H3C M9000 系列多业务安全网关

ACL 和 QoS 配置指导

新华三技术有限公司
http://www.h3c.com

资料版本：6W201-20190315

前 言

本配置指导介绍了 M9000 系列产品各软件特性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 ACL、QoS 和时间段相关的特性。

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）是一或多条规则的集合，用于识别报文流。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。网络设备依照这些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。

ACL可以应用在诸多领域，其中最基本的就是应用ACL进行报文过滤，具体配置过程请参见“1.3.5 应用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，有关ACL在这些业务中的具体应用方式，请参见相关的配置指导。

提示

ACL 本身只能识别报文，而无法对识别出的报文进行处理，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** 和 **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 0 开始，按照步长自动分配一个大于现有最大编号的最小编号。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。

如果步长发生了改变，ACL 内原有全部规则的编号都将自动从 0 开始按新步长重新排列。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。

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配置任务简介

表1-3 ACL 配置任务简介

配置任务	说明	详细配置
配置基本ACL	请根据需要匹配的报文特征选择ACL类型	1.3.1
配置高级ACL		1.3.2
配置二层ACL		1.3.3
复制ACL	可选	1.3.4

配置任务	说明	详细配置
应用ACL进行报文过滤	可选	1.3.5
使能ACL加速功能	可选	1.3.6

1.3 配置ACL

说明

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。

1.3.1 配置基本ACL

1. 配置IPv4 基本ACL

IPv4 基本 ACL 根据报文的源 IP 地址来制订规则，对 IPv4 报文进行匹配。

表1-4 配置 IPv4 基本 ACL

操作	命令	说明
进入系统视图	**system-view**	-
创建IPv4基本ACL，并进入IPv4基本ACL视图	**acl basic** { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]	缺省情况下，不存在ACL IPv4基本ACL的编号范围为2000～2999 如果以名称创建IPv4基本ACL，只能使用**acl basic name** *acl-name*命令进入其视图 如果以编号创建IPv4基本ACL，只能使用**acl basic** *acl-number*命令进入其视图 两个视图独立，只能通过各自的命令访问各自的视图
（可选）配置ACL的描述信息	**description** *text*	缺省情况下，ACL没有描述信息
（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
创建规则	**rule** [*rule-id*] { **deny** | **permit** } [**counting** | **fragment** | **logging** | **source** { **object-group** *address-group-name* | *source-address source-wildcard* | **any** } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	缺省情况下，IPv4基本ACL内不存在规则 **logging**参数需要使用该ACL的模块支持日志记录功能，例如报文过滤
（可选）为指定规则配置描述信息	**rule** *rule-id* **comment** *text*	缺省情况下，规则没有描述信息

2. 配置IPv6 基本ACL

IPv6 基本 ACL 根据报文的源 IPv6 地址来制订规则，对 IPv6 报文进行匹配。

表1-5 配置 IPv6 基本 ACL

操作	命令	说明
进入系统视图	**system-view**	-
创建IPv6基本ACL，并进入IPv6基本ACL视图	**acl ipv6 basic** { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]	缺省情况下，不存在ACL IPv6基本ACL的编号范围为2000～2999 如果以名称创建IPv6基本ACL，只能使用**acl ipv6 basic name** *acl-name*命令进入其视图 如果以编号创建IPv6基本ACL，只能使用 **acl ipv6 basic** *acl-number*命令进入其视图 两个视图独立，只能通过各自的命令访问各自的视图
（可选）配置ACL的描述信息	**description** *text*	缺省情况下，ACL没有描述信息
（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
创建规则	**rule** [*rule-id*] { **deny** | **permit** } [**counting** | **fragment** | **logging** | **routing** [**type** *routing-type*] | **source** { **object-group** *address-group-name* | *source-address source-prefix* | *source-address/source-prefix* | **any** } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	缺省情况下，IPv6基本ACL内不存在规则 **logging**参数需要使用该ACL的模块支持日志记录功能，例如报文过滤
（可选）为指定规则配置描述信息	**rule** *rule-id* **comment** *text*	缺省情况下，规则没有描述信息

1.3.2 配置高级ACL

1. 配置IPv4 高级ACL

IPv4 高级 ACL 可根据报文的源 IP 地址、目的 IP 地址、报文优先级、IP 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 协议的消息类型和消息码等）等信息来制定规则，对 IPv4 报文进行匹配。用户可利用 IPv4 高级 ACL 制订比 IPv4 基本 ACL 更准确、丰富、灵活的规则。

表1-6 配置 IPv4 高级 ACL

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
创建IPv4高级ACL，并进入IPv4高级ACL视图	**acl advanced** { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]	缺省情况下，不存在ACL IPv4高级ACL的编号范围为3000～3999 如果以名称创建IPv4高级ACL，只能使用**acl advanced name** *acl-name*命令进入其视图 如果以编号创建IPv4高级ACL，只能使用**acl advanced** *acl-number*命令进入其视图 两个视图独立，只能通过各自的命令访问各自的视图
（可选）配置ACL的描述信息	**description** *text*	缺省情况下，ACL没有描述信息
（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
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** { **object-group** *address-group-name* | *dest-address dest-wildcard* | **any** } | **destination-port** { **object-group** *port-group-name* | *operator port1* [*port2*] } | { **dscp** *dscp* | { **precedence** *precedence* | **tos** *tos* } * } | **fragment** | **icmp-type** { *icmp-type* [*icmp-code*] | *icmp-message* } | **logging** | **source** { **object-group** *address-group-name* | *source-address source-wildcard* | **any** } | **source-port** { **object-group** *port-group-name* | *operator port1* [*port2*] } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	缺省情况下，IPv4高级ACL内不存在规则 **logging**参数需要使用该ACL的模块支持日志记录功能，例如报文过滤
（可选）为指定规则配置描述信息	**rule** *rule-id* **comment** *text*	缺省情况下，规则没有描述信息

2. 配置IPv6 高级ACL

IPv6 高级 ACL 可根据报文的源 IPv6 地址、目的 IPv6 地址、报文优先级、IPv6 承载的协议类型及特性（如 TCP/UDP 的源端口和目的端口、TCP 报文标识、ICMPv6 协议的消息类型和消息码等）等信息来制定规则，对 IPv6 报文进行匹配。用户可利用 IPv6 高级 ACL 制订比 IPv6 基本 ACL 更准确、丰富、灵活的规则。

表1-7 配置 IPv6 高级 ACL

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
创建IPv6高级ACL，并进入IPv6高级ACL视图	**acl ipv6 advanced** { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]	缺省情况下，不存在ACL IPv6高级ACL的编号范围3000～3999 如果以名称创建IPv6 高级ACL，只能使用**acl ipv6 advanced name** *acl-name*命令进入其视图 如果以编号创建IPv6高级ACL，只能使用 **acl ipv6 advanced** *acl-number*命令进入其视图 两个视图独立，只能通过各自的命令访问各自的视图
（可选）配置ACL的描述信息	**description** *text*	缺省情况下，ACL没有描述信息
（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
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** { **object-group** *address-group-name* | *dest-address dest-prefix* | *dest-address/dest-prefix* | **any** } | **destination-port** { **object-group** *port-group-name* | *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** { **object-group** *address-group-name* | *source-address source-prefix* | *source-address/source-prefix* | **any** } | **source-port** { **object-group** *port-group-name* | *operator port1* [*port2*] } | **time-range** *time-range-name* | **vpn-instance** *vpn-instance-name*] *	缺省情况下，IPv6高级ACL内不存在规则 **logging**参数需要使用该ACL的模块支持日志记录功能，例如报文过滤
（可选）为指定规则配置描述信息	**rule** *rule-id* **comment** *text*	缺省情况下，规则没有描述信息

1.3.3 配置二层ACL

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

表1-8 配置二层 ACL

操作	命令	说明
进入系统视图	**system-view**	-

操作	命令	说明
创建二层ACL，并进入二层ACL视图	**acl mac** { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]	缺省情况下，不存在ACL 二层ACL的编号范围为4000～4999 如果以名称创建二层ACL，只能使用**acl mac name** *acl-name*命令进入其视图 如果以编号创建二层ACL，只能使用 **acl mac** *acl-number*命令进入其视图 两个视图独立，只能通过各自的命令访问各自的视图
（可选）配置ACL的描述信息	**description** *text*	缺省情况下，ACL没有描述信息
（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
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*] *	缺省情况下，二层ACL内不存在规则
（可选）为指定规则配置描述信息	**rule** *rule-id* **comment** *text*	缺省情况下，规则没有描述信息

1.3.4 复制ACL

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。

提示

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

表1-9 复制 ACL

操作	命令	说明
进入系统视图	**system-view**	-
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* }	-

1.3.5 应用ACL进行报文过滤

ACL 最基本的应用就是进行报文过滤，即通过将 ACL 规则应用到指定接口的入方向上，从而对该接口收到的报文进行过滤。

1. 在接口上应用ACL进行报文过滤

一个接口在一个方向上最多可应用 32 个 ACL 进行报文过滤。

表1-10 在接口上应用 ACL 进行报文过滤

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
在接口上应用ACL进行报文过滤	**packet-filter** [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } **inbound**	缺省情况下，接口不对报文进行过滤

2. 在安全域间实例上应用ACL进行报文过滤

一个安全域间实例上最多可应用 32 个 ACL 进行报文过滤。

表1-11 在安全域间实例上应用 ACL 进行报文过滤

操作	命令	说明
进入系统视图	**system-view**	-
进入安全域间实例视图	**zone-pair security source** *source-zone-name* **destination** *destination-zone-name*	-
在安全域间实例上应用ACL进行报文过滤	**packet-filter** [**ipv6**] { *acl-number* | **name** *acl-name* }	缺省情况下，安全域间实例不对报文进行过滤

说明

有关安全域间实例的详细介绍和配置，请参见“安全配置指导”中的“安全域”。

3. 配置报文过滤日志信息或告警信息的生成与发送周期

在配置了报文过滤日志的生成与发送周期之后，设备将周期性地生成报文过滤日志信息并发送到信息中心或生成告警信息并发送到 SNMP 模块，包括该周期内被匹配的报文数量以及所使用的 ACL 规则。同时还会开启报文的首包上送功能，即对匹配规则的数据流的第一个数据包进行记录并送到信息中心或 SNMP 模块。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”。有关告警信息的详细介绍，请参见“网络管理和监控配置指导”中的“SNMP”。

表1-12 配置报文过滤日志信息或告警信息的生成与发送周期

操作	命令	说明
进入系统视图	**system-view**	-
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，同时，报文首包上送功能处于关闭状态。

4. 配置报文过滤的缺省动作

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。

表1-13 配置报文过滤的缺省动作为 deny

操作	命令	说明
进入系统视图	**system-view**	-
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

说明

配置报文过滤的缺省动作在安全域间实例上不会生效。安全域间实例报文过滤的缺省动作为 deny。

5. 配置报文过滤缺省动作统计功能

提示

在接口上只有应用了 ACL 进行报文过滤，才允许使能报文过滤缺省动作统计功能。

使能了报文过滤缺省动作统计功能之后，接口将对报文过滤缺省动作的执行次数进行统计。

表1-14 配置报文过滤缺省动作统计功能

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
在接口上使能报文过滤缺省动作统计功能	**packet-filter default inbound hardware-count**	缺省情况下，报文过滤的缺省动作统计功能处于关闭状态

1.3.6 使能ACL加速功能

在对基于会话的业务报文（如 NAT、ASPF 等）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 加速功能则可以解决上述问题，当对包含大量规则的 ACL 使能了加速功能之后，其规则匹配速度将大大提高，从而提高了设备的转发性能以及新建连接的性能。

表1-15 使能 ACL 加速功能

操作	命令	说明
进入系统视图	**system-view**	-
创建ACL，并进入ACL视图	**acl** [**ipv6**] { **advanced** | **basic** } { *acl-number* | **name** *acl-name* } [**match-order** { **auto** | **config** }]	-
使能加速功能	**accelerate**	缺省情况下，所有ACL的加速功能均处于关闭状态

1.4 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1-16 ACL 显示和维护

配置	命令
显示ACL的加速状态（分布式设备－独立运行模式）	**display acl accelerate** { **summary** [**ipv6**] | **verbose** [**ipv6**] { *acl-number* | **name** *acl-name* } **slot** *slot-number* [**cpu** *cpu-number*] }
显示ACL的加速状态（分布式设备－IRF模式）	**display acl accelerate** { **summary** [**ipv6**] | **verbose** [**ipv6**] { *acl-number* | **name** *acl-name* } **chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*] }
显示ACL的配置和运行情况	**display acl** [**ipv6** | **mac**] { *acl-number* | **all** | **name** *acl-name* }
显示ACL在报文过滤中的应用情况（分布式设备－独立运行模式）	**display packet-filter** { **interface** [*interface-type interface-number*] [**inbound**] | **zone-pair security** [**source** *source-zone-name* **destination** *destination-zone-name*] } [**slot** *slot-number* [**cpu** *cpu-number*]]
显示ACL在报文过滤中的应用情况（分布式设备－IRF模式）	**display packet-filter** { **interface** [*interface-type interface-number*] [**inbound**] | **zone-pair security** [**source** *source-zone-name* **destination** *destination-zone-name*] } [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]]
显示ACL在报文过滤中应用的统计信息以及报文过滤缺省动作的统计信息	**display packet-filter statistics** { **interface** *interface-type interface-number* **inbound** [**default** | [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] | **zone-pair security source** *source-zone-name* **destination** *destination-zone-name* [[**ipv6**] { *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** [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* } [**brief**]
显示ACL在报文过滤中的详细应用情况（分布式设备－独立运行模式）	**display packet-filter verbose** { **interface** *interface-type interface-number* **inbound** [[**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] | **zone-pair security source** *source-zone-name* **destination** *destination-zone-name* [[**ipv6**] { *acl-number* | **name** *acl-name* }] } [**slot** *slot-number* [**cpu** *cpu-number*]]

配置	命令
显示ACL在报文过滤中的详细应用情况（分布式设备－IRF模式）	**display packet-filter verbose** { **interface** *interface-type interface-number* **inbound** [[**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] | **zone-pair security source** *source-zone-name* **destination** *destination-zone-name* [[**ipv6**] { *acl-number* | **name** *acl-name* }] } [**chassis** *chassis-number* **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（包括累加统计信息）以及报文过滤缺省动作的统计信息	**reset packet-filter statistics** { **interface** [*interface-type interface-number*] **inbound** [**default** | [**ipv6** | **mac**] { *acl-number* | **name** *acl-name* }] | **zone-pair security** [**source** *source-zone-name* **destination** *destination-zone-name*] [**ipv6**] { *acl-number* | **name** *acl-name* } }

1.5 ACL典型配置举例

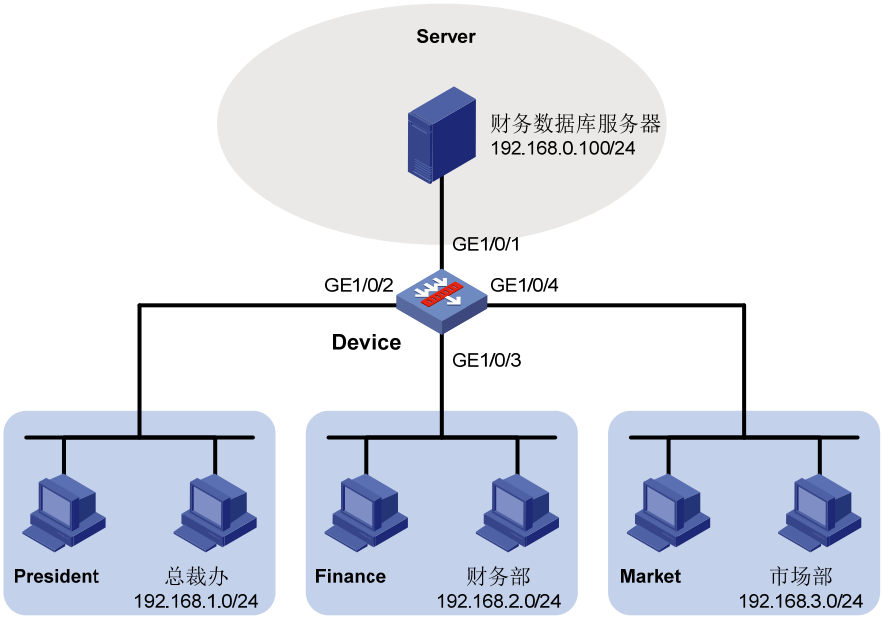
1.5.1 在安全域间实例上应用包过滤的ACL典型配置举例

1. 组网需求

- 某公司内的各部门之间通过 Device 实现互连，总裁办、财务部和市场部分别属于 President 域、Finance 域和 Market 域。该公司的工作时间为每周工作日的 8 点到 18 点。
- 通过在安全域间实例上配置包过滤，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

将接口 GigabitEthernet1/0/1 加入 Server 域。

```
<Device> system-view
[Device] security-zone name Server
[Device-security-zone-Server] import interface gigabitethernet 1/0/1
[Device-security-zone-Server] quit
```

将接口 GigabitEthernet1/0/2 加入 President 域。

```
[Device] security-zone name President
[Device-security-zone-President] import interface gigabitethernet 1/0/2
[Device-security-zone-President] quit
```

将接口 GigabitEthernet1/0/3 加入 Finance 域。

```
[Device] security-zone name Finance
[Device-security-zone-Finance] import interface gigabitethernet 1/0/3
[Device-security-zone-Finance] quit
```

将接口 GigabitEthernet1/0/4 加入 Market 域。

```
[Device] security-zone name Market
[Device-security-zone-Market] import interface gigabitethernet 1/0/4
[Device-security-zone-Market] quit
```

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] time-range work 08:00 to 18:00 working-day
```

创建 IPv4 高级 ACL 3000，允许总裁办在任意时间访问财务数据库服务器。

```
[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] quit
```

创建 IPv4 高级 ACL 3001，允许财务部在工作时间访问财务数据库服务器。

```
[Device] acl advanced 3001
[Device-acl-ipv4-adv-3001] 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-3001] quit
```

创建 IPv4 高级 ACL 3002，禁止其它部门在任何时间访问财务数据库服务器。

```
[Device] acl advanced 3002
[Device-acl-ipv4-adv-3002] rule deny ip source any destination 192.168.0.100 0
[Device-acl-ipv4-adv-3002] quit
```

创建域间实例（源域为 President、目的域为 Server），并在该域间实例上引用 ACL 3000 进行包过滤。

```
[Device] zone-pair security source president destination server
[Device-zone-pair-security-President-Server] packet-filter 3000
[Device-zone-pair-security-President-Server] quit
```

创建域间实例（源域为 Finance、目的域为 Server），并在该域间实例上引用 ACL 3001 进行包过滤。

```
[Device] zone-pair security source finance destination server
[Device-zone-pair-security-Finance-Server] packet-filter 3001
[Device-zone-pair-security-President-Server] quit
```

创建域间实例（源域为 Market、目的域为 Server），并在该域间实例上引用 ACL 3002 进行包过滤。

```
[Device] zone-pair security source market destination server
[Device-zone-pair-security-Market-Server] packet-filter 3002
[Device-zone-pair-security-Market-Server] 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 3001 和 ACL 3002 的配置和运行情况。

```
[Device] display acl 3001
Advanced IPv4 ACL 3001, 1 rule,
ACL's step is 5
 rule 0 permit ip source 192.168.2.0 0.0.0.255 destination 192.168.0.100 0 time-range work
(4 times matched) (Active)
[Device] display acl 3002
Advanced IPv4 ACL 3002, 1 rule,
ACL's step is 5
 rule 0 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 ACL 3001 的规则 0 是生效的；且由于之前使用了 **ping** 命令的缘故，ACL 3001 和 ACL 3002 的规则 0 分别被匹配了 4 次。

目 录

1 QoS简介

1.1 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，只要存在抢夺网络资源的情况，就会出现服务质量的要求。服务质量是相对网络业务而言的，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。例如，在网络总带宽固定的情况下，如果某类业务占用的带宽越多，那么其他业务能使用的带宽就越少，可能会影响其他业务的使用。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。

下面从 QoS 服务模型出发，对目前使用最多、最成熟的一些 QoS 技术逐一进行描述。在特定的环境下合理地使用这些技术，可以有效地提高服务质量。

1.2 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）

1.2.1 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。

1.2.2 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。

1.2.3 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。

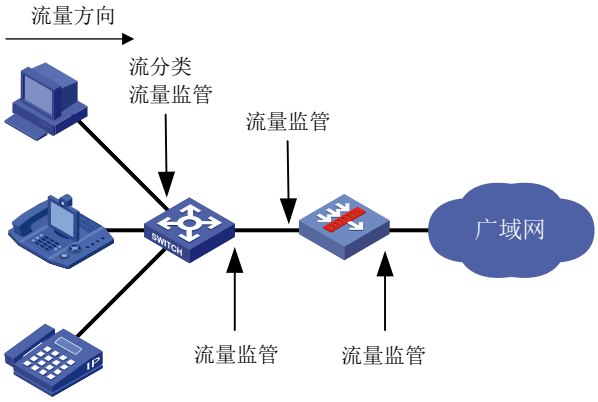
本文提到的技术都是基于 DiffServ 服务模型。

1.3 QoS技术综述

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。

1.3.1 QoS技术在网络中的位置

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。

1.3.2 QoS技术在设备中的处理顺序

图1-2 各 QoS 技术在同一网络设备中的处理顺序

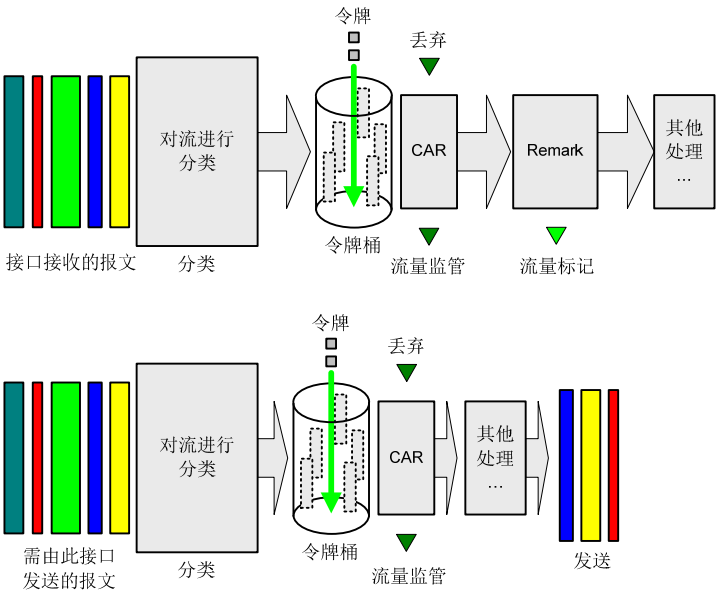


图 1-2 简要描述了各种QoS技术在网络设备中的处理顺序。

(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。

2 QoS配置方式

2.1 配置方式介绍

QoS 的配置方式分为 QoS 策略配置方式和非 QoS 策略配置方式两种。

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。

2.1.1 非QoS策略配置方式

非 QoS 策略配置方式是指不通过 QoS 策略来进行配置。

2.1.2 QoS策略配置方式

QoS 策略配置方式是指通过配置 QoS 策略来实现 QoS 功能。

QoS 策略包含了三个要素：类、流行为、策略。用户可以通过 QoS 策略将指定的类和流行为绑定起来，灵活地进行 QoS 配置。

1. 类

类的要素包括：类的名称和类的规则。

用户可以通过命令定义一系列的规则来对报文进行分类。

2. 流行为

流行为用来定义针对报文所做的 QoS 动作。

流行为的要素包括：流行为的名称和流行为中定义的动作。

用户可以通过命令在一个流行为中定义多个动作。

3. 策略

策略用来将指定的类和流行为绑定起来，对符合分类条件的报文执行流行为中定义的动作。

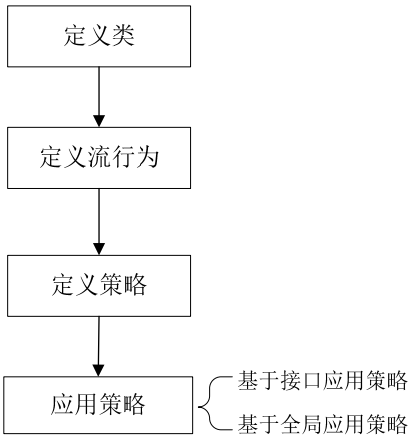
策略的要素包括：策略名称、绑定在一起的类和流行为的名称。

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策略配置方式的步骤

如 图 2-1 所示：

图2-1 QoS 策略配置方式的步骤



2.2.1 定义类

定义类首先要创建一个类，然后在该类的视图下配置匹配规则。

表2-1 定义类

操作	命令	说明
进入系统视图	**system-view**	-
创建类，并进入类视图	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍

2.2.2 定义流行为

定义流行为首先需要创建一个流行为，然后可以在该流行为视图下根据需要配置相应的 QoS 动作。每个流行为由一组 QoS 动作组成。

表2-2 定义流行为

操作	命令	说明
进入系统视图	**system-view**	-
创建流行为，并进入流行为视图	**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
配置流行为的动作	流行为就是对应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	缺省情况下，未配置流行为的动作

2.2.3 定义策略

1. 配置父策略

在策略视图下为类指定对应的流行为。以某种匹配规则将流区分为不同的类，再结合不同的流行为就能很灵活的实现各种 QoS 功能。

表2-3 定义策略

操作	命令	说明
进入系统视图	**system-view**	-
创建QoS策略，并进入策略视图	**qos policy** *policy-name*	缺省情况下，不存在QoS策略
为类指定流行为	**classifier** *classifier-name* **behavior** *behavior-name* [**insert-before** *before-classifier-name*]	缺省情况下，没有为类指定流行为

2. 配置子策略

通过在流行为视图下应用子策略，可以实现策略嵌套功能。即由 **traffic classifier** 命令定义的某一类流量，除了执行父策略中定义的行为外，还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行为。

表2-4 配置子策略

操作	命令	说明
进入系统视图	**system-view**	-
创建一个类，并进入类视图	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍
退回系统视图	**quit**	-
创建一个流行为，并进入流行为视图	**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
配置子策略	**traffic-policy** *policy-name*	缺省情况下，未配置嵌套策略
退出流行为视图	**quit**	-
创建一个策略，并进入策略视图	**qos policy** *policy-name*	缺省情况下，不存在策略
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，没有为类指定流行为

2.2.4 应用策略

QoS 策略支持以下应用方式：

- 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。
- 基于全局应用 QoS 策略：QoS 策略对所有流量生效。

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。当流分类规则中匹配的是 ACL 时，允许删除或修改该 ACL（包括向该 ACL 中添加、删除和修改规则）。。

1. 基于接口应用QoS策略

一个策略可以应用于多个接口。接口的每个方向（出和入两个方向）只能应用一个策略。

如果 QoS 策略应用在接口的出方向，则 QoS 策略对本地协议报文不起作用。本地协议报文是设备内部发起的某些报文，它是维持设备正常运行的重要协议报文。为了确保这些报文能够被不受影响的发送出去，即便在接口的出方向应用了 QoS 策略，本地协议报文也不会受到 QoS 策略的限制，从而降低了因配置 QoS 而误将这些报文丢弃或进行其他处理的风险。一些常见的本地协议报文如下：链路维护报文、IS-IS、OSPF、RIP、BGP、LDP、RSVP、SSH 等。

表2-5 在接口上应用策略

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	进入接口视图后，下面进行的配置只在当前接口生效
在接口上应用QoS策略	**qos apply policy** *policy-name* { **inbound** | **outbound** } [**enhancement**]	缺省情况下，未在接口上应用QoS策略 如果策略中与类关联的行为是重定向到Blade时，需应用增强类型策略

2. 基于全局应用QoS策略


提示

当某个单板资源不足导致全局应用 QoS 策略失败时，用户可以执行 undo qos apply policy global 命令进行手工删除。

基于全局应用 QoS 策略可以方便对设备上的所有流量进行管理。

表2-6 基于全局应用 QoS 策略

操作	命令	说明
进入系统视图	**system-view**	-
全局应用QoS策略	**qos apply policy** *policy-name* **global** { **inbound** | **outbound** } **enhancement**	缺省情况下，未在全局应用QoS策略

2.2.5 配置接口流速统计时间

提示

子接口的流速统计时间采用主接口上设置的统计时间。

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。

表2-7 配置接口流速统计时间

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
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

2.2.6 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2-8 QoS 策略显示和维护

操作	命令
显示类的配置信息（分布式设备－独立运行模式）	**display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示类的配置信息（分布式设备－IRF模式）	**display traffic classifier** { **system-defined** | **user-defined** } [*classifier-name*] [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]]
显示流行为的配置信息（分布式设备－独立运行模式）	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示流行为的配置信息（分布式设备－IRF模式）	**display traffic behavior** { **system-defined** | **user-defined** } [*behavior-name*] [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]]
显示QoS策略的配置信息（分布式设备－独立运行模式）	**display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示QoS策略的配置信息（分布式设备－IRF模式）	**display qos policy** { **system-defined** | **user-defined** } [*policy-name* [**classifier** *classifier-name*]] [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]]
显示接口上QoS策略的配置信息和运行情况（分布式设备－独立运行模式）	**display qos policy interface** [*interface-type interface-number*] [**slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**]

操作	命令
显示接口上QoS策略的配置信息和运行情况（分布式设备－IRF模式）	**display qos policy interface** [*interface-type interface-number*] [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**]
显示基于全局应用QoS策略的信息（分布式设备－独立运行模式）	**display qos policy global** [**slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**]
显示基于全局应用QoS策略的信息（分布式设备－IRF模式）	**display qos policy global** [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**]
清除全局应用QoS策略的统计信息	**reset qos policy global** [**inbound** | **outbound**]

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。比如限制每个时间间隔某个流只能得到承诺分配给它的那部分资源，防止由于过分突发所引发的网络拥塞。

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. 令牌桶的特点

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，可以配置两个令牌桶（分别称为 C 桶和 E 桶）。以流量监管为例，分为单速率单桶双色算法、单速率双桶三色算法和双速率双桶三色算法。

(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。

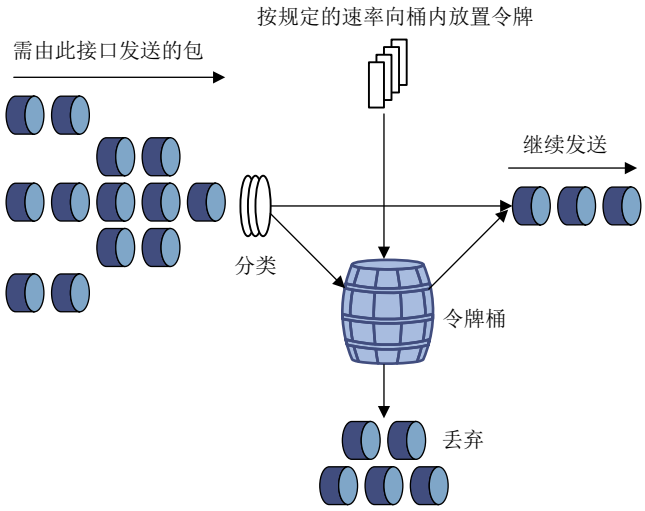
3.1.2 流量监管

说明

流量监管支持入和出两个方向，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

流量监管的配置有两种方式：QoS 策略配置方式和非 QoS 策略配置方式。

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

3.2.1 QoS策略配置方式

表3-1 配置流量监管（QoS 策略配置方式）

操作	命令	说明
进入系统视图	**system-view**	-
创建一个类，并进入类视图	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍

操作		命令	说明
退回系统视图		**quit**	-
创建一个流行为，并进入流行为视图		**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
配置流量监管动作		**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*] *	缺省情况下，未配置流量监管动作
退回系统视图		**quit**	-
创建一个策略，并进入策略视图		**qos policy** *policy-name*	缺省情况下，不存在策略
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，没有为类指定流行为
退回系统视图		**quit**	-
应用QoS策略	基于接口	2.2.4 1. 基于接口应用QoS策略	必选其一 缺省情况下，未应用QoS策略
	基于全局	2.2.4 2. 基于全局应用QoS策略	

3.2.2 非QoS策略配置方式

1. 基于CAR列表的流量监管配置

表3-2 基于 CAR 列表的流量监管配置

操作	命令	说明
进入系统视图	**system-view**	-
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**]] }	缺省情况下，不存在CAR列表
进入接口视图	**interface** *interface-type interface-number*	-

操作	命令	说明
在接口上配置基于CAR列表的CAR策略	**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*] *	缺省情况下，接口上未应用CAR策略

2. 基于ACL的流量监管配置

表3-3 基于 ACL 的流量监管配置

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
在接口上配置基于ACL规则的CAR策略	**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*] *	缺省情况下，接口上未应用CAR策略

3. 适配所有流的流量监管配置

表3-4 适配所有流的流量监管配置

操作	命令	说明
进入系统视图	**system-view**	-
进入接口视图	**interface** *interface-type interface-number*	-
在接口应用CAR策略	**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*] *	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的运行情况，通过查看显示信息验证配置的效果。

表3-5 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（分布式设备－独立运行模式）	**display qos carl** [*carl-index*] [**slot** *slot-number* [**cpu** *cpu-number*]]
显示CAR列表（分布式设备－IRF模式）	**display qos carl** [*carl-index*] [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]]
显示流量监管的相关配置信息	**display traffic behavior user-defined** [*behavior-name*]

4 流量过滤

4.1 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。

4.2 配置流量过滤

表4-1 配置流量过滤

操作		命令	说明
进入系统视图		**system-view**	-
创建一个类，并进入类视图		**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍
退回系统视图		**quit**	-
创建一个流行为，并进入流行为视图		**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
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
退回系统视图		**quit**	-
创建一个策略，并进入策略视图		**qos policy** *policy-name*	缺省情况下，不存在策略
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，没有为类指定流行为
退回系统视图		**quit**	-
应用QoS策略	基于接口	2.2.4 1. 基于接口应用QoS策略	必选其一 缺省情况下，未应用QoS策略
	基于全局	2.2.4 2. 基于全局应用QoS策略	
（可选）显示流量过滤的相关配置信息		**display traffic behavior user-defined** [*behavior-name*]	**display**命令可以在任意视图下执行

5 重标记

5.1 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。

5.2 配置重标记

表5-1 配置重标记

操作		命令	说明
进入系统视图		**system-view**	-
创建一个类，并进入类视图		**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍
退回系统视图		**quit**	-
创建一个流行为，并进入流行为视图		**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
重新标记报文的动作	重新标记报文的802.1p优先级	**remark dot1p** *dot1p-valuer*	必选其一 缺省情况下，未配置重新标记报文的动作
	重新标记报文的DSCP值	**remark dscp** *dscp-value*	
	重新标记报文的IP优先级	**remark ip-precedence** *ip-precedence-value*	
	重新标记报文的QoS本地ID值	**remark qos-local-id** *local-id-value*	
退回系统视图		**quit**	-
创建一个策略，并进入策略视图		**qos policy** *policy-name*	缺省情况下，不存在策略
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，没有为类指定流行为
退回系统视图		**quit**	-
应用QoS	基于接口	2.2.4 1. 基于接口应用QoS策略	必选其一

操作		命令	说明
策略	基于全局	2.2.4 2. 基于全局应用QoS策略	缺省情况下，未应用QoS策略
（可选）显示重标记的相关配置信息		**display traffic behavior user-defined** [*behavior-name*]	**display**命令可以在任意视图下执行

6 流量重定向

6.1 流量重定向简介

流量重定向就是将符合流分类的流重定向到其他地方进行处理。

目前支持的流量重定向包括以下几种：

- 重定向到引擎接口/引擎聚合接口：对于收到需要由某个引擎接口/引擎聚合接口处理的报文时，可以通过配置重定向到此接口。
- 重定向到备份组：对于收到需要由某个备份组处理的报文时，可以通过配置重定向到此备份组。有关备份组的详细内容，请参见“可靠性配置指导”中的“备份组”。

6.2 配置流量重定向

表6-1 配置流量重定向

操作		命令	说明
进入系统视图		**system-view**	-
创建一个类，并进入类视图		**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍
退回系统视图		**quit**	-
创建一个流行为，并进入流行为视图		**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
配置流量重定向动作		**redirect** { **failover-group** *group-name* [**channel** *channel-id*] | **interface** *interface-type interface-number* }	缺省情况下，未配置流量重定向动作 在配置重定向动作时，同一个流行为中重定向类型只能为重定向到接口、重定向到备份组中的一种，以最后一次配置为准
退回系统视图		**quit**	-
创建一个策略，并进入策略视图		**qos policy** *policy-name*	缺省情况下，不存在策略
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，没有为类指定流行为
退回系统视图		**quit**	-
应用QoS策略	基于接口	2.2.4 1. 基于接口应用QoS策略	必选其一 缺省情况下，未应用QoS策略
	基于全局	2.2.4 2. 基于全局应用QoS策略	

操作	命令	说明
（可选）显示流量重定向的相关配置信息	**display traffic behavior user-defined** [*behavior-name*]	**display**命令可以在任意视图下执行

6.3 流量重定向配置举例

6.3.1 重定向至接口配置举例

1. 组网需求

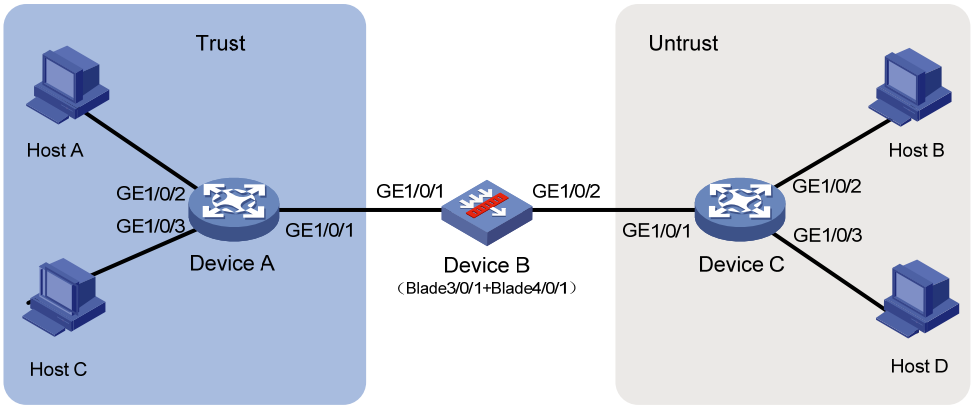
如下图所示，Device B 为 M9000 设备，Slot3 和 Slot4 上分别安装防火墙业务板，引擎口分别为 Blade3/0/1、Blade4/0/1。通过配置流量重定向至引擎口 Blade3/0/1、Blade4/0/1，以便所有通过 Device B 转发的流量都经过引擎板安全策略的处理：符合安全策略的报文，则允许放行，反之则丢弃。

具体需求如下：

- Host A 所在子网与 Host B 所在子网间的流量经过 Slot3 槽位的业务板处理；
- Host C 所在子网与 Host D 所在子网间的流量经过 Slot4 槽位的业务板处理。

2. 组网图

图6-1 重定向至接口配置组网图



设备	接口	IP地址	设备	接口	IP地址
Device A	GE1/0/1	10.1.1.1/24	Device B	GE1/0/1	10.1.1.2/24
	GE1/0/2	10.2.1.1/24		GE1/0/2	11.1.1.2/24
	GE1/0/3	10.3.1.1/24	Host A	/	10.2.1.2/24
Device C	GE1/0/1	11.1.1.1/24	Host B	/	11.2.1.2/24
	GE1/0/2	11.2.1.1/24	Host C	/	10.3.1.2/24
	GE1/0/3	11.3.1.1/24	Host D	/	11.3.1.2/24

3. 配置思路

为实现需求，需要在 Device B 上进行如下配置：

- 将端口 GigabitEthernet1/0/1 接收到的源 IP 地址为 10.2.1.0/24 的报文转发至引擎口 Blade3/0/1 处理，源 IP 地址为 10.3.1.0/24 的报文转发至引擎口 Blade4/0/1 处理；
- 将端口 GigabitEthernet1/0/2 接收到的源 IP 地址为 11.2.1.0/24 的报文转发至引擎口 Blade3/0/1 处理，源 IP 地址为 11.3.1.0/24 的报文转发至引擎口 Blade4/0/1 处理。

4. 配置步骤

配置接口 IP 地址、路由、安全域及域间安全策略保证网络可达，具体配置步骤略。

定义基本 ACL 2000，对源 IP 地址为 10.2.1.0/24、11.2.1.0/24 的报文进行分类。

```
<DeviceB> system-view
[DeviceB] acl number 2000
[DeviceB-acl-basic-2000] rule permit source 10.2.1.0 0.0.0.255
[DeviceB-acl-basic-2000] rule permit source 11.2.1.0 0.0.0.255
[DeviceB-acl-basic-2000] quit
```

定义基本 ACL 2001，对源 IP 地址为 10.3.1.0/24、11.3.1.0/24 的报文进行分类。

```
[DeviceB] acl number 2001
[DeviceB-acl-basic-2001] rule permit source 10.3.1.0 0.0.0.255
[DeviceB-acl-basic-2001] rule permit source 11.3.1.0 0.0.0.255
[DeviceB-acl-basic-2001] quit
```

定义类 classifier_1，匹配基本 ACL 2000。

```
[DeviceB] traffic classifier classifier_1
[DeviceB-classifier-classifier_1] if-match acl 2000
[DeviceB-classifier-classifier_1] quit
```

定义类 classifier_2，匹配基本 ACL 2001。

```
[DeviceB] traffic classifier classifier_2
[DeviceB-classifier-classifier_2] if-match acl 2001
[DeviceB-classifier-classifier_2] quit
```

定义流行为 behavior_1，动作为重定向至 Blade 3/0/1。

```
[DeviceB] traffic behavior behavior_1
[DeviceB-behavior-behavior_1] redirect interface Blade 3/0/1
[DeviceB-behavior-behavior_1] quit
```

定义流行为 behavior_2，动作为重定向至 Blade 4/0/1。

```
[DeviceB] traffic behavior behavior_2
[DeviceB-behavior-behavior_2] redirect interface Blade 4/0/1
[DeviceB-behavior-behavior_2] quit
```

定义策略 policy，为类 classifier_1 指定流行为 behavior_1，为类 classifier_2 指定流行为 behavior_2。

```
[DeviceB] qos policy policy
[DeviceB-qospolicy-policy] classifier classifier_1 behavior behavior_1
[DeviceB-qospolicy-policy] classifier classifier_2 behavior behavior_2
[DeviceB-qospolicy-policy] quit
```

将策略 policy 应用到端口 GigabitEthernet1/0/1 的入方向上。

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

将策略 policy 应用到端口 GigabitEthernet1/0/2 的入方向上。

```
[DeviceB] interface gigabitethernet 1/0/2
[DeviceB-GigabitEthernet1/0/2] qos apply policy policy inbound enhancement
```

7 流量统计

7.1 流量统计简介

流量统计就是通过与类关联，对符合匹配规则的流进行统计，统计报文数或字节数。例如，可以统计从某个源 IP 地址发送的报文，然后管理员对统计信息进行分析，根据分析情况采取相应的措施。

7.2 配置流量统计

表7-1 配置流量统计

操作	命令	说明
进入系统视图	**system-view**	-
创建一个类，并进入类视图	**traffic classifier** *classifier-name* [**operator** { **and** | **or** }]	缺省情况下，不存在类
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 具体规则请参见QoS命令参考中的命令**if-match**的介绍
退回系统视图	**quit**	-
创建一个流行为，并进入流行为视图	**traffic behavior** *behavior-name*	缺省情况下，不存在流行为
为流行为配置流量统计动作	**accounting** [**byte** | **packet**]	缺省情况下，未配置流量统计动作
退回系统视图	**quit**	-
创建一个策略，并进入策略视图	**qos policy** *policy-name*	缺省情况下，不存在策略
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，没有为类指定流行为
退回系统视图	**quit**	-
基于接口应用QoS策略	2.2.4 1. 基于接口应用QoS策略	缺省情况下，未应用QoS策略
（可选）显示流量统计的相关配置信息	分布式设备－独立运行模式： **display qos policy interface** [*interface-type interface-number*] [**slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**] 分布式设备－IRF模式： **display qos policy interface** [*interface-type interface-number*] [**chassis** *chassis-number* **slot** *slot-number* [**cpu** *cpu-number*]] [**inbound** | **outbound**]	**display**命令可以在任意视图下执行

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

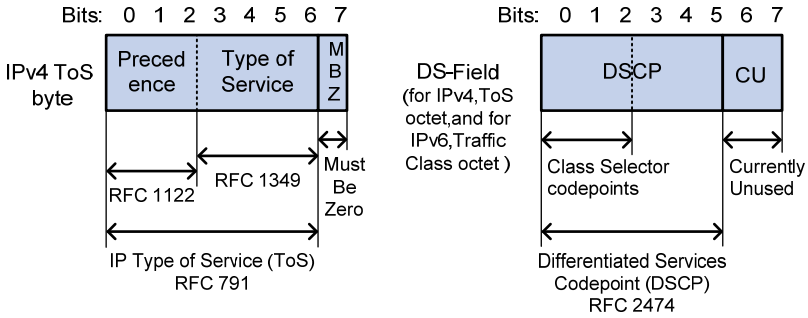
缩略语	英文全名	中文解释
LFI	Link Fragmentation and Interleaving	链路分片与交叉
LLQ	Low Latency Queuing	低时延队列
LR	Line Rate	限速
LSP	Label Switched Path	标签交换路径
MPLS	Multiprotocol Label Switching	多协议标签交换
P2P	Peer-to-Peer	对等
PE	Provider Edge	服务提供商网络边缘
PHB	Per-hop Behavior	单中继段行为
PIR	Peak Information Rate	峰值信息速率
PQ	Priority 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

缩略语	英文全名	中文解释
WRR	Weighted Round Robin	加权轮询队列

8.2 附录 B 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

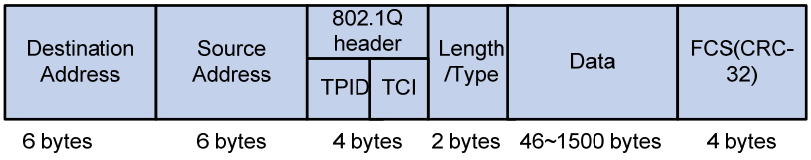
DSCP 优先级（十进制）	DSCP 优先级（二进制）	关键字
46	101110	ef
10	001010	af11
12	001100	af12

DSCP 优先级（十进制）	DSCP 优先级（二进制）	关键字
14	001110	af13
18	010010	af21
20	010100	af22
22	010110	af23
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

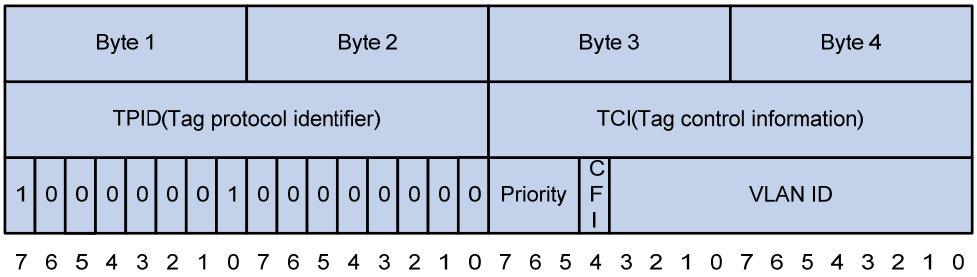


表8-4 802.1p 优先级说明

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。但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。

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）生效时间段。

每个时间段都以一个名称来标识，用户最多可创建 1024 个不同名称的时间段。一个时间段内可包含一或多个周期时间段（最多 32 个）和绝对时间段（最多 12 个），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 配置时间段

表1-1 配置时间段

操作	命令	说明
进入系统视图	**system-view**	-
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* }	缺省情况下，不存在时间段

1.3 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1-2 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** }