



A Packet's Adventures on Huawei Routers

"Introduce You How Packets are Forwarded
on Huawei High-End Routers"

HUAWEI TECHNOLOGIES CO., LTD.

Copyright © Huawei Technologies Co., Ltd. 2016. All rights reserved.

No part of this document may be reproduced or transmitted in any form or by any means without prior written consent of Huawei Technologies Co., Ltd.

Trademarks and Permissions

Huawei and other Huawei trademarks are trademarks of Huawei Technologies Co., Ltd. All other trademarks and trade names mentioned in this document are the property of their respective holders.

Notice

This document is a supplement to product documents. It only serves as a reference for you to learn information about packet forwarding processes on Huawei routers. Packet forwarding processes vary with products. If the information provided in this document conflicts with that in a specific product document, the information in the product document prevails.

The information in this document is subject to change without notice. Every effort has been made in the preparation of this document to ensure accuracy of the contents, but all statements, information, and recommendations in this document do not constitute a warranty of any kind, express or implied.

Intended Audience

- Network planning engineers
- System maintenance engineers

Huawei Technologies Co., Ltd.

Website: <http://www.huawei.com>

Email: dcinfo@huawei.com

Content

Introduction-----	1
Chapter 1 Switching, Addressing, and Forwarding-----	3
Chapter 2 Packet Receiving, Sending, Parsing, and Encapsulating-----	10
Chapter 3 Traffic Control-----	17
Chapter 4 QoS Basics-----	20
Chapter 5 QoS Processing-----	31
Chapter 6 Other Processing on the Forwarding Plane-----	37
Chapter 7 Journey of Protocol Packets-----	45
Chapter 8 IP Unicast Forwarding Process-----	51
Chapter 9 Layer 2 Ethernet Frame Forwarding-----	56
Chapter 10 IP Multicast Forwarding-----	65
Chapter 11 MPLS Forwarding-----	80

Introduction

Modern networks may require an array of high, medium, and low-level routers, each with differing functions and applications. This document explains how Huawei high-end routers (such as the NE40E, NE80E, and NE5000E) operate.

Where Does the Data Go After Being "Swallowed" by a Router?

A router continually swallows and passes communication data.



Where does data go after being swallowed by a router?

Most data input to a router from one interface is output through another interface. These data packets are only "passers-by" and therefore also called pass-by packets. A small portion of data is "absorbed" - either sent to the CPU or dropped in transit.

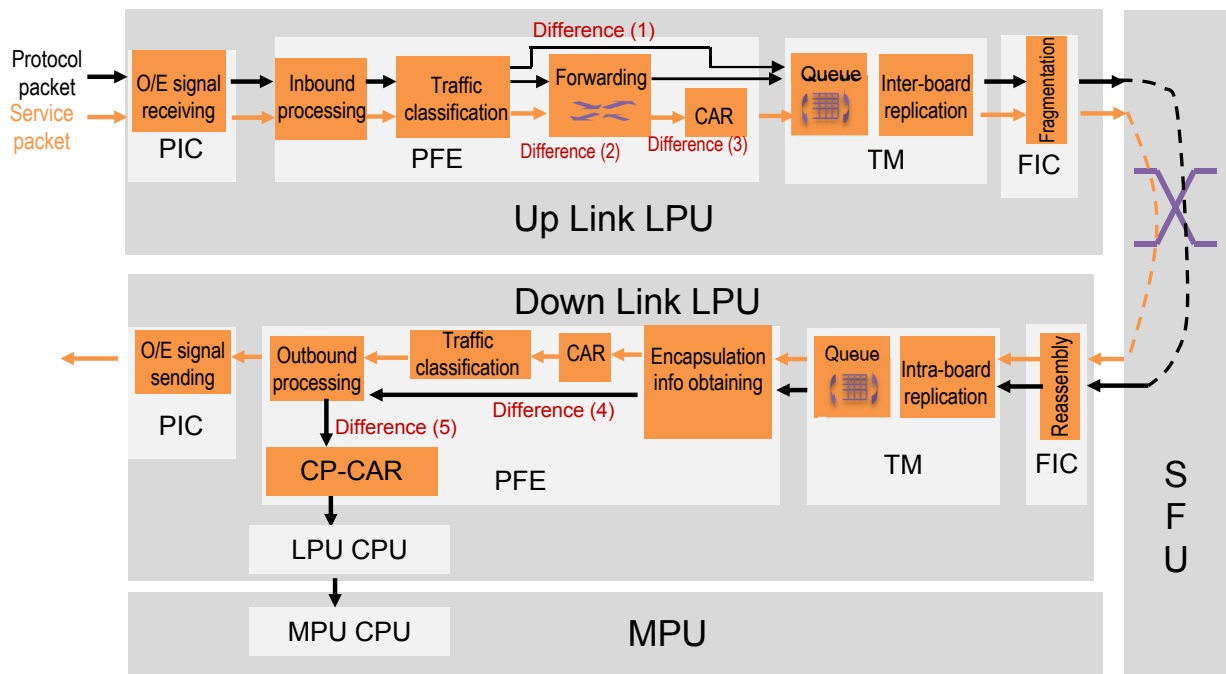


Note:

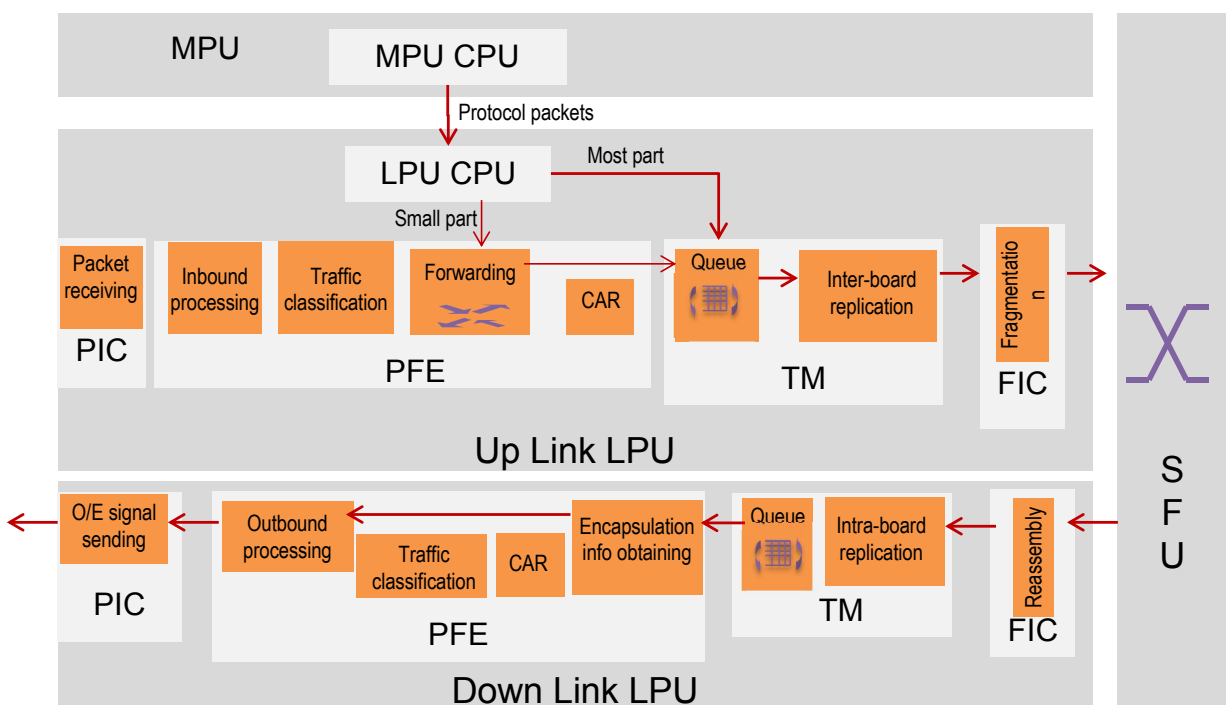
Huawei high-end routers use hardware-based forwarding that prevents service packets from being processed by device CPUs. This increases performance by preventing all packets from being processed by device CPUs. This is known as software-based forwarding.

Router Forwarding Panorama

The following figure shows how a router forwards service and protocol packets.



The following figure shows how a router's CPU forwards a protocol packet.



These processes are explained further throughout the following chapters.

Chapter 1

Switching, Addressing, and Forwarding

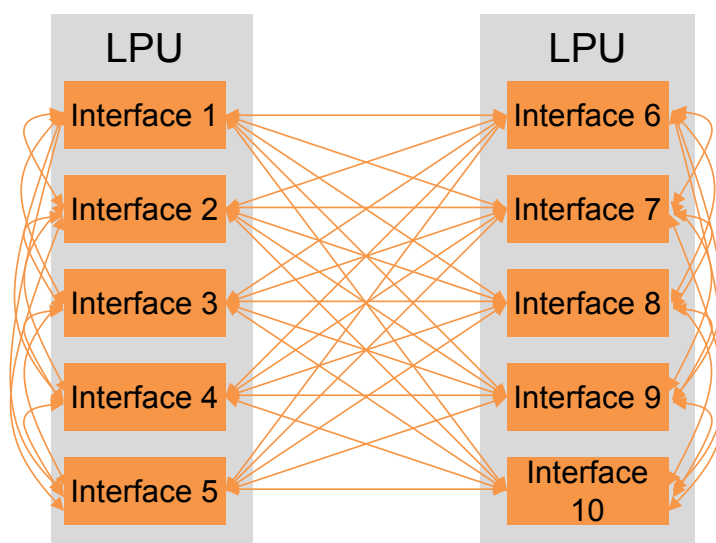
This chapter details the basic functions of a router: switching, addressing, and forwarding.

This chapter will explain:

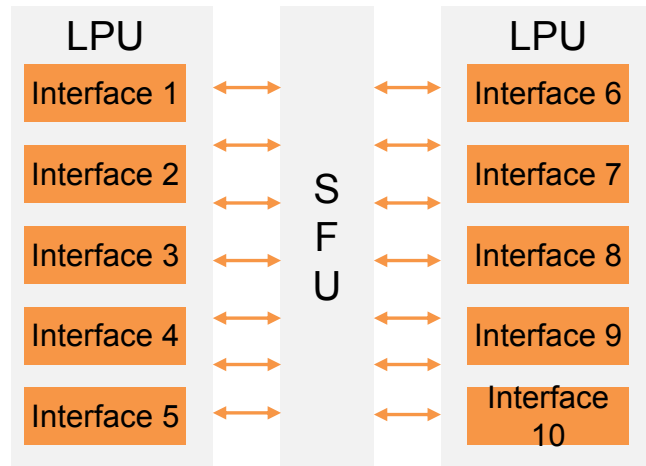
- The importance of switch fabric units (SFUs).
- Uplink and downlink processes.
- The relationship between a routing table and a forwarding information base (FIB), and their location in a router.
- FIB generation.

Starting from "Switching"

Data is transmitted and received by line processing units (LPUs) through cables inserted into LPU interfaces. Two interfaces must be connected to allow data transfer between them. In practice, data packets may be sent or received from any interface. If cables are connected through point-to-point (P2P), $N \times (N-1)/2$ cable connections are required.



SFUs simplify LPU connections by allowing interfaces to communicate with each other through the SFUs rather than through P2P connections.



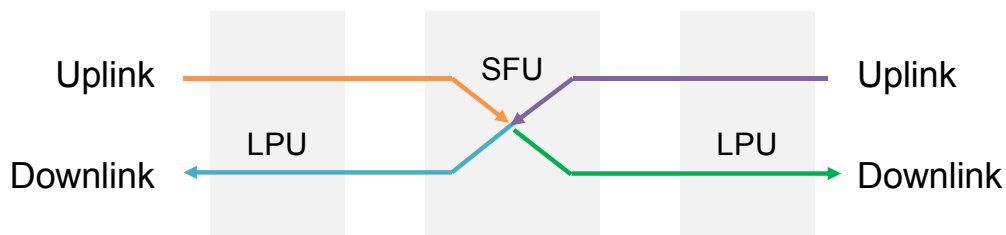
Note:

Devices such as the NE40E-X1/X2/X3, do not have SFUs. Instead, their LPUs have switching modules, which perform the same function as an SFU.

An SFU establishes connections between inbound and outbound interfaces for data switching. It operates independent of device configurations, protocols, and data packet types. For more details about SFUs, see Basic Router Hardware Concept - Switch Fabric.

Uplink and Downlink

With an SFU as the midway point, a packet's journey along a router can be cut into two parts: the former going uplink, and the latter going downlink.



Addressing and Forwarding

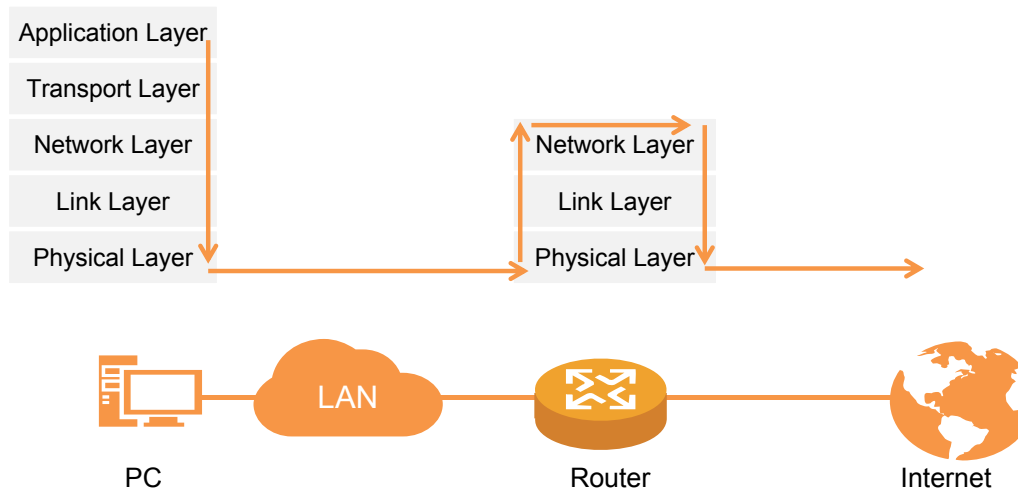
Switches are able to transfer data from one interface to another, but cannot determine the shortest possible route between the countless nodes on the Internet. This is where routers are needed.

Routers perform two tasks:

- **Addressing**: the selection of the shortest possible transmission path.
- **Forwarding**: the transfer of packets through a matching interface.

Routers improve communication efficiency, reduces network load, and conserves network resources. This is beyond a switch's capability.

Most networks follow the TCP/IP model. In the TCP/IP model, routers operate at the network layer, also known as Layer 3.



Routers working at Layer 3 in the TCP/IP model

Addressing is the process of searching for the network layer address of a data packet, known as the IP address. To search for an IP address, a router uses a routing table, which uses destination IP addresses as indexes. Each router has a routing table, which is similar to a subway station map.

What Does a Routing Table Look Like?

The following figure shows the map of a bus station, a real world example of a routing table.



The following figure shows a routing table, which contains the following fields:

Destination/Mask, NextHop, and Interface.

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.0/8	Static	60	0	RD	10.136.120.1	GigabitEthernet1/0/0
10.136.120.0/23	Direct	0	0	D	10.136.120.107	GigabitEthernet1/0/0
10.136.120.107/32	Direct	0	0	D	127.0.0.1	GigabitEthernet1/0/0
10.136.121.255/32	Direct	0	0	D	127.0.0.1	GigabitEthernet1/0/0
127.0.0.0/8	Direct	0	0	D	127.0.0.1	InLoopBack0
127.0.0.1/32	Direct	0	0	D	127.0.0.1	InLoopBack0
127.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0
192.1.1.0/30	Direct	0	0	D	192.1.1.2	GigabitEthernet2/0/0
192.1.1.2/32	Direct	0	0	D	127.0.0.1	GigabitEthernet2/0/0
192.1.1.3/32	Direct	0	0	D	127.0.0.1	GigabitEthernet2/0/0
255.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0

Destination/Mask
IP address

NextHop IP
address

Outbound
interface

This table tells a router how to forward a data packet. For example, if a router receives a packet with the destination address 10.0.0.1, the router searches the routing table, finds the first entry matches, and forwards the packet to GigabitEthernet 1/0/0.

Routing tables are generated dynamically through routing protocols or manually by configuration. In manual configuration, static routes are configured manually and do not adapt to network changes. If the network topology changes, these routes must be manually updated. Dynamic routing protocols allow routers to automatically exchange routing information and calculate routes based on the collected information. This method allows routing tables to update according to topology changes. Direct routes are another type of route discovered by the link layer protocol.

Where Is a Routing Table Placed?

The ideal location for a routing table is in a public network location, such as the MPU. Placing the routing table in the SFU causes data transfer bottlenecks. This is why the SFU cannot run routing protocols, maintain routing tables, or perform address-based forwarding.

The routing table cannot be placed on a downlink LPU, because the SFU must know the destination LPU before switching received packets. Address-based forwarding must be completed on the uplink. Placing the routing table on an uplink LPU will require that each LPU has a routing table, because packets may enter from any LPU. The MPU CPU is the ideal choice for running routing protocols, calculating routes, and generating and maintaining a routing table.

FIB and Routing Table

Huawei high-end routers use hardware-based forwarding. During this process, service packets are not processed by the MPU CPU. The MPU CPU must deliver forwarding information to LPUs after generating a routing table. This forwarding information is stored in each LPU's FIB. All forwarding information comes from the MPU and is therefore identical for all LPUs.

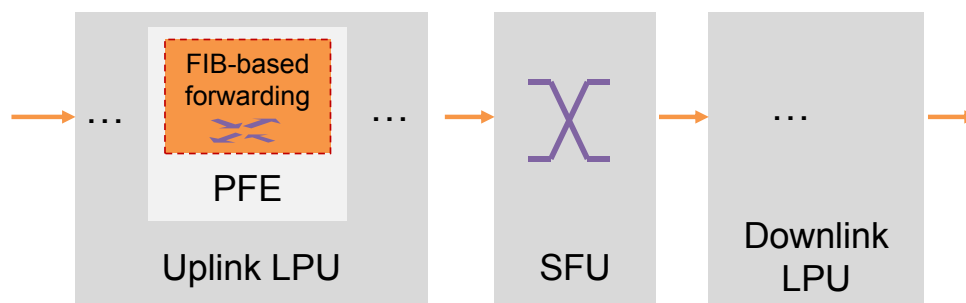
All modern high-performance routers use an architecture that separates the forwarding plane from the control plane. The control plane runs routing protocols, maintains the routing table, and delivers the FIB to the forwarding plane. The forwarding plane forwards data packets. This architecture prevents the two planes from negatively affecting each other. Traffic burdening the forwarding plane will not affect route learning in the control plane, and routing in the control plane will not affect packet transfer in the forwarding plane.

Destination/Mask	Nexthop	Flag	TimeStamp	Interface	TunnelID
192.1.1.0/30	192.1.1.2	U	15:54:32	GE2/0/0	0x0
192.1.1.3/32	127.0.0.1	HU	15:54:32	GE2/0/0	0x0
192.1.1.2/32	127.0.0.1	HU	15:54:32	GE2/0/0	0x0
10.136.120.0/23	10.136.120.107	U	00:00:00	GE0/0/0	0x0
127.0.0.0/8	127.0.0.1	HU	00:00:00	InLoop0	0x0
10.136.121.255/32	127.0.0.1	HU	00:00:00	GE0/0/0	0x0
10.136.120.107/32	127.0.0.1	HU	00:00:00	GE0/0/0	0x0
127.255.255.255/32	127.0.0.1	HU	00:00:00	InLoop0	0x0
255.255.255.255/32	127.0.0.1	HU	00:00:00	InLoop0	0x0
127.0.0.1/32	127.0.0.1	HU	00:00:00	InLoop0	0x0
10.0.0.0/8	10.136.120.1	GSU	00:00:00	GE0/0/0	0x0

A FIB, shown in the preceding figure, is similar to a routing table. Both have **Destination/Mask**, **Nexthop**, and **Interface**. This is because a FIB is generated based on a routing table.

A routing table may contain multiple routes to the same destination, but a FIB selects the optimal one. The next hop in a routing table may not be directly reachable, but the next hop in a FIB must be directly reachable. The process of finding the direct next hop from the original next hop is route iteration.

After a router is powered on, it learns the network topology and generates a routing table by running routing protocols. If LPUs successfully register, the MPU generates forwarding entries based on the routing table and delivers them to LPU FIBs. The router then forwards data packets based on the FIB. The component that forwards packets is the packet forwarding engine (PFE), which is typically an NP or ASIC chip.



What Happens If a Route Is Unreachable?

When a route is unreachable, the router searches an FIB for forwarding procedures. This forwarding mode is called pre-routing, the process of planning the route before forwarding. Most modern routers use this mode for unicast forwarding. In this mode, if a router finds no match and no default route in the FIB, the data packet cannot reach its destination. As attempts at retransmission will continue to fail, the data packet can only be discarded. This is an undesirable outcome, and the PFE records reasons for packet drops as well as packet loss statistics.

Pre-Routing and Flow-Triggering

The routers we have discussed all use pre-routing modes. The alternative is

flow-triggering mode, where packets are sent without preemptively establishing a route. When using flow-triggering mode, the router will search the FIB upon receiving a packet. If no match is found, the router will generate a forwarding entry for subsequent forwarding based on the packet.

Routers and switches use MAC address tables for Layer 2 forwarding. MAC address learning is a form of flow-triggering mode.

As it provides legitimate attack paths, flow-triggering mode is more vulnerable to traffic attacks. Attackers can launch traversal attacks by overwhelming the a router with various unknown packets. To prevent these attacks, Huawei high-end routers support a MAC address learning limit function. This function sets a maximum number of MAC addresses that a router can learn and creates a forced time interval between learning addresses. This function can also be disabled.

Chapter 2

Packet Receiving, Sending, Parsing, and Encapsulating

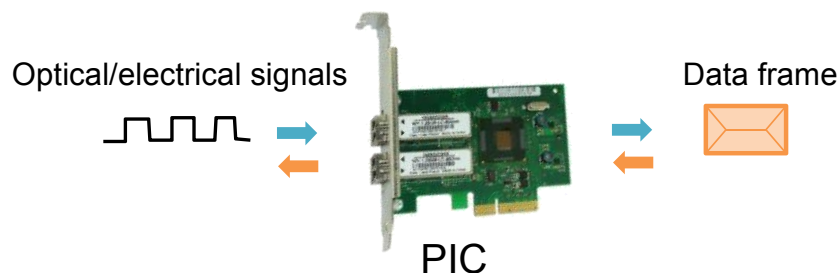
Packets are transmitted as optical/electrical signals through communication cables. Signal receiving, forwarding, switching and sending are just some aspects of data communication.

This chapter details the following:

- Conversion between optical/electrical signals and data frames
- Validity check on data frames
- Packet parsing
- Packet encapsulating

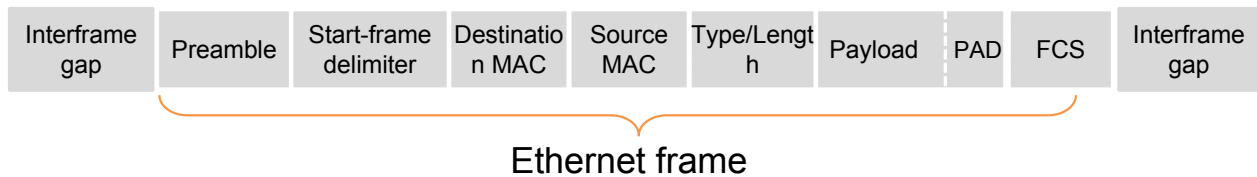
Packet Receiving and Sending on PICs

Data is transmitted as optical/electrical signals (or bit flows at the physical layer) through communication cables. To allow routers to interpret these signals for later forwarding, each cable-inserting interface contains a physical interface controller (PIC). The PIC detects and converts these signals to data frames, such as Ethernet frames, PPP frames, or ATM cells.



A PIC has two important functions: to receive and send optical/electrical signals at the physical layer and to check the validity of data frames. Data may become malformed during cable transmission. Malformed packets are called error packets, and cannot be correctly parsed by the PFE. To prevent this problem, PICs must check the validity of data frames.

The following figure shows the format of an Ethernet frame:



Ethernet standards specify that a frame is invalid if:

- The frame length is not integral bytes.
- The frame contains errors discovered by the FCS.
- The frame payload is not between 46 bytes to 1500 bytes.

Invalid frames are discarded, and not retransmitted.



Note:

Ethernet frames require an interframe gap between each frame. This means a device must wait for a configured period of time before sending another frame. The interframe gap ensures the frame receiver has enough time to process a received frame before receiving another. This allows for processes such as the adjusting of the buffer pointer or updating statistics to be performed.

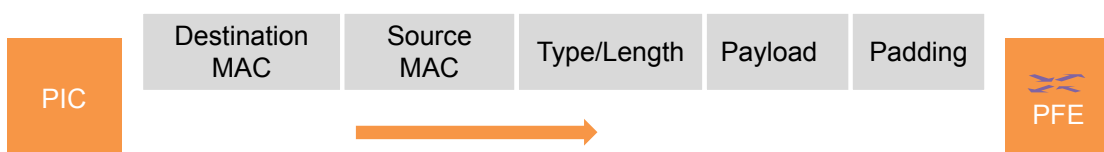
Ethernet standards set the minimum interframe gap at 12 bytes, all filled with 1s. GE interfaces may reduce the interframe gap to 64 bits. 10GE interfaces may reduce the interframe gap to 40 bits.

Ethernet standards also specify that the preamble is 7 bytes, all of them being 10101010. The start-frame delimiter is 1 byte: 10101011.

The conditions of a data frame being discarded varies according to which of the above three standards is broken.

A PIC will allow for variance in interframe gap (for example if it is not all 1s). The preamble and start-frame delimiter however, must comply with standards, otherwise, they will be processed as a part of an interframe gap.

After a PIC converts data signals to a data frame and verifies validity, it sends the frame content to the PFE. The frame's content excludes the interframe gap, preamble, and start-frame delimiter.

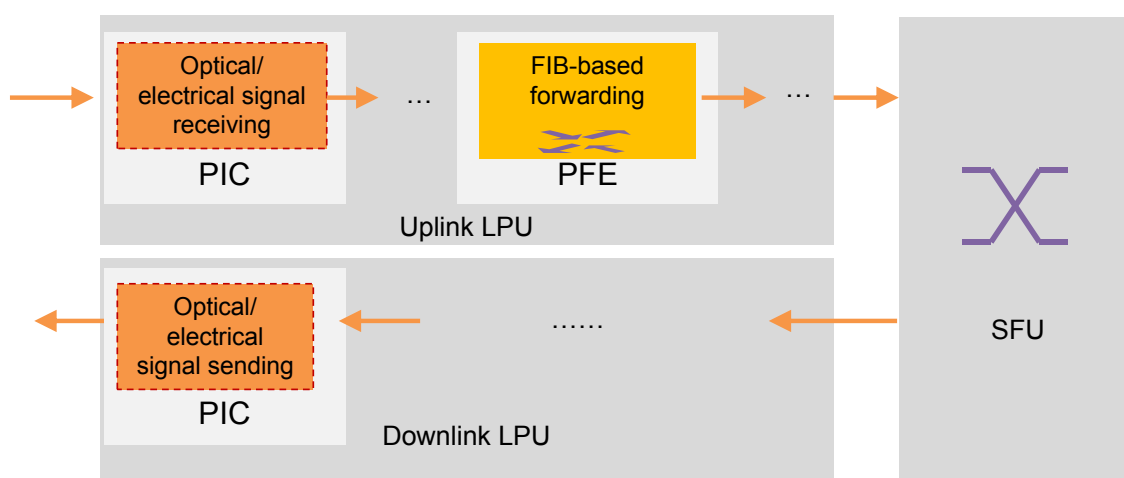




Note:

The PIC type determines the LPU service type. For example, if 4 x 2.5G POS PICs are integrated on an LPU, the LPU supports 4 x 2.5G POS services; if 10 x GE PICs are integrated on an LPU, the LPU supports 10 x GE services. After a PIC is installed on an LPU, the PFE can learn the PIC type and parse PIC-sent data based on the service type.

A data packet forwarded by a PFE and switched by an SFU will arrive at a downlink LPU, which also holds a PIC.



The downlink PIC performs the following functions:

- Calculates the FCS based on data frame content.
- Adds the interframe gap, preamble, start-frame delimiter, and FCS to the data frame.
- Converts the data frame to optical/electrical signals.
- Sends the signals through the outbound cable.

Packet Parsing

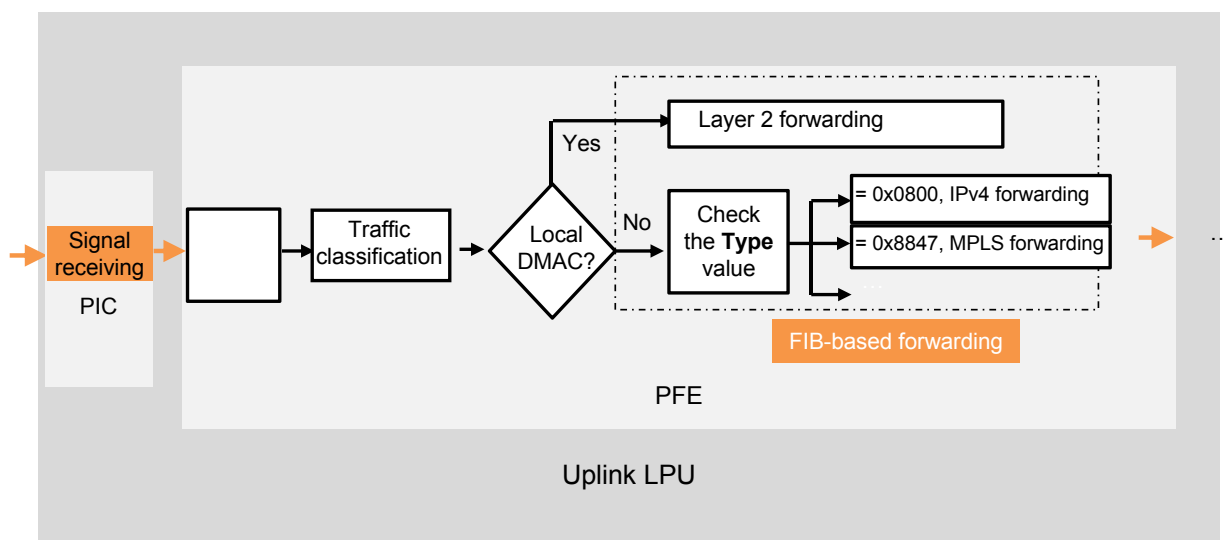
A Huawei router is diversely functional, enabling it to meet varying service requirements. Most functions may be enabled, disabled, or adjusted by engineers using commands. Upon receiving a data frame from a PIC, the PFE first parses the frame's Layer 2 header and then performs checking and processing based on the configuration.

All configuration commands are parsed by MPUs and then delivered to LPU. The LPU contains many tables, including the FIB, ingress port attribute table (IPAT), egress port attribute table (EPAT), ACL table, and traffic classification table. During LPU start up, an MPU begins to deliver configurations to the LPU. This is the process of setting table entry values on the LPU. Then, LPU components search for various entries and process the data based on found values. The LPU is updated in real time with configurations that have been added, deleted, or modified.

For example, if users require an Ethernet interface to provide access to services for VLANs 10 to 20, VLANs outside this range will be discarded. To meet this requirement, the **portswitch** command must be configured on the Ethernet interface. This command configures the interface to work in Layer 2 mode. The interface must be configured as a trunk interface and allow VLAN IDs 10 to 20 to pass. With these configurations delivered, the interface's Layer 2 bridge forwarding status is Enable, interface type is Trunk, and the VLAN ID in the IPAT ranges from 10 to 20. Now when a frame arrives, the PFE first determines the interface type is trunk. The PFE then checks whether the frame's Layer 2 header carries a VLAN tag. If no VLAN tag is present, the PFE discards the frame. If there is a VLAN tag, the PFE checks whether the VLAN ID in the tag ranges between 10 and 20. If the VLAN ID is outside the range, the PFE discards the frame.

The PFE then performs traffic classification, packet filtering, and redirection based on IPAT configurations. These processes will be detailed in later chapters.

If the frame matches configured parameters, the PFE searches the FIB and forwards the frame based on the forwarding action configured in the IPAT.



For example, when transmitting an Ethernet frame, the PFE will check the frame destination MAC address and then proceed with one of the following:

- If the destination of the MAC address is not the local device, the PFE will perform Layer 2 bridge forwarding.
- If the destination of the MAC address is the local device, the PFE performs forwarding based on the **Type** field of the frame header. These include IP, MPLS, and other forwarding protocols.

Note: If the forwarding status in the IPAT does not match parsed status, the frame will be discarded. For example, if IPv6 status is set to Disable in the IPAT and the interface receives an IPv6 packet, the interface will discard the packet.

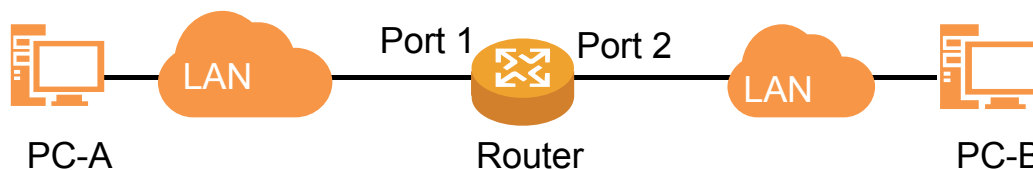


Note:

FIB-based forwarding does not apply to ARP, RARP, IS-IS, LLDP, LACP, or PPP control packets. When a PFE parses Layer 2 frame headers, it uses the **Protocol** field to determine that these protocol packets must be sent directly to the device CPU. FIB-based forwarding does not apply to protocol packets with multicast address destinations (ranging from 224.0.0.1 to 224.0.0.255).

Packet Encapsulating

Different packets use varying encapsulation methods. Illustrated in the following figure, an Ethernet packet is going to be sent from PC-A to PC-B. This is the simplest of IP forwarding scenarios, with the packet being sent from PC-A to PC-B over an intermediate router. The router is PC-A's gateway.



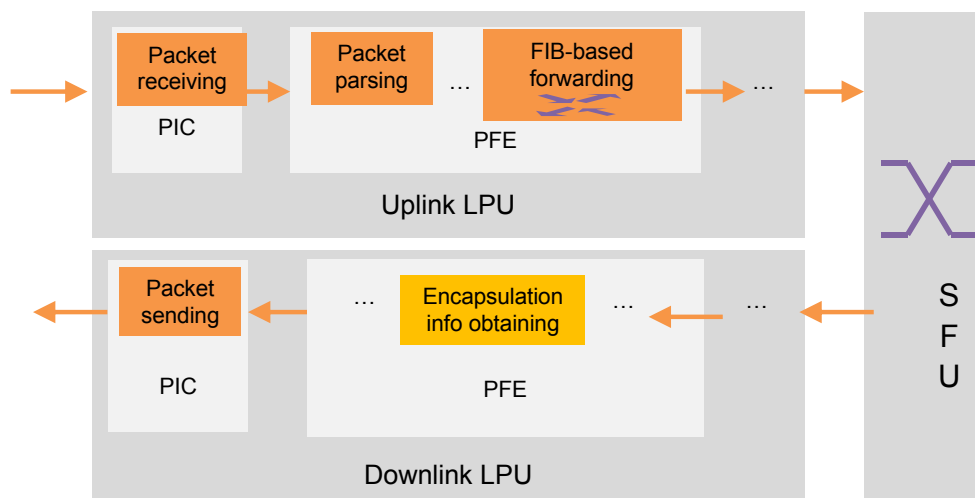
The characteristics of the packet are as follows:

- Destination IP address is PC-B's IP address.
- Source IP address is PC-A's IP address.
- Packet type is IPv4.
- Destination MAC address is the routers Port 1 MAC address.
- Source MAC address is PC-A's MAC address.

The forwarding process is as follows:

1. Upon receiving this packet, the router discovers the destination MAC address belongs to Port 1, indicating that the router needs to parse the packet further.
2. The router continues to parse the packet and finds the **Type** field is 0x800, meaning it must be forwarded in IPv4 format.
3. The router then searches the FIB table and discovers that the packet destination is outside the router, and it needs to be sent to Port 2. The router will not parse content following this destination IP address.
4. The router replaces the destination and source MAC address with PC-B's and Port 2's MAC addresses, respectively. This is known as encapsulation.
5. The router then sends the packet to the PIC. The packet is then sent out from Port 2 by the PIC.

Information that is added to a packet to be sent is known as encapsulation information. The router obtains the following encapsulation information, namely the source and destination MAC addresses from the downlink LPU's PFE in the following ways: ARP table for IP-MAC mapping and EPAT for outbound interface-MAC mapping.



An IP packet forwarding scenario with both source and destination MAC addresses encapsulated follows this process:

1. The uplink PFE obtains the outbound interface of the packet from the FIB table.

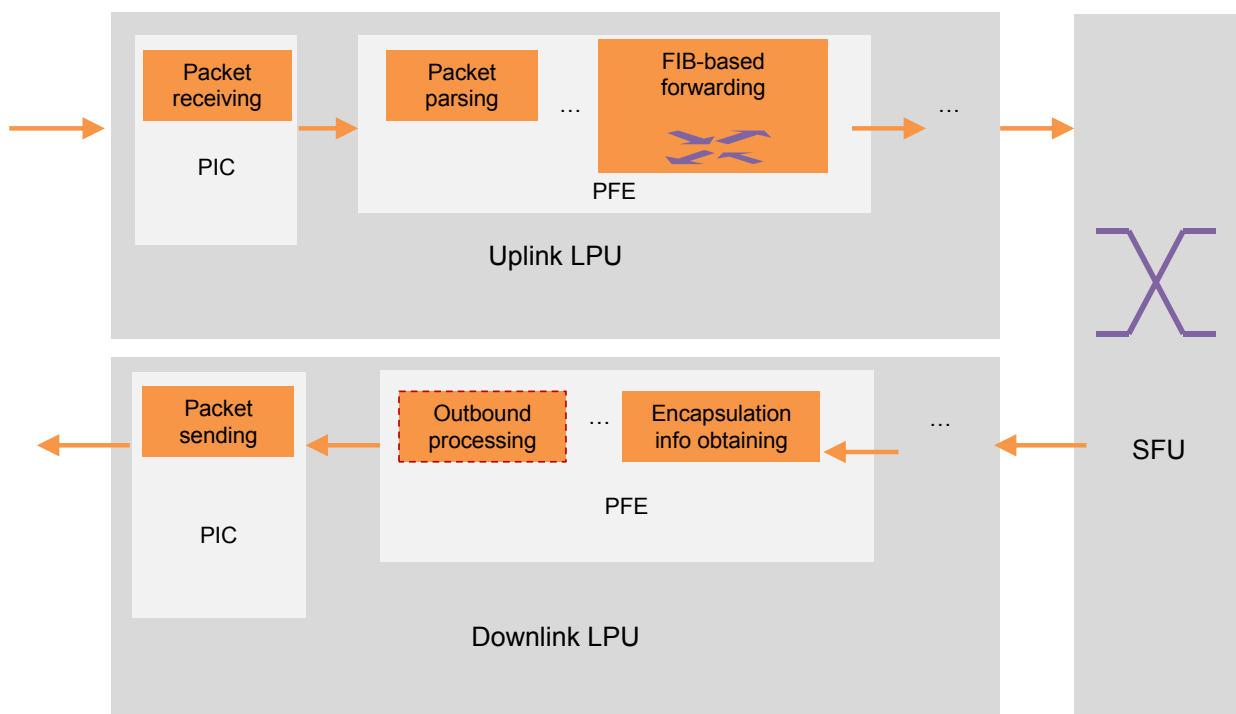
2. The packet arrives at the downlink PFE, the PFE looks up the ARP table and obtains the destination MAC address based on the destination IP address.
3. The PFE looks up the EPAT and obtains the source MAC address based on the outbound interface.

Note: Different scenarios may require additional encapsulation information.

For example, in a QinQ scenario, VLAN tags must be added. In an MPLS scenario, MPLS labels must be added. All encapsulation is performed on the downlink LPU's PFE.

Outbound Processing

Similar to uplink processing, the PFE must perform outbound checking and processing based on the EPAT before an encapsulated data frame is sent to the downlink PIC. For example, the PFE will check whether the frame length exceeds the MTU of the outbound interface. If exceeded, the PFE may fragment the frame or performs other operation. For details about MTU, see Special Topic - MTU(V2.0).



Chapter 3

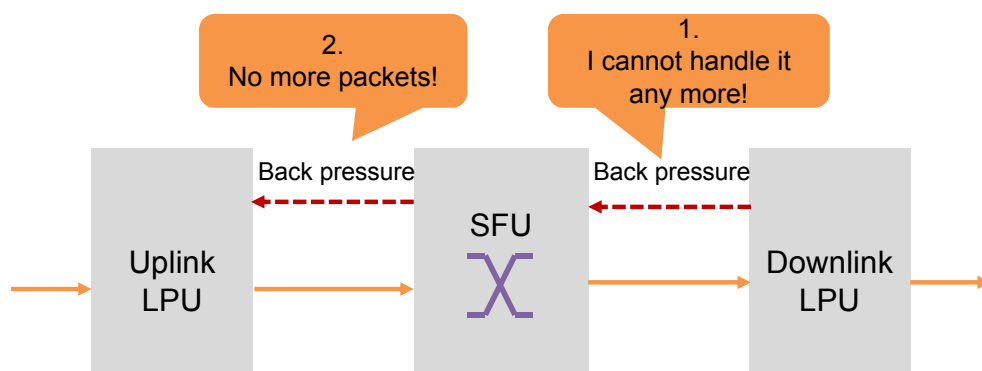
Traffic Control

In addition to packet receiving, parsing, forwarding, switching, encapsulating, and sending functions, a router has another important function: traffic control. Traffic control consists of the following mechanisms:

- Back pressure
- Queue
- Traffic policing

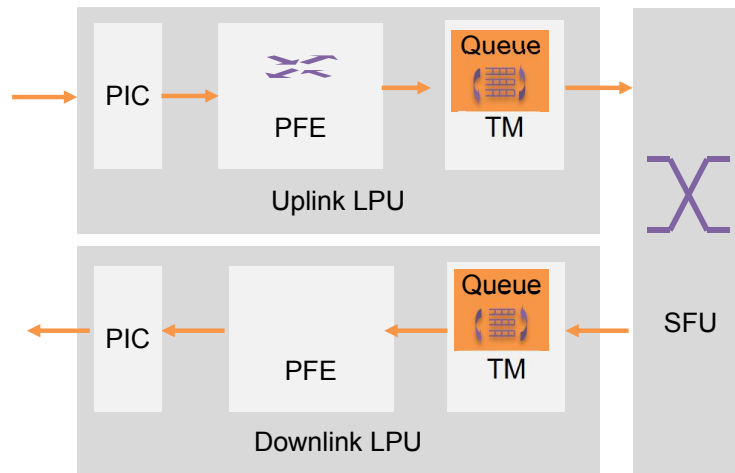
Back Pressure

In a typical link, the uplink LPU processes packets and sends packets to the SFU, and all packets are transmitted through the SFU. Therefore, the SFU is the core element that determines a router's performance and must constantly be in the non-blocking state. The forwarding capacity of an SFU is equal to the sum of all LPUs' forwarding capacities. However, the downlink LPU's forwarding capacity may be lower than the SFU's forwarding capacity. What happens if the downlink LPU is unable to handle the traffic transmitted from the SFU? To resolve this issue, the back pressure mechanism is configured on the router. With this mechanism, the downlink LPU back pressures packet flow to inform the SFU that the traffic rate is beyond its capacity. The SFU stops sending packets to the downlink LPU and buffers the packets to be transmitted.



Queue

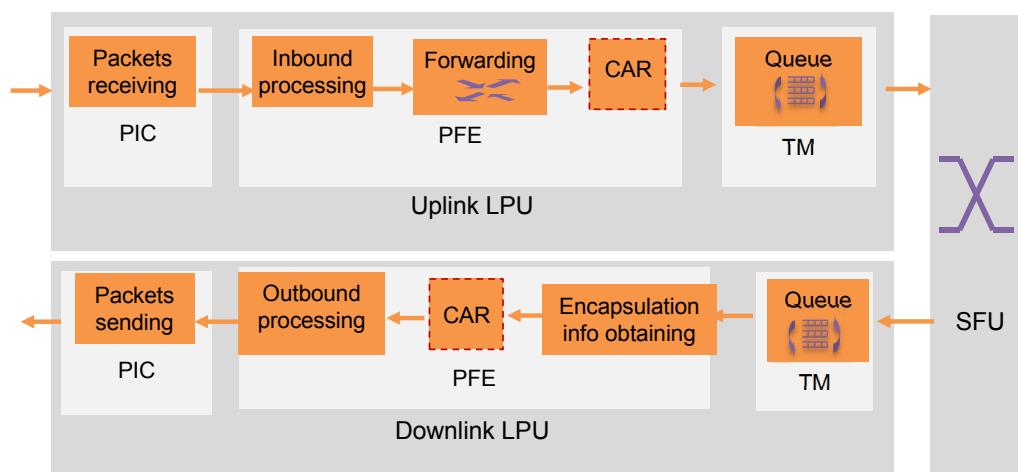
In addition to the back pressure mechanism, a high-end Huawei router is also equipped with a Traffic Management (TM) chip, which provides high-speed buffer resources and buffers packets in queues in the case of traffic congestion. When the traffic congestion is relieved, the TM chip sends the buffered packets out of queues to the SFU based on a specific rule. If the TM chip cannot buffer all packets, it drops packets based on a specific rule.



Similarly, the rate at which packets are sent to the downlink LPU may exceed the downlink LPU's forwarding capacity. Therefore, the downlink LPU is also equipped with a TM chip to buffer packets and manage queues. If traffic congestion occurs, the TM chip on the downlink LPU temporarily buffers packets in queues. After the traffic congestion is relieved, the TM chip sends packets out of queues to the outbound interface based on a specific rule.

Traffic Policing (CAR)

In addition to the back pressure and queuing mechanisms, there is another important traffic control mechanism: traffic policing. Traffic policing enables an interface to drop excess packets to ensure that the traffic rate conforms to the allowed maximum bandwidth on the interface. Currently the Committed Access Rate (CAR) technology is used to implement traffic policing. The PFE on the uplink or downlink LPU implements CAR for incoming or outgoing packets.



Protocol packets that are sent to the CPU or delivered by the CPU are not processed by CAR so that these protocol packets will not be dropped even in a traffic burst. (The packets to be sent to the CPU will experience CP-CAR processing. For details, see Chapter 7 Journey of Protocol Packets.)

CAR is not implemented for protocol packets in case they are dropped due to traffic congestion. The queue mechanism also has protection measures to prevent protocol packets from being dropped. Specifically, protocol packets are allocated high service classes and put in high-priority queues for preferential scheduling.



NOTE:

For detailed information about queue, CAR, traffic classification, and service class, see Chapter 4 QoS Basics.

Chapter 4

QoS Basics

Packet forwarding on a router involves many quality of service (QoS) concepts, such as behavior aggregate (BA) classification, multi-field (MF) classification, CAR, queue, and buffer. This chapter provides QoS basics to help you better understand QoS on a router.

Why Do We Need QoS?

When a new product is launched, users and manufacturers are typically less concerned about quality issues. After competitor products go on sale, users start to compare the qualities of the products. This scenario applies to products including the IP network. Traditionally, the best effort (BE) model is used on an IP network so that the network makes its best attempt to send packets but without any guarantee for performance.

With continuous technology improvement and fierce product competition, users have higher requirements on network quality. To meet these demands, multiple IP QoS service models are introduced, among which integrated service (IntServ) and differentiated service (DiffServ) models are typically implemented.

IntServ

IntServ requires users to apply for specific levels of service from the network before sending packets. After receiving these requests, the network reserves sufficient resources for these requests. IntServ works like a shuttle bus, with seats reserved for every ticketed passenger, and drives off even if the seats are not fully occupied.

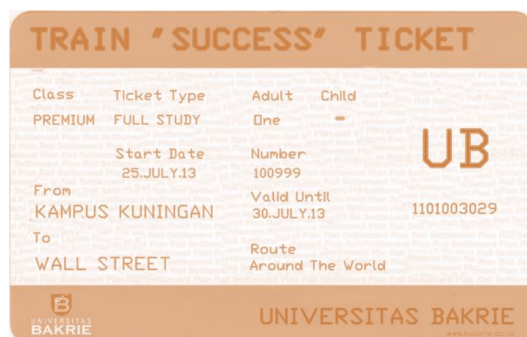
In addition, the vehicle company has to maintain great amount of booking information:

Booking ticket number	
Renter:	<input type="text"/>
Tel:	<input type="text"/>
Address:	<input type="text"/>
Location for getting a vehicle:	<input type="text"/>
Location for returning a vehicle:	<input type="text"/>
Renting period: From	<input type="text"/>
To	<input type="text"/>
Total:	<input type="text"/> days
Booked vehicle type:	<input type="text"/>
Rental:	<input type="text"/>
Advance payment:	<input type="text"/>
Balance payment:	<input type="text"/>
Renting Mode:	<input type="checkbox"/> Paid-driver <input type="checkbox"/> Self-drive

Due to these defects, the IntServ model has not been used on IP networks since 1990s. Currently the DiffServ model is widely used on IP networks.

DiffServ

The DiffServ model classifies network traffic into multiple classes for differentiated processing. To be specific, the DiffServ model implements traffic classification first and then allocates different identifiers to different classes of packets. After a network node receives these packets, it simply identifies these identifiers and processes packets based on the actions corresponding to these identifiers. The DiffServ model and train ticket service system is similar. A train ticket marks the service that you book: soft sleeper, hard sleeper, hard seat, or no seat. You get on a train and enjoy the specific service marked on your ticket. On an IP network, the relationship between an identifier and a packet is analogous to the relationship between a train ticket and a passenger.

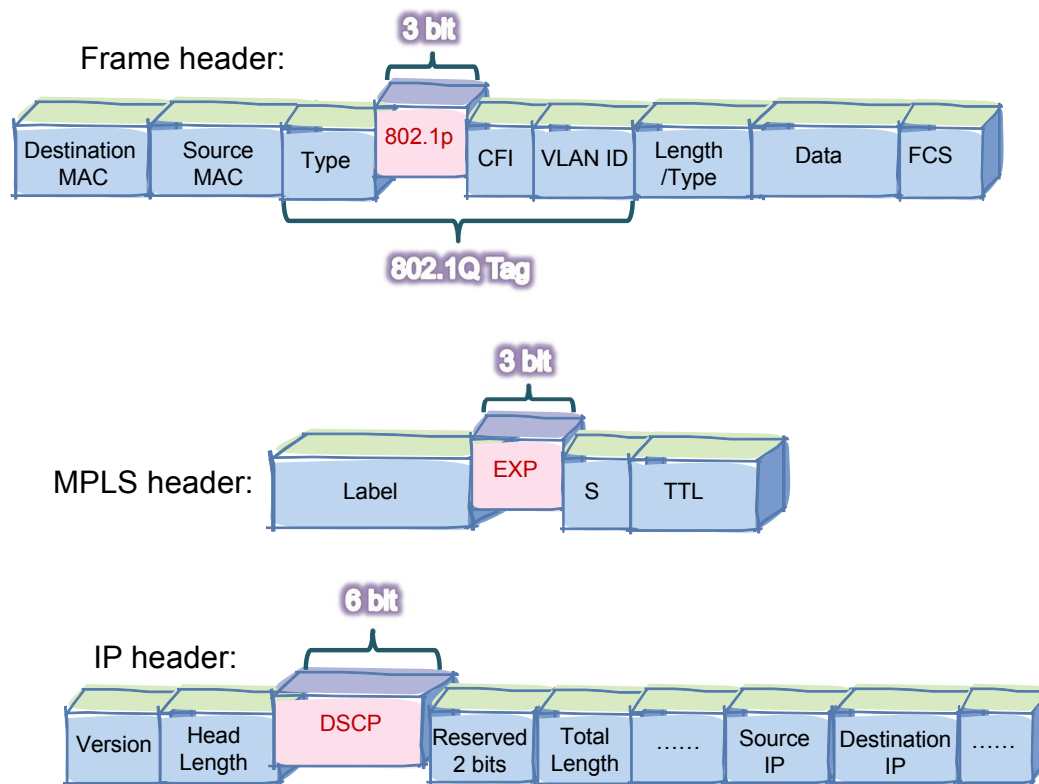


DSCP and PHB

Differentiated services code point (DSCP) and per-hop behavior (PHB) are important concepts of the DiffServ model.

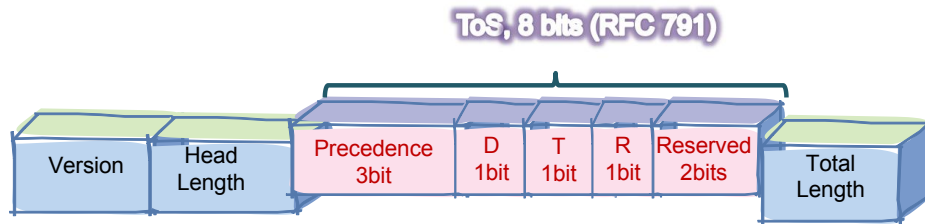
DSCP

As stated before, the identifier in a packet header is like a train ticket. Similarly, packets have various packet headers, for example, frame headers at Layer 2, MPLS headers at Layer 2.5, and IP headers at Layer 3. The 802.1p, EXP, and DSCP fields (identifiers) are used in frame, MPLS, and IP headers, respectively.

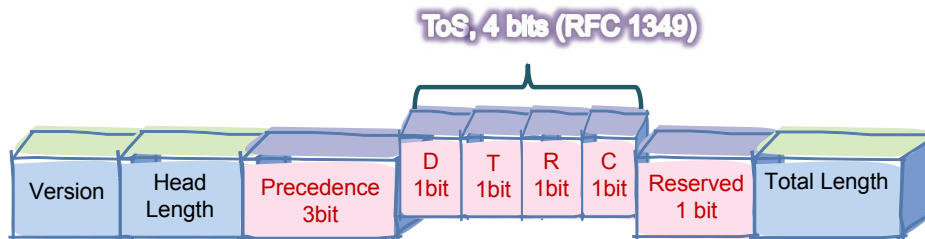


A packet may travel over multiple networks, such as an Ethernet, MPLS network, and IP network, and the forwarding behaviors vary according to networks. For example, on an Ethernet, a network node parses only the Ethernet header of a packet but ignores its MPLS or IP header. This is why each layer has its own identifier.

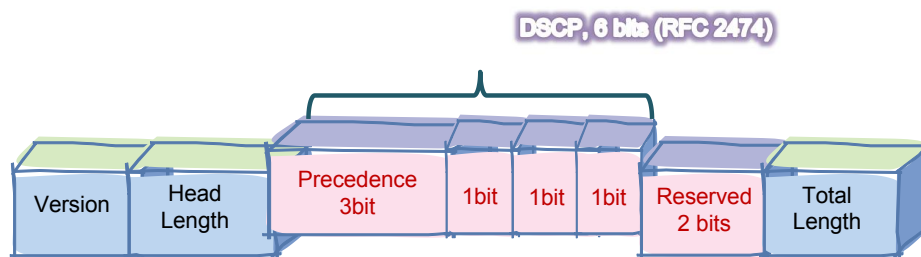
In addition to identifiers such as the 802.1p, EXP, and DSCP fields, there is another identifier called type of service (ToS), which is also a field in IP headers. Initially RFC 791 defines no DSCP value but an 8-bit ToS value and uses 3 bits from these 8 bits to indicate the Precedence. There are eight types of precedence then.



RFC 1349 redefines the ToS field and renames one of the reserved 2 bits as the C field. The D, T, R, and C fields constitute the ToS field. Due to these differences, the RFC version needs to be stated when speaking of ToS. Huawei routers comply with RFC 1349.



Subsequently, RFC 2474 redefines the 8 bits and renames the specific 6 bits as the DSCP field.



PHB

As its name suggests, per-hop behavior (PHB) refers to the behavior performed by packets on each hop. However, this does not mean that PHB is a specific action, such as traffic scheduling, packet dropping, traffic policing, traffic shaping, and re-marking. In fact, PHB only defines a forwarding behavior that is visible to users without any specific actions. In other words, PHB identifies the classes of behaviors similar to the star-rating of hotels, such as 3-star, 4-star, and 5-star. RFCs define four types of PHBs: CS, EF, AF, and BE, each of which corresponds to a DSCP value. PHB classifies packets as CS, EF, AF, and BE based on the service characteristics to which users are most sensitive, such as the delay, jitter, and packet loss rate.

- The best effort (BE) PHB focuses only on whether packets can reach the destination, regardless of the transmission performance. Traditional IP packets can be transmitted in BE mode. The BE PHB is used on IP networks by default and all routers must support the BE PHB.

- The assured forwarding (AF) PHB applies to services that require short delay, low packet loss rate, and high reliability, such as video, voice, and enterprise VPN services.
- The expedited forwarding (EF) PHB applies to real-time services that require short delay, low jitter, and low packet loss rate, such as video, voice, and videoconferencing.
- The class selector (CS) PHB indicates the same service class as the IP precedence value. RFC 2474 reserves all values of the XXX000 format to allow DiffServ-incapable devices that only parse the three leftmost bits in the ToS field to be compatible with other devices.

Typically an AF or CS PHB carries a suffix, such as AF11, AF22, CS6, and CS7, while a BE or EF PHB carries no suffix. This is because a BE or EF PHB corresponds to only one DSCP value and a CS or AF PHB corresponds to multiple DSCP values. Currently, four AF classes with three levels of drop precedence in each AF class are defined for general use. AF is expressed in the format of AF1x to AF4x, with x indicating the drop precedence and ranging from 1 to 3.

For example, assume that four community networks are connected to the same edge router on an ISP network. If a community sends a large number of FTP packets, traffic congestion may occur, affecting the FTP transmission of other communities. To address this issue, set the maximum FTP traffic rate to 500 Mbit/s for each community. After that, re-mark the DSCP field of packets on each inbound interface to compensate for the scenario where the traffic rate of one community exceeds 1 Gbit/s. Specifically, traffic transmitted at a rate of 500 Mbit/s or lower is marked AF11, traffic transmitted at a rate of 500 Mbit/s to 1 Gbit/s is marked AF12, and traffic transmitted at a rate higher than 1 Gbit/s is marked AF13. When traffic congestion occurs, the AF13 traffic is dropped first. If traffic congestion persists after the AF13 traffic is dropped, the AF12 traffic is dropped. If traffic congestion is still detected after that, the AF11 traffic has to be dropped at last. In this manner, all communities are fairly treated.

PHB = Service Quality?

PHB reflects the service class of packets but not the service quality. In other words, CS is higher than BE in service class, which does not mean that the service quality of CS traffic is higher than that of BE traffic. PHB is only a hop-by-hop behavior, whereas service quality is an end-to-end service guarantee. Service quality is typically measured based on the following specifications:

- Bandwidth/throughput
- Delay

- Delay variation (jitter)
- Packet loss rate
- Availability

In addition to PHB, many other factors affect service quality, such as link bandwidth, device processing capability, network stability, and transmission distance.

PHB = Queue?

There are BE, AF1, AF2, AF3, AF4, EF, CS6, and CS7 queues on Huawei routers.

This does not mean that PHB is equal to all queues. To be specific, the name of a queue does not reflect the priority of the queue in service class. Similarly, if all hard seats in a hard seat coach are replaced with hard sleepers, this coach is actually a hard sleeper coach although it is still named a hard seat coach.

Queues can be treated in a similar fashion. For example, if strict priority (SP) scheduling is implemented for a BE queue (rarely happens) and weighted fair queuing (WFQ) scheduling is implemented for the other seven queues, the BE queue has the highest priority among all other queues in service class. In this case, the PHB of the BE queue is no longer BE. (The SP and WFQ concepts will be provided in the Queue Mechanism.) Although the name of a queue does not represent its PHB, a queue is still named similarly to PHB as it can vividly show a queue's priority. If the queues were named queue 1 to queue 8 instead, it would be hard figure out what these queues are.

How Are Packets Placed in Queues Based on DSCP Values?

Both the 802.1p and EXP fields are 3 bits long, which correspond to exactly eight values and can have one-to-one mappings with eight PHBs or queues. What about DSCP values? How do DSCP values map eight PHBs or queues as a DSCP field corresponds to 64 values? Behavior aggregate (BA) classification is used to resolve this problem.

BA Classification (Mapping and Reverse Mapping)

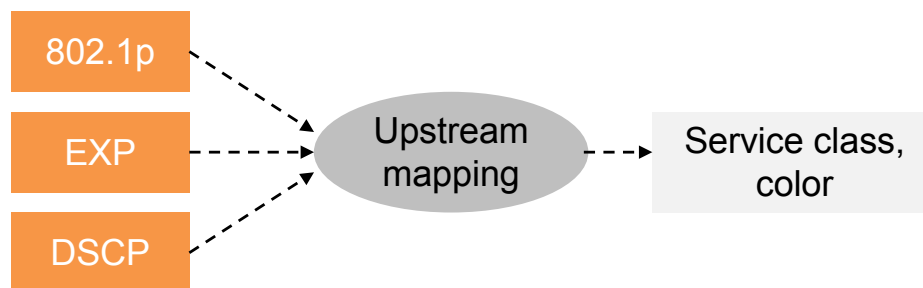
Traffic classification consists of behavior aggregate (BA) classification and multi-field (MF) classification. BA classification enables a device to simply identify the traffic that has the specific priority or service class for mapping between external and internal priorities.

For example, BA classification enables a device to classify packets based on the DSCP values of IP packets, EXP values of MPLS packets, or 802.1p values of VLAN frames. MF classification has a more complex classification rule and enables a device to classify packets in a finely granular way based on fields other than priority identifiers, such as 5-tuple, MAC address, protocol number, label, or TTL information.

As stated before, in the DiffServ model, packets are classified and marked, and PHBs are implemented for classified packets. Regardless of whether BA or MF classification is implemented, packets are classified based on packet headers' fields, which are obtained by parsing packet headers. It is impossible for a device to parse packet headers every time a QoS operation is implemented, such as placing packets in queues, dropping packets, and sending packets out of queues.

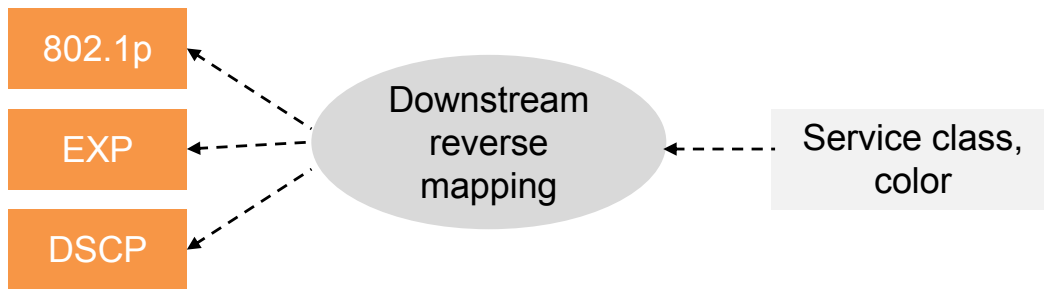
To resolve this problem, designers use two internal markers: service class and color, which are also called scheduling precedence and drop precedence, respectively. When a device parses a packet header, the device sets these two internal markers for the packet based on the packet header priority. In this manner, the device only needs to read these two markers before implementing any QoS operation. To be specific, a device implements a QoS operation based on the service class and color of packets.

The default PHB on an IP network is BE. Therefore, the initial values of the two internal markers are BE and Green. If the **trust upstream** command is run on an inbound interface, the interface maps the external priorities (802.1p, EXP, and DSCP) of packets to the internal priorities (service class and color). This process is called mapping.



If re-marking or re-marking after CAR is configured on a device, the device re-marks the service class and color of packets, regardless of whether the inbound or outbound interface receives the packets. After that, the device implements QoS operations for packets based on the service class and color. After QoS operations are complete, the downstream board (outbound interface) maps the internal priorities of packets to the external priorities.

This process is called reverse mapping. Reverse mapping is an optional configuration and is unnecessary if the external priorities of packets must remain unchanged.



MF Classification (Traffic Policy)

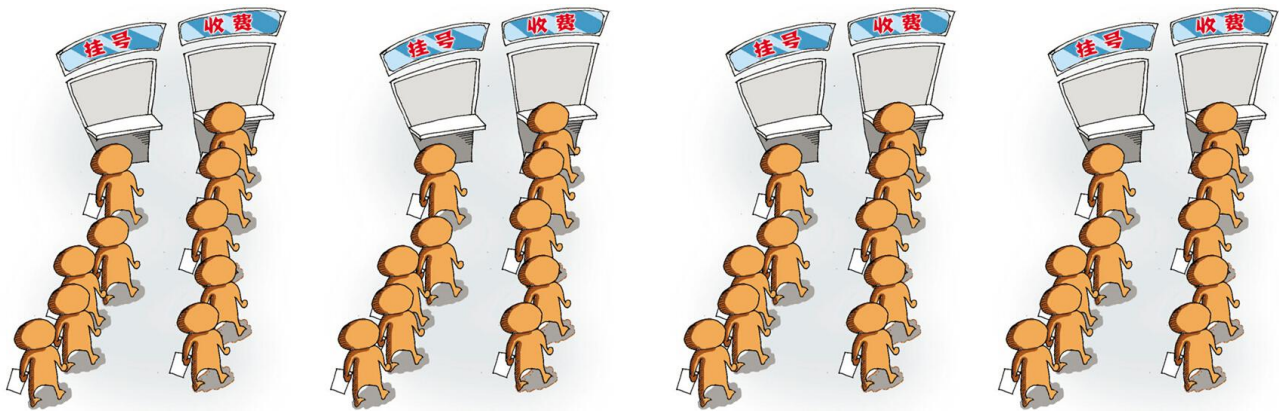
Multi-field (MF) classification enables a device to classify packets based on complex rules, such as 5-tuple. After traffic classification is complete, various behaviors must be implemented for classified packets (traffic classifiers). Therefore, a traffic policy is used to associate traffic behaviors with traffic classifiers. A traffic policy is configured in a profile which supports batch modification and reduces configuration workloads.

A traffic policy profile covers the following concepts:

- Traffic classifier: defines if-match clauses to match packets.
- Traffic behavior: defines actions for specific traffic classifiers, such as re-marking, redirecting, balancing, fragmenting, rate limiting, and traffic statistics collecting.
- Traffic policy: associates traffic classifiers with traffic behaviors. The traffic policy takes effect after being applied to an inbound or outbound interface.

Congestion Management (Queuing)

In addition to its traffic classification and marking features, DiffServ has a queuing mechanism. When network congestion occurs on a device, the device buffers packets in queues and sends the packets from those queues when network congestion is relieved. While queues in everyday life are an inconvenience, the queuing mechanism in DiffServ is highly efficient, with each interface on a Huawei router possessing eight queues, a situation comparable to having eight registration windows at a hospital.



Scheduling Algorithms

First In First Out (FIFO) allows packets that come earlier to enter the queue first.

However, a router interface can process only one of the eight queues at a time. The scheduling algorithm determines which queue the interface will preferentially process.

Strict Priority (SP) schedules packets based on queue priorities. Packets in low-priority queues can be scheduled only after all packets in high-priority queues have been scheduled. To prevent low-priority queues from being neglected, rate limiting is generally implemented for high-priority queues.

Round Robin (RR) schedules multiple queues in ring mode. If the queue on which RR is performed is not empty, the scheduler sends one packet out of the queue. If the queue is empty, it is simply skipped. Because of this, a lot of time may elapse before high-priority queues can be scheduled. To address this, designers introduced Weighted Fair Queuing (WFQ). For example, let us say there are three non-VIP queues with the weight rate being 5:3:1. The scheduler will send five packets out of the queue with a weight value of 5 at a time, three with a weight value of 3, and one with a weight value of 1. Numerous scheduling algorithms not described in this document have been formulated for the queuing mechanism. This is why the queuing mechanism is more efficient than the process of registering at a hospital.

Congestion Avoidance (Drop Policy)

In everyday life, doctors do not see any more patients after the number of registered patients reaches a maximum threshold, say, for example, 200 registrations, or outside of normal working hours that extend from 7 a.m. to 11 a.m. and from 2 p.m. to 4 p.m.. Similarly on a router, when traffic congestion intensifies and queues that buffer packets are almost full, a router will use a drop policy to counteract the effects of congestion.

Packet dropping! UDP is highly concerned about packet dropping, but TCP has no such worries because TCP allows for packet retransmission.

Currently two drop policies exist: tail drop and Weighted Random Early Detection (WRED). Tail drop is the traditional congestion avoidance mechanism used to drop all newly arriving packets when congestion occurs. With tail drop mechanisms, all newly arriving packets are dropped when congestion occurs, causing all Transmission Control Protocol (TCP) sessions to simultaneously enter the slow start state and packet transmission to slow down. Then all TCP sessions restart their transmission at roughly the same time and when congestion occurs again, another burst of packet drops is triggered, whereupon all TCP sessions enter the slow start state once more. This cycle repeats itself again and again. This phenomenon is called TCP global synchronization. In short, tail drop is not the optimum solution for either TCP or User Datagram Protocol (UDP).

To better serve TCP and UDP, WRED is used. WRED sets two lines for each queue, as shown in the following figure.



Please stand behind the yellow safety line while waiting for the train.

When the length of a queue is lower than the threshold marked by the yellow line, no packets are dropped. When the length of a queue exceeds the threshold marked by the yellow line, newly arriving packets are randomly dropped at a rate increasing with the queue length. When the length of a queue exceeds the threshold marked by the red line, all newly arriving packets are dropped.

Tail drop applies to SP queues for services that have high real-time performance demands. Tail drop drops packets only when the queue overflows and therefore it provides the highest bandwidth for real-time services when traffic congestion occurs. WRED is generally applied to WFQ queues. WFQ queues share bandwidth based on weight and are prone to traffic congestion. Using WRED for WFQ queues effectively resolves TCP global synchronization when traffic congestion occurs.

Some users, however may still feel hard done by. For example, let us say Jerry and Tom are surfing the net. Jerry sends packets at a rate of 2 Mbit/s, and Tom sends packets at a rate of 200 Mbit/s. If traffic congestion occurs, Jerry thinks that Tom's packets should have been dropped first as Tom sends more packets and thus contributing more to traffic congestion. To resolve this, designers introduced drop precedence. The Internet Engineering Task Force (IETF) defines three types of drop precedence: red, yellow, and green, indicating the order in which packets buffered in queues are dropped during traffic congestion. In the case of Jerry and Tom surfing the net at rates of 2 Mbit/s and 200 Mbit/s respectively, you can set an upper limit of 100 Mbit/s. If Tom sends packets at a rate lower than 100 Mbit/s, Tom's packets will be colored green and thus will not be dropped. If Jerry sends packets at a rate higher than 100 Mbit/s, Jerry's excess packets will be colored red and will therefore be dropped preferentially.

Rate Limiting (CAR and Traffic Shaping)

Rate limiting is one of the most important QoS mechanisms. Rate limiting restricts the rate at which packets are sent to or from a router.

Both CAR and traffic shaping use the token bucket to measure the traffic rate, but differ in terms of packet processing. Traffic shaping is implemented based on the queuing mechanism, so it buffers excess packets in queues and sends them out of queues only when traffic congestion has been relieved. CAR is not implemented based on the queuing mechanism and simply drops excess packets without buffering them.

Traffic classification (BA and MF classification), rate limiting (CAR and traffic shaping), congestion management (queuing), and congestion avoidance (drop policy) constitute the four QoS components. Chapter 5 describes how these components process traffic.

Chapter 5

QoS Processing

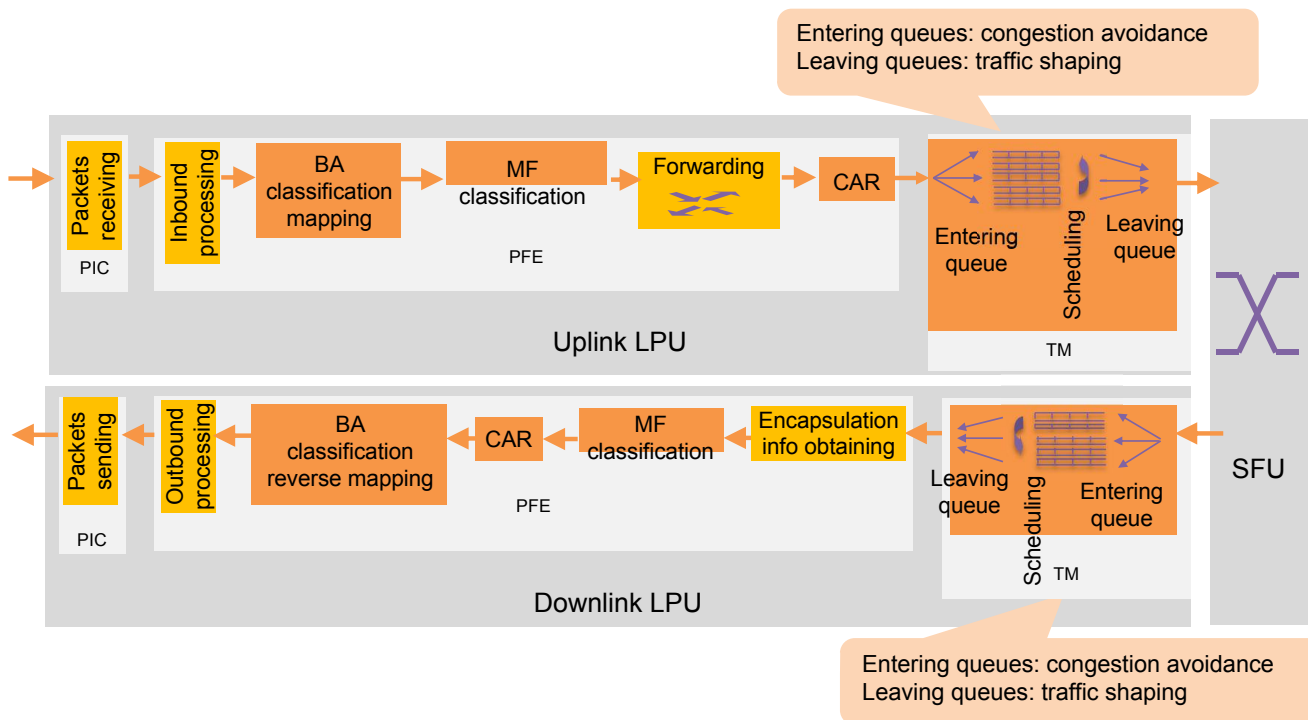
This chapter describes how QoS processes packets on the forwarding plane.

QoS Processing Sequence

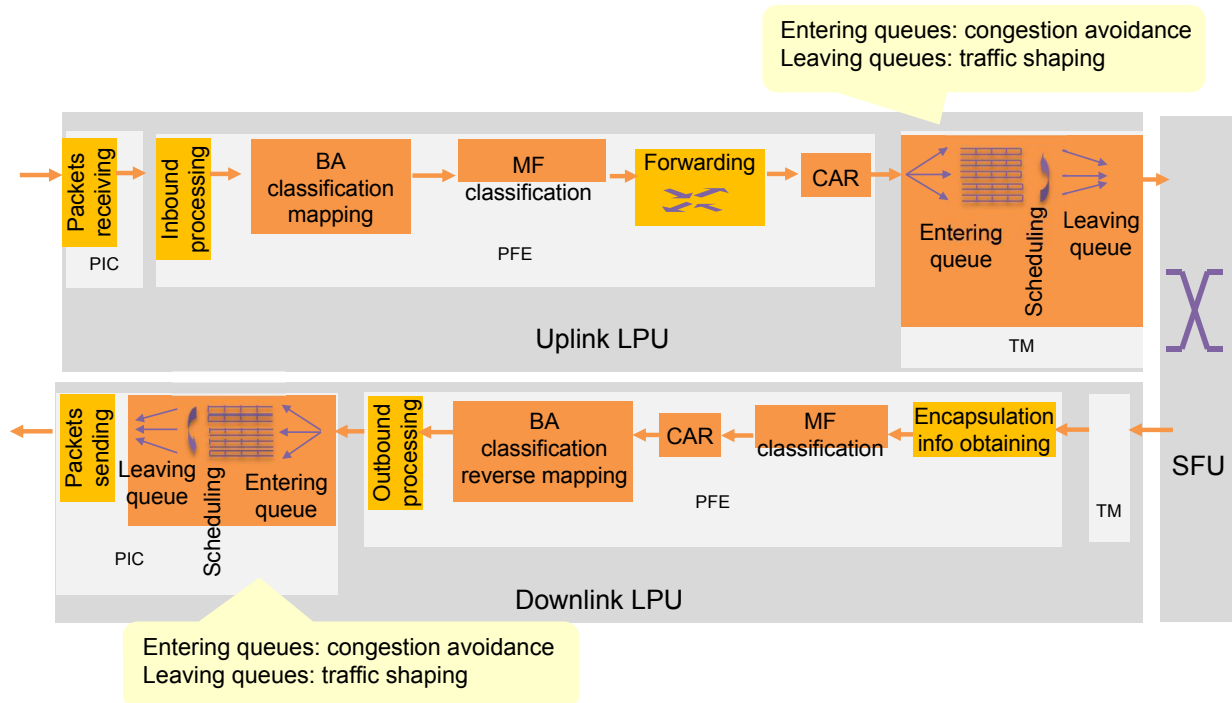
The four QoS components, traffic classification (BA and MF classification), rate limiting (CAR and traffic shaping), congestion management (queuing), and congestion avoidance (drop policy), process packets in a specific order.

PICs on some LPUs are equipped with egress Traffic Managers (eTMs), while those on other LPUs are not. The eTM-equipped PIC and non-eTM-equipped PIC differ only in terms of queue scheduling for downstream packets. To be specific, queue scheduling is implemented on the eTM when the PIC is equipped with an eTM and is implemented on the TM when the PIC is not equipped with an eTM.

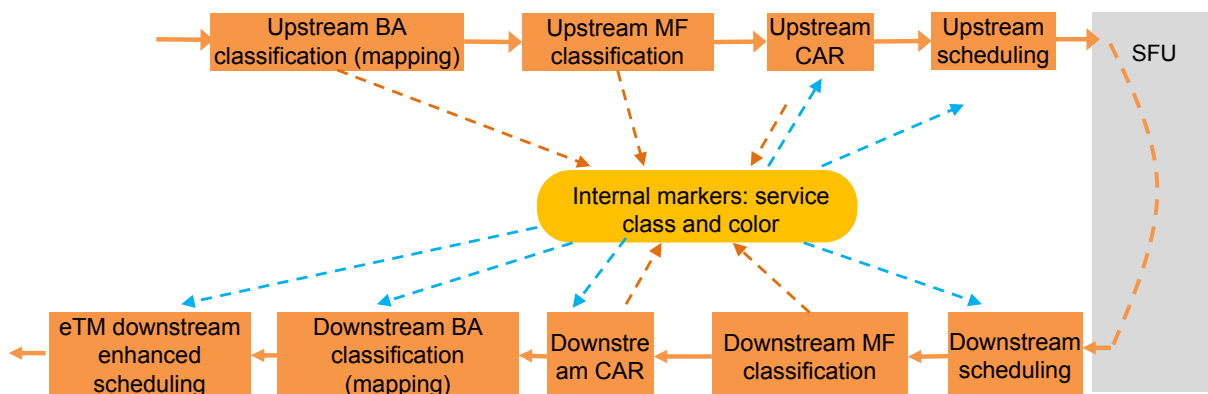
QoS processing when the PIC is not equipped with an eTM:



QoS processing when the PIC is equipped with an eTM:

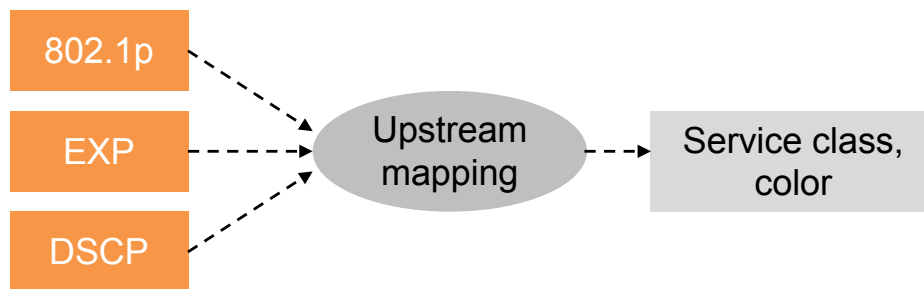


Traffic classification and marking provide a basis for DiffServ, and traffic policing, traffic shaping, congestion management, and congestion avoidance are implemented to provide DiffServ. As stated in Chapter 4 QoS Basics, Huawei routers set two internal markers for each packet: service class and color, which correspond to the scheduling precedence and drop precedence, respectively. QoS operations are performed for packets based on those two internal markers.

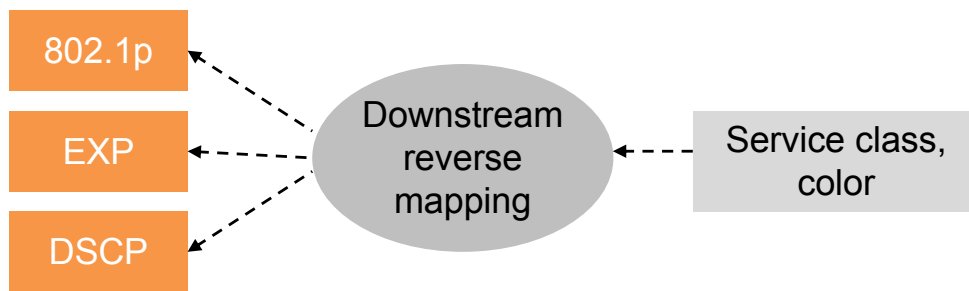


The process in detail is as follows:

1. The upstream PFE initializes the internal priority of packets (service class as BE and color as green).
2. The upstream PFE implements BA classification based on the inbound interface configuration. BA classification requires the upstream PFE to obtain the priority field value (802.1p, DSCP, or MPLS EXP) for traffic classification and reconfigure the internal priority of packets (service class and color).



3. The upstream PFE obtains packets' field information, classifies packets based on the field information, and performs behaviors, such as filter, re-mark, and re-direct, for packets based on the classification results. If the behavior is re-mark, the upstream PFE reconfigures the internal priority of packets (service class and color).
4. The upstream PFE obtains the outbound interface and next hop of the packets.
5. The upstream PFE implements CAR for packets based on the inbound interface configuration or MF classification configuration. If both interface-based CAR and MF classification-based CAR are configured, MF classification-based CAR takes effect. In a CAR operation, a pass, drop, or pass+re-mark behavior can be performed for incoming traffic. If the behavior is pass+re-mark, the upstream PFE reconfigures the internal priority of packets (service class and color).
6. Packets enter queues on the upstream TM for scheduling, and WRED is implemented for packets based on the color if configured.
7. Packets are switched to the downstream TM through the SFU.
8. (This step is skipped if the downstream PIC is equipped with an eTM) Packets enter queues on the downstream TM for scheduling, and WRED is implemented for packets based on the color if configured.
9. Packets enter the downstream PFE, and the downstream PFE obtains the encapsulation information of the packets.
10. The downstream PFE obtains packets' field information, classifies packets based on the field information, and performs behaviors, such as filter, re-mark, and re-direct, for packets based on the classification results. If the behavior is re-mark, the downstream PFE reconfigures the internal priority of packets (service class and color).
11. The downstream PFE implements CAR for packets based on the outbound interface configuration or MF traffic classification configuration. If both interface-based CAR and MF traffic classification-based CAR are configured, MF traffic classification-based CAR takes effect. In a CAR operation, a pass, drop, or pass+re-mark behavior can be performed for incoming traffic. If the behavior is pass+re-mark, the downstream PFE reconfigures the internal priority of packets (service class and color).
12. The priorities of outgoing packets are set based on service class and color for newly added packet headers and are modified for existing packet headers.



13. After processed by the downstream PFE, packets enter the downstream PIC.

- On a PIC not equipped with an eTM, the link layer interframe gap, preamble, start-frame delimiter, and Frame Check Sequence (FCS) are added to the packets so that the packets are forwarded to the physical links.
- On a PIC equipped with an eTM, after the link layer interframe gap, preamble, start-frame delimiter, and FCS are added to the packets, the PIC implements queue scheduling by placing packets in queues on the downstream eTM based on service class and implementing WRED for the packets based on the color if WRED is configured.

FAQs

Q: When Does the Priority Field Value of a Packet Change?

A: As stated before, downstream reverse mapping is implemented for packets based on the service class and color to add a new priority field or modify the existing priority field. Service class and color, however, are prone to change during the QoS process. You can determine whether the priority field value of a packet has been changed as follows:

1. Check whether service class and color have changed during the QoS process.
2. Check whether the mapping and reverse mapping rules are consistent.

For example, if DSCP value 12 is mapped to service class AF1 and color yellow in mapping, check whether service class AF1 and color yellow are mapped to DSCP value 12 in reverse mapping.

3. Check whether reverse mapping is implemented for outgoing packets.

On most boards, if the **remark** command is run in a traffic policy that is applied to incoming or outgoing packets, the priorities of incoming or outgoing packets are reconfigured based on command configuration. On an LPUF-21/40, however, if the **remark** command is run in a traffic policy applied to incoming packets, the priorities of incoming packets are not reconfigured, but the priorities of outgoing packets are reconfigured if this command is run in a traffic policy for outgoing packets.

Q: How Do I Check Whether Reverse Mapping Is Implemented for Outgoing Packets?

A: A device sets two markers on each interface to determine whether to implement reverse mapping for outgoing packets.

- Marker 1: BA on the inbound interface. BA is carried in internally added packet headers and transmitted to the outbound interface through the SFU.
- Marker 2: PHB on the outbound interface. Reverse mapping is implemented for packets only when both BA and PHB are enabled (on an LPUF-41/100, reverse mapping is implemented for packets when PHB is enabled, regardless of whether BA is enabled).

By default, BA is disabled, and PHB is enabled in V600R002 or earlier and is disabled in V600R003 or later.

- To enable BA, run the **trust upstream**, **remark**, **service-class**, **diffserv-mode pipe**, or **diffserv-mode short-pipe** command. The **diffserv-mode pipe** and **diffserv-mode short-pipe** commands apply only to ingress and egress PEs in MPLS scenarios. You can enable BA on any board other than an LPUF-21/40 or LPUF-41/100 by simply running the **diffserv-mode pipe** or **diffserv-mode short-pipe** command. To enable BA on an LPUF-21/40 or LPUF-41/100, run the **diffserv-mode pipe** or **diffserv-mode short-pipe** command and specific other commands.
- To disable BA, run the **service-class class-value color color-value no-remark** command on the inbound interface, or delete the preceding four commands.
- To enable PHB, run the **trust upstream** or **qos phb enable** command on the outbound interface.
- To disable PHB, run the **qos phb disable** or **undo trust upstream** command on the outbound interface.

Q: Which Priority Is Used for Mapping If DSCP, 802.1p, and EXP Values Are Carried in Packets?

A: It depends on the inbound interface configuration.

- If the **trust upstream** command is not run on the inbound interface, the external priorities of packets are not trusted and mapped to the default internal priority BE and green.
- If both the **trust upstream** and **trust 802.1p** commands are run on the inbound interface, priority mapping is implemented for VLAN-tagged packets based on their 802.1p values in the outer VLAN tags, and the external priorities of non-VLAN-tagged packets are mapped to the default internal priority BE and green.
- If only the **trust upstream** command is run on the inbound interface, priority mapping is implemented based on EXP values for MPLS packets and based on DSCP values for non-MPLS packets. For the other packets, if the device identifies a protocol packet, it maps the external priority of the protocol packet to the internal priority CS6 and green; if the device does not identify a protocol packet, it maps the external priority of the protocol packet to the internal priority BE and green.

Q: Which Priority Is Reconfigured During Reverse Mapping If DSCP, 802.1p, and EXP Values Are Carried in a Packet?

A: It depends on the type of the downlink LPU and the inbound interface configuration.

- If not both BA and PHB are enabled, the priorities of packets are not reconfigured.
- If both BA and PHB are enabled and the **trust upstream** and **trust 802.1p** commands are run on the inbound interface, 802.1p and EXP values are reconfigured for MPLS packets, and only 802.1p values are reconfigured for non-MPLS packets.
- If both BA and PHB are enabled but the **trust 802.1p** command is not run on the inbound interface, the priority reconfiguration varies according to board types. For details, see *Classification and Marking in QoS Special Issue*.

Q: How Is the Priority Field Set for a Newly Added Packet Header?

A: The settings vary according to board types. For details, see *Classification and Marking in QoS Special Issue*.

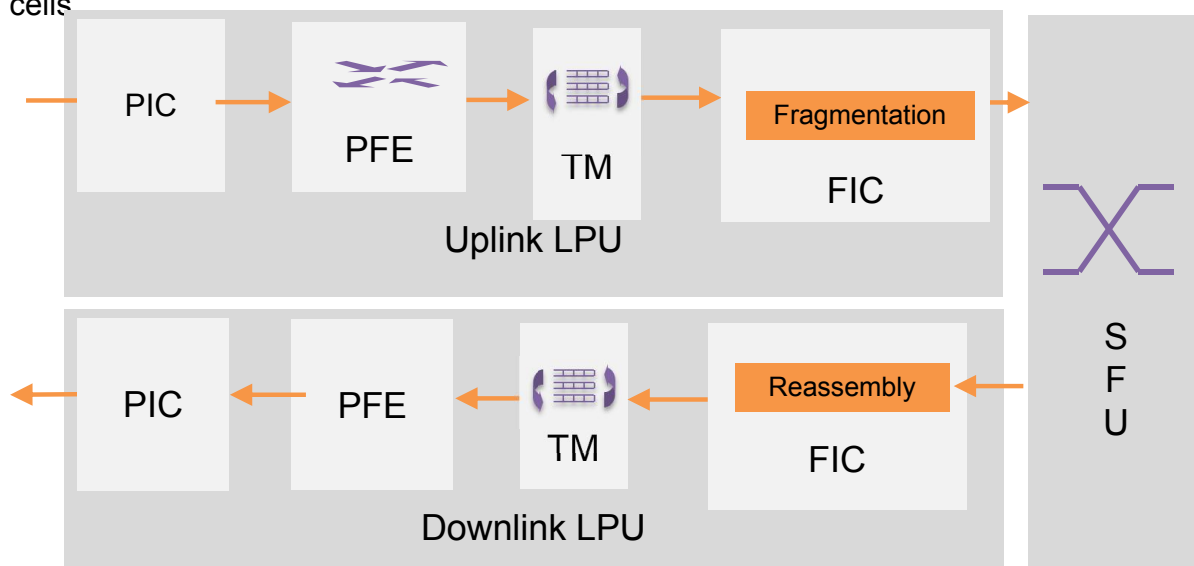
Chapter 6

Other Processing on the Forwarding Plane

In addition to packet receiving, sending, parsing, forwarding, exchanging, and encapsulation, as well as QoS processing, the forwarding plane of a router implements fragmentation, reassembly, multicast and broadcast replication, network address translation (NAT), packet filtering, and redirection.

Fragmentation and Reassembly

Huawei high-end routers have multiple SFUs for M+N backup. To implement load balancing, SFUs use a switching technology that is based on cells of fixed length. Before an LPU sends packets to an SFU, the fabric interface controller (FIC) of the LPU fragments them into cells of fixed length, which is similar to ATM cells. Upon receipt of the cells, the SFU switches them. Finally, the FIC of a downlink LPU reassembles the cells



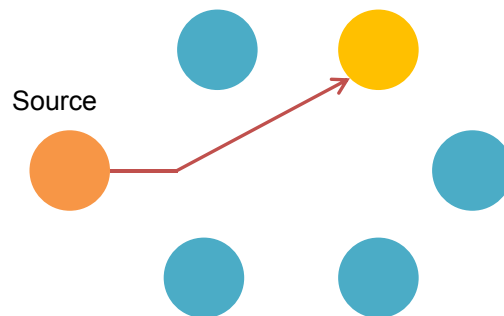
Note:

According to the type of data sent to an SFU, the SFU performs cell-based or packet-based switching. Packet-based switching does not require fragmentation or reassembly. However, load imbalance may occur among SFUs because data packets vary in length. Huawei high-end routers perform switching that is based on cells of fixed length, preventing such load imbalance.

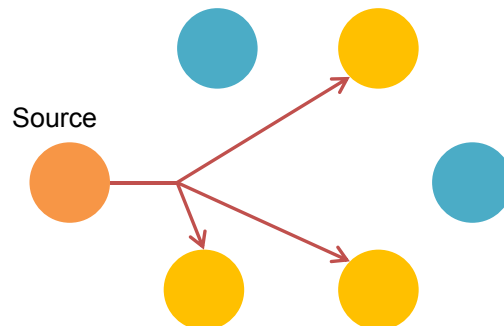
Multicast and Broadcast Replication

First, let's review the unicast, multicast, and broadcast communication modes.

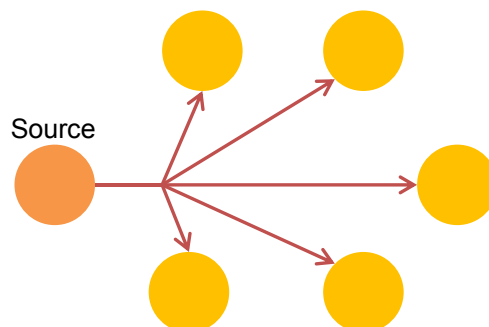
Unicast: Point-to-point (P2P) communication. Data packets are forwarded, without being copied.



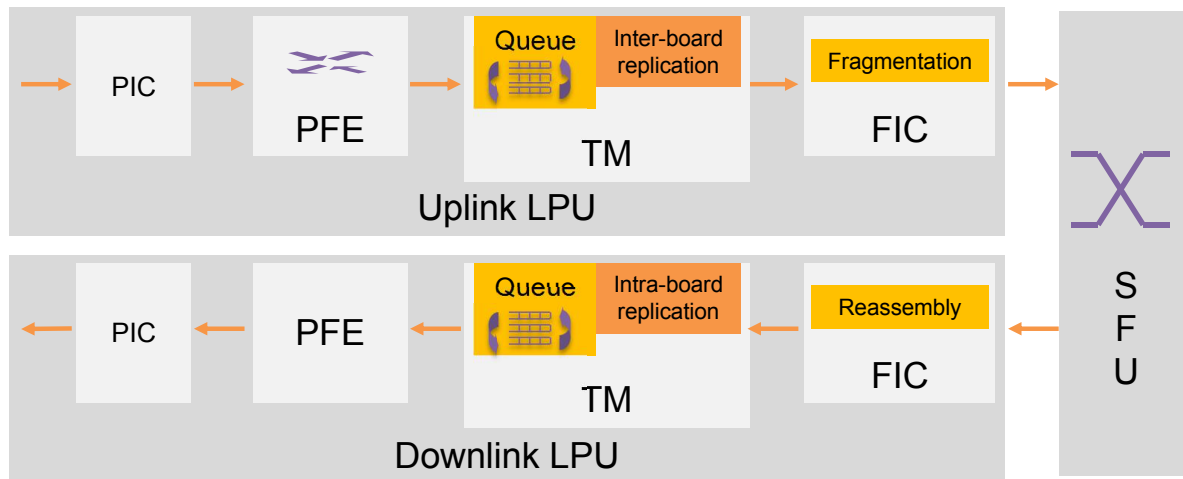
Multicast: Point-to-multipoint (P2MP) communication. Data packets are copied and sent only to requesters. The hosts that request the packets need to join in the same group (multicast group) so that they can receive all data sent to this group.



Broadcast: One-to-all communication. Data packets are copied and forwarded unconditionally, and all hosts on the network can receive them, regardless of whether the hosts need them. Broadcast traffic is restricted within LANs, preventing broadcast data from affecting a large number of hosts.



Multicast or broadcast packets may be sent by multiple outbound interfaces on different LPUs. To forward these packets, the traffic management (TM) chip on the uplink LPU copies the packets to different downlink LPUs, and the TM chips on the downlink LPUs copy the packets to different interfaces on the same board.



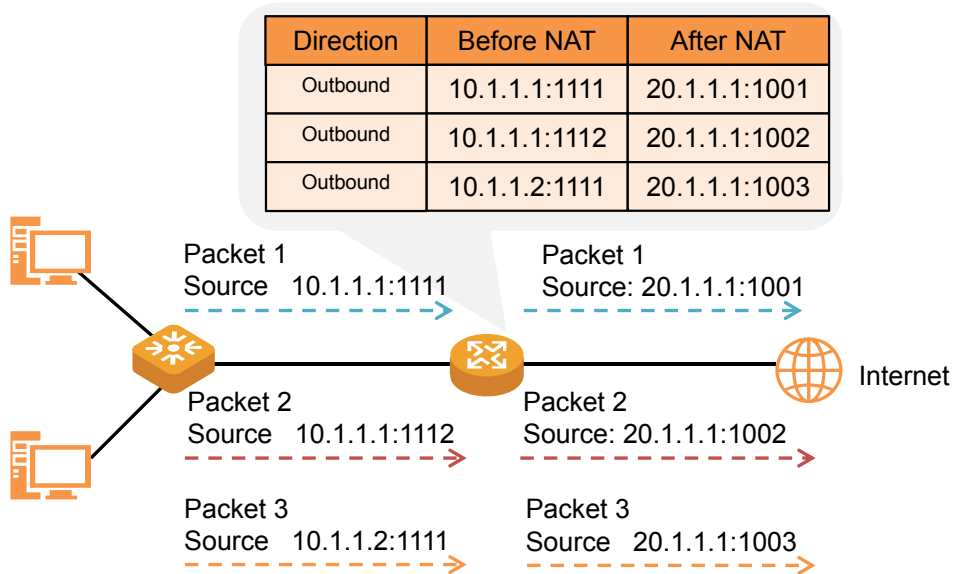
This chapter describes only broadcast and multicast replication on hardware components. For details about broadcast and multicast forwarding processes, see the following chapters.

NAT

Basic Principles

NAT converts IP addresses in IP datagram headers between private and public networks. Hosts in a LAN can use private addresses for internal communication. If they need to access the public network, their source addresses (private IP addresses) are translated into public IP addresses. For the return traffic to reach the hosts, the public IP addresses are translated back to the private IP addresses. In this way, all hosts in the LAN can use limited public IP addresses (even one) to access the Internet.

NAT has multiple implementation modes, among which Huawei high-end routers implement network address port translation (NAPT). In NAPT mode, a NAT device translates different source IP addresses carried in received packets into the same public IP address, and port numbers into different port numbers. This mode allows a large number of hosts to access the public network using only one public IP address.



In the preceding figure, packets 1 and 2 carry the same source private IP address but different port numbers. Packets 1 and 3 carry different source private IP addresses but the same port number. The NAT device translates the source IP addresses in packets 1, 2, and 3 into the same public IP address, and port numbers into different port numbers. Packets 1, 2, and 3 can be identified by their new port numbers. After receiving the return traffic from the public network, the NAT device identifies the destination hosts based on the destination IP address and port numbers.

NAPT is based on a NAT address pool which contains multiple public IP addresses. When a user needs to access the public network, the user's packets are sent to the NAT device. The NAT device selects one public IP address from the NAT address pool, maps it to the source IP address, and performs NAT accordingly. If the NAT device fails to receive any packets from the user within a period, it reclaims the public IP address previously used by the user.

NAT Address Pool Route Advertisement

After users on a private network access the Internet, the return traffic from the Internet carry the public IP addresses of the NAT address pool as destination IP addresses, and devices on the Internet need to search their routing tables for routes to the public IP addresses to forward the traffic. To ensure the forwarding, the NAT device needs to advertise NAT address pool routes to the Internet. However, these public IP addresses in the address pool are dynamically allocated by the NAT device. How can the routes to these public IP addresses be advertised?

In fact, after a NAT public address pool is created on a Huawei high-end router, NAT public address pool user network routes (UNRs) are generated. For example, if the network segment of a NAT public address pool is 10.0.0.0/22, the following UNR is generated:

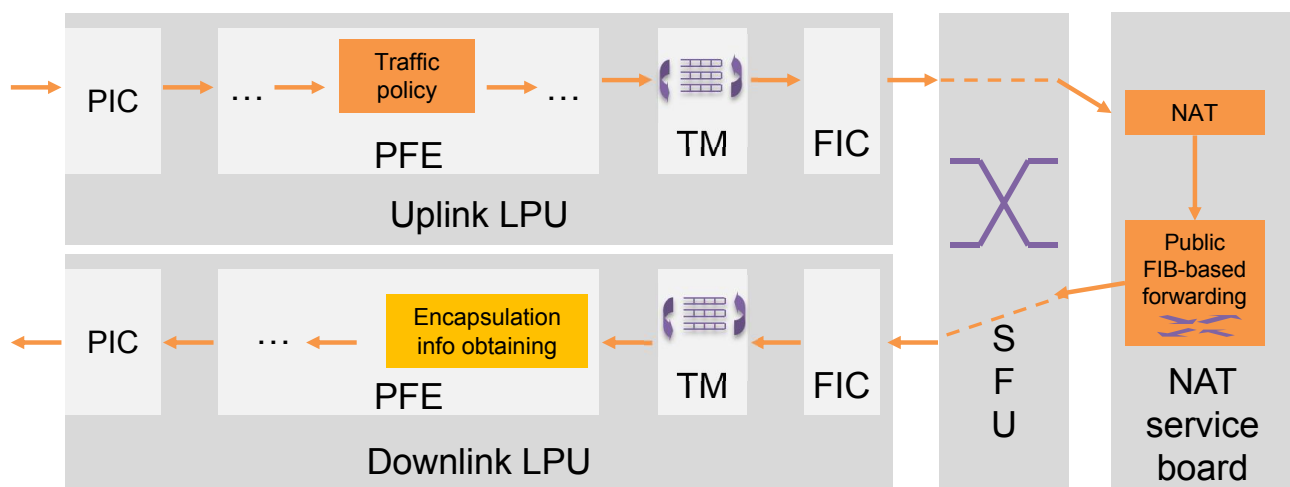
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
100.0.0.0/22	Unr	64	65535	D	127.0.0.1	InLoopBack0

If you want routes in the NAT address pool to be advertised to the public network, import the UNR to a dynamic routing protocol using the **import-route unr** command.

Implementation

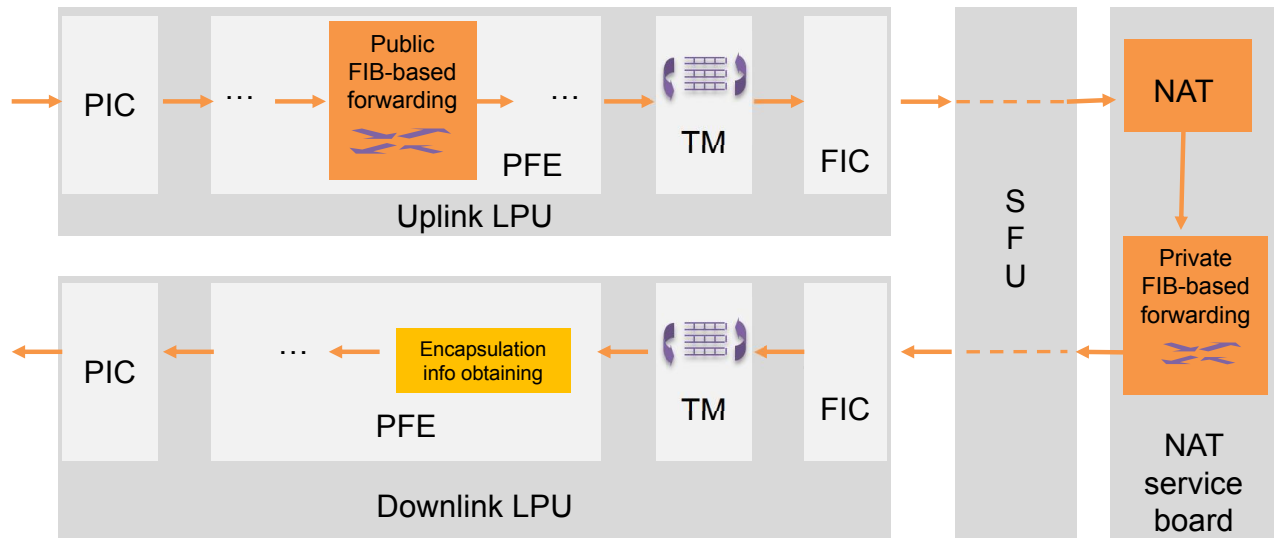
On Huawei high-end routers, the NAT service board is responsible for NAT. The service process is as follows: An uplink LPU distributes traffic to the NAT service board which performs NAT and sends the traffic to a downlink LPU. Finally, the downlink LPU forwards the traffic. The detailed process is as follows:

- Outgoing traffic forwarding process (from a private network to the public network)



The process is similar to that of other service flows except for the following differences:

1. The packet forwarding engine (PFE) on the uplink LPU uses multi-field (MF) classification for traffic diversion to the NAT service board.
 2. The SFU sends packets to the NAT service board.
 3. The NAT service board first performs NAT. After NAT, the private source IP addresses of the packets are translated into public IP addresses, and port numbers are also translated.
 4. Then, the NAT service board searches the public forwarding table for the destination LPU and outbound interface information. Note that the NAT service board has the same forwarding table as LPUs.
 5. The SFU switches the packets to a downlink LPU. The rest of the process is the same as that of other service flows.
- Incoming traffic forwarding process (from the public network to a private network)



The process is similar to that of other service flows except for the following differences:

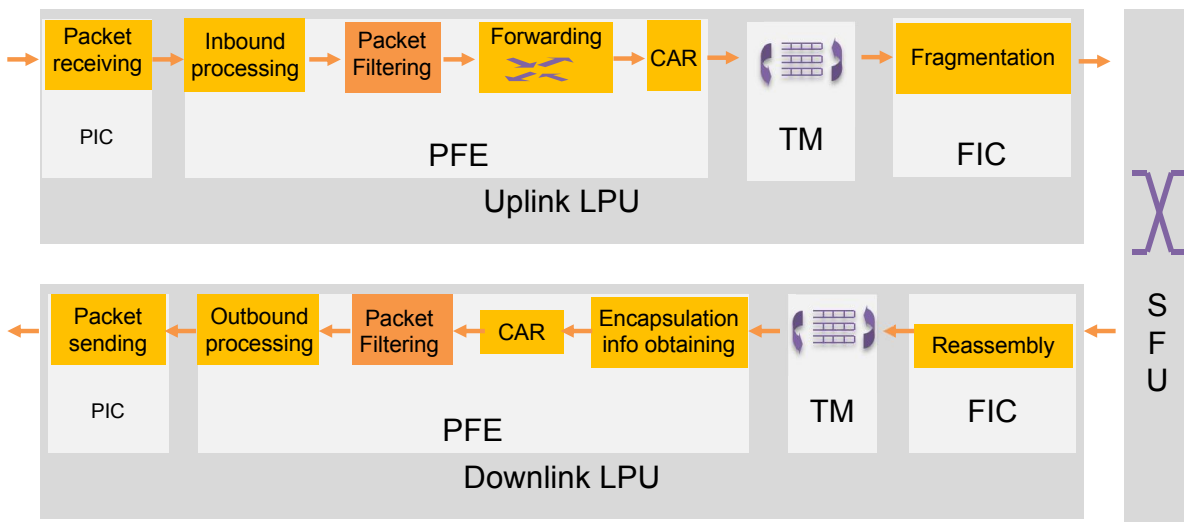
1. When the PFE on the uplink LPU searches the public routing table, it obtains the NAT address pool UNR, with the NAT service board as the destination LPU. In a centralized NAT scenario, the UNR carries destination LPU and interface information. In a distributed NAT scenario, the uplink LPU sends the packets to the CPU for processing to obtain the destination LPU and interface information.
2. The SFU sends packets to the NAT service board.
3. The NAT service board translates the private source IP addresses of the packets into public IP addresses and also translates port numbers.
4. Then, the NAT service board searches the VPN forwarding table for the destination LPU and outbound interface information.
5. The SFU switches the packets to a downlink LPU. The rest of the process is the same as that of other service flows.

Packet Filtering

Routers filter packets based on ACLs. Specifically, the routers obtain the packet header information, such as the Ethernet frame header, MPLS header, IP header, and TCP/IP port number, match the information against ACL rules, and forward or discard the packets based on the matching result.

To associate ACL rules with packet processing behavior (forwarding or discarding), MF classification is used.

Packet filtering is implemented on the forwarding plane of the uplink or downlink LPU, as shown in the following figure:



As described in chapter 4, traffic policies are configured in profiles. The profiles consist of the following parts:

- Traffic classifier: sets ACL matching rules that are used in if-match clauses.
- Traffic behavior: determines whether the traffic meeting the matching rules is forwarded (permit) or discarded (deny).
- Traffic policy: associates the traffic classifier with the traffic behavior. The traffic policy is applied to the inbound or outbound interface.



Note:

For details about ACL, see *Special Topic - ACL*.

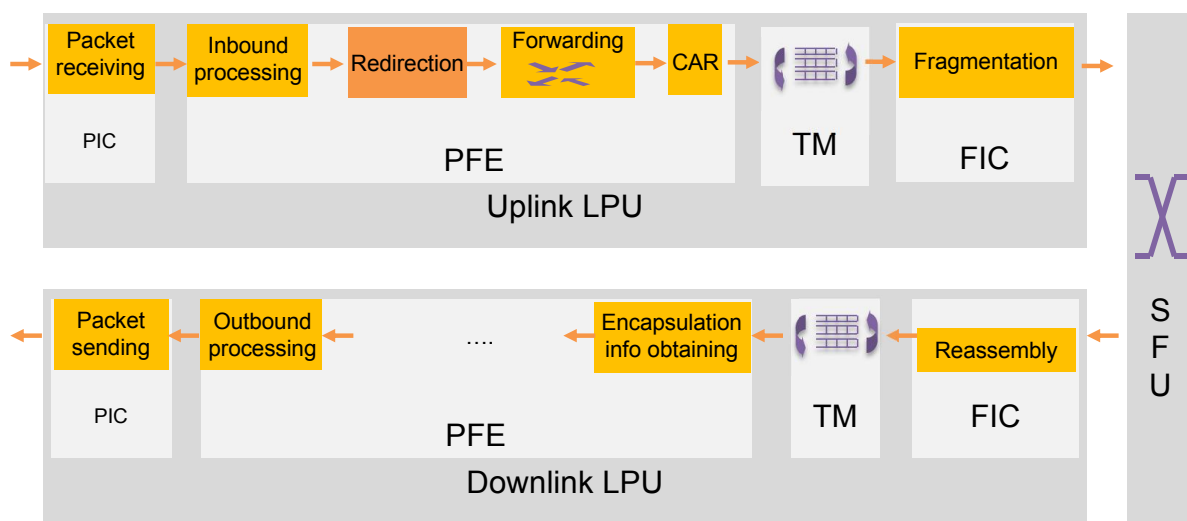
Policy-based Routing (Redirection)

Policy-based Routing (PBR) is also called route redirection. In most cases, routers search the routing table for routes to forward packets based on the destination IP addresses of the packets. In contrast, PBR selects routes based on a user-defined policy and can be used for security and load balancing purposes. PBR allows routers to select forwarding paths based on more packet attributes, such as the source IP address, destination IP address, and packet length.

PBR is different from routing policies. PBR routes data packets based on a user-defined policy instead of routes in the existing routing table. Routing policies control route generation, advertisement, and selection by following rules and changing route attributes.

PBR supports discard and forward policies. To configure a discard policy, run the **redirect** command with an outbound interface specified in the traffic behavior view. To configure a forward policy, run the command without specifying any outbound interface.

- In the case of a discard policy, routers forward packets based on the specified next hop and outbound interface if the outbound interface is Up. If the outbound interface is an Ethernet interface, ARP entries must also be available; otherwise, packets are discarded. If the next hop or outbound interface does not exist, packets are discarded.
- In the case of a forward policy, routers search the forwarding table for an outbound interface based on a specified IP address. If a corresponding outbound interface or even a default route is available, packets are forwarded; otherwise, packets are forwarded based on the packets' destination IP addresses.



On the forwarding plane, PBR is implemented in the MF classification phase. Forward PBR depends on the forwarding table.

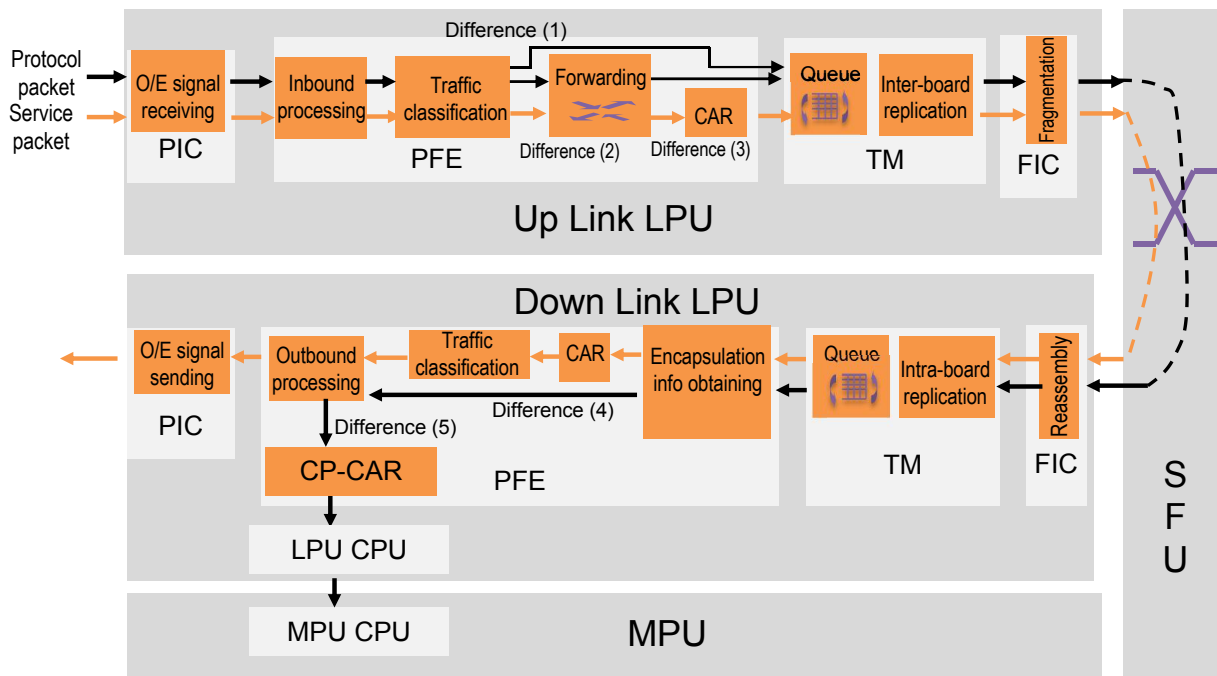
Chapter 7

Journey of Protocol Packets

Most of the packets received by routers are service packets. These packets are processed by service boards and SFUs. Routers also receive a small number of special packets, such as routing protocol packets, user login and logout packets, and exception or error packets. These special packets are sent by service boards to the CPU of the master control board for processing. This chapter describes how protocol packets are forwarded.

Journey of Incoming Protocol Packets

Processing of the protocol packets that need to be sent to the CPU is similar to that of service packets. The following figure shows how protocol packets are processed.



The five differences are as follows:

Difference 1: The system does not search the forwarding table if packets are identified as protocol packets.

The PFE (NP or ASIC chip) parses packets. If from the protocol field in the Layer 2 frame header, the PFE identifies a packet as a protocol packet that needs to be sent to the CPU for processing, such as an ARP, RARP, IS-IS, LLDP, LACP, or PPP control packet. If the destination IP address of the protocol packet is a reserved multicast IP address (ranging from 224.0.0.1 to 224.0.0.255), the uplink LPU does not search the forwarding table for packet forwarding.

As mentioned in the preceding chapter, the uplink LPU searches the forwarding table for the destination LPU and outbound interface information. The SFU switches the packets to a downlink LPU based on the destination LPU. Finally, the downlink LPU forwards the packets based on the outbound interface information. If an uplink LPU identifies a packet as a protocol packet that needs to be sent to the CPU for processing, the board does not search the forwarding table. Instead, it fills the destination LPU field with its slot and fills the outbound interface field with the CPU.

Difference 2: The packets with next-hop IP address 127.0.0.1 are sent to the CPU.

Routes of the following types carry a fixed next-hop IP address (127.0.0.1), and the packets with such a next-hop address need to be sent to the CPU for processing.

1. Interface host routes and direct subnet routes with a broadcast address

If directly connected interfaces are configured with IP addresses and the link layer protocol and IP layer protocol are Up, three routes are generated. For example:

Routing table:						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.2.5.0/24	Direct	0	0	D	10.2.5.5	GigabitEthernet1/0/0
10.2.5.5/32	Direct	0	0	D	127.0.0.1	GigabitEthernet1/0/0
10.2.5.255/32	Direct	0	0	D	127.0.0.1	GigabitEthernet1/0/0

Forwarding table:					
Destination/Mask	Nexthop	Flag	TimeStamp	Interface	TunnelID
10.2.5.0/24	10.2.5.5	U	t[5847]	GigabitEthernet1/0/0	0x0
10.2.5.5/32	127.0.0.1	HU	t[5847]	InLoop0	0x0
10.2.5.255/32	127.0.0.1	HU	t[5847]	InLoop0	0x0

- The first route is a network segment route, indicating that the GE1/0/0 of the router is directly connected to the network segment 10.2.5.0 and that the outbound interface to the network segment is GE1/0/0.
- The second route is a host route, and the destination IP address 10.2.5.5 is the IP address of GE1/0/0. When a router receives a packet destined for the IP address of a local interface, it sends the packet to the application protocol stack. On Huawei routers, the outbound interface of host routes displayed in the forwarding table is InLoopBack0, indicating that the corresponding packets need to be sent to the CPU for processing.
- The third route carries a broadcast address of 10.2.5.255/32 (one subnet of 10.2.5.0/24). According to IP standards, all Layer 3 interfaces on network segment 10.2.5.0/24 need to accept the packets with this address. The outbound interface of such routes is also InLoopBack0. Upon receipt of such packets, routers send them to the CPU for processing.

In addition, loopback and virtual template (VT) interfaces are logical, and IP addresses with a 32-bit mask are usually configured for such interfaces. Each interface of this type has a host route, as shown in the following figure:

Routing table:						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.0.0.5/32	Direct	0	0	D	127.0.0.1	LoopBack1
10.2.3.9/32	Direct	0	0	D	127.0.0.1	Virtual-Template5

Forwarding table:					
Destination/Mask	Nexthop	Flag	TimeStamp	Interface	TunnelID
10.0.0.5/32	127.0.0.1	HU	t[142]	InLoop0	0x0
10.2.3.9/32	127.0.0.1	HU	t[28733]	InLoop0	0x0

The next-hop IP address of such host routes is 127.0.0.1, and the outbound interface in the forwarding table is InLoopBack0, indicating that the corresponding packets need to be sent to the CPU for processing.

2. Routes with a network-wide broadcast address

IP address 255.255.255.255/32 is a network-wide broadcast address and is used to configure host startup information. During startup, a host may not know its network mask or even its IP address. In this case, the host sends a DHCP Request message with IP address 255.255.255.255/32 to obtain an IP address from the DHCP or BOOTP server. Such messages exist only on the local network. Upon receipt of such messages, routers send them to the CPU for processing instead of forwarding them.

Routing table:						
Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
255.255.255.255/32	Direct	0	0	D	127.0.0.1	InLoopBack0

Forwarding table:					
Destination/Mask	Nexthop	Flag	TimeStamp	Interface	TunnelID
255.255.255.255/32	127.0.0.1	HU	t[128]	InLoop0	0x0

3. UNRs

When a user dials up to a broadband remote access server (BRAS) or broadband network gateway (BNG) using PPPoE, the BRAS or BNG requests an IP address from the RADIUS server and allocates the address to the user. If the IP address allocated to the user is 10.111.111.1/32, the BRAS or BNG generates the route 10.111.111.1/32. After receiving a network-to-user packet, the BRAS or BNG sends it to the CPU for processing so that accounting can be implemented. The next-hop IP address of this route is also 127.0.0.1.

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.111.111.1/32	Unr	61	0	D	127.0.0.1	InLoopBack0

This route is not learned through any routing protocol, nor is it a direct or static route. It is a UNR.

The BRAS or BNG needs to advertise this route so that the user can receive packets from the network. However, if the BRAS or BNG has a large number of access users, it needs to advertise the same number of UNRs. To prevent this problem, the BRAS or BNG generates a UNR based on the address pool, with 127.0.0.1 as the next-hop IP address, and Null0 as the outbound interface.

Destination/Mask	Proto	Pre	Cost	Flags	NextHop	Interface
10.111.111.0/24	Unr	61	0	D	127.0.0.1	NULL0

Difference 3: The committed access rate (CAR) of protocol packets is not limited.

The CAR of protocol packets that are sent to the CPU is not limited, preventing packet loss in the case of traffic bursts.

Difference 4: Traffic classification is not performed on protocol packets.

Protocol packets are sent to the CPU on the downlink LPU. Therefore, traffic policy-related functions, such as traffic classification and marking, are meaningless for the packets.

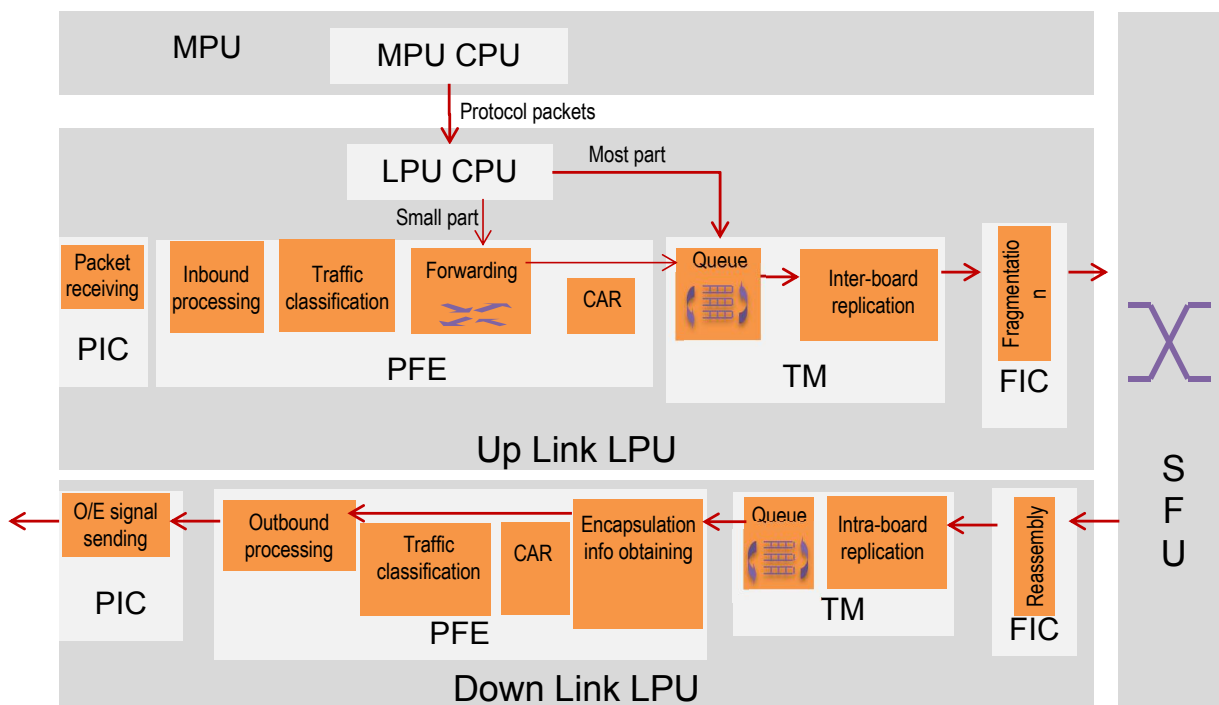
Difference 5: Control plane (CP)-CAR is performed on protocol packets before they are sent to the CPU for processing.

If a large number of packets are sent to the CPU for processing, the CPU will be overloaded. To prevent this problem, CP-CAR is performed on the packets before they are sent to the CPU. The mechanism of CP-CAR is similar to that of flow-based CAR. For details, see Chapter 4. Packets are separated in different channels based on the protocol type, VLAN, or user. Each channel uses a token bucket to limit the packet rate. If the bandwidth of the packets that are sent to the CPU exceeds a specified rate, the packets will be randomly discarded.

Journey of Outgoing Protocol Packets

The protocol packets sent by the CPU are directly delivered to the PFE, without being processed by the PIC. Because most of the protocol packets sent by the CPU carry destination LPU and outbound interface information, the forwarding plane does not need to search the forwarding table. The packets directly enter a queue. As for a small number of special packets, such as ping packets with a specified source interface (triggered by the **ping x.x.x.x-si interface-name** command), the forwarding plane needs to search the forwarding table because the IP address of the source interface is unknown. Then, the special packets are sent to the TM, without going through CAR limitation.

The subsequent processing of protocol packets is similar to that of service packets except that CAR limitation and traffic classification are not performed on the protocol packets on the downlink LPU.

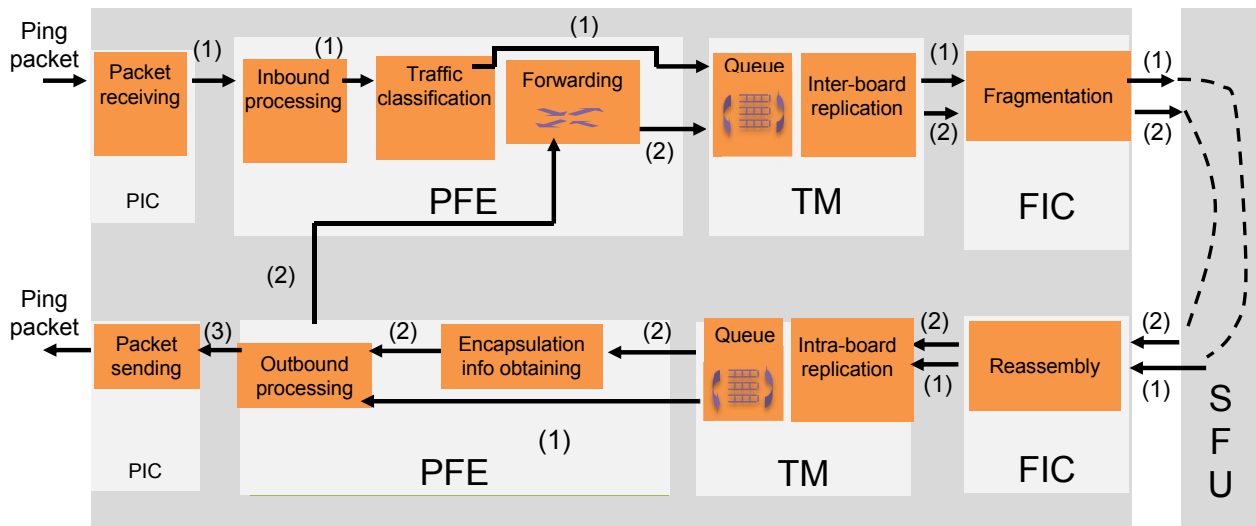


Fast Reply Packets Are Not Sent to the CPU

Ping (ICMP request) packets are usually used to check the connectivity of the link to a gateway or network-side IP address. Before ICMP request packets are sent to the CPU for parsing, CP-CAR is implemented.

Upon receipt of the packets, the CPU constructs ICMP reply packets and sends them to the source end. If a large number of ICMP request packets are sent to the CPU, the CPU will be overloaded, increasing the ping delay. To solve this problem, Huawei high-end routers support the ICMP fast reply function. With this function, the received ICMP request packets are not sent to the CPU for processing. Instead, the PFE of the LPU responds to the source end with ICMP reply packets, greatly shortening the ping delay.

The forwarding process using ICMP fast reply is as follows:



As shown in the preceding figure, ping packets are not sent to the CPU on the downlink LPU. Instead, the PFE swaps the source and destination IP addresses of the packets and loops the packets back to the uplink LPU.

However, if the size of fast reply packets is greater than the MTU, the packets are fragmented, and the fragmented packets are regarded as common ping packets for further CPU processing.

By default, ICMP fast reply is enabled in most versions. If ping packets are simulated as service packets during troubleshooting, ICMP fast reply needs to be disabled. To disable it, run the **undo icmp-reply fast** command in the slot view or system view.

Similarly to ICMP fast reply, there is ARP fast reply and web fast reply, but these are not detailed in this section.

Chapter 8

IP Unicast Forwarding Process

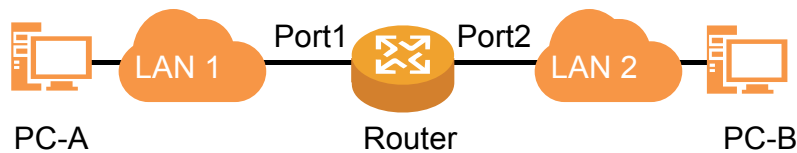
The first six chapters describe how a packet is processed on the forwarding plane. In performing this action, the forwarding process is the most important process and varies with the service. The forwarding processes of different services will be detailed in the following chapters. This chapter describes IPv4 and IPv6 unicast forwarding processes.

IPv4 Unicast Forwarding

P2P IPv4 Unicast Forwarding

First, let's review the IP P2P unicast forwarding process using Ethernet frames as an example.

The following figure shows a simple IP forwarding scenario. PC-A in LAN 1 sends an IP packet to PC-B in LAN 2 through a router. This router is the gateway of PC-A.



The destination IP address of the packet is the IP address of PC-B, the source IP address is the IP address of PC-A, the destination MAC address is the MAC address of port 1 on the router, and the source MAC address is the MAC address of PC-A.

DestinationMAC = Port1	Source MAC = PC-A	Protocol type = IPv4	Source IP = PC-A	Destination IP = PC-B
---------------------------	----------------------	----------------------------	---------------------	--------------------------

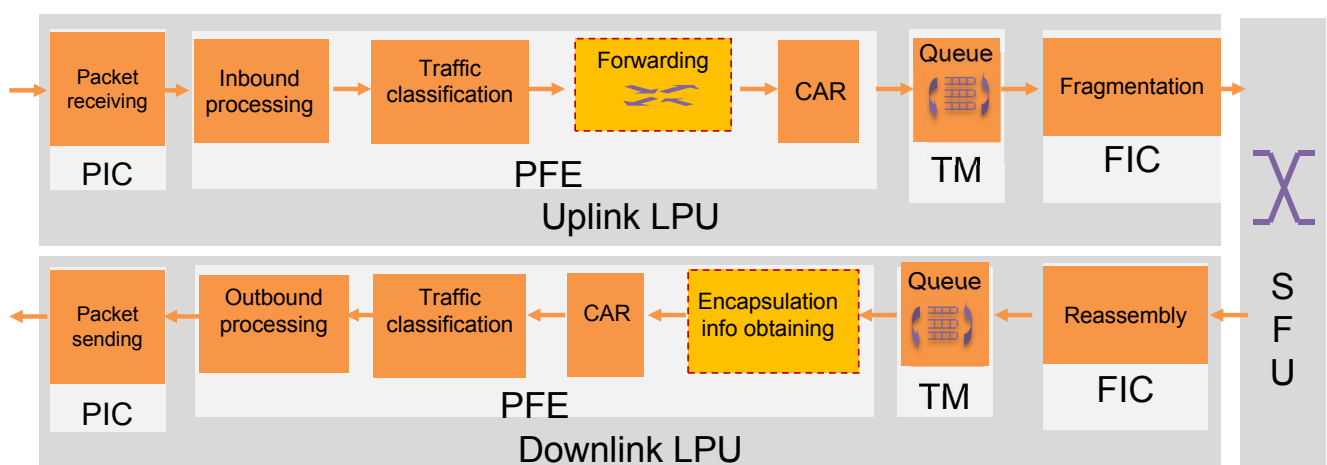
Forwarding process on the router:

1. Upon receipt of the packet, the router parses it because the destination MAC address is the MAC address of Port 1. If the destination MAC is not a local MAC address, the router directly performs Layer 2 forwarding without parsing the packet.
2. Finding that the protocol carried in the packet is IPv4 (the value of **eth_type** being 0x800), the router performs IPv4 forwarding accordingly.

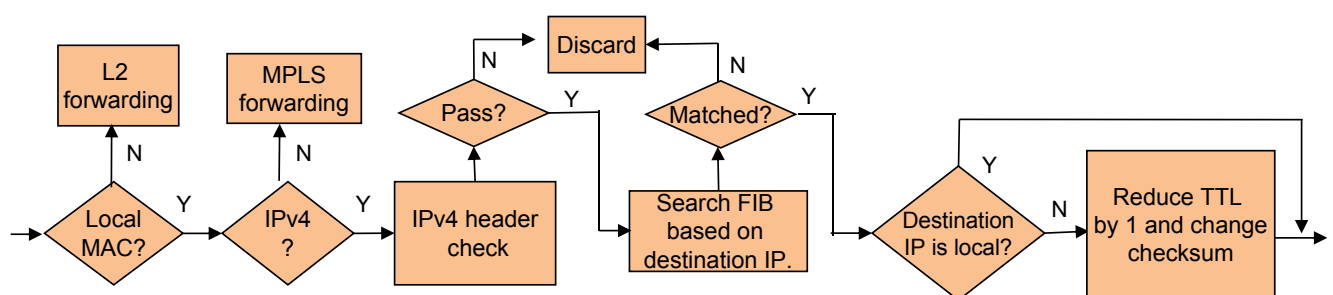
3. The router searches the IP forwarding table (FIB). Knowing that the packet is not destined for itself and that the outbound interface is Port 2, the router does not parse the rest content in the IP header.
4. The router changes the destination MAC address to the MAC address of PC-B, and the source MAC address to the MAC address of Port 2, and forwards the packet through Port 2.

Complete IPv4 Forwarding Process on the Router

The following figure shows the complete IPv4 forwarding process. In the process, we should pay attention to FIB-based forwarding and encapsulation info obtaining.



FIB-based forwarding:



The router performs the following operations:

Step 1. Checks whether the destination MAC address is a local MAC address. If not, the router performs L2 forwarding. If so, the router proceeds to the next step.

Step 2. Checks whether the protocol type of the packet is IPv4.

For example, in the case of an Ethernet frame, the router checks whether the value of **eth_type** is 0x800. If not, the router performs the corresponding forwarding process. If so, the router proceeds to the next step.

Step 3. Checks whether the packet length, IP address, and checksum are correct. If not, the router discards the packet. If so, the router proceeds to the next step.

Step 4. Checks whether the destination IP address is a unicast IP address. If not, the router performs the corresponding forwarding process. If so, the router proceeds to the next step.

Step 5. Searches the FIB for the next-hop IP address and outbound interface corresponding to the destination IP address. In the case of a public packet, the router searches the public FIB. In the case of a VPN packet, the router searches the FIB of the corresponding VPN.

FIB:

Destination/Mask	Nexthop	Flag	TimeStamp	Interface	TunnelID
10.2.5.0/24	10.2.5.5	U	t[5847]	GigabitEthernet1/0/0	0x0
10.2.5.5/32	127.0.0.1	HU	t[5847]	InLoop0	0x0

- If equal-cost routes are available for load balancing, they are all displayed in the FIB. The router uses the load balancing hash algorithm to select one from them. For details about load balancing, see *Special Topic - Load Balancing*.
- If fast reroute (FRR) is enabled, the router selects the primary or backup route based on the status of the outbound interface. If the interface is Up, it selects the primary route. Otherwise, it selects the backup route.
- If the outbound interface is a trunk interface, the router uses the trunk load balancing hash algorithm to select one trunk member interface as the outbound interface.

Step 6: If unicast reverse path forwarding (URPF) check is enabled, the router searches the FIB based on the source IP address. In the case of loose URPF check, a packet passes the check as long as the outbound interface is a physical interface. Specifically, the outbound interface cannot be the CPU or a Null, TE, or IPv4 tunnel interface. In the case of strict URPF check, the router searches the FIB based on the inbound interface and source IP address of each packet. If a corresponding route exists in the FIB and the inbound interface of the packet is the outbound interface of the route, the packet passes the check; otherwise, the router discards the packet. If the inbound interface is a VLAN sub-interface, the outbound interface must be the inbound interface, and the two interfaces must have the same VLAN ID.

Note:

In most cases, after a router receives a packet, it searches for a route based on the destination IP address of the packet. If a route is available, the router forwards the packet accordingly. If no route is available, the router discards the packet.

If URPF check is enabled, the router obtains the source address and inbound interface of the packet, searches the FIB for a route destined for the source IP address, and checks whether the outbound interface of the route is the inbound interface of the packet. URPF check prevents attacks that use spoofed source IP addresses.

However, multiple routes to the same destination IP address may exist in the FIB in some scenarios, such as in a load balancing scenario. The outbound interfaces of the routes are different. If URPF is configured in this case, packet loss will occur. To prevent this problem, use loose URPF check. In loose URPF mode, a packet can pass the URPF check as long as there is a route destined for the source IP address of the packet, regardless of whether the outbound interface of the route matches the inbound interface of the packet.

Step 7. If the destination IP address is not a local IP address, the router decreases the TTL in the packet header by 1, recalculates and modifies the checksum value, and performs subsequent operations, such as CAR. If the destination IP address is a local IP address (the next-hop IP address being 127.0.0.1), the router sends the packet to the upstream TM component.

Finally, the SFU sends the packet to the downlink LPU based on the outbound interface information (including the destination LPU and outbound interface).

Encapsulation info obtaining

On the downlink LPU, the PFE searches for an ARP entry based on the next-hop or destination IP address and the VLAN ID to obtain the destination MAC address, and searches for the MAC address of the outbound interface. Then, the PFE replaces the destination MAC address with the MAC address of the next hop, and replaces the source MAC address with the MAC address of the local outbound interface.

ARP table:					
IP ADDRESS	MAC ADDRESS	EXPIRE (M)	TYPE	INTERFACE	VPN-INSTANCE
100.2.150.51	0018-8201-4daa		I -	GE0/0/0	
100.2.200.7	0013-d326-a32f	1	D-0	GE0/0/0	
192.1.23.1	00e0-fcd5-c877		I -	GE1/0/2	
37.1.3.1	00e0-fcd5-c863		I -	GE1/0/3	

If no corresponding ARP entry exists, the ARP learning function is triggered, with the steps detailed as follows:

1. The router sends an ARP request packet. The destination MAC address of the packet is a broadcast address, the destination IP address is the IP address of the next hop, and the source IP address is a local IP address.
2. Because the destination MAC address of the packet is a broadcast address, all devices or hosts (including the next-hop device) in the LAN can receive the packet. Upon receipt of the packet, the next-hop device parses it and finds that the destination IP address is its own IP address, it replies with an ARP response packet carrying its own MAC address.

3. After the router receives the response packet, it obtains and adds the next-hop MAC address to the ARP entry table.

After ARP learning, the router performs subsequent processing based on the next-hop MAC address.

Outbound Check and Encapsulation

If the destination IP address of the packet is a local IP address, the outbound interface processing module sends the packet to the CPU of the LPU. Finally, the packet is sent to the CPU of the MPU.

If the destination IP address of the packet is not a local IP address, the outbound interface processing module checks whether the packet length is greater than the MTU. If the packet length is less than the MTU, the module sends the packet to the PIC. The PIC calculates the frame check sequence (FCS) based on the content of the data frame to be sent, and encapsulates the interframe space, preamble, start-of-frame delimiter (SFD), and FCS to the frame. Then, the PIC converts the data frame to optical or electrical signals, and sends the signals to the outbound interface.

If the packet length exceeds the MTU, the router checks the DF bit in the packet header. If the DF bit is 0, the router fragments the packet and then sends the fragments to the PIC. If the DF bit is 1, the source end of the packet does not allow fragmentation. In this case, the router performs CP-CAR check and sends the packet to the CPU of the LPU, and then to the CPU of the MPU. Finally, the router responds to the source end with an ICMP Too-Big message.

IPv6 Unicast Forwarding

IPv4 and IPv6 forwarding processes are similar, with the following differences:

- In the IPv4 forwarding process, the router searches the FIBv4 and ARP entries. In the IPv6 forwarding process, the router searches the FIBv6 and neighbor table.
- In the IPv6 forwarding process, if the length of a packet exceeds the interface IPv6 MTU, the router does not fragment the packet. Instead, it sends it to the CPU and responds to the source end with an ICMP Too-Big message.

IPv6 neighbor table:

```
[Router] display ipv6 neighbors
-----IPv6 Address : 2012::2
Link-layer   : 00e0-fcc2-13b6           State : STALE
Interface    : GE0/0/0                 Age   : 0
VLAN         : -                       CEVLAN: -
VPN name     :                         Is Router: TRUE
Secure FLAG  : UN-SECURE

IPv6 Address : FE80::2E0:FCFF:FEC2:13B6
Link-layer   : 00e0-fcc2-13b6           State : STALE
Interface    : GE0/0/0                 Age   : 0
VLAN         : -                       CEVLAN: -
VPN name     :                         Is Router: TRUE
Secure FLAG  : UN-SECURE
-----
```

Chapter 9

Layer 2 Ethernet Frame Forwarding

This chapter describes the Ethernet frame forwarding process for unicast, multicast, and broadcast traffic.

Basics of Layer 2 Ethernet Frame Forwarding

What Is Layer 2 Ethernet Frame Forwarding?

Layer 2 Ethernet frame forwarding describes how data frames are forwarded on the data link layer through network bridges or switches.

The data link layer has different network protocols, such as token ring, Ethernet, and FDDI. Among these protocols, Ethernet is most widely used. This chapter focuses on how Ethernet frames are forwarded.

An Ethernet forwards frames based on Layer 2 Ethernet frame headers or - to be more specific - MAC addresses.

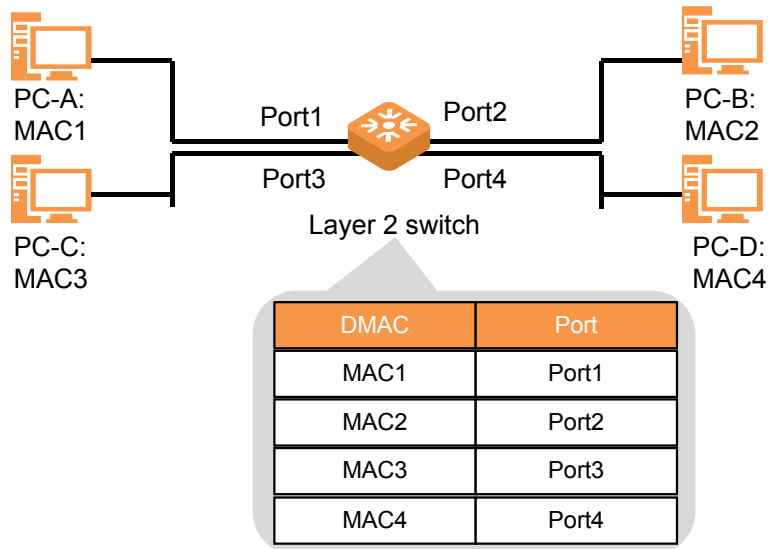
MAC Address

A MAC address is a globally unique 48-bit binary address. MAC addresses are uniformly managed and assigned by the IEEE. For easier readability, MAC addresses are represented in hexadecimal notation, such as 00-e0-fc-00-00-06. They can be classified into three types:

- Unicast address: The least significant bit of the first octet is 0, such as 00-e0-fc-00-00-06.
- Multicast address: The least significant bit of the first octet is 1, such as 01-e0-fc-00-00-06.
- Broadcast address: All 48 bits are 1s, represented as ff-ff-ff-ff-ff-ff.

Frame Forwarding Process for Layer 2 Unicast Traffic

In the following example, PC-A in a LAN sends an Ethernet frame to PC-B over a Layer 2 switch. The Ethernet frame's destination MAC address is MAC2 and source MAC address is MAC1.

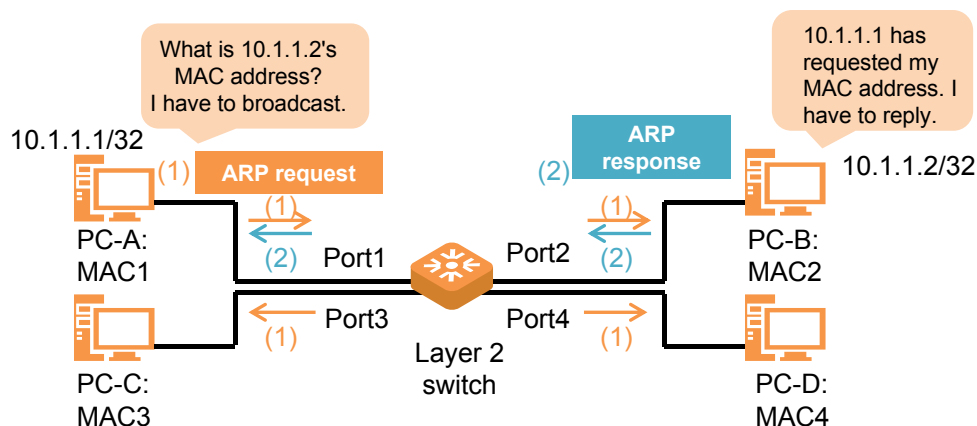


Upon receipt of the Ethernet frame, the Layer 2 switch performs the following operations:

1. Parses the Ethernet frame and reads the destination MAC address. In this example, the destination MAC address is MAC2.
2. Looks up its MAC address table and finds the corresponding outbound port. In this example, MAC2 is mapped to Port2.
3. Forwards the Ethernet frame. In this example, the Layer 2 switch forwards the Ethernet frame from Port2 to PC-B.

Frame Forwarding Process for Layer 2 Broadcast Traffic

In the preceding example, what will PC-A do if it does not know PC-B's MAC address? PC-A will broadcast an ARP request, in which the destination MAC address is a broadcast address and the source MAC address is PC-A's own MAC address. Upon receipt of the ARP request, the switch sends the request to all ports except Port1. All hosts in the LAN receive the broadcast ARP request.



After PC-B receives the ARP request, it returns an ARP response. The switch forwards the ARP response only to PC-A.

MAC Address Learning

We've seen the MAC address table on the Layer 2 switch that contains mappings between MAC addresses and ports. Where does this table come from? We've learned that a router learns IP-MAC mappings through ARP, and now let's discover how the MAC address table is built.

In the preceding example, the switch's MAC address table starts out empty. After PC-A sends an Ethernet frame to PC-B, the switch receives the frame and performs the following operations:

1. Reads the frame's source MAC address, maps the address to the port that received the frame, and adds the mapping to its MAC address table.
2. Reads the frame's destination MAC address and searches its MAC address table for the corresponding port. Since the switch has not learned PC-B's MAC address, the switch floods the data frame to all ports except to the port that received the frame. PC-B then receives the data frame.

The switch learns every device's MAC address after PC-B, PC-C, and PC-D send data frames to the switch.

Aging of MAC Address Table Entries

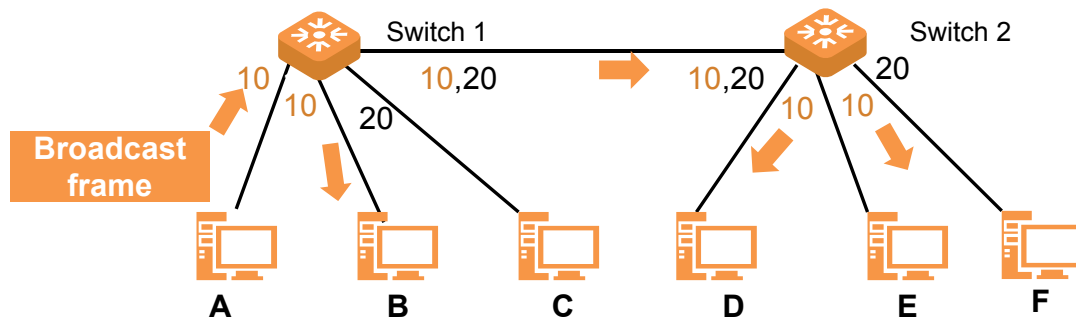
In the preceding example, if PC-D is moved or the ports connecting to PC-D and PC-C are switched, the switch may send frames to incorrect PCs if its MAC address table is not updated in real time. The switch prevents this problem by setting a timer for each MAC address entry. If the switch does not receive any frames from a particular PC before the timer expires, the switch considers the PC's MAC address entry invalid and removes it from its MAC address table. If the PC wants to send frames again, the switch has to re-learn the PC's MAC address.

VLAN Basics

In a LAN, broadcasting is unavoidable. We've learned that a switch floods broadcast or unknown unicast frames to a network. Other protocols, such as DHCP or RIP, also frequently send broadcast frames. Flooding consumes link resources and burdens hosts.

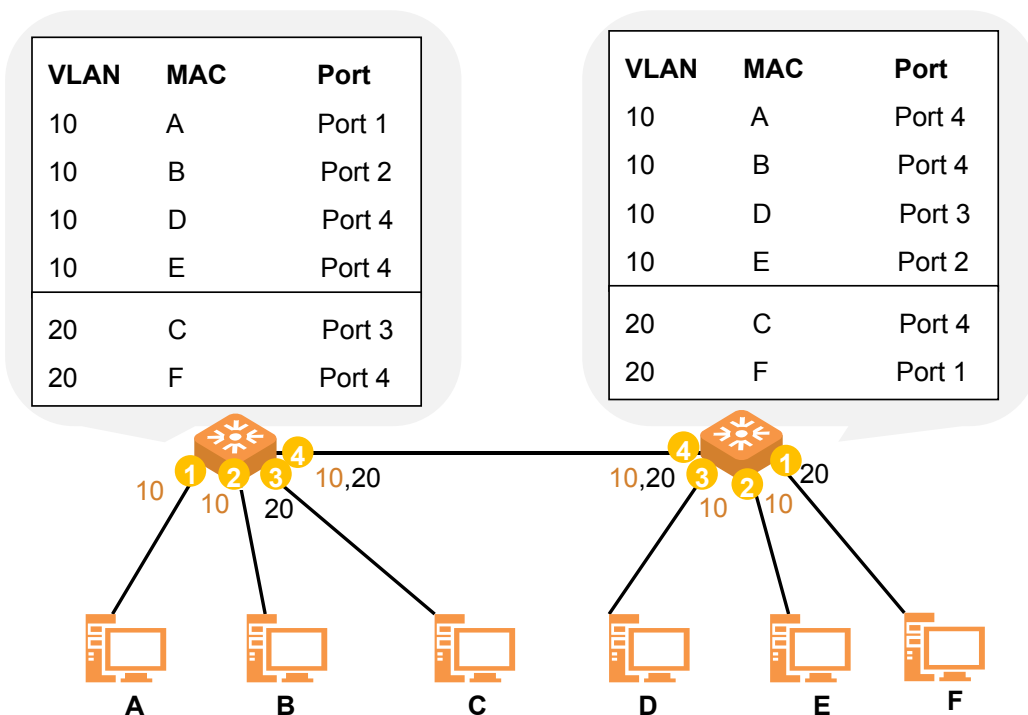
To reduce broadcast frame traffic, the virtual local area network (VLAN) was created. VLANs logically divide LANs into multiple broadcast domains. Hosts in the same VLAN can communicate with each other but hosts in different VLANs cannot. How does a VLAN isolate broadcast frames?

To isolate broadcast frames, switch ports are assigned with VLANs. Ports between multiple switches can belong to more than one VLAN. In the following figure, after Switch 1 receives a broadcast frame from host A, it adds a VLAN 10 tag to the frame and forwards the frame only to ports in VLAN 10. Ports in VLAN 20 cannot receive this frame.

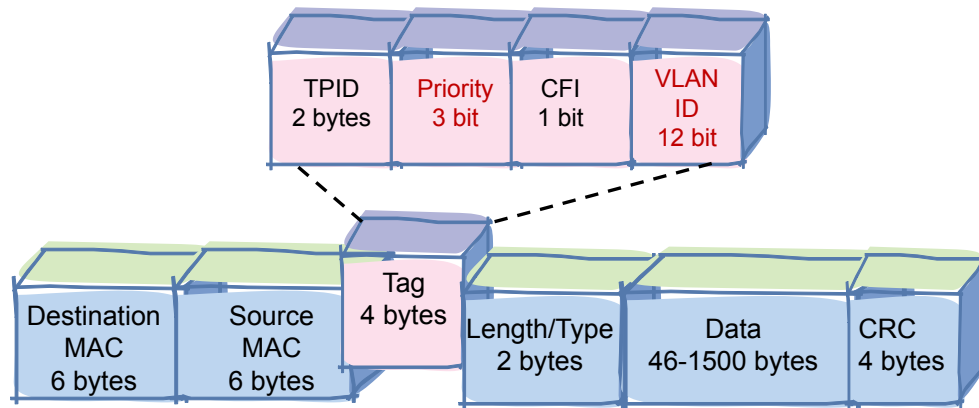


How do hosts in a VLAN communicate?

After VLANs are assigned to the switch's ports, VLAN IDs appear in the original MAC address table.



After the switch receives a frame, the switch adds a VLAN tag to the frame based on the inbound interface and forwards the frame based on the VLAN ID and destination MAC address. The format of a VLAN-tagged Ethernet frame is as follows:



After the peer switch receives the VLAN-tagged frame, the switch removes the VLAN tag and forwards the frame based on the VLAN ID and destination MAC address in its forwarding table. (Note that hosts in different VLANs can only communicate through routers.)

In the preceding example, some ports allow only one VLAN to pass, while some allow multiple VLANs to pass. VLAN ports can be classified into three types:

- **Access port:** belongs to only one VLAN and connects devices that do not support 802.1Q encapsulation, such as a user computer.
- **Trunk port:** can belong to multiple VLANs and receive and send frames from multiple VLANs. A trunk port is used to connect network devices.
- **Hybrid port:** can belong to multiple VLANs and receive and send frames from multiple VLANs. A hybrid port can connect network devices or devices that do not support 802.1Q encapsulation.

Frame processing mechanism:

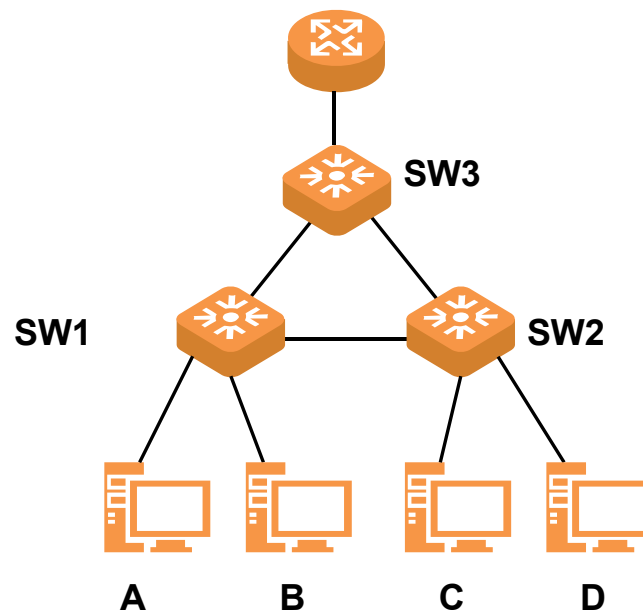
To ensure efficient frame processing, switches perform VLAN-based forwarding and frame processing differently for inbound and outbound directions. Vendor-specific devices and different VLAN ports also perform differently when processing VLAN frames. Huawei high-end routers process VLAN frames as follows.

Interface Type	Inbound		Outbound
	Receiving an untagged frame	Receiving a tagged frame	Sending a frame
Access port	Accepts and adds the default VLAN tag to the frame before forwarding.	Forwards if the frame's VLAN ID is same as the default VLAN ID. Discards if not.	Strips the VLAN tag.
Trunk port	Discards the frame.	Forwards if the VLAN ID is permitted. Discards if not permitted.	Directly sends the frame.

Hybrid port	Accepts and adds the default VLAN tag to the frame. If the default VLAN ID in the tag is permitted, the port forwards the frame. Otherwise, the port discards it.	Forwards if the VLAN ID is permitted. Discards if not permitted.	If the VLAN ID carried in the frame is the same as the default VLAN ID, the port strips the VLAN tag before forwarding. Otherwise, the port directly forwards the frame without changing the VLAN tag.
-------------	---	--	--

Layer 2 Loop Prevention - Spanning Tree Protocol

Ring networks are very common in Ethernet LAN. However, the ring network topology is prone to broadcast storms.



How do broadcast storms occur? If host A wants to communicate with host D without knowing D's MAC address, host A broadcasts an ARP request. Upon receipt, SW1 floods the frame. Both SW2 and SW3 receive the ARP request and flood the frame. SW1 receives its own ARP request and restarts the flooding process. All the three devices continuously receive and flood the frame, causing a broadcast storm.

Spanning Tree Protocol (STP), Rapid Spanning Tree Protocol (RSTP), and Multiple Spanning Tree Protocol (MSTP) were developed to detect and eliminate Layer 2 loops. These STPs are used to probe link-layer topologies and control link-layer forwarding behaviors of switches. If a network loop exists, these STPs block a selected port from forwarding or receiving Ethernet frames to eliminate the loop.

Since this chapter focuses on data frame forwarding, STP-defined port status and forwarding behavior will be discussed in detail.

STP defines five port states as follows:

- Forwarding: Forwards both user traffic and STP protocol packets - BPDUs.
- Learning: Creates a MAC address table based on the received user traffic but does not forward the user traffic.
- Listening: Determines the root bridge, root port, and designated port but does not forward the user traffic.
- Blocking: Receives and forwards BPDUs only.
- Disabled: Forwards neither BPDUs nor user traffic.

MSTP and RSTP streamline the five port states into the following three states:

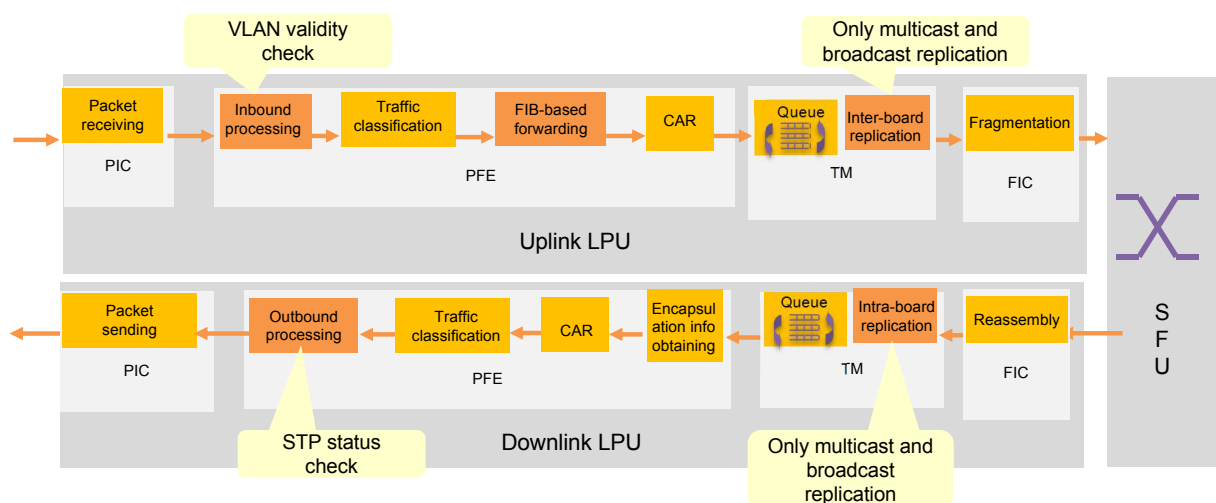
- Forwarding: Forwards both user traffic and BPDUs.
- Listening: Only receives and processes BPDUs.
- Discarding: Forwards neither BPDUs nor user traffic.

Only ports in Forwarding state forward user traffic.

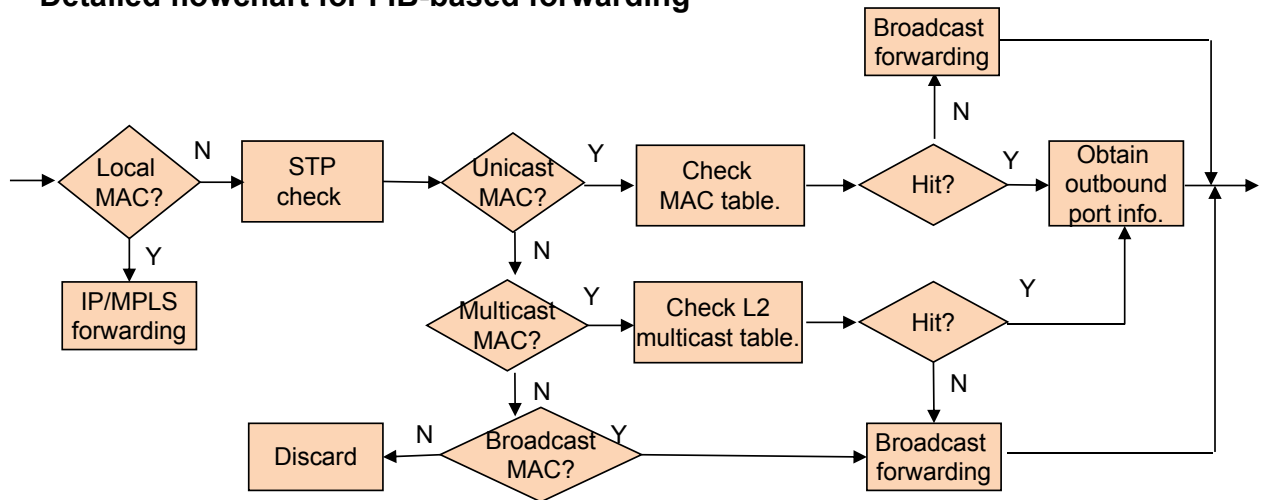
The following describes how Ethernet frames are forwarded.

Layer 2 Ethernet Frame Forwarding Process

On Huawei high-end routers, if Ethernet interfaces are switched from Layer 3 to Layer 2 using the **portswitch** command, the Layer 2 Ethernet interfaces support Layer 2 bridge forwarding. The following figure shows the complete Layer 2 bridge forwarding process, with a focus on FIB-based forwarding and encapsulation information obtainment. Other processes that are shown have been described in Chapters 1 through 6.



Detailed flowchart for FIB-based forwarding



Process description:

1. The uplink PFE (NP or ASIC chip) parses the received packet, checks VLAN validity based on port types (access, trunk, or hybrid), and discards the packet if it is invalid. For processing details, see the table description on pages 60 and 61.
2. If the packet is valid, the PFE determines whether the packet's destination MAC is a local MAC. If so, the PFE performs IP or MPLS forwarding; If not, the PFE proceeds to the next step.
3. The PFE performs STP status check. If an interface stays in the Forwarding state or does not have STP enabled, the interface forwards the packet. If an interface stays in other states, it discards the packet.
4. The PFE then determines whether the destination MAC is unicast, multicast, or broadcast according to the following:
 - Unicast: Looks up the MAC address table based on port+VLAN. If a match exists, the PFE obtains the outbound interface and VLAN ID, based on which the SFU switches the packet to the correct downlink LPU. If no match exists, the PFE broadcasts the packet.
 - Multicast: Looks up the Layer 2 multicast MAC address table based on port+VLAN for the outbound interface and VLAN ID. If no match exists, the PFE broadcasts the packet.
 - Broadcast: Continues the subsequent processing.
(If the packet is a unicast or multicast one, and the outbound interface is a trunk, the packet will be hashed to one trunk member interface.)
5. The uplink TM chip copies unknown unicast and multicast packets, and broadcast packets to all destination LPUs. It copies known multicast packets to destination LPUs where the multicast members reside, but does not copy unicast packets.
6. The downlink TM chip copies unknown unicast and multicast packets, and broadcast packets on the board receiving these packets.

7. If MAC address learning is enabled, the downlink forwarding engine learns MAC addresses.
8. The downlink forwarding engine obtains encapsulation information and performs processing based on the VLAN ID and port type (access, trunk, or hybrid). For processing details, see the table description on pages 60 and 61.
9. The downlink forwarding engine performs STP status check at the exit. If an interface stays in the Forwarding state or does not have STP enabled, the interface forwards the packet. If the interface stays in the Learning or Listening state, it discards the packet.

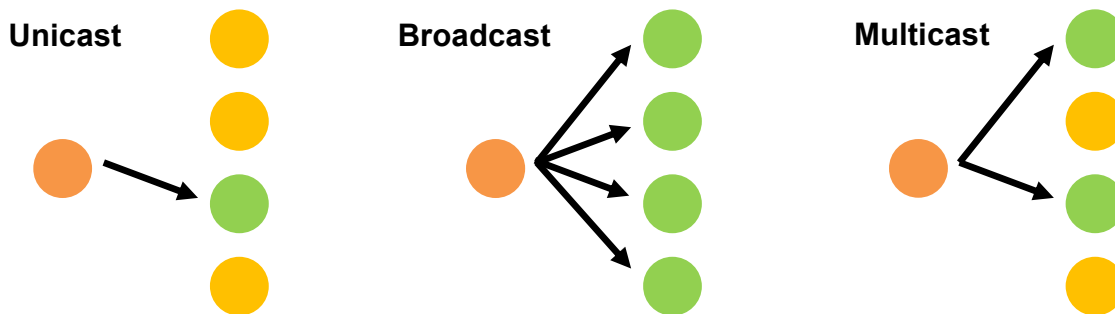
Chapter 10

IP Multicast Forwarding

Getting Started with IP Multicast

Unicast, Broadcast, and Multicast Communications

The three methods of IP communication are illustrated as follows:

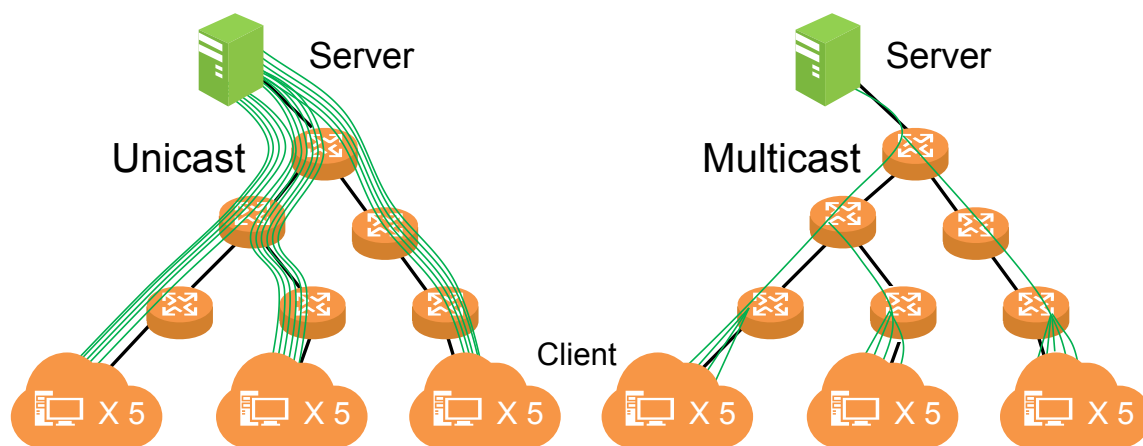


- Unicast is a one-to-one data transmission method used for communication between one source host and one destination host. Using this method, a switch or router forwards data but does not replicate data. Most data on a network, such as email, web browsing, and video on demand (VoD), is transmitted using unicast.
- Broadcast is a one-to-all data transmission method used for communication between a source IP host and all other IP hosts on the same local area network (LAN). Using this method, a switch or router replicates and forwards data unconditionally, and all hosts can receive the data even if a host does not require the data. In this manner, broadcast communication not only wastes bandwidth, but can also cause broadcast storms due to routing loops. Therefore, broadcast data in a LAN cannot be transmitted to other LANs, and cannot be forwarded by routers. However, broadcast communication is still necessary in some scenarios, such as dynamic IP address obtaining through DHCP and MAC address learning through ARP.
- Multicast is a one-to-many data transmission method used for communication between a source IP host and a selected group of IP hosts. Using this method, a switch or router replicates and forwards data only for hosts that require the data, not for all hosts. Typical multicast application scenarios include video/voice conferences, IPTV, and stock quotation releases.

Compared to unicast transmissions, multicast transmissions have significant advantages. For example, consider the situations illustrated below where 15 users have requested the same web page to view.

In unicast transmission mode, the server sends 15 identical copies of the data, which places a lot of pressure on the server and network devices involved (see the first illustration).

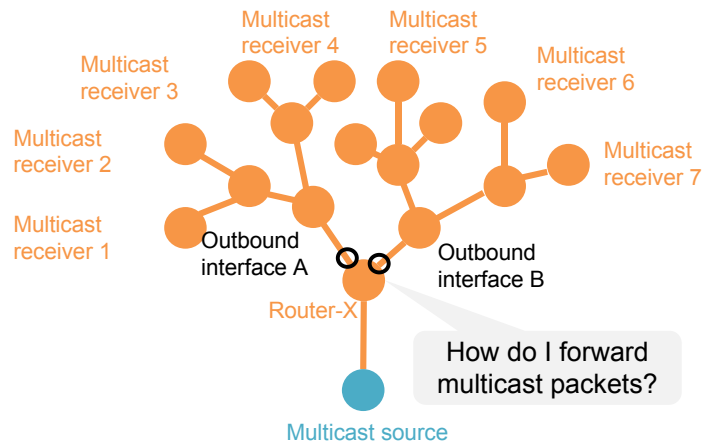
In multicast transmissions mode, the server sends only 1 copy of the data to all the 15 receivers (see the second illustration).



Multicast Distribution Tree (MDT)

A multicast distribution tree (MDT) is named as such due to the layout of the network topology. The root is the multicast source (the server) and the leaves are multicast receivers (clients). The total load of multicast data on the network does not increase with the addition of more clients, which is the biggest advantage of using multicast.

In the following example you will learn how multicast data packets transmit along an MDT on an IP network.



In unicast transmission mode, after receiving a data packet from a source, a router will look up the packet's destination IP address and then determine the packet's outbound interface recorded in the forwarding table.

Use the preceding figure as an example. If a data packet from the source is requested by 7 receivers, Router-X will read 7 destination IP addresses and find 2 outbound interfaces for the data packet. This destination IP address and outbound interface determination process is time-consuming and may overload the router when a large number of data packets are requested by numerous users. To resolve this issue, multicast groups were introduced.

Multicast Group

A multicast group consists of a group of receivers that require the same data stream, which does not identify specific hosts on a network. Multicast groups are unidirectional and useful only in the data transmission direction from a multicast source to receivers.

Multicast Group Address

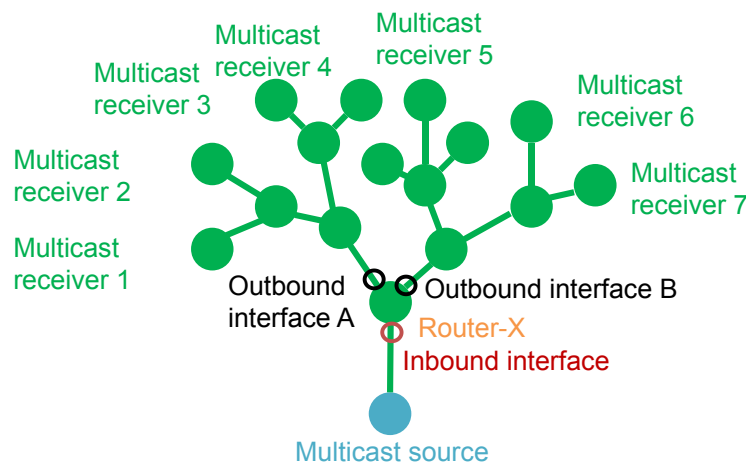
Multicast IP addresses belong to Class D addresses (ranging from 224.0.0.0 and 239.255.255.255). They are used to identify multicast groups to implement correct IP packet forwarding at the network layer.

Addresses 224.0.0.0 to 224.0.0.255 are reserved for local network protocols. For example, 224.0.0.5 and 224.0.0.6 are used by OSPF. A router does not forward packets whose destinations are these two addresses, regardless of what the packet TTL values are.

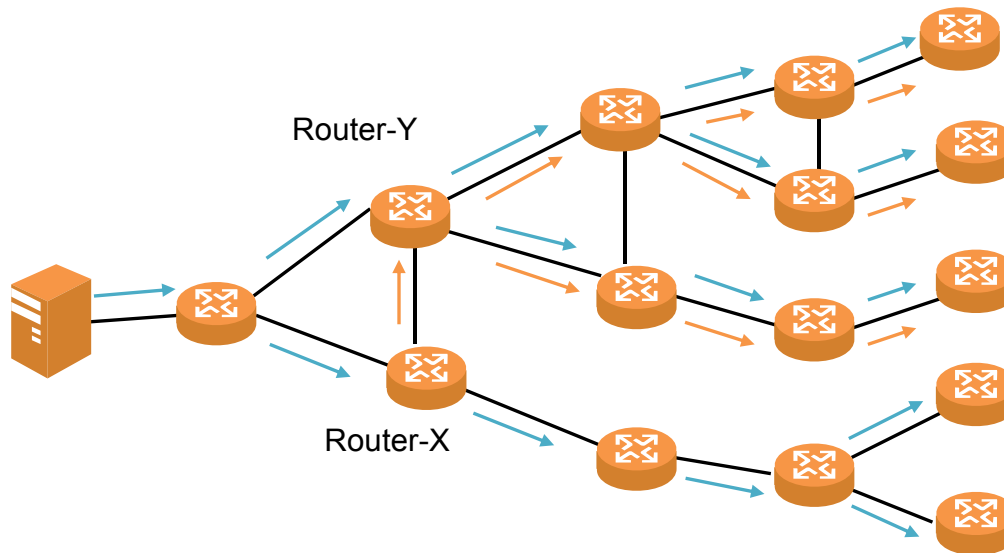
MAC addresses are required for multicast packet forwarding at Layer 2 (data link layer). Similar to unicast MAC addresses, the length of a multicast MAC address is 48 bits, written as XX-XX-XX-XX-XX in hexadecimal notation. According to IEEE 802.3, the first bit of a MAC address indicates whether the address is a unicast, broadcast, or multicast address. If the first bit is 0, it indicates a unicast address; if the first bit is 1, it indicates a multicast or broadcast address. A broadcast address is written only in the format of FF-FF-FF-FF-FF. Most multicast MAC addresses start with 01-00-5E and are written in the format of 01-00-5E-XX-XX-XX.

Multicast Forwarding Table Elements

In the example figure below, upon the receipt of a multicast data packet, Router-X first looks up the packet's destination IP address in the forwarding table to find the matching outbound interface. According to the preceding multicast group address explanation, the destination address in the multicast forwarding table is the multicast group address and the outbound interface list is {interface A, interface B}. If the multicast group address matches the destination address of the multicast packet, the multicast packet is then sent out from interfaces A or B.



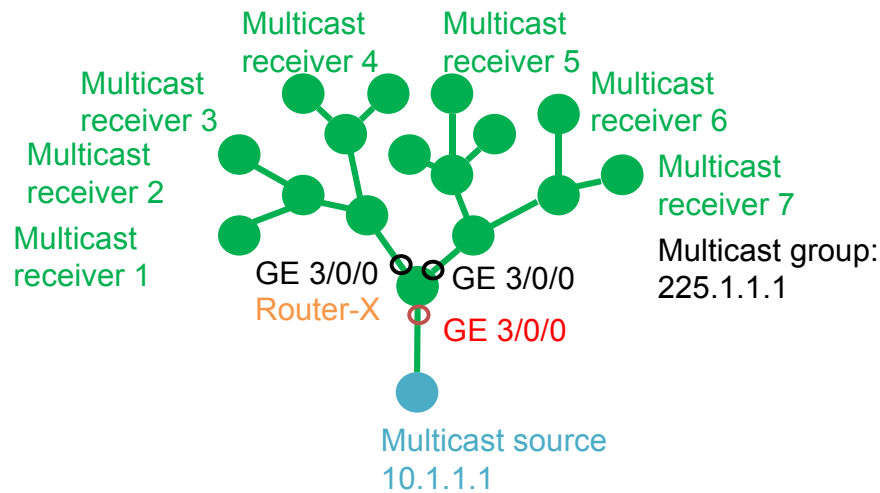
However, because multicast packets are sent to a group of receivers, the potential for incorrect multicast data forwarding increases, which can severely interrupt the network. In the following figure, the blue arrows represent correct multicast data forwarding. If Router-X sends one copy of the multicast data to Router-Y by mistake, a lot of unnecessary multicast data (represented by red arrows) will flood the network, and waste resources (the negative impact further increases if routing loops exist).



To ensure correct forwarding, multicast data must be forwarded along the MDT. Reverse Path Forwarding (RPF) check is implemented to ensure the correct forwarding path is used, and to avoid routing loops. During RPF checks, when a router receives a multicast packet, it will look up the route of the multicast source in the unicast forwarding table. If the outbound interface used in this route is the inbound interface of the packet, the multicast packet passes the RPF check and is forwarded. Therefore, in multicast forwarding, both the source and destination of a multicast data packet are taken care of. RPF check for every multicast packet, however, increases the router's workload. To solve this problem, unicast reverse path forwarding (uRPF) checks are implemented before the forwarding entry is generated in the routing table. In this way, the multicast source address, and the interface toward the multicast source, are also included in the forwarding entry. When multicast packets are forwarded, the router then only needs to check whether the source address of the packet is the multicast source of the forwarding table, and the destination address is the multicast group of the forwarding table.

If the result is correct, the router then checks whether the inbound interface of the packet is the interface toward the multicast source. If the result remains correct, the multicast packet is then forwarded; otherwise the packet is discarded.

In summary, a multicast forwarding table contains the following items: a multicast source, a multicast group, an interface connected to the multicast source, and the outbound interface list. The multicast source and group addresses are usually matched in pairs and displayed in the form of (S, G), in which S stands for a multicast source, and G for a multicast group. As shown in the figure below, the multicast forwarding entry generated by Router-X is (10.1.1.1, 255.1.1.1), GE1/0/0, {GE2/0/0, GE3/0/0}.



Multicast Forwarding table:

```
Multicast Forwarding Table of VPN-Instance: public net
Total 1 entry, 1 matched
00001. (10.1.1.1, 225.1.1.1), MID: 0, Flags: 0x0:0
  Uptime: 00:08:32, Timeout in: 00:03:26
  Incoming interface: GigabitEthernet1/0/0
  List of 1 outgoing interfaces:
    1: GigabitEthernet2/0/0
    2: GigabitEthernet3/0/0
  Matched 18696 packets(523488 bytes), Wrong If 0 packets
  Forwarded 18696 packets(523488 bytes)
```

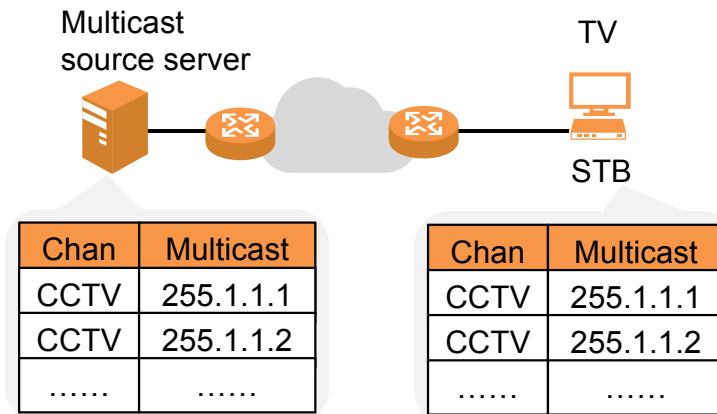
To generate a multicast forwarding entry such as the above example, the router needs to discover the following:

- Multicast groups
- Receivers in the multicast groups
- Multicast group sources

How Are Multicast Groups Established?

To establish multicast groups, routers must first learn the destination IP addresses in unicast transmissions. IP addresses are manually configured or dynamically allocated by the RADIUS server for hosts. These IP addresses are then advertised on the network for routers to learn the network topology and generate routes through network protocols. The router then selects the optimal routes and places them in the routing and forwarding tables. To forward a packet, the router looks up the IP address of the packet in the forwarding table to find the appropriate outbound interface.

The destination address of a multicast packet, however, is a multicast group address that does not represent a host address. As a result, the multicast group address cannot be allocated to a host or configured on an interface. Instead, a multicast group is a convention between the multicast source and receivers, similar in means as a TV channel. Using the figure below as an example, there is a list of TV channels corresponding with multicast group addresses on the set-top box (STB).

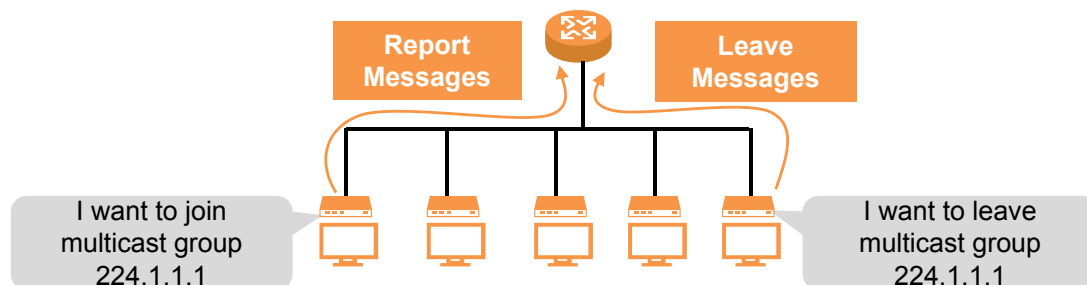


As shown in the above tables, the multicast group addresses of both the STB and the server are identical.

What Are Receivers in a Multicast Group?

Receivers in a multicast group are identified using the Internet Group Management Protocol (IGMP). IGMP sets up and manages the membership in a multicast group on directly connected network segments. To be more specific, it manages the group members on different interfaces.

IPTV is used as an example to illustrate how IGMP works:

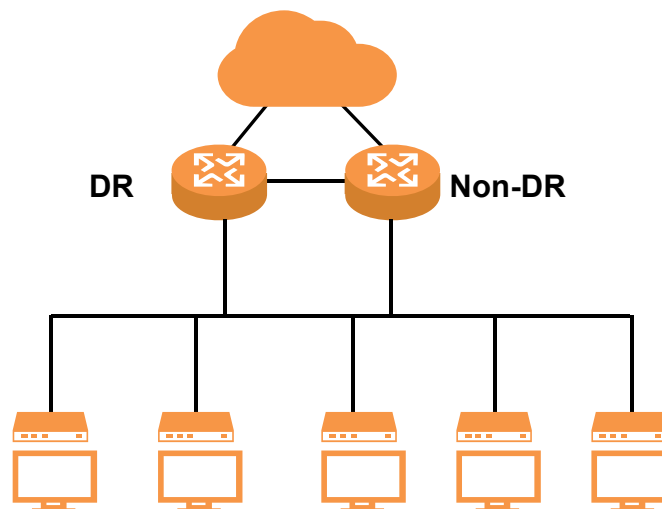


- When an STB user wants to watch a TV channel, the STB sends an IGMP Report message to join the multicast group of the selected TV channel (the multicast group 224.1.1.1 in this example).
- When the viewer stops watching the TV channel, the STB sends an IGMP Leave message to leave the multicast group of this channel.

- The router sends IGMP Query messages to learn whether there are receivers in a multicast group through the Report or Leave messages from hosts.

Similar to an audience selecting TV channels (an audience can turn the TV on and off, or switch between channels at any time), the STB can join or leave a multicast group anywhere and at anytime, without limit.

To enhance reliability, two or more routers are deployed on a LAN. A "representative" for these routers is selected by IGMP to forward the multicast data to all multicast receivers. This "representative" is called a receiver's Designated Router (DR). All multicast receivers receive multicast data and send IGMP messages through this DR. An example of this network is shown below:



Multicast routing protocols are also needed to ensure that the addresses and information about members of multicast groups are sent to all multicast routers on the network.

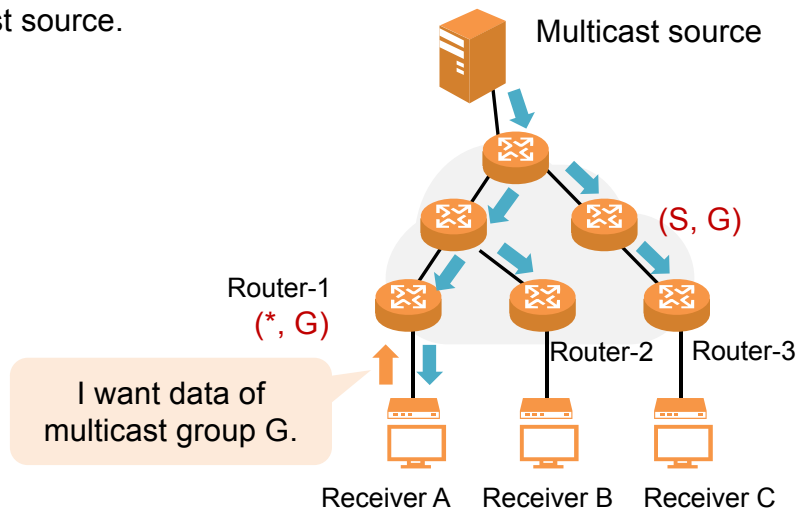
Among various multicast routing protocols, the most widely used one is Protocol Independent Multicast (PIM). Protocol independent means that there is no special protocol requirement on unicast routing.

The PIM protocol has two modes: PIM-DM and PIM-SM.

PIM-DM

In the figure below, the viewer (Receiver A) turns on the TV to watch a TV channel. The STB reports that it wants to receive the multicast data of group G, using IGMP, to the receiver's DR (Router-1).

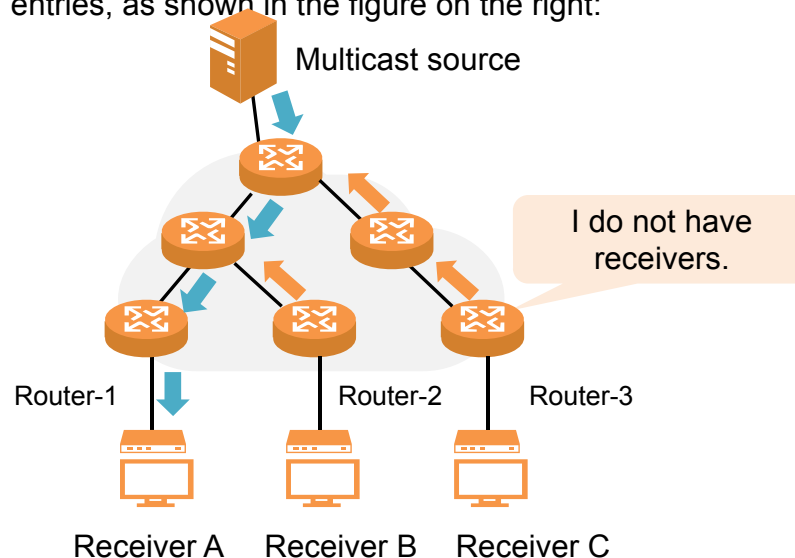
Router-1 receives the IP address of multicast group G. However, Router-1 does not know where the multicast source is, so a (*, G) entry is generated where * represents a multicast source.



When the multicast source sends multicast data to the network, all multicast routers forward the data. As a result, Router-1, Router-2, and Router-3 all receive the same multicast data. Because Receiver A wants the data, Router-1 sends the data to Receiver A. Router-2 and Router-3 do not forward the data to Receiver B or C.

When multicast data is transmitted on the network, multicast routers can obtain the IP address of the multicast source, the inbound interfaces to the multicast source, and the IP addresses of multicast groups. The outbound interface list includes all interfaces connected to the downstream interfaces. At this point, multicast entries (S, G) can be generated. The receiver DR can identify its receivers and update the outbound interface list to get the correct multicast forwarding entries. However, in the process detailed above,

Router-2 and Router-3 do not have any receivers. To save network resources, they inform the upstream router to delete interfaces connected to them from the outbound interface list, so that the multicast data is not forwarded to them. This process is called Prune. When the Prune process is complete, all routers on the network have the correct forwarding entries, as shown in the figure on the right:



In addition, the multicast source may connect to the network with multiple routers. Using PIM-DM, a router is selected as the source's DR to send and receive multicast data. However, some routers still have (S, G) entries listed after the Prune process has been completed, even when they do not have any receivers. Only their outbound interface lists are empty. To resolve this problem, PIM-SM was introduced.

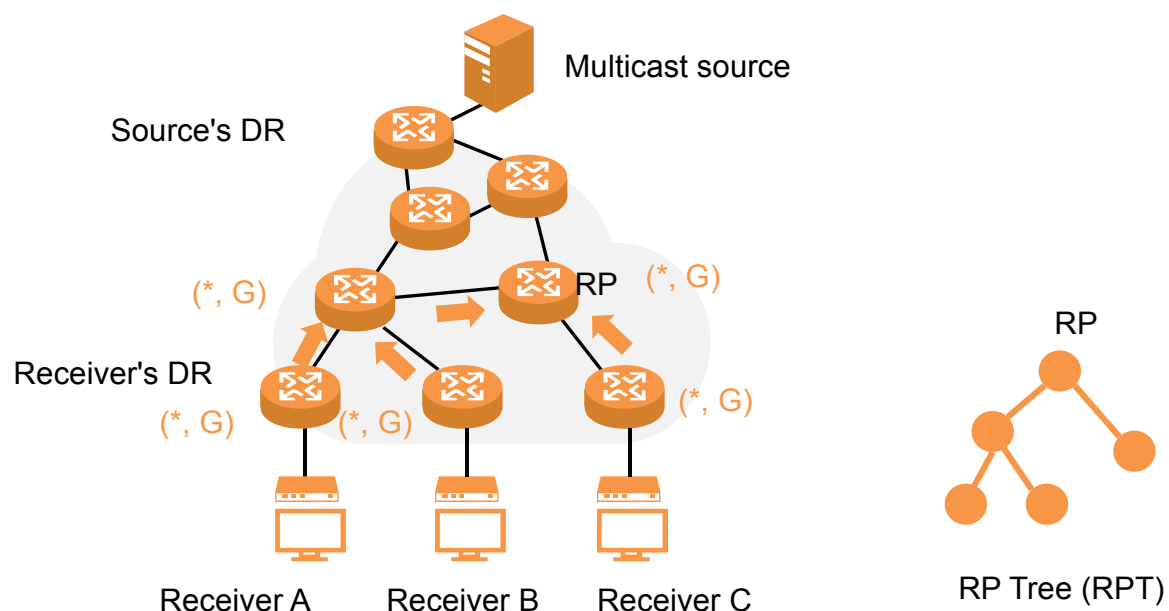
PIM-SM

Selecting an "Agent"

A multicast router on the network is selected as the Rendezvous Point (RP) using PIM-SM. The RP works as an agent. All other routers on the network need to learn the RP position. By using PIM-SM, another router is selected as the source's DR to send and receive multicast data.

Establishing an "RP Tree"

Once the receiver's DR is selected, it sends a PIM Join message to the RP. The (*, G) entry, used to identify the RP address and the group address, is advertised to all routers between the receiver's DR and the RP. As a result, the RP is established as the root of the shared tree. Similar to PIM-DM, the receiver then sends an IGMP Join message to the receiver's DR to join multicast group G, and the receiver router generates the (*, G) entry.

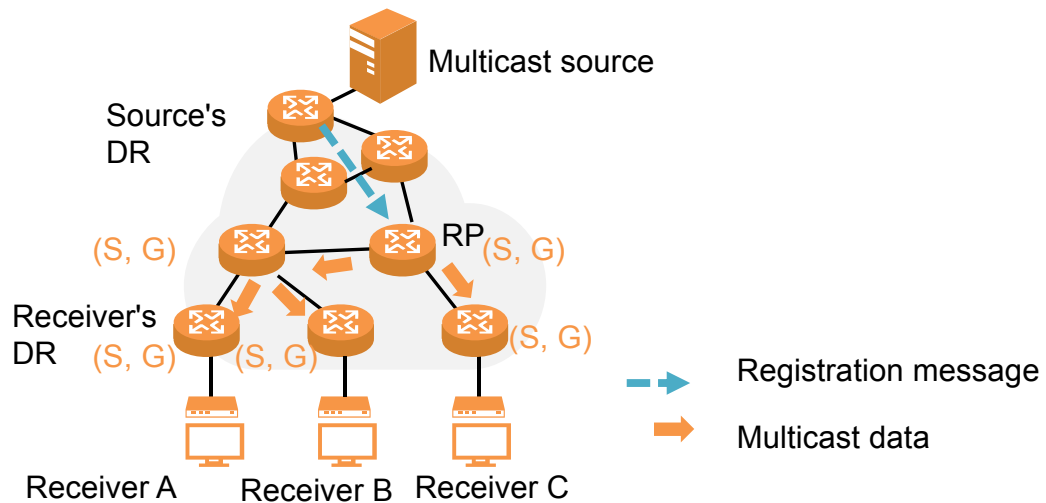


Multicast Source Registration

When the source's DR receives multicast data from the multicast source, as it does not have the multicast forwarding entry, it encapsulates the received multicast data packet in a Register message and sends this message to the RP for notification. The source's DR then creates an (S, G) entry locally with an empty interface list.

After receiving the Register message, the RP decapsulates the message and sends the multicast data packet to receivers along the RPT. As a result, every router on the path can generate an (S, G) entry correctly and replicate all interface information from the (*, G) entry.

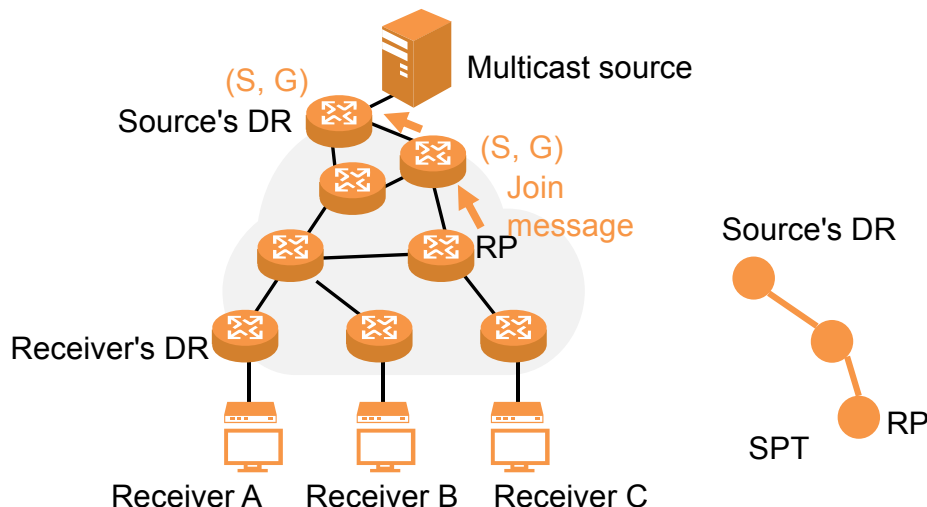
If there is no RPT when the RP receives the Register message, the RP discards the packet.



Establishing the Shortest Path Tree

If a lot of multicast data needs to be sent to the RP, the source's DR must first encapsulate the multicast data in Register messages.

The RP then has to decapsulate every Register message to extract the multicast data. This is a very inefficient process. To address this issue, the RP creates a shortest path tree (SPT) between itself and the source's DR to receive multicast data directly through the SPT. To establish an SPT, the RP creates an (S, G) entry in the multicast table, in which the inbound interface is the interface that receives Join messages. It then sends a Join message to the source's DR. This means that every RP's upstream routers can receive the correct (S, G) entry and that an SPT (with the source's DR as its root) is also created.



After receiving the RP's Join message and correct (S, G) entry, the source's DR can forward multicast data based on the (S, G) entry.

After the SPT is established, the RP can receive the multicast data of a multicast group through the SPT and changes the inbound interface to the interface that receives the multicast data.

The RP is also able to send a Register Stop message to the source's DR to stop the source's DR from sending multicast data in Register messages.

This allows the source's DR to send multicast data packets only through the SPT, stopping multicast data packets being encapsulated and sent as Register messages.



Note:

A source's DR will keep sending Register messages and multicast data packets through the SPT, until receiving a Register Stop message. This is due to the following scenario:

Join messages are multicast messages that contain:

- A source IP address identical to the local interface
- A destination address 224.0.0.13
- A TTL value of 1

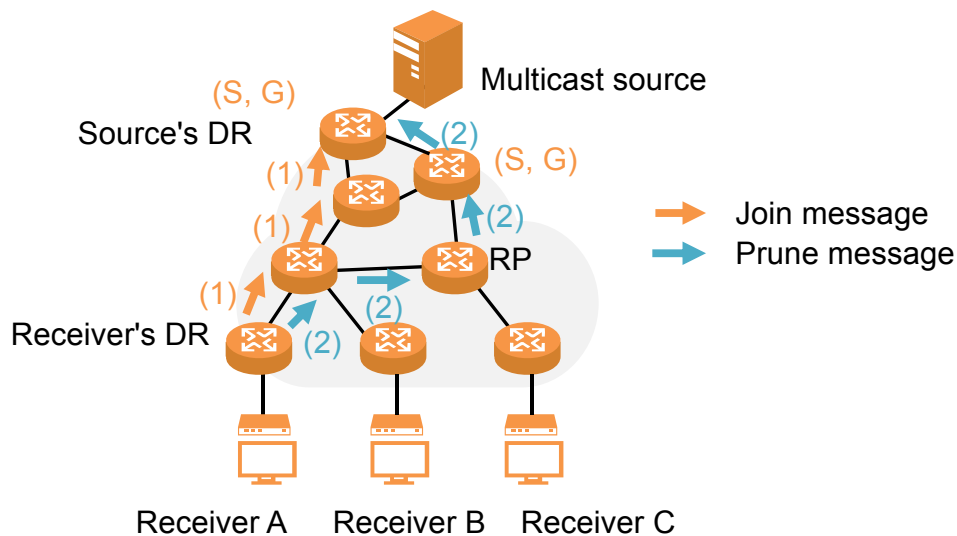
When a Join message sent upstream by the RP reaches the source's DR, the source IP address of the message is not that of the RP. As a result, the source's DR cannot identify whether the Register message was initially sent by the RP or the receiver's DR (the receiver's DR is also able to send a Join message to the source's DR). The source's DR also cannot determine whether the multicast packets it sent can be received by the RP. Only when the source's DR receives a Register Stop message can the source's DR confirm that its multicast packets can be received by the RP.

The source's DR also needs to confirm if its multicast data packets are received by the RP. In the (S, G) entries generated by the RP, when the RP receives Register messages from the source's DR, the inbound interface is the one that receives the Register message. However, the RP may not receive multicast data packets and Register messages on the same interface. As a result, the source's DR needs to confirm that the multicast packets it sent can be received by the RP before the source's DR stops sending Register messages.

SPT Switchover

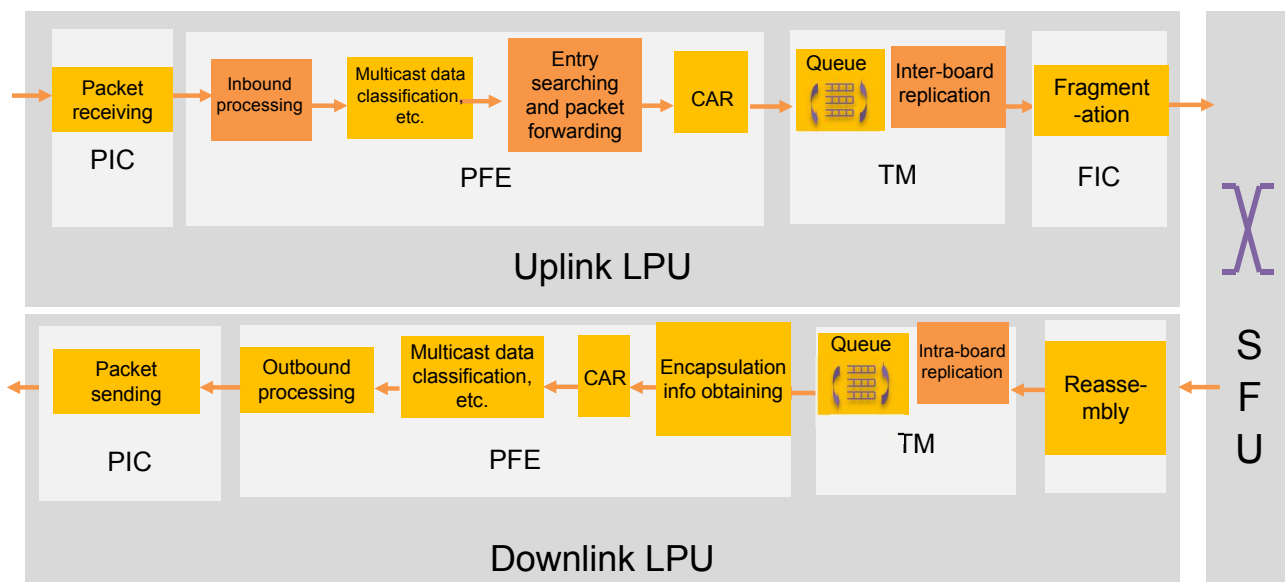
In the preceding process, all multicast data must be forwarded by the RP, which may overload the RP when the multicast packet sending rate is high.

To resolve this issue, PIM-SM allows the receiver's DR to send Join messages along the shortest path tree to the source's DR. This generates the SPT from the receiver's DR to the source's DR. The receiver's DR then sends a Prune message to the RP to prune the original RPT. The source's DR prunes the SPT between the source's DR and the RP, allowing multicast data to be forwarded along its SPT instead of through the RP. The process is illustrated in the following figure:

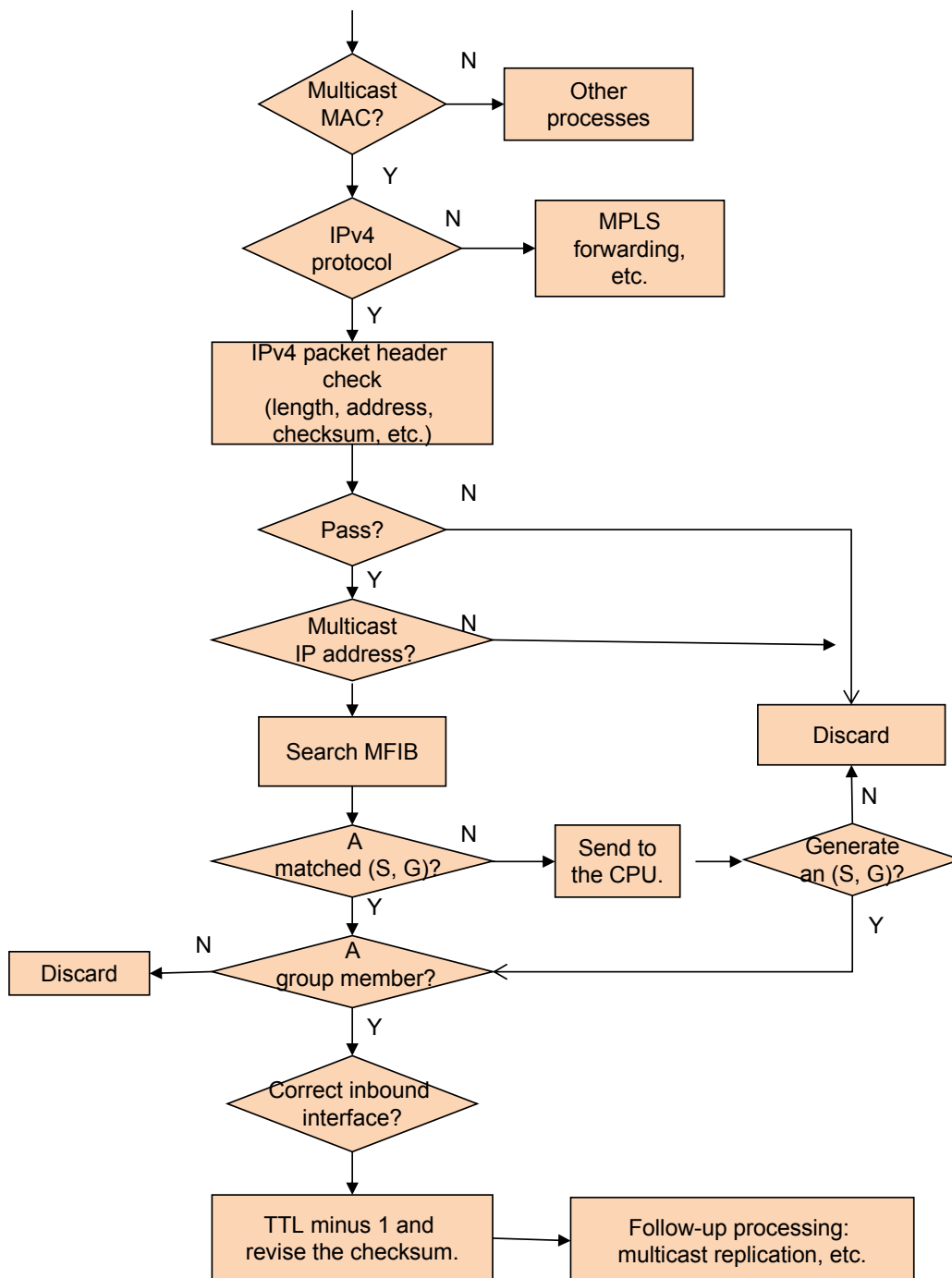


IP Multicast Forwarding Process

The IP multicast forwarding process is shown in the following figures, with an emphasis on multicast forwarding based on the forwarding table.



IP multicast forwarding process (based on the routing table):



Step 1. Determine whether the destination MAC of the packet is a multicast MAC. If it is, the packet proceeds to the next step. If not, unicast forwarding is performed.

Step 2. Determine whether the packet is an IP packet. If it is, the packet proceeds to the next step. If not, other forwarding processes (such as MPLS forwarding) are performed.

Step 3. Check whether the length, IP address, and checksum of the packet are correct. If they are, the packet proceeds to the next step. If not, the packet is discarded.

Step 4. Check whether the IP address is a multicast address. If it is, the packet proceeds to the next step. If not, the packet is discarded.

Step 5. Check whether IP multicast is enabled on the inbound interface. If it is, the packet proceeds to Step 7. If not, the packet proceeds to the next step.

Step 6. Check whether there is an (S, G) entry in the multicast FIB (MFIB) table of the public network or the VPN that matches the packet. If there is a matching (S, G) entry, the following scenarios may occur:

- If the inbound interface of the forwarding entry is the same as that on which the packet was received, the packet proceeds to Step 7. However, if the outbound interface of a device is in Register state, the device is the source's DR, and has not received a Register-Stop message. The multicast packets received by this device first needs to be sent to the CPU, encapsulated in Register messages, and then sent to the RP.
- If the inbound interface is not the same as that on which the packet was received, the packet is sent to the CPU for processing. The CPU performs RPF check. If the interface toward the multicast source is the same as the inbound interface in the (S, G) entry based on the unicast routing table, the (S, G) entry is correct, but the packet is forwarded along an incorrect path, and therefore discarded. If not, the (S, G) entry is aged out. The inbound interface in the (S, G) entry, based on the unicast routing table, is then updated, as well as the forwarding table. The router then checks whether the interface on which the packet is received is the updated interface. If it is, the packet proceeds to Step 7. If not, the packet is discarded.

When there is no matching (S, G) entry, the following scenarios may occur:

- ◆ If the router is the source's DR and receives the first multicast packet from the multicast source, this means that (S, G) entries have not been created. Multicast packets must first be sent to the CPU to be encapsulated into Register messages and then sent to the RP. The CPU sends (S, G) entries in which there is no outbound interface. Upon the receipt of RP's Register messages, the CPU adds outbound interfaces to the (S, G) entries. If the source's DR fails to register to the RP, to alleviate the workload of CPU, the multicast data afterwards is forwarded directly, not in Register messages. The outbound interface in the (S, G) entry, however, is empty, which causes the packet to be discarded.
- ◆ If the routers along the RPT only have (*, G) entries but no (S, G) entries when the first packet of a multicast group is sent from the RP to the receiver, multicast packets are sent to the CPU for processing. The CPU generates the (S, G) entry and copies the outbound interface list from the (*, G) entry. It then performs the RPF check on the packet. If it passes the RPF check, the packet proceeds to Step 7. If it fails, the packet is discarded.

Step 7. Check whether there is a multicast group member matching the (S, G) entry. If there is a matching entry, the packet proceeds to the next step. If no match is found, or the outbound interface list is empty, the packet is discarded.

Step 8. Check whether the inbound interface of the packet is the same as that in the (S, G) entry. If it is, the packet proceeds to the next step. If not, the packet is discarded.

Step 9. Perform follow-up procedures (such as for uplink TMs, intra-board multicast replication is performed, and for downlink TMs, inter-board multicast replication is performed). See Chapters 1 to 6 for details about these procedures.

Chapter 11 MPLS Forwarding

MPLS Basics

MPLS Overview and Background

Multiprotocol Label Switching (MPLS) uses labels, not routes, to forward packets and combines the advantages of IP and asynchronous transfer mode (ATM) technology.

IP technology, while simple and cheap to deploy, relies on the longest match algorithm, which is not the most efficient choice for forwarding packets. In comparison, ATM is much more efficient at forwarding packets. ATM uses fixed-length labels (called cells) and maintains a label table, which is much smaller than a routing table. However, ATM is a complex protocol with a high deployment cost, which has hindered its widespread popularity and growth.

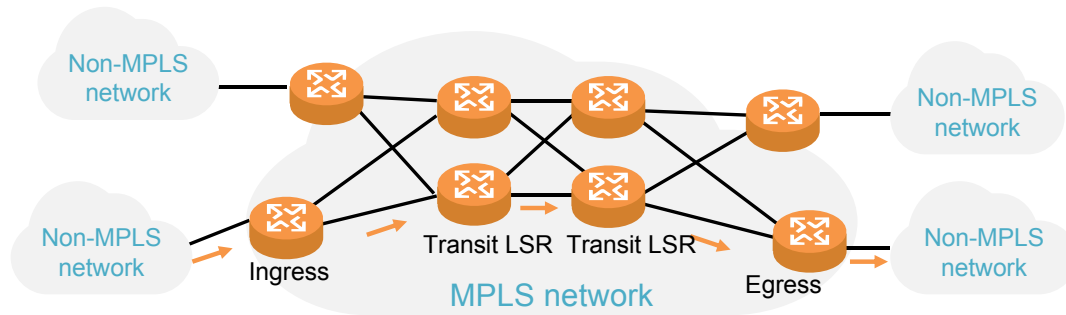
MPLS was developed to satisfy users who wanted a technology that combines the best features of both IP and ATM.

MPLS is designed to increase forwarding rates. Unlike IP, MPLS analyzes packet headers on the edge of a network, not at each hop. Therefore, packet processing time is shortened.

Although losing its advantage in accelerating the forwarding speed, MPLS supports multi-layer labels, and its forwarding plane is connection-oriented. MPLS is widely used in virtual private network (VPN), traffic engineering (TE), and quality of service (QoS) scenarios.

Typical MPLS Network Structure

The following figure illustrates the typical MPLS network structure.



Label switching routers (LSRs) are the basic elements of an MPLS network. There are three main types of LSRs:

An Ingress is on the MPLS network edge and receives packets from another network. It analyzes data packets and adds a label to them.

A transit LSR is within an MPLS domain and it forwards packets based on labels.

An egress is on the MPLS network edge and sends packets to another network. It removes the label from the packet before sending the packet out of the MPLS network.

MPLS Header and Label

An MPLS header is 4 bytes long and contains the following fields.

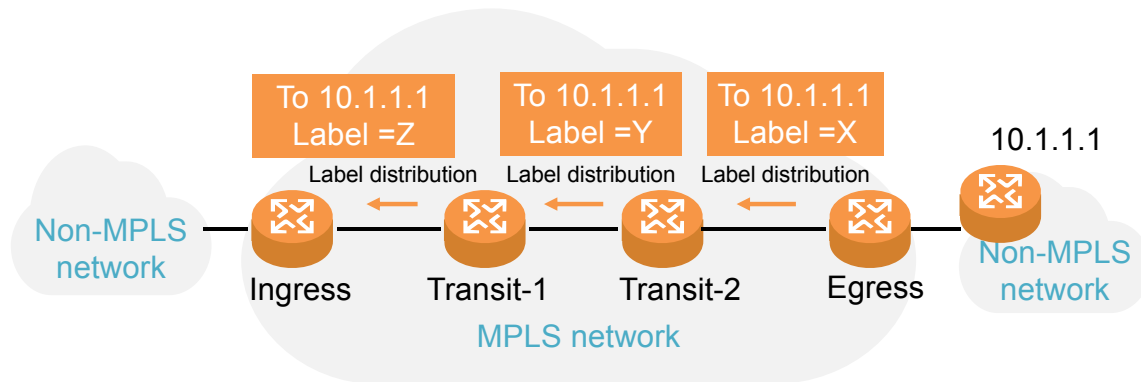
20 bits	3 bits	1 bit	8 bits
Label value	EXP	S	TTL

- **Label:** a label value.
- **EXP:** used for extension. This field is used to implement the class of service (CoS) function, which is similar to Ethernet 802.1p.
- **S:** whether a label is at the bottom of a label stack. MPLS supports multiple labels that can be stacked. Value 1 indicates a label at the bottom of a label stack.
- **TTL:** short for time to live. This field is the same as the TTL in IP packets.

Label Distribution Process

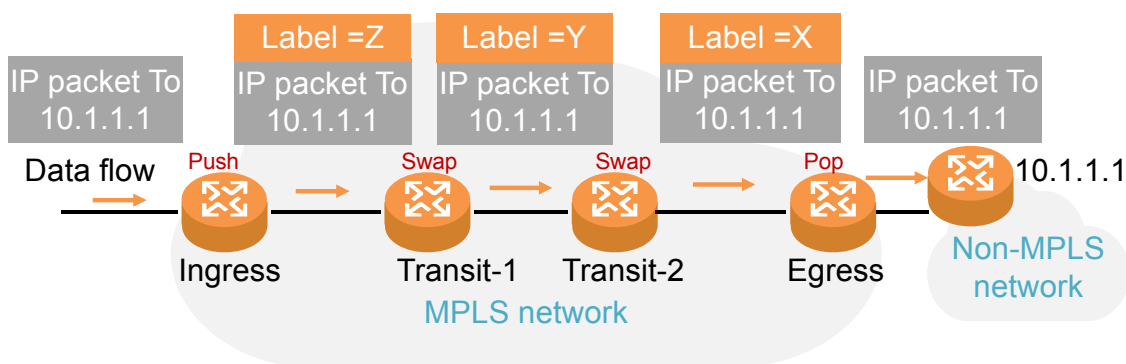
Data packets travel along label switched paths (LSPs) across an MPLS network. LSPs that are unidirectional originate from the ingress and terminate at the egress. MPLS forwarding is performed in a "road is built first, and cars go" way. Before packets are forwarded, labels must be distributed to establish an LSP.

A label is assigned by a downstream LSR to an upstream LSR. In the following figure, Transit-2 is downstream to Transit-1 and Egress is downstream to Transit-2. Conversely, Transit-1 is upstream to Transit-2 and Ingress is upstream to Transit-1.



Downstream LSRs assign labels because the downstream LSR itself uses these labels to identify a packet that can be forwarded. If an upstream LSR assigns a label, the downstream LSRs do not know how to use the label to forward a packet unless they agree on this label. To simplify the process, the downstream LSR itself assigns the label.

Packet Forwarding Process



Step 1: The ingress receives an IP packet destined for 10.1.1.1, adds label Z to the packet, and forwards it downstream.

Step 2: Transit-1 receives the labeled packet, swaps label Z for label Y, and forwards the packet downstream.

Step 3: Transit-2 receives the labeled packet, swaps label Y for label X, and also forwards the packet downstream.

Step 4: The egress receives the packet, removes label X, and forwards the packet over an IP route to 10.1.1.1.

Label Operation — Push, Swap, and Pop

- **Push:** Adding a label to a packet, as shown in Step 1.
- **Swap:** Swapping a label at the top of the label stack in an MPLS packet for another label assigned by a next hop, as shown in Steps 2 and 3.
- **Pop:** Removing a label from an MPLS packet before the packet leaves the MPLS network, as shown in Step 4. In addition, the penultimate LSR can also remove a label from an MPLS packet. This process is called penultimate hop popping (PHP), and is described in the following section.

PHP Mechanism and Implicit Null Label

Assume that an MPLS packet arrives at the egress, the last hop of an LSP. The egress looks up its MPLS forwarding table for a matching entry and removes the label from the MPLS packet. After the egress finds that the MPLS packet becomes an unlabeled IP packet, the egress re-looks up for an entry in the IP forwarding table and forwards the packet. Obviously, the lookup in the MPLS forwarding table is unnecessary and reduces forwarding efficiency. To make an improvement, PHP enables the egress to instruct the penultimate LSR to remove the last label from the MPLS packet before sending the packet to the egress. After receiving the packet, the egress directly forwards the unlabeled IP packet or single-labeled packet. PHP helps reduce the burden on the egress.

PHP enables the egress to assign only implicit null label (label 3), to the penultimate LSR. The implicit null label is removed before appearing in the label stack of a packet reaching the egress. When an implicit null label is distributed to a penultimate LSR, the LSR directly removes the label without having to swap an existing label for it at the top of the label stack.

MPLS VPN Overview

As mentioned in the MPLS background, MPLS is widely used in virtual private network (VPN) services. Well, what is VPN?

Before the advent of VPN, telecom carriers rented Layer 2 leased lines to enterprises. Each leased line was exclusive to a specific enterprise.

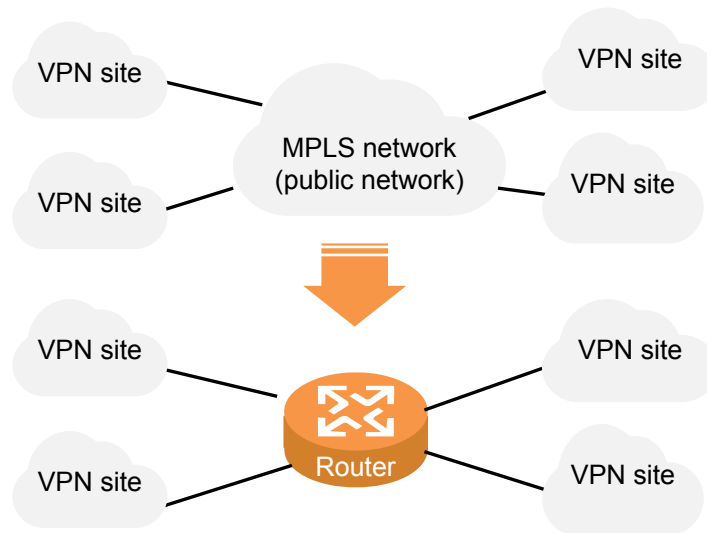
Each time a leased line was added, a new physical link needed to be built, which was time consuming and costly. ATM and Frame Relay (FR), following the leased line technique, enabled carriers to use virtual circuits to provide P2P leased lines. Such leased lines were time-saving and of low cost. Virtual circuits depended on dedicated transmission media. Either ATM or FR if used must be supported by all devices in all service areas. Costly network construction and low transmission rates made virtual circuit-based services lag behind the development speed of applications on the Internet.

A substitute solution was invented to use VPNs over an existing IP network. The nature of VPNs is to provide virtual leased line services over a shared network (known as a public network), which poses a problem. No enterprise wants its data to be exposed on the shared public network, and their VPNs must be isolated from one another. Packets of a specific VPN must be transparently transmitted over the public network. To tackle this issue, VPNs use the tunneling technique to transmit data.

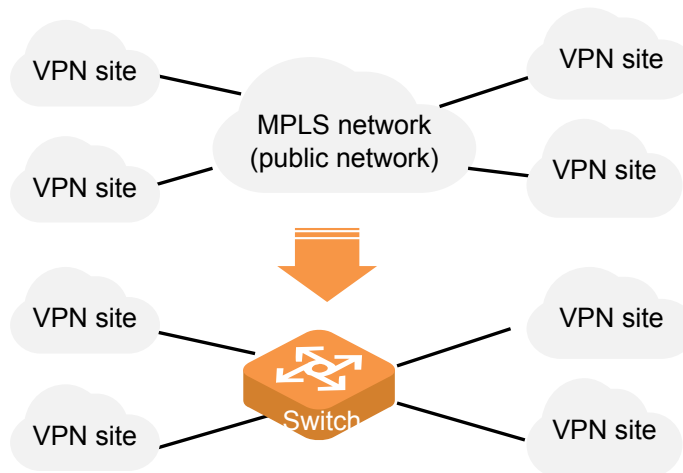
The tunneling technique provides a tunnel between two network nodes to transparently exchange data. Tunnel protocols are used to establish tunnels. They include Generic Routing Encapsulation (GRE), Layer 2 Tunneling Protocol (L2TP), and MPLS that is what we are talking about. After a tunnel is established, one end adds a tunnel protocol header to each packet and forwards the packet to the other end. Upon receipt of it, the other end removes the header and forwards the packet. Tunnels, including MPLS LSPs, are the integral part of VPNs. MPLS LSPs are the most commonly used on carrier networks. VPNs that transmit data along MPLS LSPs are called MPLS VPNs.

MPLS VPNs are classified as MPLS L3VPNs or MPLS L2VPNs. MPLS L2VPNs involve virtual private LAN service (VPLS) tunnels and virtual leased line (VLL) and pseudo wire emulation edge-to-edge (PWE3) tunnels.

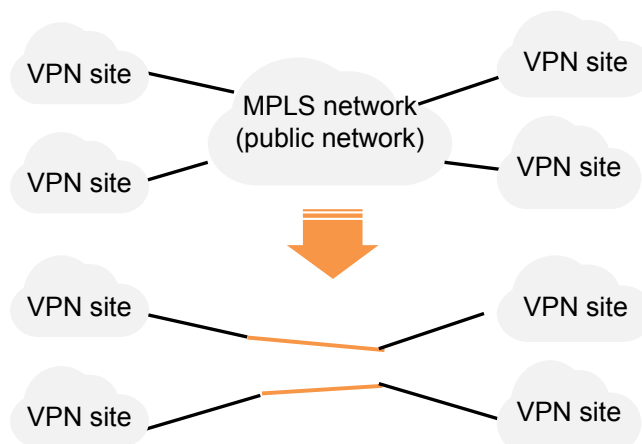
- **MPLS L3VPN:** L3VPN users consider that a shared network (public network) is like a router connecting to VPN sites to one another. The public network builds dedicated routing and forwarding tables for each VPN.



- **VPLS:** uses virtual leased network segments to connect LANs to one another. For VPLS users, a public network is like an Ethernet switch connecting VPN sites to each other. VPLS is also called E-LAN.



- **VLL and PWE3:** use an IP network to simulate traditional leased lines. VLL users take a public network as a P2P link to connect VPN sites to one another. VLL is also called virtual private wire service (VPWS) or E-Line. PWE3 is an extension to VLL.



MPLS Label Position

Labels are encapsulated between the data link and network layers. Their position in a data packet is illustrated in the following figure.



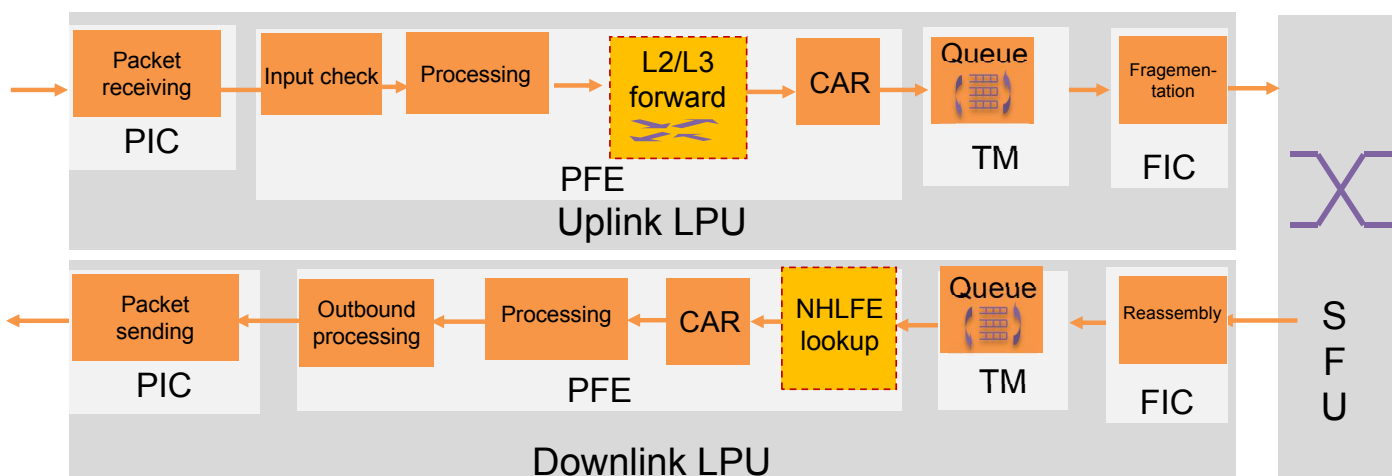
An MPLS packet can carry multiple labels. The label next to the physical layer is called the outer label or stack top label. The label next to the network layer is called the inner label or stack bottom label. Theoretically, an unlimited number of MPLS labels can be stacked in an MPLS packet.

On an MPLS VPN, the most inner label is the VPN label, also called a private network label. The most outer label is the tunnel label, also called a public network label.

MPLS Forwarding Process

Processing on the Ingress

After a data packet enters an MPLS network, the ingress analyzes it and adds a label to the packet. Transit LSRs forward the packet based on labels carried in the packet. The egress removes the label from the packet before sending the packet out of the MPLS network.



The processing on the ingress is as follows:

Step 1: The uplink packet forwarding engine (PFE) parses a received packet and determines the forwarding type. The process is the same as the Layer 2 or Layer 3 IP forwarding process for handling an incoming packet. Layer 2 forwarding is performed for an incoming packet in a VPLS, VLL, or PWE3 scenario, and Layer 3 IP forwarding in an MPLS VPN scenario.

- **MPLS L3VPN scenario:** Layer 3 IP forwarding is performed for incoming packets. The ingress searches the forwarding information base (FIB) table for a matching entry. If the tunnel ID is 0x0, common IP forwarding is performed. If the tunnel ID is not 0x0, MPLS L3VPN forwarding is performed.

Destination/Mask	Nexthop	Flag	TimeStamp	Interface	TunnelID
1.1.1.1/32	10.0.0.1	DGU	t[347299]	GE1/0/0	0x0
10.0.0.0/24	10.0.0.1	U	t[257502]	GE1/0/0	0x0
192.0.0.0/24	10.0.0.1	DGHUT	t[670625]	GE1/0/0	0x2000001
127.0.0.0/8	127.0.0.1	U	t[102]	InLoop0	0x0

VPN packets are transparently transmitted along LSPs in an MPLS domain. The outbound interface of VPN packets on the ingress connects to an LSP. To provide a uniform interface for upper-layer applications (such as VPN and route management) that use tunnels, the ingress automatically assigns an ID to each tunnel. Such ID is called a tunnel ID that is valid only on a local node. The following figure shows the format of a tunnel ID.

LSP Token	Sequence-number	Slot-ID	Allocation Method
-----------	-----------------	---------	-------------------

The **LSP Token** field is used to search the MPLS forwarding table for matching entries. The LSP token value is merely an index used in MPLS forwarding entry lookup.

- **VPLS:** The ingress performs Layer 2 bridge forwarding for incoming packets. The ingress searches the MAC entry table based on the destination MAC address and VLAN ID in a packet and finds a matching outbound interface name and LSP token. The following figure shows an example of a MAC entry table:

MAC Address	VLAN/ VSI/SI	PEVLAN	CEVLAN	Port	Type	LSP/LSR-ID MAC-Tunnel
0005-0005-0005	b	-	-	GE3/1/6	static	3/-

- **VLL/PWE3:** Layer 2 VPN forwarding is performed for incoming packets.

Step 1: The ingress searches the L2VPN forwarding table and finds the matching control word, outbound interface information, and LSP token.

Step 2: The ingress performs common processing and uses a switch fabric unit (SFU) to an outbound interface.

Step 3: The downlink packet forwarding engine (PFE) searches for the next hop label forwarding entry (NHLFE) based on the LSP token. The NHLFE table is used to guide MPLS forwarding.

Pictured to the right is an example of an NHLFE table:

```
NHLFE:
LSR Type      : Ingress
Tunnel id     : 0x2000001
Out interface  : GigabitEthernet1/0/0
Nexthop       : 10.0.0.1
Out label     : 4096
Label operation : PUSH
```

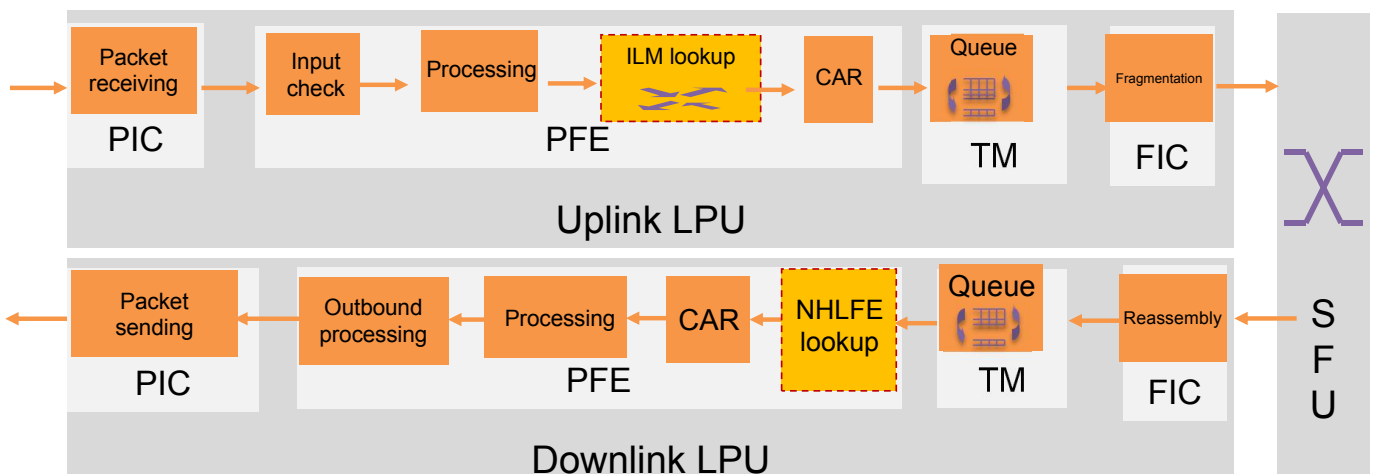
The NHLFE entry contains an inner label value, an outer label value, a label operation type, an outbound name, and a next-hop IP address.

Step 4: The ingress performs common downlink processing.

Step 5: The outbound interface module encapsulates the recently processed information into the packet. The module pushes two labels into the packet. The S field value in the inner label is 1, indicating the stack bottom label. The S field value in the outer label is 0. If VLL or PWE3 is used, the module determines whether to add a control word between the inner label and payload based on the control word before encapsulating data link layer information. If the Ethernet type is used at the data link layer, the Eth-Type field value is 0x8847. The module forwards the packet to a physical interface card (PIC), and the PIC converts the packet into electrical or optical signals and forwarding them.

Processing on a Transit LSR

Packets in MPLS L3VPN, VPLS, VLL, and PWE3 scenarios are processed in the same way on a transit LSR.



Step 1: The transit LSR parses the received packet and finds its protocol type is MPLS. The transit LSR uses the stack top label to look for a match in the incoming label mapping (ILM) table and obtain an entry that contains the tunnel ID and outbound interface information. The outbound interface information contains the target blade (TB) and target port (TP).

If load balancing is used, multiple ILM entries are found. The transit LSR uses a hash algorithm to select one ILM entry. The following figure shows an example of an ILM entry.

```
ILM:
In Label      : Ingress
Swap label    : --
Load-balance Count: 2
Tunnel id [0] : 0x2000002
Out interface [0] : GigabitEthernet2/0/0
Nexthop [0]   : 20.2.1.2
Tunnel id [1] : 0x2000003
Out interface [1] : GigabitEthernet2/0/1
Nexthop [1]   : 20.2.2.2
Has FRR LSP   : No
FRR inner label : --
FRR tunnel id  : 0
FRR out interface : no
FRR nexthop    : no
```

If fast reroute (FRR) is used, the transit LSR determines the active and standby routes based on the LSP status and outbound interface status. If the primary LSP and its outbound interface are working properly, the transit LSR selects the primary LSP. If not, the FRR LSP (backup LSP) is selected.

Step 2: If a trunk interface is used as an outbound interface, the transit LSR uses a trunk hash algorithm to select a trunk member interfaces as the outbound interface.

Step 3: The transit LSR performs common processing and uses an SFU to forward the packet downstream based on target board information.

Step 4: The downstream PFE searches for an NHLFE entry based on the tunnel ID and LSP token. The matching NHLFE entry contains outbound interface information, a next-hop IP address, an outgoing label value, and a label operation type. The label operation type is "swap" for a label with value non-3 or "pop" for a label with value 3.

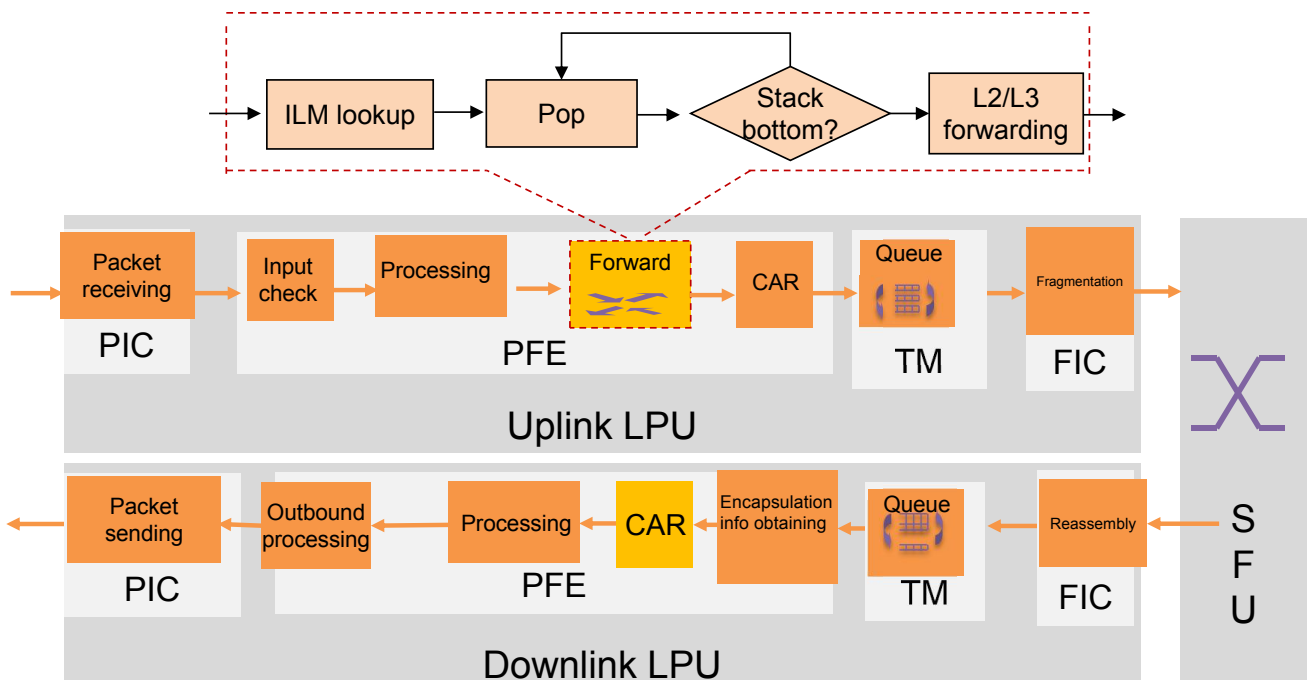
```
NHLFE:
LSR Type      : Transit
Tunnel id     : 0x2000002
Out interface  : GigabitEthernet2/0/0
Nexthop       : 20.2.1.2
Out label     : 3
Label operation : SWAP
```

Step 5: If the outgoing label value is not 3, the transit LSR swaps the outer incoming label for another outgoing label value, reduces the TTL value by one, and encapsulates data link layer information into the packet. If Ethernet is used at the data link layer, the Eth-Type value is 0x8847. If the outgoing label value is 3 (implicit null label), the transit LSR removes the outer label, sets the MPLS TTL carried in the inner label to the outer MPLS TTL value minus one, and encapsulates data link layer information into the packet.

Step 6: The transit LSR performs common processing, such as implementing the committed access rate (CAR).

Step 7: The downlink PFE uses the traffic classification QoS policy to re-set the EXP value in the MPLS header of the packet and checks the packet with the outbound interface. The transit LSR forwards the packet to the PIC, and the PIC converts the packet into electrical or optical signals and forwards them.

Processing on the Egress



Step 1: The egress searches the ILM table for an entry matching the outer label in the packet. If the label operation type is "pop" in the entry, the egress removes the label from the packet.

Step 2: The egress checks the S field in the removed label.

- If the S field is 0, the egress repeats Step 1 to remove the label from the stack.
- If the S field is 1, the egress then performs Layer 2 or Layer 3 forwarding based on payload. For details about Layer 2 and Layer 3 forwarding, see "L2 Bridge Forwarding Process" and "IP Unicast Forwarding Process."

MPLS TTL Processing

RFC 3443 defines two TTL processing modes:

Uniform mode: After an IP packet reaches the ingress, its IP TTL decreases by 1 and is mapped to its MPLS TTL field. The MPLS TTL decreases by 1 at each hop along an LSP, but the IP TTL remains constant. The egress reduces the MPLS TTL by 1 and maps the value to the IP TTL field before forwarding the packet.

Pipe mode: After an IP packet reaches the ingress, its IP TTL decreases by 1, and the MPLS TTL is fixed at a value (255 by default). The MPLS TTL decreases by 1 at each hop along an LSP, but the IP TTL remains constant. After the packet reaches the egress, the egress removes the MPLS label and reduces the IP TTL by 1 before forwarding the IP packet. In pipe mode, the IP TTL in each packet decreases by 1 only on the ingress and egress along the LSP.

Huawei high-end routers allow you to set TTL processing modes for both outer (tunnel) and inner (L3VPN) labels. The following table lists four mode combinations.

Combination	Outer (Tunnel) TTL Mode	Inner (L3VPN) TTL Mode
1	Uniform	Uniform
2	Pipes	Uniform
3	Uniform	Pipe
4	Pipe	Pipe

Each combination of TTL processing modes indicates a specific processing method.

TTL Processing on the Ingress

- If the uniform mode is used to process outer MPLS TTLs, the outer MPLS TTL copies the inner MPLS TTL value.
- If the pipe mode is used to process outer MPLS TTLs, the outer MPLS TTL is fixed at 255.
- If the uniform mode is used to process inner MPLS TTLs, the inner MPLS TTL copies the IP TTL value.
- If the pipe mode is used to process inner MPLS TTLs, the inner MPLS TTL is fixed at 255.
- In all cases, the IP TTL decreases by 1 on the ingress.

Processing on the ingress:

Combination	Outer MPLS TTL Mode	Inner MPLS TTL Mode	IP TTL Value	Inner MPLS TTL Value	Outer MPLS TTL Value
1	Uniform	Uniform	Decreases by 1.	Equal to the IP TTL.	Equal to the inner MPLS TTL.
2	Pipe	Uniform	Decreases by 1.	Equal to the IP TTL.	255
3	Uniform	Pipe	Decreases by 1.	255	Equal to the inner MPLS TTL.
4	Pipe	Pipe	Decreases by 1.	255	255

TTL Processing on a Transit LSR

- The IP TTL is unchanged.
- The outer MPLS TTL is decreases by 1 at each hop, and the inner MPLS TTL remains.
- If the penultimate LSR is assigned an implicit null label, the outer MPLS TTL is unchanged, and the inner MPLS TTL copies the outer MPLS TTL and decreases by 1.

Processing on a transit LSR in most cases:

Combination	Outer MPLS TTL Mode	Inner MPLS TTL Mode	Outer MPLS TTL Value	Inner MPLS TTL Value	IP TTL Value
1	Uniform	Uniform	Decreases by 1.	Unchanged.	Unchanged.
2	Pipe	Uniform	Decreases by 1.	Unchanged.	Unchanged.
3	Uniform	Pipe	Decreases by 1.	Unchanged.	Unchanged.
4	Pipe	Pipe	Decreases by 1.	Unchanged.	Unchanged.

Processing on the PHP-capable penultimate LSR:

Combination	Outer MPLS TTL Mode	Inner MPLS TTL Mode	Outer MPLS TTL Value	Inner MPLS TTL Value	IP TTL Value
1	Uniform	Uniform	Pop	Equal to the outer MPLS TTL minus 1.	Unchanged.
2	Pipe	Uniform	Pop	Equal to the outer MPLS TTL minus 1.	Unchanged.
3	Uniform	Pipe	Pop	Equal to the outer MPLS TTL minus 1.	Unchanged.
4	Pipe	Pipe	Pop	Equal to the outer MPLS TTL minus 1.	Unchanged.

TTL Processing on the Egress

- If the outer MPLS TTL mode is set to uniform, the inner MPLS TTL copies the outer MPLS TTL if there is an outer label. If there is no outer label, the inner MPLS TTL is unchanged (it does not copy the outer MPLS TTL value or decrease by 1).
- If the outer TTL mode is set to pipe, the inner MPLS TTL is unchanged (it does not copy the outer MPLS TTL value or decrease by 1).
- If the inner MPLS TTL mode is set to uniform, the IP TTL copies the inner MPLS TTL value and decreases by 1.
- If the inner MPLS TTL mode is set to pipe, the IP TTL decreases by 1.

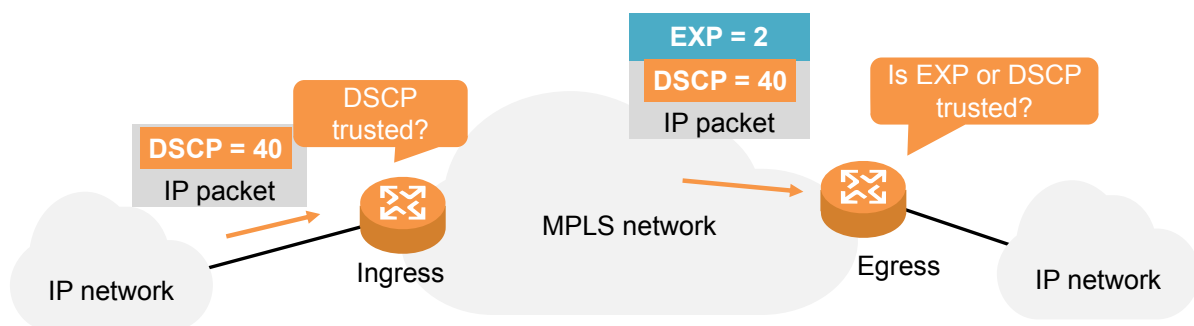
Processing on the egress:

Combination	Outer MPLS TTL Mode	Inner MPLS TTL Mode	Outer MPLS TTL Value	Inner MPLS TTL Value	IP TTL Value
1	Uniform	Uniform	Pop	Equal to the outer MPLS TTL (if there is an outer label) or unchanged (if no outer label is used).	Equal to the inner MPLS TTL minus 1.
2	Pipe	Uniform	Pop	Unchanged.	Equal to the inner MPLS TTL minus 1.
3	Uniform	Pipe	Pop	Equal to the outer MPLS TTL (if there is an outer label) or unchanged (if no outer label is used).	Decreases by 1.
4	Pipe	Pipe	Pop	Unchanged.	Decreases by 1.

MPLS CoS Processing Modes

The QoS differentiated service (DiffServ) model allows transit LSRs in a DiffServ domain to check and modify the class of service (CoS) values. CoS values include IP precedence, differentiated services code point (DSCP), and EXP values. CoS values vary during transmission on an IP or MPLS network.

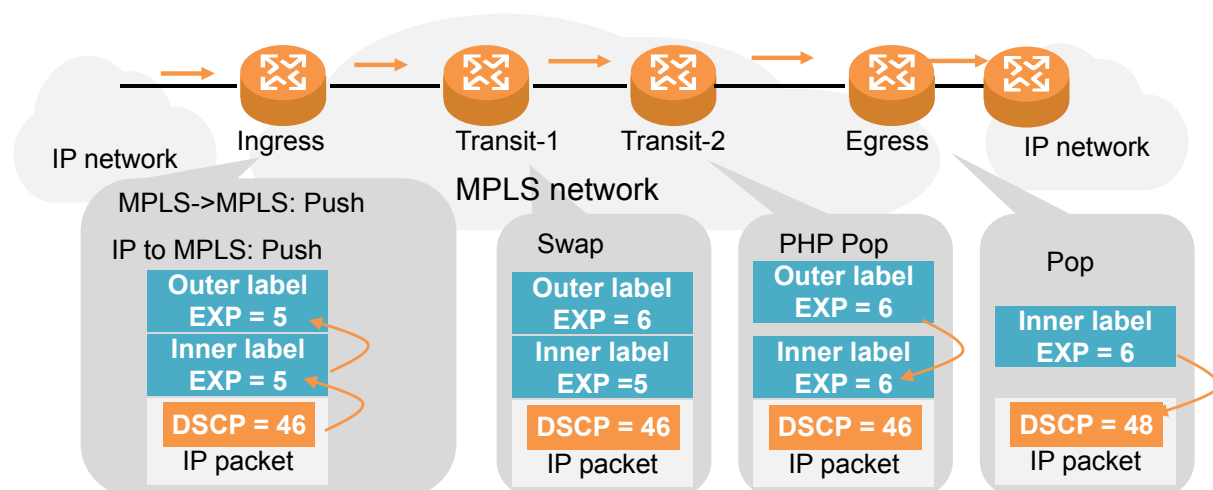
Carriers need to determine whether to trust CoS information in an IP or MPLS packet that enters an MPLS network or leaves an MPLS network for an IP network.



RFC 3270 defines three modes for processing CoS values: uniform, pipe, and short pipe.

Uniform Mode

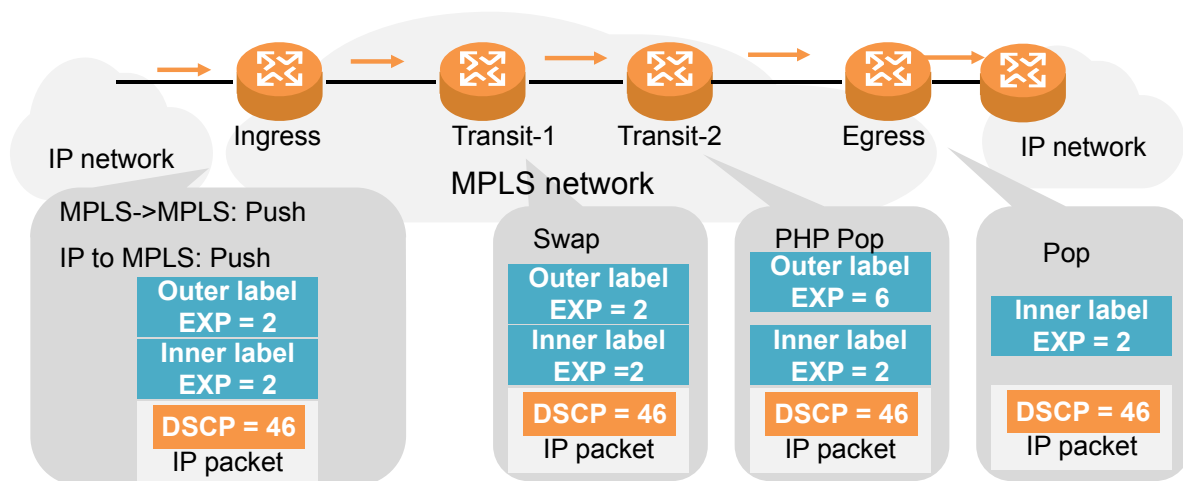
Used when carriers trust the CoS value (IP precedence or DSCP) in a packet originating from an IP network. The ingress copies the CoS value in the packet and pastes it in the EXP field of the outer label to implement the same QoS guarantee on the MPLS network. When the packet leaves the MPLS network, the egress copies the EXP value and pastes it in the IP precedence or DSCP field in the IP packet.



Uniform mode uses the same priority identifier on both the IP and MPLS networks. Priority mapping is performed when packets enter and exit an MPLS domain. One drawback of this mode is that if the EXP value in a packet changes on an MPLS network, the per-hop behavior (PHB) for the packet that leaves the MPLS network also changes. In this case, the original CoS value in the packet does not take effect.

Pipe Mode

Used when a carrier determines not to trust the CoS value in a packet sent by an IP network. The ingress sets a new EXP value in the outer MPLS label, independent of the existing CoS value. QoS guarantee is provided for the packet sent from the ingress to egress. After the packet leaves the MPLS network, the packet is scheduled based on the original CoS value.



In pipe mode, the ingress does not copy the IP precedence or DSCP value to the EXP field in each packet that enters an MPLS network. The egress does not copy the EXP value to the IP precedence or DSCP field in a packet to leave an MPLS network, either. If the EXP value in a packet changes on an MPLS network, the change takes effect only on the MPLS network. When a packet leaves an MPLS network, the original CoS value continues to take effect.

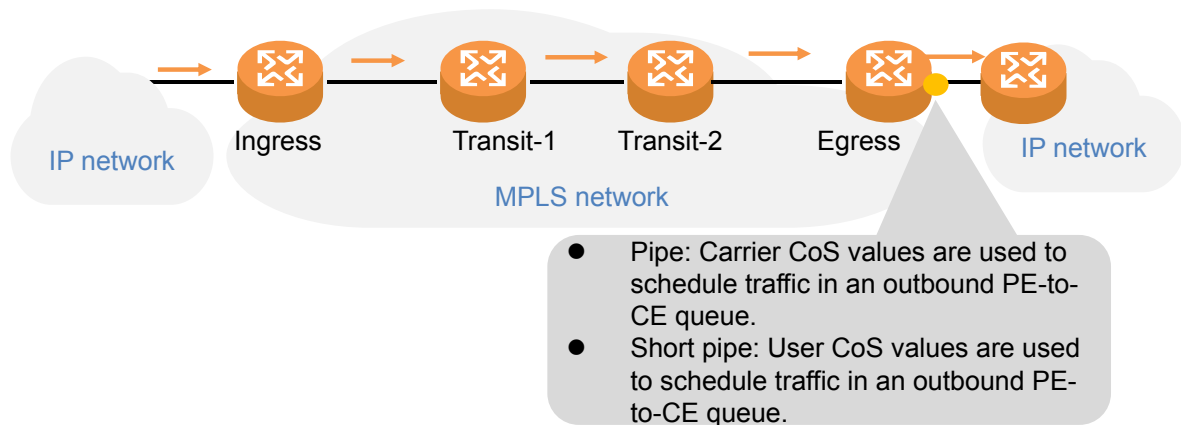
Short Pipe Mode

The short pipe mode is an enhancement of the pipe mode. Packet processing on the ingress in short pipe mode is the same as that in pipe mode. In pipe or short pipe mode, a carrier can define a desired CoS value for QoS implementation on a carrier network, without changing user-side CoS values in packets.

One difference between pipe and short pipe modes is that, in short pipe mode, the egress pops the label before implementing QoS scheduling. Packets forwarded from the ingress to the penultimate LSR are scheduled based on CoS values that a carrier defines, and packets that arrive on the egress are scheduled based on the original CoS values.

Other differences between pipe and short pipe modes include:

- In pipe mode, carrier QoS re-marking is enabled for PE-to-CE traffic.
- In short pipe mode, user QoS re-marking is used for PE-to-CE traffic.





TECH SUPPORT APP



HUAWEI

Data Documentation Dept.

<http://support.huawei.com>

