон Технической поддержки +7 (495) 797-33-11 доб. 0