H3C SecPath 系列虚拟负载均衡产品

ACL 和 QoS 配置指导(V7)

新华三技术有限公司
http://www.h3c.com

资料版本：6W400-20210104
产品版本：E1171

前 言

本配置指导介绍了产品各软件的原理及其配置方法，包含原理简介、配置任务描述和配置举例。《ACL 和 QoS 配置指导》主要介绍 QoS 相关协议的原理和配置，包括流分类、流量监管、流量整形、QoS 策略、拥塞管理、拥塞避免、MPLS QoS 等。

前言部分包含如下内容：

读者对象

本手册主要适用于如下工程师：

- 网络规划人员
- 现场技术支持与维护人员
- 负责网络配置和维护的网络管理员

本书约定

1. 命令行格式约定

格　式	意　义
粗体	命令行关键字（命令中保持不变、必须照输的部分）采用**加粗**字体表示。
斜体	命令行参数（命令中必须由实际值进行替代的部分）采用*斜体*表示。
[ ]	表示用“[ ]”括起来的部分在命令配置时是可选的。
{ x | y | ... }	表示从多个选项中仅选取一个。
[x | y | ...]	表示从多个选项中选取一个或者不选。
{ x | y | ... }*	表示从多个选项中至少选取一个。
[x | y | ...]*	表示从多个选项中选取一个、多个或者不选。
&<1-n>	表示符号&前面的参数可以重复输入1～n次。
#	由“#”号开始的行表示为注释行。

2. 图形界面格式约定

格　式	意　义
< >	带尖括号“< >”表示按钮名，如“单击<确定>按钮”。
[ ]	带方括号“[ ]”表示窗口名、菜单名和数据表，如“弹出[新建用户]窗口”。
/	多级菜单用“/”隔开。如[文件/新建/文件夹]多级菜单表示[文件]菜单下的[新建]子菜单下

格　式	意　义
	的[文件夹]菜单项。

3. 各类标志

本书还采用各种醒目标志来表示在操作过程中应该特别注意的地方，这些标志的意义如下：

 警告	该标志后的注释需给予格外关注，不当的操作可能会对人身造成伤害。
 注意	提醒操作中应注意的事项，不当的操作可能会导致数据丢失或者设备损坏。
 提示	为确保设备配置成功或者正常工作而需要特别关注的操作或信息。
 说明	对操作内容的描述进行必要的补充和说明。
 窍门	配置、操作、或使用设备的技巧、小窍门。

4. 图标约定

本书使用的图标及其含义如下：

	该图标及其相关描述文字代表一般网络设备，如路由器、交换机、防火墙等。
	该图标及其相关描述文字代表一般意义下的路由器，以及其他运行了路由协议的设备。
	该图标及其相关描述文字代表二、三层以太网交换机，以及运行了二层协议的设备。
	该图标及其相关描述文字代表无线控制器、无线控制器业务板和有线无线一体化交换机的无线控制引擎设备。
	该图标及其相关描述文字代表无线接入点设备。
	该图标及其相关描述文字代表无线终结单元。
	该图标及其相关描述文字代表无线终结者。
	该图标及其相关描述文字代表无线Mesh设备。
	该图标代表发散的无线射频信号。
	该图标代表点到点的无线射频信号。
	该图标及其相关描述文字代表防火墙、UTM、多业务安全网关、负载均衡等安全设备。

	该图标及其相关描述文字代表一般网络设备，如路由器、交换机、防火墙等。
	该图标及其相关描述文字代表防火墙插卡、负载均衡插卡、NetStream插卡、SSL VPN插卡、IPS插卡、ACG插卡等安全插卡。

5. 示例约定

由于设备型号不同、配置不同、版本升级等原因，可能造成本手册中的内容与用户使用的设备显示信息不一致。实际使用中请以设备显示的内容为准。

本手册中出现的端口编号仅作示例，并不代表设备上实际具有此编号的端口，实际使用中请以设备上存在的端口编号为准。

资料意见反馈

如果您在使用过程中发现产品资料的任何问题，可以通过以下方式反馈：

E-mail：info@h3c.com

感谢您的反馈，让我们做得更好！

目 录

1 ACL

1.1 ACL简介

ACL（Access Control List，访问控制列表）是一系列用于识别报文流的规则的集合。这里的规则是指描述报文匹配条件的判断语句，匹配条件可以是报文的源地址、目的地址、端口号等。设备依据 ACL 规则识别出特定的报文，并根据预先设定的策略对其进行处理，最常见的应用就是使用 ACL 进行报文过滤。此外，ACL 还可应用于诸如路由、安全、QoS 等业务中识别报文，对这些报文的具体处理方式由应用 ACL 的业务模块来决定。

1.1.1 ACL 的编号和名称

用户在创建 ACL 时必须为其指定编号或名称，不同的编号对应不同类型的 ACL，如表 1-1 所示；当 ACL 创建完成后，用户就可以通过指定编号或名称的方式来应用和编辑该 ACL。

对于编号相同的基本 ACL 或高级 ACL，必须通过 **ipv6** 关键字进行区分。对于名称相同的 ACL，必须通过 **ipv6**、**mac** 关键字进行区分。

1.1.2 ACL 的分类

根据规则制订依据的不同，可以将 ACL 分为如表 1-1 所示的几种类型。

表1-1 ACL 的分类

ACL 类型	编号范围	适用的 IP 版本	规则制订依据
基本ACL	2000～2999	IPv4	报文的源IPv4地址
		IPv6	报文的源IPv6地址
高级ACL	3000～3999	IPv4	报文的源IPv4地址、目的IPv4地址、报文优先级、IPv4承载的协议类型及特性等三、四层信息
		IPv6	报文的源IPv6地址、目的IPv6地址、报文优先级、IPv6承载的协议类型及特性等三、四层信息
二层ACL	4000～4999	IPv4和IPv6	报文的源MAC地址、目的MAC地址、802.1p优先级、链路层协议类型等二层信息

1.1.3 ACL 的规则匹配顺序

当一个 ACL 中包含多条规则时，报文会按照一定的顺序与这些规则进行匹配，一旦匹配上某条规则便结束匹配过程。ACL 的规则匹配顺序有以下两种：

- 配置顺序：按照规则编号由小到大进行匹配。
- 自动排序：按照“深度优先”原则由深到浅进行匹配，各类型 ACL 的“深度优先”排序法则如表 1-2 所示。

表1-2 各类型 ACL 的“深度优先”排序法则

ACL 类型	“深度优先”排序法则
IPv4基本ACL	**1.** 先判断规则的匹配条件中是否包含 VPN 实例，包含者优先 **2.** 如果 VPN 实例的包含情况相同，再比较源 IPv4 地址范围，较小者优先 **3.** 如果源 IPv4 地址范围也相同，再比较配置的先后次序，先配置者优先
IPv4高级ACL	**1.** 先判断规则的匹配条件中是否包含 VPN 实例，包含者优先 **2.** 如果 VPN 实例的包含情况相同，再比较协议范围，指定有 IPv4 承载的协议类型者优先 **3.** 如果协议范围也相同，再比较源 IPv4 地址范围，较小者优先 **4.** 如果源 IPv4 地址范围也相同，再比较目的 IPv4 地址范围，较小者优先 **5.** 如果目的 IPv4 地址范围也相同，再比较四层端口（即 TCP/UDP 端口）号的覆盖范围，较小者优先 **6.** 如果四层端口号的覆盖范围无法比较，再比较配置的先后次序，先配置者优先
IPv6基本ACL	**1.** 先判断规则的匹配条件中是否包含 VPN 实例，包含者优先 **2.** 如果 VPN 实例的包含情况相同，再比较源 IPv6 地址范围，较小者优先 **3.** 如果源 IPv6 地址范围也相同，再比较配置的先后次序，先配置者优先
IPv6高级ACL	**1.** 先判断规则的匹配条件中是否包含 VPN 实例，包含者优先 **2.** 如果 VPN 实例的包含情况相同，再比较协议范围，指定有 IPv6 承载的协议类型者优先 **3.** 如果协议范围相同，再比较源 IPv6 地址范围，较小者优先 **4.** 如果源 IPv6 地址范围也相同，再比较目的 IPv6 地址范围，较小者优先 **5.** 如果目的 IPv6 地址范围也相同，再比较四层端口（即 TCP/UDP 端口）号的覆盖范围，较小者优先 **6.** 如果四层端口号的覆盖范围无法比较，再比较配置的先后次序，先配置者优先
二层ACL	**1.** 先比较源 MAC 地址范围，较小者优先 **2.** 如果源 MAC 地址范围相同，再比较目的 MAC 地址范围，较小者优先 **3.** 如果目的 MAC 地址范围也相同，再比较配置的先后次序，先配置者优先

说明

- 比较 IPv4 地址范围的大小，就是比较 IPv4 地址通配符掩码中“0”位的多少：“0”位越多，范围越小。通配符掩码（又称反向掩码）以点分十进制表示，并以二进制的“0”表示“匹配”，“1”表示“不关心”，这与子网掩码恰好相反，譬如子网掩码 255.255.255.0 对应的通配符掩码就是 0.0.0.255。此外，通配符掩码中的“0”或“1”可以是不连续的，这样可以更加灵活地进行匹配，譬如 0.255.0.255 就是一个合法的通配符掩码。
- 比较 IPv6 地址范围的大小，就是比较 IPv6 地址前缀的长短：前缀越长，范围越小。
- 比较 MAC 地址范围的大小，就是比较 MAC 地址掩码中“1”位的多少：“1”位越多，范围越小。

1.1.4 ACL 的步长

ACL 中的每条规则都有自己的编号，这个编号在该 ACL 中是唯一的。在创建规则时，可以手工为其指定一个编号，如未手工指定编号，则由系统为其自动分配一个编号。由于规则的编号可能影响规则匹配的顺序，因此当由系统自动分配编号时，为了方便后续在已有规则之前插入新的规则，系统通常会在相邻编号之间留下一定的空间，这个空间的大小（即相邻编号之间的差值）就称为 ACL 的步长。譬如，当步长为 5 时，系统会将编号 0、5、10、15……依次分配给新创建的规则。

系统为规则自动分配编号的方式如下：系统从规则编号的起始值开始，自动分配一个大于现有最大编号的步长最小倍数。譬如原有编号为 0、5、9、10 和 12 的五条规则，步长为 5，此时如果创建一条规则且不指定编号，那么系统将自动为其分配编号 15。

如果步长或规则编号的起始值发生了改变，ACL 内原有全部规则的编号都将自动从规则编号的起始值开始按步长重新排列。譬如，某 ACL 内原有编号为 0、5、9、10 和 15 的五条规则，当修改步长为 2 之后，这些规则的编号将依次变为 0、2、4、6 和 8。

需要注意的是，ACL 规则的匹配顺序为自动排序时，修改步长后新的编号是按照规则的匹配顺序（即“深度优先”原则）重新排序的，并非按照规则的原有编号顺序排序。譬如，步长为 5 时，规则的匹配顺序为 0、10、5……修改步长为 2 后，各规则对应的新编号为 0、2、4……

1.1.5 ACL 对分片报文的处理

传统报文过滤只对分片报文的首个分片进行匹配过滤，对后续分片一律放行，因此网络攻击者通常会构造后续分片进行流量攻击。为提高网络安全性，ACL 规则缺省会匹配所有非分片报文和分片报文的全部分片，但这样又带来效率低下的问题。为了兼顾网络安全和匹配效率，可将过滤规则配置为仅对后续分片有效。

1.2 ACL配置限制和指导

通过编号创建的 ACL，可以通过如下命令进入其视图：

- **acl** [**ipv6**] **number** *acl-number*；
- **acl** { [**ipv6**] { **advanced** | **basic** } | **mac** } *acl-number* 命令进入其视图。

通过 **acl** [**ipv6**] **number** *acl-number* **name** *acl-name* 命令指定编号和名称创建的 ACL，可以使用如下命令进入其视图：

- **acl** [**ipv6**] **name** *acl-name*，本命令仅支持进入已创建的基本或高级 ACL 视图；
- **acl** [**ipv6**] **number** *acl-number* [**name** *acl-name*]；
- **acl** { [**ipv6**] { **advanced** | **basic** } | **mac** } **name** *acl-name*。

通过 **acl** { [**ipv6**] { **advanced** | **basic** } | **mac** } **name** *acl-name* 命令指定名称创建的 ACL，可以使用如下命令进入其视图：

- **acl** [**ipv6**] **name** *acl-name*，本命令仅支持进入已创建的基本或高级 ACL 视图；
- **acl** { [**ipv6**] { **advanced** | **basic** } | **mac** } **name** *acl-name*。

如果 ACL 规则的匹配项中包含了除 IP 五元组（源 IP 地址、源端口号、目的 IP 地址、目的端口号、传输层协议）、ICMP 报文的消息类型和消息码信息、VPN 实例、日志操作和时间段之外的其它匹配项，则设备转发 ACL 匹配的这类报文时会启用慢转发流程。慢转发时设备会将报文上送控制平面，计算报文相应的表项信息。执行慢转发流程时，设备的转发能力将会有所降低。

1.3 ACL配置任务简介

ACL 配置任务如下

- 配置不同类型的 ACL
  - 配置基本 ACL
  - 配置高级 ACL
  - 配置二层 ACL
- （可选）复制 ACL
- （可选）配置 ACL 规则的加速匹配功能
- （可选）应用 ACL 进行报文过滤

1.4 配置基本ACL

1.4.1 功能简介

基本 ACL 根据报文的源 IP 地址来制订规则，对报文进行匹配。

1.4.2 配置限制和指导

当 ACL 规则中配置了 **logging** 参数，且引用该 ACL 的模块支持并开启了日志记录功能时，logging 功能生成的日志信息不会输出到控制台和监视终端。此时如需获取该日志，可通过执行 **display logbuffer** 命令进行查看。有关 **display logbuffer** 命令的详细介绍，请参见“网络管理和监控命令参考”中的“信息中心”。

1.4.3 配置 IPv4 基本 ACL

(1) 进入系统视图。

system-view

(2) 创建 IPv4 基本 ACL。请至少选择其中一项进行配置。

- 通过编号创建 IPv4 基本 ACL。

  acl number *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]

- 通过关键字创建 IPv4 基本 ACL。

  acl basic { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]

(3) （可选）配置 ACL 的描述信息。

description *text*

缺省情况下，未配置 ACL 的描述信息。

(4) （可选）配置规则编号的步长。

step *step-value*

缺省情况下，规则编号的步长为 5，起始值为 0。

(5) 创建规则。

rule [*rule-id*] { **deny** | **permit** } [**counting** | **fragment** | **logging** | **source** { *source-address source-wildcard* | **any** } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *

logging 参数是否生效取决于引用该 ACL 的模块是否支持日志记录功能，例如报文过滤支持日志记录功能，如果其引用的 ACL 规则中配置了 **logging** 参数，该参数可以生效。

(6) （可选）为规则配置描述信息。

rule *rule-id* **comment** *text*

缺省情况下，未配置规则的描述信息。

1.4.4 配置 IPv6 基本 ACL

(1) 进入系统视图。

system-view

(2) 创建 IPv6 基本 ACL。请至少选择其中一项进行配置。

- 通过编号创建 IPv6 基本 ACL。

  acl ipv6 number *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]

- 通过关键字创建 IPv6 基本 ACL。

  acl ipv6 basic { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]

(3) （可选）配置 ACL 的描述信息。

description *text*

缺省情况下，未配置 ACL 的描述信息。

(4) （可选）配置规则编号的步长。

step *step-value*

缺省情况下，规则编号的步长为 5，起始值为 0。

(5) 创建规则。

rule [*rule-id*] { **deny** | **permit** } [**counting** | **fragment** | **logging** | **routing** [**type** *routing-type*] | **source** { *source-address source-prefix* | *source-address*/*source-prefix* | **any** } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *

logging 参数是否生效取决于引用该 ACL 的模块是否支持日志记录功能，例如报文过滤支持日志记录功能，如果其引用的 ACL 规则中配置了 **logging** 参数，该参数可以生效。

(6) （可选）为规则配置描述信息。

rule *rule-id* **comment** *text*

缺省情况下，未配置规则的描述信息。

1.5 配置高级ACL

1.5.1 功能简介

高级 ACL 可根据报文的源地址、目的地址、报文优先级、QoS 本地值、承载的协议类型及特性（如 TCP/UDP 的源端口和目的端口、TCP 报文标识、ICMP 或 ICMPv6 协议的消息类型和消息码等），对报文进行匹配。用户可利用高级 ACL 制订比基本 ACL 更准确、丰富、灵活的规则。

1.5.2 配置限制和指导

当 ACL 规则中配置了 **logging** 参数，且引用该 ACL 的模块支持并开启了日志记录功能时，logging 功能生成的日志信息不会输出到控制台和监视终端。此时如需获取该日志，可通过执行 **display logbuffer** 命令进行查看。有关 **display logbuffer** 命令的详细介绍，请参见“网络管理和监控命令参考”中的“信息中心”。

1.5.3 配置 IPv4 高级 ACL

(1) 进入系统视图。

system-view

(2) 创建 IPv4 高级 ACL。请至少选择其中一项进行配置。

- 通过编号创建 IPv4 高级 ACL。

  acl number *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]

- 通过关键字创建 IPv4 高级 ACL。

  acl advanced { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]

(3) （可选）配置 ACL 的描述信息。

description *text*

缺省情况下，未配置 ACL 的描述信息。

(4) （可选）配置规则编号的步长。

step *step-value*

缺省情况下，规则编号的步长为 5，起始值为 0。

(5) 创建规则。

rule [*rule-id*] { **deny** | **permit** } *protocol* [{ { **ack** *ack-value* | **fin** *fin-value* | **psh** *psh-value* | **rst** *rst-value* | **syn** *syn-value* | **urg** *urg-value* } * | **established** } | **counting** | **destination** { *dest-address dest-wildcard* | **any** } | **destination-port** *operator port1* [*port2*] | { **dscp** *dscp* | { **precedence** *precedence* | **tos** *tos* } * } | **fragment** | **icmp-type** { *icmp-type* [*icmp-code*] | *icmp-message* } | **logging** | **source** { *source-address source-wildcard* | **any** } | **source-port** | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *

logging 参数是否生效取决于引用该 ACL 的模块是否支持日志记录功能，例如报文过滤支持日志记录功能，如果其引用的 ACL 规则中配置了 **logging** 参数，该参数可以生效。

(6) （可选）为规则配置描述信息。

rule *rule-id* **comment** *text*

缺省情况下，未配置规则的描述信息。

1.5.4 配置 IPv6 高级 ACL

(1) 进入系统视图。

system-view

(2) 创建 IPv6 高级 ACL。请至少选择其中一项进行配置。

- 通过编号创建 IPv6 高级 ACL。

  acl ipv6 number *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]

- 通过关键字创建 IPv6 高级 ACL。

  acl ipv6 advanced { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]

(3) （可选）配置 ACL 的描述信息。

description *text*

缺省情况下，未配置 ACL 的描述信息。

(4) （可选）配置规则编号的步长。

step *step-value*

缺省情况下，规则编号的步长为 5，起始值为 0。

(5) 创建规则。

rule [*rule-id*] { **deny** | **permit** } *protocol* [{ { **ack** *ack-value* | **fin** *fin-value* | **psh** *psh-value* | **rst** *rst-value* | **syn** *syn-value* | **urg** *urg-value* } * | **established** } | **counting** | **destination** { *dest-address dest-prefix* | *dest-address/dest-prefix* | **any** } | **destination-port** *operator port1* [*port2*] | **dscp** *dscp* | **flow-label** *flow-label-value* | **fragment** | **icmp6-type** { *icmp6-type icmp6-code* | *icmp6-message* } | **logging** | **routing** [**type** *routing-type*] | **hop-by-hop** [**type** *hop-type*] | **source** { *source-address source-prefix* | *source-address/source-prefix* | **any** } | **source-port** *operator port1* [*port2*] | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *

logging 参数是否生效取决于引用该 ACL 的模块是否支持日志记录功能，例如报文过滤支持日志记录功能，如果其引用的 ACL 规则中配置了 **logging** 参数，该参数可以生效。

(6) （可选）为规则配置描述信息。

rule *rule-id* **comment** *text*

缺省情况下，未配置规则的描述信息。

1.6 配置二层ACL

1. 功能简介

二层 ACL 可根据报文的源 MAC 地址、目的 MAC 地址、802.1p 优先级、链路层协议类型、报文的封装类型等二层信息来制订规则，对报文进行匹配。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 创建二层 ACL。请至少选择其中一项进行配置。

- 通过编号创建二层 ACL。

  acl number *acl-number* [**name** *acl-name*] [**match-order** { **auto** | **config** }]

- 通过关键字创建二层 ACL。

  acl mac { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]

(3) （可选）配置 ACL 的描述信息。

description *text*

缺省情况下，未配置 ACL 的描述信息。

(4) （可选）配置规则编号的步长。

step *step-value*

缺省情况下，规则编号的步长为 5，起始值为 0。

(5) 创建规则。

rule [*rule-id*] { **deny** | **permit** } [**cos** *dot1p* | **counting** | **dest-mac** *dest-address dest-mask* | { **lsap** *lsap-type lsap-type-mask* | **type** *protocol-type protocol-type-mask* } | **source-mac** *source-address source-mask* | **time-range** *time-range-name*] *

(6) （可选）为规则配置描述信息。

rule *rule-id* **comment** *text*

缺省情况下，未配置规则的描述信息。

1.7 复制ACL

1. 功能简介

用户可通过复制一个已存在的 ACL（即源 ACL），来生成一个新的同类型 ACL（即目的 ACL）。除了 ACL 的编号和名称不同外，目的 ACL 与源 ACL 完全相同。

2. 配置限制和指导

目的 ACL 要与源 ACL 的类型相同，且目的 ACL 必须不存在，否则将导致复制 ACL 失败。

3. 配置步骤

(1) 进入系统视图。

system-view

(2) 复制并生成一个新的 ACL。

acl [**ipv6** | **mac**] **copy** { *source-acl-number* | **name** *source-acl-name* } **to** { *dest-acl-number* | **name** *dest-acl-name* }

1.8 配置ACL规则的加速匹配功能

1. 功能简介

在对基于会话的业务报文进行规则匹配时，通常只对首个报文进行匹配以加快报文的处理速度，但这有时并不足以解决报文匹配的效率问题。譬如，当有大量用户同时与设备新建连接时，需要对每个新建连接都进行规则匹配，如果 ACL 内包含有大量规则，那么这个匹配过程将很长，这会导致用户建立连接时间超长，从而影响设备新建连接的性能。

ACL 规则的加速匹配功能则可以解决上述问题，当对包含大量规则的 ACL 开启了加速匹配功能之后，其规则匹配速度将大大提高，从而提升设备的转发性能以及新建连接的性能。

当设备支持软件加速，并开启本功能后，添加、删除和修改规则时，并不会立即加速，而是延迟一定时间后加速。如果在该时间内，规则又发生变化，则重新计时。当 ACL 中的规则小于等于 100 条时，此时间为 2 秒；当 ACL 中的规则大于 100 条时，此时间为 20 秒。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 创建 ACL，并进入 ACL 视图。

acl { [**ipv6**] { **advanced** | **basic** } { *acl-number* | **name** *acl-name* } | **mac** { *acl-number* | **name** *acl-name* } } [**match-order** { **auto** | **config** }]

(3) 配置 ACL 规则的加速匹配功能。

accelerate

缺省情况下，ACL 规则的加速匹配功能处于关闭状态。

1.9 应用ACL进行报文过滤

1.9.1 功能简介

ACL 最基本的应用就是进行报文过滤。例如，将 ACL 规则应用到指定接口的入或出方向上，从而对该接口收到或发出的报文进行过滤。

1.9.2 在接口上应用 ACL 进行报文过滤

1. 配置限制和指导

一个接口在一个方向上最多可应用 32 个 ACL 进行报文过滤。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 在接口上应用 ACL 进行报文过滤。

packet-filter [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } { **inbound** | **outbound** }

缺省情况下，未配置接口的报文过滤。

1.9.3 配置报文过滤日志信息或告警信息的生成与发送周期

1. 功能简介

报文过滤日志或告警信息的生成与发送周期起始于报文过滤中 ACL 匹配数据流的第一个数据包，报文过滤日志或告警信息包括周期内被匹配的报文数量以及所使用的 ACL 规则。在一个周期内：

- 对于规则匹配数据流的第一个数据包，设备会立即生成报文过滤日志或告警信息；
- 对于规则匹配数据流的其他数据包，设备将在周期结束后生成报文过滤日志或告警信息。

设备生成的报文过滤日志将发送给信息中心，有关信息中心的详细介绍，请参见“网络管理和监控配置指导”中的“信息中心”。

设备生成的告警信息将发送给 SNMP，有关 SNMP 的详细介绍，请参见“网络管理和监控配置指导”中的“SNMP”。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 配置报文过滤日志信息或告警信息的生成与发送周期。

acl { **logging** | **trap** } **interval** *interval*

缺省情况下，报文过滤日志信息或告警信息的生成与发送周期为 0 分钟，即不记录报文过滤的日志和告警信息。

1.9.4 配置报文过滤的缺省动作

1. 功能简介

系统缺省的报文过滤动作为 Permit，即允许未匹配上 ACL 规则的报文通过。通过本配置可更改报文过滤的缺省动作为 Deny，即禁止未匹配上 ACL 规则的报文通过。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 配置报文过滤的缺省动作为 Deny。

packet-filter default deny

缺省情况下，报文过滤的缺省动作为 Permit，即允许未匹配上 ACL 规则的报文通过。

1.10 ACL显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示 ACL 配置后的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除 ACL 的统计信息。

表1-3 ACL 显示和维护

配置	命令
显示ACL的配置和运行情况	**display acl** [**ipv6** | **mac**] { *acl-number* | **all** | **name** *acl-name* }
显示ACL的加速状态	（独立运行模式） **display acl accelerate** { **summary** [**ipv6** | **mac**] | **verbose** [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } } （IRF模式） **display acl accelerate** { **summary** [**ipv6** | **mac**] | **verbose** [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } **slot** *slot-number* }
显示ACL在报文过滤中的应用情况	（独立运行模式） **display packet-filter interface** [*interface-type interface-number*] [**inbound** | **outbound**] （IRF模式） **display packet-filter interface** [*interface-type interface-number*] [**inbound** | **outbound**] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示ACL在报文过滤中应用的统计信息	**display packet-filter statistics interface** *interface-type interface-number* { **inbound** | **outbound** } [**default** | [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] [**brief**]
显示ACL在报文过滤中应用的累加统计信息	**display packet-filter statistics sum** { **inbound** | **outbound** } [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } [**brief**]
显示ACL在报文过滤中的详细应用情况	（独立运行模式） **display packet-filter verbose interface** *interface-type interface-number* { **inbound** | **outbound** } [[**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] （IRF模式） **display packet-filter verbose interface** *interface-type interface-number* { **inbound** | **outbound** } [[**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] [**slot** *slot-number* [**cpu** *cpu-number*]]
清除ACL的统计信息	**reset acl** [**ipv6** | **mac**] **counter** { *acl-number* | **all** | **name** *acl-name* }
清除ACL在报文过滤中应用的统计信息	**reset packet-filter statistics interface** [*interface-type interface-number*] { **inbound** | **outbound** } [**default** | [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }

1.11 ACL典型配置举例

1.11.1 在接口上应用包过滤的 ACL 配置举例

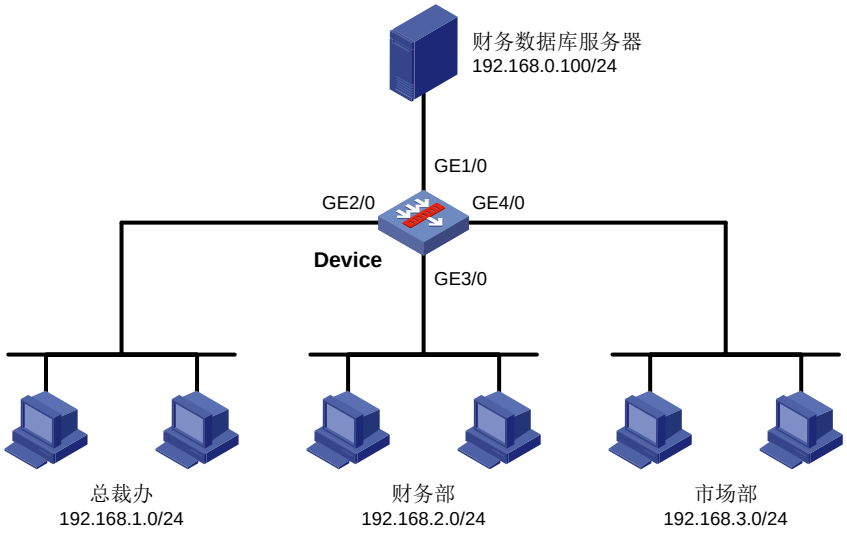
1. 组网需求

- 某公司内的各部门之间通过 Device 实现互连，该公司的工作时间为每周工作日的 8 点到 18 点。

- 通过配置，允许总裁办在任意时间、财务部在工作时间访问财务数据库服务器，禁止其它部门在任何时间、财务部在非工作时间访问该服务器。

2. 组网图

图1-1 ACL 典型配置组网图



3. 配置步骤

创建名为 work 的时间段，其时间范围为每周工作日的 8 点到 18 点。

```
<Device> system-view
[Device] time-range work 08:00 to 18:00 working-day
```

创建 IPv4 高级 ACL 3000，并制订如下规则：允许总裁办在任意时间、财务部在工作时间访问财务数据库服务器，禁止其它部门在任何时间、财务部在非工作时间访问该服务器。

```
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit ip source 192.168.1.0 0.0.0.255 destination
192.168.0.100 0
[Device-acl-ipv4-adv-3000] rule permit ip source 192.168.2.0 0.0.0.255 destination
192.168.0.100 0 time-range work
[Device-acl-ipv4-adv-3000] rule deny ip source any destination 192.168.0.100 0
[Device-acl-ipv4-adv-3000] quit
```

应用 IPv4 高级 ACL 3000 对接口 GigabitEthernet1/0 出方向上的报文进行过滤。

```
[Device] interface gigabitethernet 1/0
[Device-GigabitEthernet1/0] packet-filter 3000 outbound
[Device-GigabitEthernet1/0] quit
```

4. 验证配置

配置完成后，在各部门的 PC（假设均为 Windows XP 操作系统）上可以使用 **ping** 命令检验配置效果，在 Device 上可以使用 **display acl** 命令查看 ACL 的配置和运行情况。例如在工作时间：

在财务部的 PC 上检查到财务数据库服务器是否可达。

```
C:\> ping 192.168.0.100

Pinging 192.168.0.100 with 32 bytes of data:
```

```
Reply from 192.168.0.100: bytes=32 time=1ms TTL=255
Reply from 192.168.0.100: bytes=32 time<1ms TTL=255
Reply from 192.168.0.100: bytes=32 time<1ms TTL=255
Reply from 192.168.0.100: bytes=32 time<1ms TTL=255

Ping statistics for 192.168.0.100:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 1ms, Average = 0ms
```

由此可见，财务部的 PC 能够在工作时间访问财务数据库服务器。

在市场部的 PC 上检查财务数据库服务器是否可达。

```
C:\> ping 192.168.0.100

Pinging 192.168.0.100 with 32 bytes of data:

Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.0.100:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

由此可见，市场部的 PC 不能在工作时间访问财务数据库服务器。

查看 IPv4 高级 ACL 3000 的配置和运行情况。

```
[Device] display acl 3000
Advanced IPv4 ACL 3000, 3 rules,
ACL's step is 5
 rule 0 permit ip source 192.168.1.0 0.0.0.255 destination 192.168.0.100 0
 rule 5 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.0.100 0 time-range work
(4 times matched) (Active)
 rule 10 deny ip destination 192.168.0.100 0 (4 times matched)
```

由此可见，由于目前是工作时间，因此规则 5 是生效的；且由于之前使用了 **ping** 命令的缘故，规则 5 和规则 10 分别被匹配了 4 次。

目 录

1 QoS 概述

QoS 即服务质量。对于网络业务，影响服务质量的因素包括传输的带宽、传送的时延、数据的丢包率等。在网络中可以通过保证传输的带宽、降低传送的时延、降低数据的丢包率以及时延抖动等措施来提高服务质量。网络资源总是有限的，在保证某类业务的服务质量的同时，可能就是在损害其它业务的服务质量。因此，网络管理者需要根据各种业务的特点来对网络资源进行合理的规划和分配，从而使网络资源得到高效利用。

1.1 QoS服务模型简介

通常 QoS 提供以下三种服务模型：

- Best-Effort service（尽力而为服务模型）
- Integrated service（综合服务模型，简称 IntServ）
- Differentiated service（区分服务模型，简称 DiffServ）

1.1.1 Best-Effort 服务模型

Best-Effort 是一个单一的服务模型，也是最简单的服务模型。对 Best-Effort 服务模型，网络尽最大的可能性来发送报文。但对时延、可靠性等性能不提供任何保证。

Best-Effort 服务模型是网络的缺省服务模型，通过 FIFO 队列来实现。它适用于绝大多数网络应用，如 FTP、E-Mail 等。

1.1.2 IntServ 服务模型

IntServ 是一个综合服务模型，它可以满足多种 QoS 需求。该模型使用 RSVP 协议，RSVP 运行在从源端到目的端的每个设备上，可以监视每个流，以防止其消耗资源过多。这种体系能够明确区分并保证每一个业务流的服务质量，为网络提供最细粒度化的服务质量区分。

但是，IntServ 模型对设备的要求很高，当网络中的数据流数量很大时，设备的存储和处理能力会遇到很大的压力。IntServ 模型可扩展性很差，难以在 Internet 核心网络实施。

1.1.3 DiffServ 服务模型

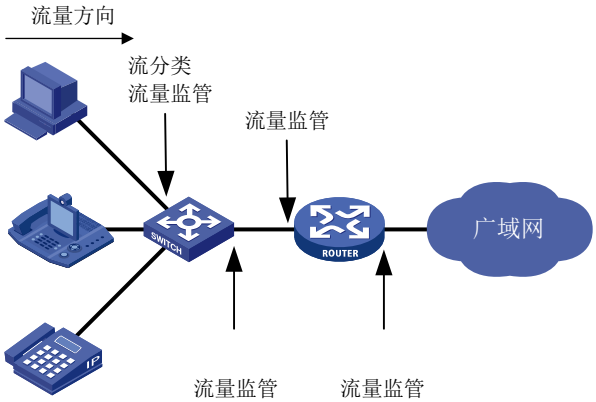
DiffServ 是一个多服务模型，它可以满足不同的 QoS 需求。与 IntServ 不同，它不需要通知网络为每个业务预留资源。区分服务实现简单，扩展性较好。

本文提到的技术都是基于 DiffServ 服务模型。

1.2 QoS技术在网络中的位置

QoS 技术包括流分类、流量监管等。下面对常用的技术进行简单地介绍。

图1-1 常用 QoS 技术在网络中的位置



如图 1-1 所示，流分类和流量监管主要完成如下功能：

- 流分类：采用一定的规则识别符合某类特征的报文，它是对网络业务进行区分服务的前提和基础。
- 流量监管：对进入或流出设备的特定流量进行监管，以保护网络资源不受损害。可以作用在接口入方向和出方向。

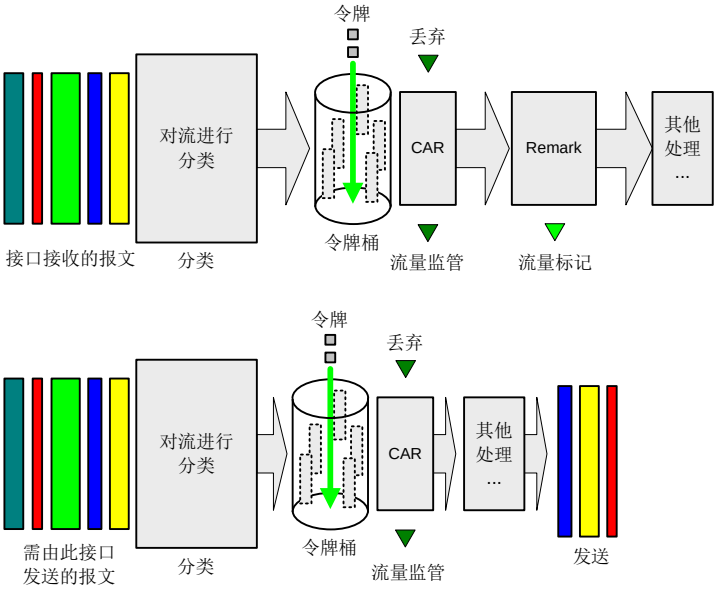
1.3 QoS技术在设备中的处理顺序

图 1-2 简要描述了各种 QoS 技术在网络设备中的处理顺序。

(1) 首先通过流分类对各种业务进行识别和区分，它是后续各种动作的基础；

(2) 通过各种动作对特定的业务进行处理。这些动作需要和流分类关联起来才有意义。具体采取何种动作，与所处的阶段以及网络当前的负载状况有关。例如，当报文进入网络时进行流量监管等。

图1-2 各 QoS 技术在同一网络设备中的处理顺序



1.4 QoS配置方式

QoS 的配置方式分为 MQC 方式（模块化 QoS 配置，Modular QoS Configuration）和非 MQC 方式。

MQC 方式通过 QoS 策略定义不同类别的流量要采取的动作，并将 QoS 策略应用到不同的目标位置（例如接口）来实现对业务流量的控制。

非 MQC 方式则通过直接在目标位置上配置 QoS 参数来实现对业务流量的控制。

有些 QoS 功能只能使用其中一种方式来配置，有些使用两种方式都可以进行配置。在实际应用中，两种配置方式也可以结合起来使用。

2 QoS 策略

2.1 QoS策略简介

QoS 策略由如下部分组成：

- 类，定义了对报文进行识别的规则。
- 流行为，定义了一组针对类识别后的报文所做的 QoS 动作。

通过将类和流行为关联起来，QoS 策略可对符合分类规则的报文执行流行为中定义的动作。

用户可以在一个策略中定义多个类与流行为的绑定关系。

2.2 QoS策略配置任务简介

QoS 策略配置任务如下：

(1) 定义类

(2) 定义流行为

(3) 定义策略

(4) （可选）配置策略嵌套

(5) 应用策略

- 基于接口应用 QoS 策略
- 基于控制平面应用 QoS 策略
- 基于管理口控制平面应用 QoS 策略

(6) （可选）配置接口流速统计时间

2.3 定义类

(1) 进入系统视图。

system-view

(2) 创建类，并进入类视图。

traffic classifier *classifier-name* [**operator** { **and** | **or** }]

(3) 定义匹配数据包的规则。

if-match [**not**] *match-criteria*

缺省情况下，未定义匹配数据包的规则。

具体规则的介绍，请参见“QoS 命令”中的 **if-match** 命令。

2.4 定义流行为

(1) 进入系统视图。

system-view

(2) 创建流行为，并进入流行为视图。

traffic behavior *behavior-name*

(3) 配置流行为的动作。

缺省情况下，未配置流行为的动作。

流行为动作就是对符合流分类的报文做出相应的 QoS 动作，例如流量监管、流量过滤、重标记、流量统计等，具体情况请参见本文相关章节。

2.5 定义策略

(1) 进入系统视图。

system-view

(2) 创建 QoS 策略，并进入策略视图。

qos policy *policy-name*

(3) 为类指定流行为。

classifier *classifier-name* **behavior** *behavior-name* [**insert-before** *before-classifier-name*]

缺省情况下，未指定类对应的流行为。

2.6 配置策略嵌套

1. 功能简介

QoS 策略分为两种：父策略和子策略，其中父策略即为普通的 QoS 策略。通过在父策略流行为视图下创建一个新的策略，即创建子策略，可以实现策略嵌套功能。

配置策略嵌套后，**traffic classifier** 命令定义的某一类流量，除了执行父策略中定义的流行为外，还会由子策略再次对该类流量进行分类，并执行子策略中定义的流行为。

2. 配置准备

配置策略嵌套时，请先定义子策略。关于定义子策略配置，请参见“2.5 定义策略”。

3. 配置步骤

(1) 进入系统视图。

system-view

(2) 定义父策略的类。

a. 创建父策略的类，并进入父策略的类视图。

traffic classifier *classifier-name* [**operator** { **and** | **or** }]

b. 定义父策略匹配数据包的规则。

if-match [**not**] *match-criteria*

缺省情况下，未定义匹配数据包的规则。

具体规则的配置介绍，请参见 QoS 命令中的 **if-match** 命令。

c. 退回系统视图。

quit

(3) 在父策略流行为中嵌套子策略。

a. 创建父策略流行为，并进入父策略的流行为视图。

traffic behavior *behavior-name*

b. 指定子策略，配置策略嵌套。

traffic-policy *policy-name*

c. 退出流行为视图。

quit

(4) 创建父策略，并进入父策略视图。

qos policy *policy-name*

(5) 在父策略中为类指定采用的流行为。

classifier *classifier-name* **behavior** *behavior-name*

缺省情况下，没有为类指定流行为。

2.7 应用策略

2.7.1 设备支持的策略应用位置

QoS 策略支持应用在如下位置：

- 基于接口应用 QoS 策略，QoS 策略对通过接口接收或发送的流量生效。
- 基于控制平面应用 QoS 策略，QoS 策略对通过控制平面接收的流量生效。
- 基于管理口控制平面应用 QoS 策略，QoS 策略对通过管理口接收的流量生效。

2.7.2 策略应用限制和指导

QoS 策略应用后，用户仍然可以修改 QoS 策略中的流分类规则和流行为，以及二者的对应关系。当流分类规则中使用 ACL 匹配报文时，允许删除或修改该 ACL（包括向该 ACL 中添加、删除和修改匹配规则）。

2.7.3 基于接口应用 QoS 策略

1. 配置限制和指导

基于接口应用 QoS 策略时需要注意的是：

- 一个 QoS 策略可以应用于多个接口，但在接口的每个方向（出和入两个方向）只能应用一个策略。
- QoS 策略应用在出方向时，对设备发出的协议报文不起作用，以确保这些报文在策略误配置时仍然能够正常发出，维持设备的正常运行。常见的本地协议报文如下：链路维护报文、RIP、LDP、SSH 等。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 在接口上应用已创建的 QoS 策略。

qos apply policy *policy-name* { **inbound** | **outbound** }

缺省情况下，未在接口上应用 QoS 策略。

2.7.4 基于控制平面应用 QoS 策略

1. 功能简介

设备上存在用户平面和控制平面：

- 用户平面（User Plane）：是指对报文进行收发、交换的处理单元，它的主要工作是转发报文。在设备上，与之相对应的核心物理实体就是各种专用转发芯片，它们有极高的处理速度和很强的数据吞吐能力。
- 控制平面（Control Plane）：是指运行大部分路由交换协议进程的处理单元，它的主要工作是进行协议报文的解析和协议的计算。在设备上，与之相对应的核心物理实体就是 CPU，它具备灵活的报文处理能力，但数据吞吐能力有限。

用户平面接收到无法识别或处理的报文会送到控制平面进行进一步处理。如果上送控制平面的报文速率超过了控制平面的处理能力，那么上送控制平面的报文会得不到正确转发或及时处理，从而影响协议的正常运行。

为了解决此问题，用户可以把 QoS 策略应用在控制平面上，通过对上送控制平面的报文进行过滤、限速等 QoS 处理，达到保护控制平面正常报文的收发、维护控制平面正常处理状态的目的。

预定义的 QoS 策略中通过协议类型或者协议组类型来标识各种上送控制平面的报文类型，用户也可以在流分类视图下通过 **if-match** 命令引用这些协议类型或者协议组类型来进行报文分类，然后根据需要为这些报文重新配置流行为。系统预定义的 QoS 策略信息可以通过 **display qos policy control-plane pre-defined** 命令查看。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 进入控制平面视图。

（独立运行模式）

control-plane

（IRF 模式）

control-plane slot *slot-number*

(3) 在控制平面上应用已创建的 QoS 策略。

qos apply policy *policy-name* **inbound**

缺省情况下，未在控制平面上应用 QoS 策略。

2.7.5 基于管理口控制平面应用 QoS 策略

1. 功能简介

管理口控制平面仅针对管理口上送给控制平面的报文。

如果管理口上送给控制平面的报文速率超过其处理能力，报文会得不到正确转发或及时处理，从而影响协议的正常运行。

为了解决此问题，用户可以把 QoS 策略应用在管理口控制平面上，通过对管理口上送给控制平面的报文进行 QoS 限速处理，达到保护管理口正常报文的收发、维护管理口正常处理状态的目的。

预定义的 QoS 策略中通过协议类型或者协议组类型来标识各种上送管理口控制平面的报文类型，用户也可以在流分类视图下通过 **if-match** 命令引用这些协议类型或者协议组类型来进行报文分类，然后根据需要为这些报文重新配置流行为。系统预定义的 QoS 策略信息可以通过 **display qos policy control-plane management pre-defined** 命令查看。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 进入管理口控制平面视图。

control-plane management

(3) 在管理口控制平面上应用已创建的 QoS 策略。

qos apply policy *policy-name* **inbound**

缺省情况下，未在管理口控制平面上应用 QoS 策略。

2.8 配置接口流速统计时间

1. 功能简介

通过配置接口流速统计时间，我们可以统计经过 QoS 策略流分类后每类报文的发送和丢弃速率。假设流速统计时间为 t（t 默认为 5 分钟），则系统将统计最近 t 时间内每类报文发送和丢弃的平均速率，且每 t/5 分钟刷新一次统计速率。流速统计的结果可以通过命令 **display qos policy interface** 查看。

2. 配置限制和指导

配置接口流速统计时间时需要注意的是：子接口的流速统计时间采用主接口上设置的统计时间。

3. 配置步骤

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 配置接口流速统计时间。

qos flow-interval *interval*

缺省情况下，接口流速统计时间为 5 分钟。

2.9 QoS策略显示和维护

在任意视图下执行 **display** 命令可以显示 QoS 策略的运行情况，通过查看显示信息验证配置的效果。

在用户视图下执行 **reset** 命令可以清除 QoS 策略的统计信息。

表2-1 QoS 策略显示和维护

操作	命令
显示QoS策略的配置信息	（独立运行模式） **display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] （IRF模式） **display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] [**slot** *slot-number*]
显示Tunnel接口Hub-Spoke隧道应用QoS策略的配置信息和运行情况	**display qos policy advpn tunnel** *number* [*ipv4-address* | *ipv6-address*] [**outbound**]
显示基于控制平面应用QoS策略的信息	（独立运行模式） **display qos policy control-plane** （IRF模式） **display qos policy control-plane slot** *slot-number*
显示管理口控制平面应用的QoS策略信息	**display qos policy control-plane management**
显示系统预定义的管理口控制平面应用QoS策略的信息	**display qos policy control-plane management pre-defined**
显示系统预定义的控制平面应用QoS策略的信息	（独立运行模式） **display qos policy control-plane pre-defined** （IRF模式） **display qos policy control-plane pre-defined** [**slot** *slot-number*]
显示接口上QoS策略的配置信息和运行情况	（独立运行模式） **display qos policy interface** [*interface-type interface-number*] [**inbound** | **outbound**] （IRF模式） **display qos policy interface** [*interface-type interface-number*] [**slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**]
显示流行为的配置信息	（独立运行模式） **display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] （IRF模式） **display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**slot** *slot-number*]
显示类的配置信息	（独立运行模式） **display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] （IRF模式） **display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] [**slot** *slot-number*]

操作	命令
清除控制平面应用QoS策略的统计信息	（独立运行模式） **reset qos policy control-plane** （IRF模式） **reset qos policy control-plane slot** *slot-number*
清除Tunnel接口Hub-Spoke隧道应用QoS策略的统计信息	**reset qos policy advpn tunnel** *number* [*ipv4-address* | *ipv6-address*] [**outbound**]
清除管理口控制平面应用QoS策略的统计信息	**reset qos policy control-plane management**

3 流量监管

3.1 流量监管简介

如果不限制用户发送的流量，那么大量用户不断突发的数据只会使网络更拥挤。为了使有限的网络资源能够更好地发挥效用，更好地为更多的用户服务，必须对用户的流量加以限制。流量监管可以实现流量的速率限制功能，而要实现此功能就必须对通过设备的流量进行度量。一般采用令牌桶（Token Bucket）对流量进行度量。

3.1.1 流量评估与令牌桶

1. 令牌桶

令牌桶可以看作是一个存放一定数量令牌的容器。系统按设定的速度向桶中放置令牌，当桶中令牌满时，多出的令牌溢出，桶中令牌不再增加。

2. 用令牌桶评估流量

在用令牌桶评估流量规格时，是以令牌桶中的令牌数量是否足够满足报文的转发为依据的。如果桶中存在足够的令牌可以用来转发报文，称流量遵守或符合这个规格，否则称为不符合或超标。

评估流量时令牌桶的参数包括：

- 平均速率：向桶中放置令牌的速率，即允许的流的平均速度。通常配置为 CIR。
- 突发尺寸：令牌桶的容量，即每次突发所允许的最大的流量尺寸。通常配置为 CBS，突发尺寸必须大于最大报文长度。

每到达一个报文就进行一次评估。每次评估，如果桶中有足够的令牌可供使用，则说明流量控制在允许的范围内，此时要从桶中取走满足报文的转发的令牌；否则说明已经耗费太多令牌，流量超标了。

3. 复杂评估

为了评估更复杂的情况，实施更灵活的调控策略，可以使用两个令牌桶（分别称为 C 桶和 E 桶）对流量进行评估。主要有如下三种算法。

(1) 单速率单桶双色算法

- CIR：表示向 C 桶中投放令牌的速率，即 C 桶允许传输或转发报文的平均速率；
- CBS：表示 C 桶的容量，即 C 桶瞬间能够通过的承诺突发流量。

每次评估时，依据下面的情况，可以分别实施不同的流控策略：

- 如果 C 桶有足够的令牌，报文被标记为 green，即绿色报文；
- 如果 C 桶令牌不足，报文被标记为 red，即红色报文。

(2) 单速率双桶三色算法

- CIR：表示向 C 桶中投放令牌的速率，即 C 桶允许传输或转发报文的平均速率；
- CBS：表示 C 桶的容量，即 C 桶瞬间能够通过的承诺突发流量；
- EBS：表示 E 桶的容量的增量，即 E 桶瞬间能够通过的超出突发流量，取值不为 0。E 桶的容量等于 CBS 与 EBS 的和。

每次评估时，依据下面的情况，可以分别实施不同的流控策略：

- 如果 C 桶有足够的令牌，报文被标记为 green，即绿色报文；
- 如果 C 桶令牌不足，但 E 桶有足够的令牌，报文被标记为 yellow，即黄色报文；
- 如果 C 桶和 E 桶都没有足够的令牌，报文被标记为 red，即红色报文。

(3) 双速率双桶三色算法

- CIR：表示向 C 桶中投放令牌的速率，即 C 桶允许传输或转发报文的平均速率；
- CBS：表示 C 桶的容量，即 C 桶瞬间能够通过的承诺突发流量；
- PIR：表示向 E 桶中投放令牌的速率，即 E 桶允许传输或转发报文的最大速率；
- EBS：表示 E 桶的容量，即 E 桶瞬间能够通过的超出突发流量。

每次评估时，依据下面的情况，可以分别实施不同的流控策略：

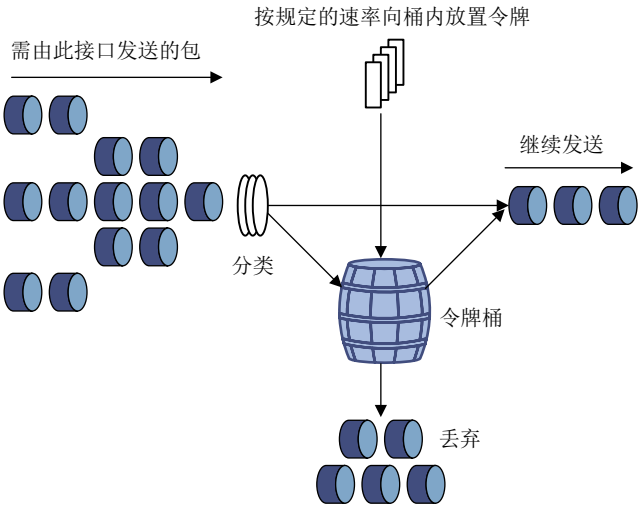
- 如果 C 桶有足够的令牌，报文被标记为 green，即绿色报文；
- 如果 C 桶令牌不足，但 E 桶有足够的令牌，报文被标记为 yellow，即黄色报文；
- 如果 C 桶和 E 桶都没有足够的令牌，报文被标记为 red，即红色报文。

3.1.2 流量监管

流量监管分为入和出两个方向，为了方便描述，下文以出方向为例。

流量监管就是对流量进行控制，通过监督进入网络的流量速率，对超出部分的流量进行“惩罚”，使进入的流量被限制在一个合理的范围之内，以保护网络资源和运营商的利益。例如可以限制 HTTP 报文不能占用超过 50%的网络带宽。如果发现某个连接的流量超标，流量监管可以选择丢弃报文，或重新配置报文的优先级。

图3-1 TP 示意图



流量监管广泛的用于监管进入 Internet 服务提供商 ISP 的网络流量。流量监管还包括对所监管流量的流分类服务，并依据不同的评估结果，实施预先设定好的监管动作。这些动作可以是：

- 转发：比如对评估结果为“符合”的报文继续转发。
- 丢弃：比如对评估结果为“不符合”的报文进行丢弃。

- 改变优先级并转发：比如对评估结果为“符合”的报文，将其优先级进行重标记后再进行转发。
- 改变优先级并进入下一级监管：比如对评估结果为“符合”的报文，将其优先级进行重标记后再进入下一级的监管。
- 进入下一级的监管：流量监管可以进行分级，每级关注和监管更具体的目标。

3.2 配置流量监管

3.2.1 流量监管配置方式介绍

可以通过MQC方式和非MQC方式配置流量监管，其中非MQC方式配置流量监管时分为以下几种：

- 基于CAR列表的流量监管配置。
- 基于ACL的流量监管配置。
- 适配所有流的流量监管配置。
- 基于上线用户的流量监管配置。
- 基于家庭用户的流量监管配置。

如果接口上同时采用了MQC方式和非MQC方式配置了流量监管，那么只有前者会生效。

3.2.2 配置流量监管（MQC方式）

1. 配置限制和指导

设备支持基于接口、控制平面和管理口控制平面应用QoS策略配置流量监管。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 定义类。

a. 创建类，并进入类视图。

traffic classifier *classifier-name* [**operator** { **and** | **or** }]

b. 定义匹配数据包的规则。

if-match [**not**] *match-criteria*

缺省情况下，未定义匹配数据包的规则。

具体规则的介绍，请参见“QoS命令”中的**if-match**命令。

c. 退回系统视图。

quit

(3) 定义流行为。

a. 创建一个流行为并进入流行为视图。

traffic behavior *behavior-name*

b. 配置流量监管动作。

（绝对值配置方式）

car cir *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action* | **red** *action* | **yellow** *action*] *

car cir *committed-information-rate* [**cbs** *committed-burst-size*] **pir** *peak-information-rate* [**ebs** *excess-burst-size*] [**green** *action* | **red** *action* | **yellow** *action*] *

（百分比配置方式）

car cir percent *cir-percent* [**cbs** *cbs-time* [**ebs** *ebs-time*]] [**green** *action* | **red** *action* | **yellow** *action*] *

car cir percent *cir-percent* [**cbs** *cbs-time*] **pir percent** *pir-percent* [**ebs** *ebs-time*] [**green** *action* | **red** *action* | **yellow** *action*] *

缺省情况下，未配置流量监管动作。

c. 退回系统视图。

quit

(4) 定义策略。

a. 创建策略并进入策略视图。

qos policy *policy-name*

b. 在策略中为类指定采用的流行为。

classifier *classifier-name* **behavior** *behavior-name*

缺省情况下，未指定类对应的流行为。

c. 退回系统视图。

quit

(5) 应用 QoS 策略。

具体配置请参见“2.7 应用策略”

缺省情况下，未应用 QoS 策略。

3.2.3 配置基于 CAR 列表的流量监管

(1) 进入系统视图。

system-view

(2) 创建 CAR 列表并配置匹配规则。

qos carl *carl-index* { **dscp** *dscp-list* | **mac** *mac-address* | **mpls-exp** *mpls-exp-value* | **precedence** *precedence-value* | { **destination-ip-address** | **source-ip-address** } { **range** *start-ip-address* **to** *end-ip-address* | **subnet** *ip-address mask-length* } [**per-address** [**shared-bandwidth**]] }

(3) 进入接口视图。

interface *interface-type interface-number*

(4) 在接口上配置基于 CAR 列表的 CAR 策略。

（绝对值配置方式）

qos car { **inbound** | **outbound** } **carl** *carl-index* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action* | **red** *action* | **yellow** *action*] *

qos car { **inbound** | **outbound** } **carl** *carl-index* **cir** *committed-information-rate* [**cbs** *committed-burst-size*] **pir** *peak-information-rate* [**ebs** *excess-burst-size*] [**green** *action* | **red** *action* | **yellow** *action*] *

（百分比配置方式）

qos car { **inbound** | **outbound** } **carl** *carl-index* **percent cir** *cir-percent* [**cbs** *cbs-time* [**ebs** *ebs-time*]] [**green** *action* | **red** *action* | **yellow** *action*] *

qos car { **inbound** | **outbound** } **carl** *carl-index* **percent cir** *cir-percent* [**cbs** *cbs-time*] **pir** *pir-percent* [**ebs** *ebs-time*] [**green** *action* | **red** *action* | **yellow** *action*] *

缺省情况下，接口上未应用 CAR 策略。

3.2.4 配置基于 ACL 的流量监管

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 在接口上配置基于 ACL 规则的 CAR 策略。

（绝对值配置方式）

qos car { **inbound** | **outbound** } **acl** [**ipv6**] *acl-number* **cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action* | **red** *action* | **yellow** *action*] *

qos car { **inbound** | **outbound** } **acl** [**ipv6**] *acl-number* **cir** *committed-information-rate* [**cbs** *committed-burst-size*] **pir** *peak-information-rate* [**ebs** *excess-burst-size*] [**green** *action* | **red** *action* | **yellow** *action*] *

（百分比配置方式）

qos car { **inbound** | **outbound** } **acl** [**ipv6**] *acl-number* **percent cir** *cir-percent* [**cbs** *cbs-time* [**ebs** *ebs-time*]] [**green** *action* | **red** *action* | **yellow** *action*] *

qos car { **inbound** | **outbound** } **acl** [**ipv6**] *acl-number* **percent cir** *cir-percent* [**cbs** *cbs-time*] **pir** *pir-percent* [**ebs** *ebs-time*] [**green** *action* | **red** *action* | **yellow** *action*] *

缺省情况下，接口上未应用 CAR 策略。

3.2.5 配置适配所有流的流量监管

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 在接口应用 CAR 策略。

（绝对值配置方式）

qos car { **inbound** | **outbound** } **any cir** *committed-information-rate* [**cbs** *committed-burst-size* [**ebs** *excess-burst-size*]] [**green** *action* | **red** *action* | **yellow** *action*] *

qos car { **inbound** | **outbound** } **any cir** *committed-information-rate* [**cbs** *committed-burst-size*] **pir** *peak-information-rate* [**ebs** *excess-burst-size*] [**green** *action* | **red** *action* | **yellow** *action*] *

（百分比配置方式）

qos car { **inbound** | **outbound** } **any percent cir** *cir-percent* [**cbs** *cbs-time* [**ebs** *ebs-time*]] [**green** *action* | **red** *action* | **yellow** *action*] *

qos car { **inbound** | **outbound** } **any percent cir** *cir-percent* [**cbs** *cbs-time*] **pir** *pir-percent* [**ebs** *ebs-time*] [**green** *action* | **red** *action* | **yellow** *action*] *

缺省情况下，接口上没有应用 CAR 策略。

3.3 流量监管显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示配置后流量监管运行情况，通过查看显示信息验证配置的效果。

表3-1 流量监管显示和维护

操作	命令
显示接口的流量监管配置情况和统计信息	**display qos car interface** [*interface-type interface-number*]
显示CAR列表	（独立运行模式） **display qos carl** [*carl-index*] （IRF模式） **display qos carl** [*carl-index*] [**slot** *slot-number*]
显示QoS和ACL资源的使用情况（本命令的详细介绍，请参见“ACL和QoS命令参考”中的“ACL”）	（独立运行模式） **display qos-acl resource** （IRF模式） **display qos-acl resource** [**slot** *slot-number*]

操作	命令
显示流量监管的相关配置信息	（独立运行模式） **display traffic behavior user-defined** [*behavior-name*] （IRF模式） **display traffic behavior user-defined** [*behavior-name*] [**slot** *slot-number*]

3.4 流量监管典型配置举例

3.4.1 流量监管典型配置举例

1. 配置需求

- 设备 Device A 通过接口 GigabitEthernet3/0 和设备 Device B 的接口 GigabitEthernet1/0 互连
- Server、Host A、Host B 可经由 Device A 和 Device B 访问 Internet
- Server、Host A 与 Device A 的 GigabitEthernet1/0 接口在同一网段
- Host B 与 Device A 的 GigabitEthernet2/0 接口在同一网段

要求在设备 Device A 上对接口 GigabitEthernet1/0 接收到的源自 Server 和 Host A 的报文流分别实施流量控制如下：

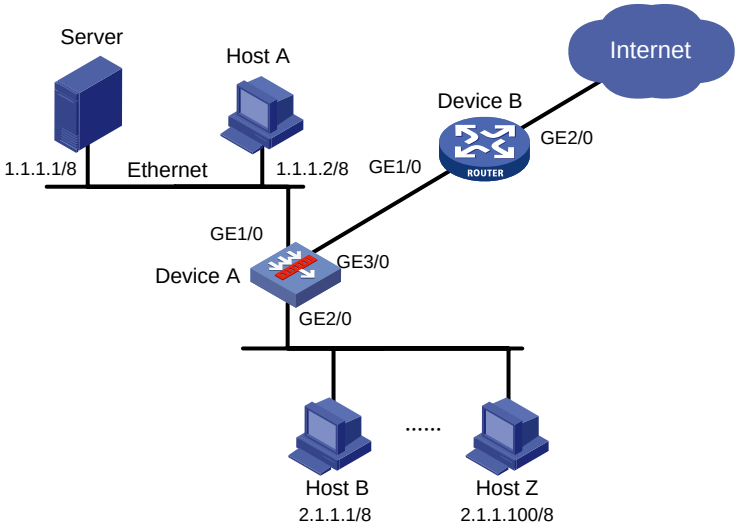
- 来自 Server 的报文流量约束为 10240kbps，流量小于 10240kbps 时可以正常发送，流量超过 10240kbps 时则将违规报文的优先级设置为 0 后进行发送；
- 来自 Host A 的报文流量约束为 2560kbps，流量小于 2560kbps 时可以正常发送，流量超过 2560kbps 时则丢弃违规报文；

对设备 Device B 的 GigabitEthernet1/0 和 GigabitEthernet2/0 接口收发报文有如下要求：

- Device B 的 GigabitEthernet1/0 接口接收报文的总流量限制为 20480kbps，如果超过流量限制则将违规报文丢弃；
- 经由 Device B 的 GigabitEthernet2/0 接口进入 Internet 的报文流量限制为 10240kbps，如果超过流量限制则将违规报文丢弃。

2. 组网图

图3-2 流量监管配置组网图



3. 配置步骤

(1) 配置设备 Device A

配置 ACL 规则列表，分别匹配来源于 Server 和 Host A 的报文流。

```
<DeviceA> system-view
[DeviceA] acl basic 2001
[DeviceA-acl-ipv4-basic-2001] rule permit source 1.1.1.1 0
[DeviceA-acl-ipv4-basic-2001] quit
[DeviceA] acl basic 2002
[DeviceA-acl-ipv4-basic-2002] rule permit source 1.1.1.2 0
[DeviceA-acl-ipv4-basic-2002] quit
```

创建流分类 server，匹配 Server 发出的报文流。

```
[DeviceA] traffic classifier server
[DeviceA-classifier-server] if-match acl 2001
[DeviceA-classifier-server] quit
```

创建流分类 host，匹配 Host 发出的报文流。

```
[DeviceA] traffic classifier host
[DeviceA-classifier-host] if-match acl 2002
[DeviceA-classifier-host] quit
```

创建流行为 server，动作为流量监管，cir 为 10240kbps，对超出限制的报文（红色报文）将其 DSCP 优先级设置为 0 后发送。

```
[DeviceA] traffic behavior server
[DeviceA-behavior-server] car cir 10240 red remark-dscp-pass 0
[DeviceA-behavior-server] quit
```

创建流行为 host，动作为流量监管，cir 为 2560kbps，由于默认对红色报文的处理方式就是丢弃，因此无需配置。

```
[DeviceA] traffic behavior host
[DeviceA-behavior-host] car cir 2560
```

```
[DeviceA-behavior-host] quit
```

创建 QoS 策略，命名为 car，将流分类 server 和流行为 server 进行关联；将流分类 host 和流行为 host 进行关联。

```
[DeviceA] qos policy car
[DeviceA-qospolicy-car] classifier server behavior server
[DeviceA-qospolicy-car] classifier host behavior host
[DeviceA-qospolicy-car] quit
```

将 QoS 策略 car 应用到接口 GigabitEthernet1/0 的入方向上。

```
[DeviceA] interface gigabitethernet 1/0
[DeviceA-GigabitEthernet1/0] qos apply policy car inbound
```

(2) 配置设备 Device B

配置高级 ACL3001，匹配 HTTP 报文。

```
<DeviceB> system-view
[DeviceB] acl advanced 3001
[DeviceB-acl-adv-3001] rule permit tcp destination-port eq 80
[DeviceB-acl-adv-3001] quit
```

创建流分类 http，匹配 ACL 3001。

```
[DeviceB] traffic classifier http
[DeviceB-classifier-http] if-match acl 3001
[DeviceB-classifier-http] quit
```

创建流分类 class，匹配所有报文。

```
[DeviceB] traffic classifier class
[DeviceB-classifier-class] if-match any
[DeviceB-classifier-class] quit
```

创建流行为 car_inbound，动作为流量监管，cir 为 20480kbps，由于默认对红色报文的处理方式就是丢弃，因此无需配置。

```
[DeviceB] traffic behavior car_inbound
[DeviceB-behavior-car_inbound] car cir 20480
[DeviceB-behavior-car_inbound] quit
```

创建流行为 car_outbound，动作为流量监管，cir 为 10240kbps。

```
[DeviceB] traffic behavior car_outbound
[DeviceB-behavior-car_outbound] car cir 10240
[DeviceB-behavior-car_outbound] quit
```

创建 QoS 策略，命名为 car_inbound，将流分类 class 和流行为 car_inbound 进行关联。

```
[DeviceB] qos policy car_inbound
[DeviceB-qospolicy-car_inbound] classifier class behavior car_inbound
[DeviceB-qospolicy-car_inbound] quit
```

创建 QoS 策略，命名为 car_outbound，将流分类 http 和流行为 car_outbound 进行关联。

```
[DeviceB] qos policy car_outbound
[DeviceB-qospolicy-car_outbound] classifier http behavior car_outbound
[DeviceB-qospolicy-car_outbound] quit
```

将 QoS 策略 car_inbound 应用到接口 GigabitEthernet1/0 的入方向上。

```
[DeviceB] interface gigabitethernet 1/0
[DeviceB-GigabitEthernet1/0] qos apply policy car_inbound inbound
```

将 QoS 策略 car_outbound 应用到接口 GigabitEthernet2/0 的出方向上。

```
[DeviceB] interface gigabitethernet 2/0
[DeviceB-GigabitEthernet2/0] qos apply policy car_outbound outbound
```

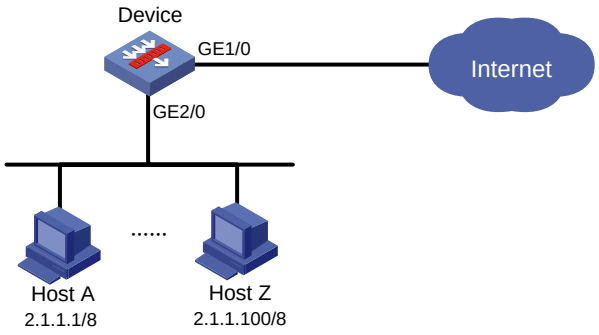
3.4.2 IP 限速配置举例

1. 配置需求

要求在设备 Device 上对接口 GigabitEthernet2/0 接收到的报文流进行限速：对 HostA～HostZ（源地址属于 IP 地址段 2.1.1.1～2.1.1.100）进行 IP 限速，逐 IP 地址流量限速 5kbps，网段内各 IP 地址的流量共享剩余带宽。

2. 组网图

图3-3 IP 限速配置组网图



3. 配置步骤

在接口 GigabitEthernet2/0 上对源地址属于 IP 地址段 2.1.1.1～2.1.1.100 内所有 PC 进行限速，网段内各 IP 地址的流量共享剩余带宽。

```
<Device> system-view
[Device] qos carl 1 source-ip-address range 2.1.1.1 to 2.1.1.100 per-address shared-bandwidth
[Device] interface gigabitethernet 2/0
[Device-GigabitEthernet2/0] qos car inbound carl 1 cir 500 cbs 1875 ebs 0 green pass red
discard
[Device-GigabitEthernet2/0] quit
```

4 拥塞管理

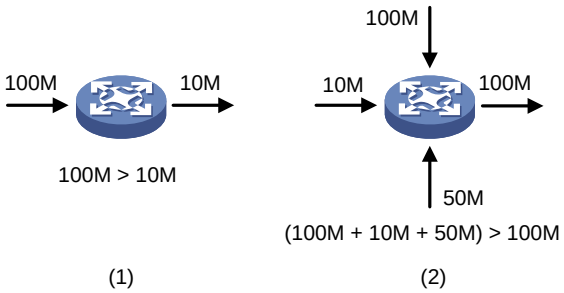
4.1 拥塞管理简介

4.1.1 拥塞的产生、影响和对策

所谓拥塞，是指当前供给资源相对于正常转发处理需要资源的不足，从而导致服务质量下降的一种现象。

在复杂的 Internet 分组交换环境下，拥塞极为常见。以图 4-1 中的两种情况为例：

图4-1 流量拥塞示意图



拥塞有可能会引发一系列的负面影响：

- 拥塞增加了报文传输的延迟和抖动，可能会引起报文重传，从而导致更多的拥塞产生。
- 拥塞使网络的有效吞吐率降低，造成网络资源的利用率降低。
- 拥塞加剧会耗费大量的网络资源（特别是存储资源），不合理的资源分配甚至可能导致系统陷入资源死锁而崩溃。

在分组交换以及多用户业务并存的复杂环境下，拥塞又是不可避免的，因此必须采用适当的方法来解决拥塞。

拥塞管理的中心内容就是当拥塞发生时如何制定一个资源的调度策略，以决定报文转发的处理次序。

4.1.2 设备支持的拥塞管理方法

对于拥塞管理，一般采用队列技术，使用一个队列算法对流量进行分类，之后用某种优先级别算法将这些流量发送出去。每种队列算法都是用以解决特定的网络流量问题，并对带宽资源的分配、延迟、抖动等有着十分重要的影响。

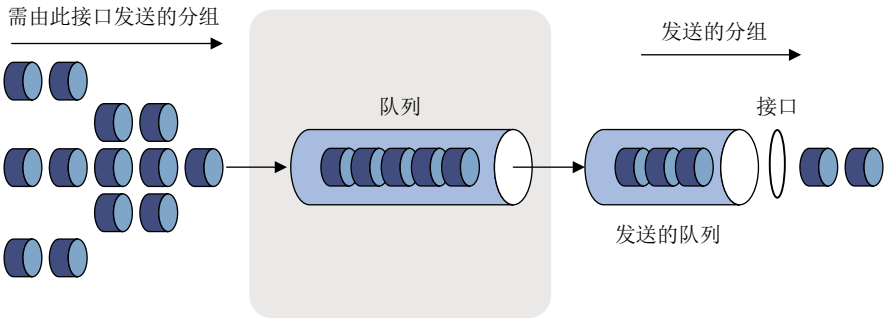
拥塞管理的处理包括队列的创建、报文的分类、将报文送入不同的队列、队列调度等。

目前，设备支持如下几种队列：

- FIFO 队列
- RTP 优先队列

4.1.3 FIFO 队列

图4-2 先入先出队列示意图



如图 4-2 所示，FIFO 按照时间到达的先后决定分组的转发次序，先进的先出，后进的后出，不需要进行流分类和队列调度，FIFO 关心的只是队列的长度，队列的长度对延迟和丢包率的影响。用户的业务流在某个设备能够获得的资源取决于分组的到达时机及当时的负载情况。Best-Effort 报文转发方式采用的就是 FIFO 的排队策略。

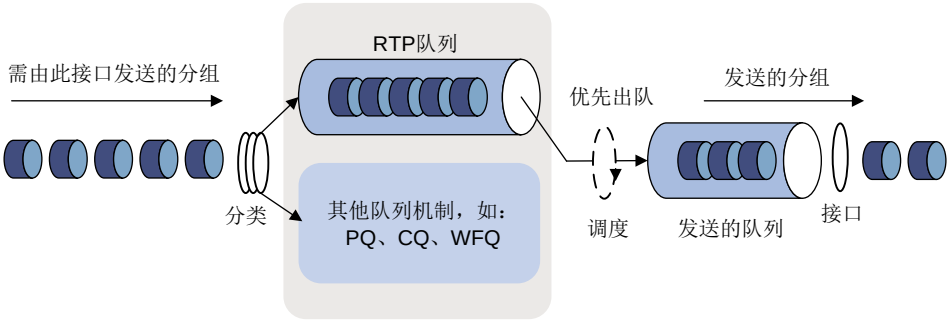
如果设备的每个端口只有一个基于 FIFO 的输入或输出队列，那么恶性的应用可能会占用所有的网络资源，严重影响关键业务数据的传送。所以还需要配置一些其他的队列调度机制与 FIFO 配合对流量进行调度和拥塞控制。

每个队列内部报文的发送次序缺省是 FIFO。

4.1.4 RTP 优先队列

RTP 优先队列是一种保证实时业务（包括语音与视频业务）服务质量的简单的队列技术。其原理就是将承载语音或视频的 RTP 报文送入高优先级队列，使其得到优先发送，保证时延和抖动降低为最低限度，从而保证了语音或视频这种对时延敏感业务的服务质量。

图4-3 RTP 优先队列示意图



如图 4-3 所示，RTP 优先队列将 RTP 报文送入一个具有较高优先级的队列。RTP 报文是端口号在一定范围内为偶数的UDP报文，端口号的范围可以配置。RTP 优先队列可以同其他队列（包括 FIFO、PQ、CQ 和 WFQ）结合使用，而它的优先级是最高的。

4.1.5 拥塞管理技术的对比

设备上提供了以上拥塞管理技术，突破了传统 IP 设备的单一 FIFO 拥塞管理策略，提供了强大的 QoS 能力，使得 IP 设备可以满足不同业务所要求的不同服务质量的要求。为了用户更好地利用拥塞管理技术，现对各种队列技术做一比较。

表4-1 拥塞管理技术对比

类型	队列数	优点	缺点
FIFO	1	• 不需要配置，易于使用 • 处理简单，延迟小	• 所有的报文均进入一个“先进先出”的队列，发送报文所占用的带宽、延迟时间、丢失的概率均由报文到达队列的先后顺序决定 • 对不匹配的数据源（即没有流控机制的流，如 UDP 报文发送）无约束力，不匹配的数据源会造成匹配的数据源（如 TCP 报文发送）带宽受损失 • 对时间敏感的实时应用（如 VoIP）的延迟得不到保证
RTP	1	• 保证了实时业务优先处理 • 在入队前进行流量监管的处理，避免出现其他队列得不到处理的情况	适用范围较窄，一般仅适用于对时延敏感的业务（如语音和视频业务）

说明

如果流量突发较大，可以通过增加队列长度的方法来改善队列调度的准确率。

4.2 配置先进先出队列的长度

FIFO 是接口缺省使用的队列调度机制，可以通过配置命令改变其队列长度。

4.2.1 配置接口先进先出队列的长度

1. 配置限制和指导

在子接口上配置 FIFO 队列时，接口上需要开启接口限速功能以保证队列功能生效。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 配置先进先出队列的长度。

qos fifo queue-length *queue-length*

缺省情况下，FIFO 队列的长度为 75。

如果流量突发较大，可以通过增加队列长度的方法来改善队列调度的准确率。

4.3 配置RTP优先队列

4.3.1 配置接口的 RTP 优先队列

(1) 进入系统视图。

system-view

(2) 进入接口视图。

interface *interface-type interface-number*

(3) 配置 RTP 队列。

qos rtpq start-port *first-rtp-port-number* **end-port** *last-rtp-port-number* **bandwidth** *bandwidth* [**cbs** *cbs*]

缺省情况下，接口上未开启 RTP 队列特性。

4.4 拥塞管理显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示拥塞管理各种队列的运行情况，通过查看显示信息验证配置的效果。

表4-2 拥塞管理的显示和维护

操作	命令
显示指定策略中指定类及与类关联的流行为的配置信息	**display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]]
显示接口上策略的配置信息和运行情况（独立运行模式）	**display qos policy interface** [*interface-type interface-number*] [**inbound** | **outbound**]
显示接口上策略的配置信息和运行情况（IRF模式）	**display qos policy interface** [*interface-type interface-number*] [**slot** *slot-number*] [**inbound** | **outbound**]
显示接口上先进先出队列配置信息和运行情况	**display qos queue fifo interface** [*interface-type interface-number*]
显示接口上所有队列配置情况和统计信息	**display qos queue interface** [*interface-type interface-number*]
显示接口实时传输协议队列配置信息和运行情况	**display qos queue rtpq interface** [*interface-type interface-number*]
显示设备配置的流行为信息	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*]
显示设备配置的类信息	**display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*]

5 流量过滤

5.1 流量过滤简介

流量过滤是指对符合流分类的流进行过滤的动作。例如，可以根据网络的实际情况禁止从某个源 IP 地址发送的报文通过。

5.2 流量过滤配置限制和指导

设备支持基于接口和控制平面应用 QoS 策略配置流量过滤。

5.3 配置流量过滤

(1) 进入系统视图。

system-view

(2) 定义类。

a. 创建一个类，并进入类视图。

traffic classifier *classifier-name* [**operator** { **and** | **or** }]

b. 定义匹配数据包的规则。

if-match [**not**] *match-criteria*

缺省情况下，未定义匹配数据包的规则。

具体规则的介绍，请参见“QoS 命令”中的 **if-match** 命令。

c. 退回系统视图。

quit

(3) 定义流行为。

a. 创建一个流行为，并进入流行为视图。

traffic behavior *behavior-name*

b. 配置流量过滤动作。

filter { **deny** | **permit** }

缺省情况下，未配置流量过滤动作。

如果配置了 **filter deny** 命令，则在该流行为视图下配置的其他流行为（除流量统计外）都不会生效。

c. 退回系统视图。

quit

(4) 定义策略。

a. 创建策略并进入策略视图。

qos policy *policy-name*

b. 在策略中为类指定采用的流行为。

classifier *classifier-name* **behavior** *behavior-name*

缺省情况下，未指定类对应的流行为。

c. 退回系统视图。

quit

(5) 应用 QoS 策略。

具体配置请参见“2.7 应用策略”

缺省情况下，未应用 QoS 策略。

(6) （可选）显示流量过滤的相关配置信息。

（独立运行模式）

display traffic behavior user-defined [*behavior-name*]

（IRF 模式）

display traffic behavior user-defined [*behavior-name*] [**slot** *slot-number*]

5.4 流量过滤典型配置举例

5.4.1 流量过滤基本组网配置举例

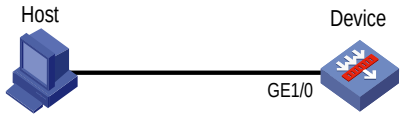
1. 组网需求

Host 通过接口 GigabitEthernet1/0 接入设备 Device。

配置流量过滤功能，对接口 GigabitEthernet1/0 接收的源端口号不等于 21 的 TCP 报文进行丢弃。

2. 组网图

图5-1 流量过滤基本组网图



3. 配置步骤

定义高级 ACL 3000，匹配源端口号不等于 21 的数据流。

```
<Device> system-view
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule 0 permit tcp source-port neq 21
[Device-acl-ipv4-adv-3000] quit
```

定义类 classifier_1，匹配高级 ACL 3000。

```
[Device] traffic classifier classifier_1
[Device-classifier-classifier_1] if-match acl 3000
[Device-classifier-classifier_1] quit
```

定义流行为 behavior_1，动作为流量过滤（**deny**），对数据包进行丢弃。

```
[Device] traffic behavior behavior_1
[Device-behavior-behavior_1] filter deny
[Device-behavior-behavior_1] quit
```

定义策略 policy，为类 classifier_1 指定流行为 behavior_1。

```
[Device] qos policy policy
[Device-qospolicy-policy] classifier classifier_1 behavior behavior_1
[Device-qospolicy-policy] quit
```

将策略 policy 应用到端口 GigabitEthernet1/0 的入方向上。

```
[Device] interface gigabitethernet 1/0
[Device-GigabitEthernet1/0] qos apply policy policy inbound
```

6 协议报文限速

6.1 协议报文限速简介

网络中的协议报文会上送 CPU 进行处理，但是 CPU 处理协议报文的速度有限，如果大量的协议报文同时上送 CPU，会使 CPU 一直忙于处理协议报文，而无法顾及其它任务，最终导致设备瘫痪。协议报文限速功能可以对上送 CPU 的协议报文速率进行限制，保证 CPU 的正常运转。

6.2 配置协议报文限速

(1) 进入系统视图。

system-view

(2) 定义类。

a. 创建一个类，并进入类视图。

traffic classifier *classifier-name* [**operator** { **and** | **or** }]

b. 定义匹配数据包的规则。

if-match [**not**] *match-criteria*

缺省情况下，未定义匹配数据包的规则。

具体规则的介绍，请参见“QoS 命令”中的 **if-match** 命令。

c. 退回系统视图。

quit

(3) 定义流行为。

a. 创建一个流行为，并进入流行为视图。

traffic behavior *behavior-name*

b. 配置协议报文限速动作。

packet-rate *value*

缺省情况下，未配置协议报文限速动作。

c. 退回系统视图。

quit

(4) 定义策略。

a. 创建一个策略，并进入策略视图。

qos policy *policy-name*

b. 在策略中为类指定采用的流行为。

classifier *classifier-name* **behavior** *behavior-name*

缺省情况下，未指定类对应的流行为。

c. 退回系统视图。

quit

(5) 基于控制平面应用 QoS 策略。

具体配置请参见“2.7.4 基于控制平面应用 QoS 策略”

缺省情况下，未应用 QoS 策略。

(6) （可选）显示协议报文限速的相关配置信息。

（独立运行模式）

display traffic behavior user-defined [*behavior-name*]

（IRF 模式）

display traffic behavior user-defined [*behavior-name*] [**slot** *slot-number*]

6.3 协议报文限速典型配置举例

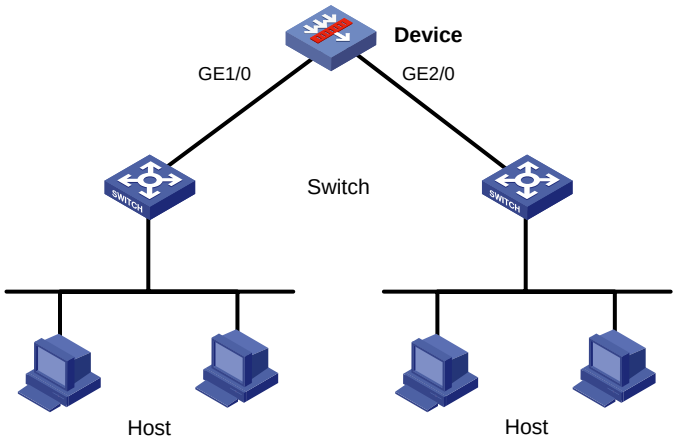
6.3.1 协议报文限速基本组网配置举例

1. 组网需求

多台 Host 通过二层交换机接入设备 Device。配置协议报文限速功能，对设备 CPU 接收的 DHCP 报文限速为每秒 500 个。

2. 组网图

图6-1 协议报文限速基本组网图



3. 配置步骤

定义类 classifier_1，匹配控制平面 DHCP 协议。

```
<Device> system-view
[Device] traffic classifier classifier_1
[Device-classifier-classifier_1] if-match control-plane protocol dhcp
[Device-classifier-classifier_1] quit
```

定义流行为 behavior_1，动作为报文限速，速率为 500 个报文每秒。

```
[Device] traffic behavior behavior_1
[Device-behavior-behavior_1] packet-rate 500
[Device-behavior-behavior_1] quit
```

定义策略 policy，为类 classifier_1 指定流行为 behavior_1。

```
[Device] qos policy policy
[Device-qospolicy-policy] classifier classifier_1 behavior behavior_1
[Device-qospolicy-policy] quit
```

将策略 policy 应用到控制平面。

```
[Device] control-plane slot 1
[Device-cp] qos apply policy policy inbound
```

7 重标记

7.1 重标记简介

重标记是将报文的优先级或者标志位进行设置，重新定义报文的优先级等。例如，对于 IP 报文来说，可以利用重标记对 IP 报文中的 IP 优先级或 DSCP 值进行重新设置，控制 IP 报文的转发。

重标记动作的配置，可以通过与类关联，将原来报文的优先级或标志位重新进行标记。

7.2 配置重标记

1. 配置限制和指导

设备支持基于接口和控制平面应用 QoS 策略配置重标记。

2. 配置步骤

(1) 进入系统视图。

system-view

(2) 定义类。

a. 创建一个类，并进入类视图。

traffic classifier *classifier-name* [**operator** { **and** | **or** }]

b. 定义匹配数据包的规则。

if-match [**not**] *match-criteria*

缺省情况下，未定义匹配数据包的规则。

具体规则的介绍，请参见“QoS 命令”中的 **if-match** 命令。

c. 退回系统视图。

quit

(3) 定义流行为

a. 创建一个流行为，并进入流行为视图。

traffic behavior *behavior-name*

b. 重新标记报文的动作。

具体重标记动作的介绍，请查看“QoS 命令”中的 **remark** 命令。

c. 退回系统视图。

quit

(4) 定义策略。

a. 创建一个策略，并进入策略视图。

qos policy *policy-name*

b. 在策略中为类指定采用的流行为。

classifier *classifier-name* **behavior** *behavior-name*

缺省情况下，未指定类对应的流行为。

c. 退回系统视图。

quit

(5) 应用 QoS 策略。

具体配置请参见“2.7 应用策略”

缺省情况下，未应用 QoS 策略。

(6) （可选）显示重标记的相关配置信息。

（独立运行模式）

display traffic behavior user-defined [*behavior-name*]

（IRF 模式）

display traffic behavior user-defined [*behavior-name*] [**slot** *slot-number*]

7.3 重标记典型配置举例

7.3.1 重标记基本组网配置举例

1. 组网需求

公司企业网通过 Device 实现互连。网络环境描述如下：

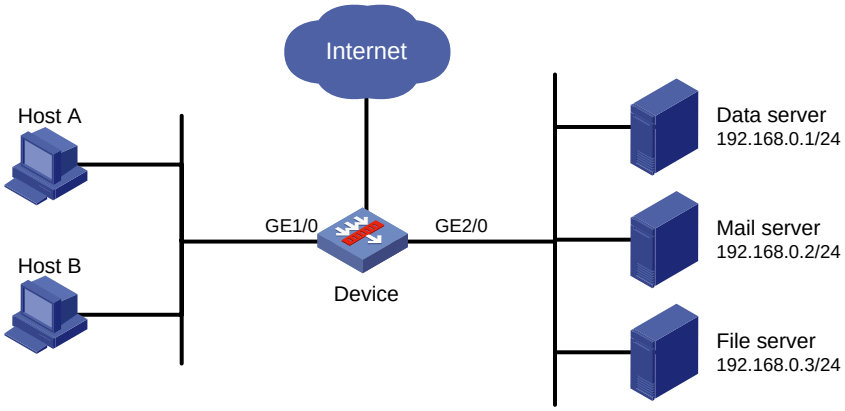
- Host A 和 Host B 通过端口 GigabitEthernet1/0 接入 Device；
- 数据库服务器、邮件服务器和文件服务器通过端口 GigabitEthernet2/0 接入 Device。

通过配置重标记功能，Device 上实现如下需求：

- 优先处理 Host A 和 Host B 访问数据库服务器的报文；
- 其次处理 Host A 和 Host B 访问邮件服务器的报文；
- 最后处理 Host A 和 Host B 访问文件服务器的报文。

2. 组网图

图7-1 重标记基本组网图



3. 配置步骤

定义高级 ACL 3000，对目的 IP 地址为 192.168.0.1 的报文进行分类。

```
<Device> system-view
```

```
[Device] acl advanced 3000
[Device-acl-ipv4-adv-3000] rule permit ip destination 192.168.0.1 0
[Device-acl-ipv4-adv-3000] quit
```

定义高级 ACL 3001，对目的 IP 地址为 192.168.0.2 的报文进行分类。

```
[Device] acl advanced 3001
[Device-acl-ipv4-adv-3001] rule permit ip destination 192.168.0.2 0
[Device-acl-ipv4-adv-3001] quit
```

定义高级 ACL 3002，对目的 IP 地址为 192.168.0.3 的报文进行分类。

```
[Device] acl advanced 3002
[Device-acl-ipv4-adv-3002] rule permit ip destination 192.168.0.3 0
[Device-acl-ipv4-adv-3002] quit
```

定义类 classifier_dbserver，匹配高级 ACL 3000。

```
[Device] traffic classifier classifier_dbserver
[Device-classifier-classifier_dbserver] if-match acl 3000
[Device-classifier-classifier_dbserver] quit
```

定义类 classifier_mserver，匹配高级 ACL 3001。

```
[Device] traffic classifier classifier_mserver
[Device-classifier-classifier_mserver] if-match acl 3001
[Device-classifier-classifier_mserver] quit
```

定义类 classifier_fserver，匹配高级 ACL 3002。

```
[Device] traffic classifier classifier_fserver
[Device-classifier-classifier_fserver] if-match acl 3002
[Device-classifier-classifier_fserver] quit
```

定义流行为 behavior_dbserver，动作为重标记报文的本地优先级为 4。

```
[Device] traffic behavior behavior_dbserver
[Device-behavior-behavior_dbserver] remark local-precedence 4
[Device-behavior-behavior_dbserver] quit
```

定义流行为 behavior_mserver，动作为重标记报文的本地优先级为 3。

```
[Device] traffic behavior behavior_mserver
[Device-behavior-behavior_mserver] remark local-precedence 3
[Device-behavior-behavior_mserver] quit
```

定义流行为 behavior_fserver，动作为重标记报文的本地优先级为 2。

```
[Device] traffic behavior behavior_fserver
[Device-behavior-behavior_fserver] remark local-precedence 2
[Device-behavior-behavior_fserver] quit
```

定义策略 policy_server，为类指定流行为。

```
[Device] qos policy policy_server
[Device-qospolicy-policy_server] classifier classifier_dbserver behavior behavior_dbserver
[Device-qospolicy-policy_server] classifier classifier_mserver behavior behavior_mserver
[Device-qospolicy-policy_server] classifier classifier_fserver behavior behavior_fserver
[Device-qospolicy-policy_server] quit
```

将策略 policy_server 应用到端口 GigabitEthernet1/0 上。

```
[Device] interface gigabitethernet 1/0
[Device-GigabitEthernet1/0] qos apply policy policy_server inbound
[Device-GigabitEthernet1/0] quit
```

8 附录

8.1 附录 A 缩略语表

表8-1 附录 A 缩略语表

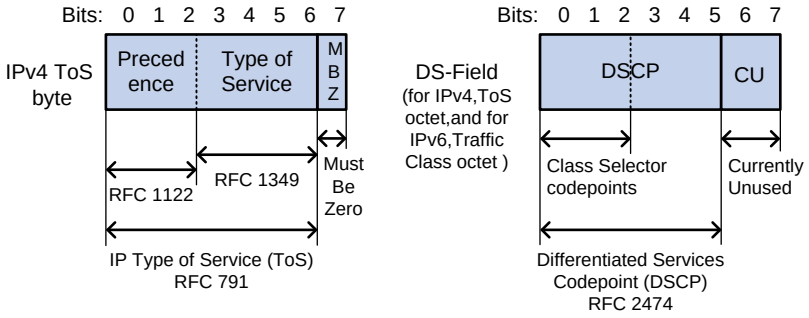
缩略语	英文全名	中文解释
AF	Assured Forwarding	确保转发
BE	Best Effort	尽力转发
BQ	Bandwidth Queuing	带宽队列
CAR	Committed Access Rate	承诺访问速率
CBQ	Class Based Queuing	基于类的队列
CBS	Committed Burst Size	承诺突发尺寸
CBWFQ	Class Based Weighted Fair Queuing	基于类的加权公平队列
CE	Customer Edge	用户边缘设备
CIR	Committed Information Rate	承诺信息速率
CQ	Custom Queuing	定制队列
DAR	Deeper Application Recognition	深度应用识别
DCBX	Data Center Bridging Exchange Protocol	数据中心桥能力交换协议
DiffServ	Differentiated Service	区分服务
DoS	Denial of Service	拒绝服务
DSCP	Differentiated Services Code Point	区分服务编码点
EACL	Enhanced ACL	增强型ACL
EBS	Excess Burst Size	超出突发尺寸
ECN	Explicit Congestion Notification	显示拥塞通知
EF	Expedited Forwarding	加速转发
FEC	Forwarding Equivalance Class	转发等价类
FIFO	First in First out	先入先出
FQ	Fair Queuing	公平队列
GMB	Guaranteed Minimum Bandwidth	最小带宽保证队列
GTS	Generic Traffic Shaping	通用流量整形
IntServ	Integrated Service	综合服务
ISP	Internet Service Provider	互联网服务提供商
LFI	Link Fragmentation and Interleaving	链路分片与交叉

缩略语	英文全名	中文解释
LLQ	Low Latency Queuing	低时延队列
LR	Line Rate	限速
LSP	Label Switched Path	标签交换路径
MPLS	Multiprotocol Label Switching	多协议标签交换
P2P	Peer-to-Peer	对等
PE	Provider Edge	服务提供商网络边缘
PHB	Per-hop Behavior	单中继段行为
PIR	Peak Information Rate	峰值信息速率
PQ	Priority Queuing	优先队列
PW	Pseudowire	伪线
QoS	Quality of Service	服务质量
QPPB	QoS Policy Propagation Through the Border Gateway Protocol	通过BGP传播QoS策略
RED	Random Early Detection	随机早期检测
RSVP	Resource Reservation Protocol	资源预留协议
RTP	Real-time Transport Protocol	实时传输协议
SLA	Service Level Agreement	服务水平协议
SP	Strict Priority	严格优先级队列
TE	Traffic Engineering	流量工程
ToS	Type of Service	服务类型
TP	Traffic Policing	流量监管
TS	Traffic Shaping	流量整形
VoIP	Voice over IP	在IP网络上传送语音
VPN	Virtual Private Network	虚拟专用网络
VSI	Virtual Station Interface	虚拟服务器接口
WFQ	Weighted Fair Queuing	加权公平队列
WRED	Weighted Random Early Detection	加权随机早期检测
WRR	Weighted Round Robin	加权轮询队列

8.2 附录 C 各种优先级介绍

8.2.1 IP 优先级和 DSCP 优先级

图8-1 ToS 和 DS 域



如图 8-1 所示，IP 报文头的 ToS 字段有 8 个 bit，其中前 3 个 bit 表示的就是 IP 优先级，取值范围为 0～7。RFC 2474 中，重新定义了 IP 报文头部的 ToS 域，称之为 DS（Differentiated Services，差分服务）域，其中 DSCP 优先级用该域的前 6 位（0～5 位）表示，取值范围为 0～63，后 2 位（6、7 位）是保留位。

表8-2 IP 优先级说明

IP 优先级（十进制）	IP 优先级（二进制）	关键字
0	000	routine
1	001	priority
2	010	immediate
3	011	flash
4	100	flash-override
5	101	critical
6	110	internet
7	111	network

表8-3 DSCP 优先级说明

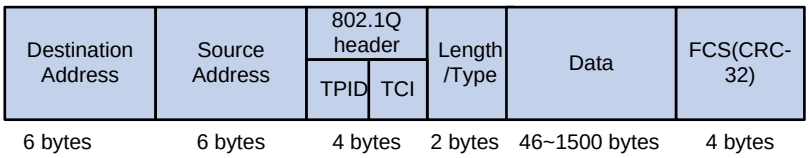
DSCP 优先级（十进制）	DSCP 优先级（二进制）	关键字
46	101110	ef
10	001010	af11
12	001100	af12
14	001110	af13
18	010010	af21
20	010100	af22
22	010110	af23

DSCP 优先级（十进制）	DSCP 优先级（二进制）	关键字
26	011010	af31
28	011100	af32
30	011110	af33
34	100010	af41
36	100100	af42
38	100110	af43
8	001000	cs1
16	010000	cs2
24	011000	cs3
32	100000	cs4
40	101000	cs5
48	110000	cs6
56	111000	cs7
0	000000	be（default）

8.2.2 802.1p 优先级

802.1p 优先级位于二层报文头部，适用于不需要分析三层报头，而需要在二层环境下保证 QoS 的场合。

图8-2 带有 802.1Q 标签头的以太网帧



如图 8-2 所示，4 个字节的 802.1Q 标签头包含了 2 个字节的 TPID（Tag Protocol Identifier，标签协议标识符）和 2 个字节的 TCI（Tag Control Information，标签控制信息），TPID 取值为 0x8100，图 8-3 显示了 802.1Q 标签头的详细内容，Priority 字段就是 802.1p 优先级。之所以称此优先级为 802.1p 优先级，是因为有关这些优先级的应用是在 802.1p 规范中被详细定义的。

图8-3 802.1Q 标签头

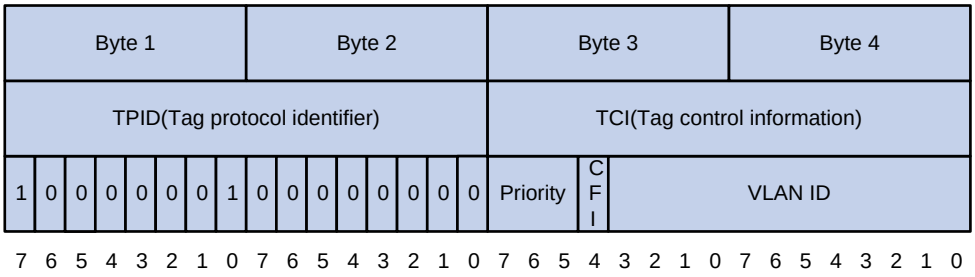


表8-4 802.1p 优先级说明

802.1p 优先级（十进制）	802.1p 优先级（二进制）	关键字
0	000	best-effort
1	001	background
2	010	spare
3	011	excellent-effort
4	100	controlled-load
5	101	video
6	110	voice
7	111	network-management

8.2.3 EXP 优先级

EXP 优先级位于 MPLS 标签内，用于标记 MPLS QoS。

图8-4 MPLS 标签的封装结构

在图 8-4 中，Exp 字段就是 EXP 优先级，长度为 3 比特，取值范围为 0～7。

目 录

1 时间段

1.1 时间段简介

时间段（Time Range）定义了一个时间范围。用户通过创建一个时间段并在某业务中将其引用，就可使该业务在此时间段定义的时间范围内生效。

譬如，当一个 ACL 规则只需在某个特定时间范围内生效时，就可以先配置好这个时间段，然后在配置该 ACL 规则时引用此时间段，这样该 ACL 规则就只能在该时间段定义的时间范围内生效。

在一个时间段中，可以使用以下两种方式定义时间范围：

- 周期时间段：表示以一周为周期（如每周一的 8 至 12 点）循环生效的时间段。
- 绝对时间段：表示在指定时间范围内（如 2015 年 1 月 1 日 8 点至 2015 年 1 月 3 日 18 点）生效的时间段。

当一个时间段内包含有多个周期时间段和绝对时间段时，系统将先分别取各周期时间段的并集和各绝对时间段的并集，再取这两个并集的交集作为该时间段最终生效的时间范围。

1.2 时间段配置限制和指导

如果一个业务所引用的时间段尚未配置或已被删除，该业务将不会生效。

用户最多可创建 1024 个不同名称的时间段。一个时间段内最多可以包含 32 个周期时间段和 12 个绝对时间段。

1.3 配置时间段

(1) 进入系统视图。

system-view

(2) 创建时间段。

time-range *time-range-name* { *start-time* **to** *end-time days* [**from** *time1 date1*] [**to** *time2 date2*] | **from** *time1 date1* [**to** *time2 date2*] | **to** *time2 date2* }

如果指定的时间段已经创建，则本命令可以修改时间段的时间范围。

1.4 时间段显示和维护

在完成上述配置后，在任意视图下执行 **display** 命令可以显示时间段配置后的运行情况，通过查看显示信息验证配置的效果。

表1-1 时间段显示和维护

配置	命令
显示时间段的配置和状态信息	**display time-range** { *time-range-name* | **all** }

1.5 时间段典型配置举例

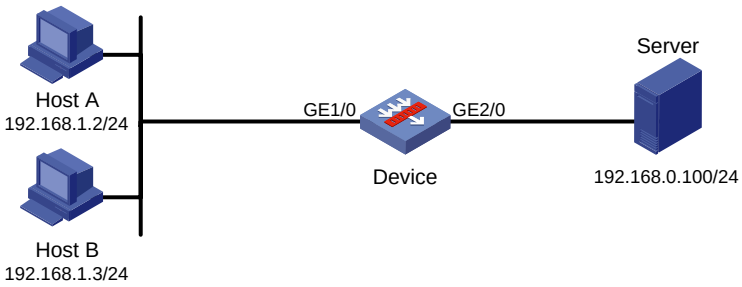
1.5.1 时间段基本组网配置举例

1. 组网需求

要求通过在 Device 上配置 ACL 规则，实现在 2015 年 6 月到 2015 年 12 月之间每周工作日的 8 点到 18 点只允许 Host A 访问 Server。

2. 组网图

图1-1 时间段典型配置组网图



3. 配置步骤

创建名为 work 的时间段，其时间范围为 2015 年 6 月到 2015 年 12 月之间每周工作日的 8 点到 18 点。

```
<Device> system-view
[Device] time-range work 08:00 to 18:00 working-day from 00:00 6/1/2015 to 24:00 12/31/2015
```

创建 IPv4 基本 ACL 2001，并制订如下规则：在名为 work 的时间段内只允许来自 192.168.1.2/32 的报文通过、禁止来自其它 IP 地址的报文通过。

```
[Device] acl basic 2001
[Device-acl-ipv4-basic-2001] rule permit source 192.168.1.2 0 time-range work
[Device-acl-ipv4-basic-2001] rule deny source any time-range work
[Device-acl-ipv4-basic-2001] quit
```

应用 IPv4 基本 ACL 2001 对接口 GigabitEthernet2/0 出方向上的报文进行过滤。

```
[Device] interface gigabitethernet 2/0
[Device-GigabitEthernet2/0] packet-filter 2001 outbound
[Device-GigabitEthernet2/0] quit
```

4. 验证配置

配置完成后，在 Device 上可以使用 **display time-range** 命令查看时间段的配置和状态信息：

显示所有时间段的配置和状态信息。

```
[Device] display time-range all
Current time is 13:58:35 6/19/2015 Friday

Time-range : work (Active)
 08:00 to 18:00 working-day
 from 00:00:00 6/1/2015 to 00:00:00 1/1/2012
```

由此可见，时间段 work 已经生效。