HCIE-R&S 面试理论之葵花宝典

欲练此功

无需自宫

数通之道

全在于此

熟读此典

IE 必成

目录

二层技术

2016 年 7 月 19 日 9:42

一，STP 与 RSTP 的区别（后 6 点可以作为 RSTP 比 STP 收敛快的原因）

1，端口角色

STP　　RSTP

DP RP blocking　　DP RP AP BP EP

2，端口状态

STP

Blocking disable listening leanring forwarding

RSTP

Discarding listening forwarding

3，Flag 位

STP

TCA　　TC

RSTP

TCA A F D pole-state P TC

4，保护机制

BPDU保护	在交换设备上，通常将直接与用户终端（如PC机）或文件服务器等非交换设备相连的端口配置为边缘端口。 正常情况下，边缘端口不会收到RST BPDU。如果有人伪造RST BPDU恶意攻击交换设备，当边缘端口接收到RST BPDU时，交换设备会自动将边缘端口设置为非边缘端口，并重新进行生成树计算，从而引起网络震荡。	交换设备上启动了BPDU保护功能后，如果边缘端口收到RST BPDU，边缘端口将被error-down，但是边缘端口属性不变，同时通知网管系统。
根保护	由于维护人员的错误配置或网络中的恶意攻击，网络中合法根桥有可能会收到优先级更高的RST BPDU，使得合法根桥失去根地位，从而引起网络拓扑结构的错误变动。这种不合法的拓扑变化，会导致原来应该通过高速链路的流量被牵引到低速链路上，造成网络拥塞。	对于启用Root保护功能的指定端口，其端口角色只能保持为指定端口。一旦启用Root保护功能的指定端口收到优先级更高的RST BPDU时，端口状态将进入Discarding状态，不再转发报文。在经过一段时间（通常为两倍的Forward Delay），如果端口一直没有再收到优先级较高的RST BPDU，端口会自动恢复到正常的Forwarding状态。 **说明：** Root保护功能只能在指定端口上配置生效。
环路保护	在运行RSTP协议的网络中，根端口和其他阻塞端口状态是依靠不断接收来自上游交换设备的RST BPDU维持。 当由于链路拥塞或者单向链路故障导致这些端口收不到来自上游交换设备的RST BPDU时，此时交换设备会重新选择根端口。原先的根端口会转变为指定端口，而原先的阻塞端口会迁移到转发状态，从而造成交换网络中可能产生环路。	在启动了环路保护功能后，如果根端口或Alternate端口长时间收不到来自上游的RST BPDU时，则向网管发出通知信息（如果是根端口则进入Discarding状态）。而阻塞端口则会一直保持在阻塞状态，不转发报文，从而不会在网络中形成环路。直到根端口或Alternate端口收到RST BPDU，端口状态才恢复正常到Forwarding状态。 **说明：** 环路保护功能只能在根端口或Alternate端口上配置生效。
防TC-BPDU攻击	交换设备在接收到TC BPDU报文后，会执行MAC地址表项和ARP表项的删除操作。如果有人伪造TC BPDU报文恶意攻击交换设备时，交换设备短时间内会收到很多TC BPDU报文，频繁的删除操作会给设备造成很大的负担，给网络的稳定带来很大隐患。	启用防TC-BPDU报文攻击功能后，在单位时间内，交换设备处理TC BPDU报文的次数可配置。如果在单位时间内，交换设备在收到TC BPDU报文数量大于配置的阈值，那么设备只会处理阈值指定的次数。对于其他超出阈值的TCN BPDU报文，定时器到期后设备只对其统一处理一次。这样可以避免频繁的删除MAC地址表项和ARP表项，从而达到保护设备的目的。

5，RSTP EP 端口

(1) 接入设备进入 Forwarding 状态，避免等待 30 秒延迟，直接转发

(2) 边缘端口 UP 时不产生 TC

(3) 生成树拓扑变化时，不会进入阻塞端口

(4) 收到 TC 不刷新 MAC 地址表

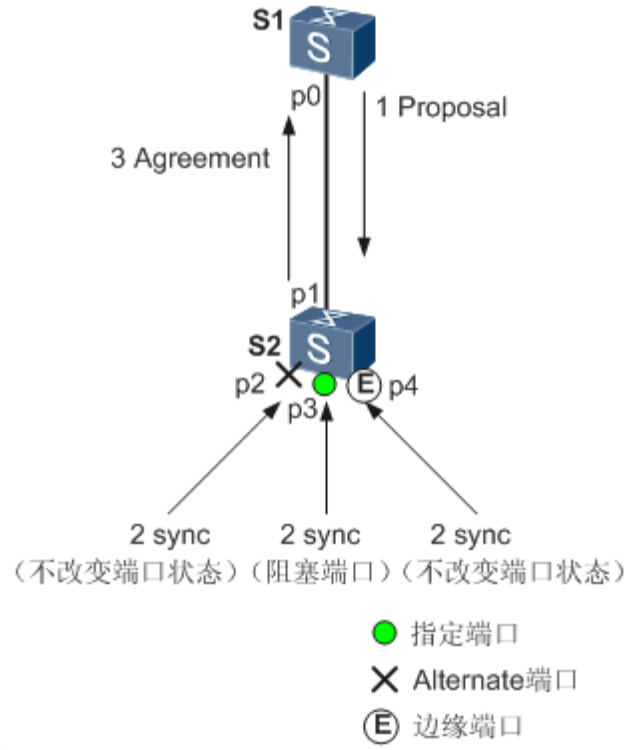
(5) 不向边缘端口转发 TC 报文

(6) 收到 BPDU 变为普通端口

6，端口快速切换机制

RSTP 中，AP 变为 RP 秒切 EP 端口没有两个 forwarding daly，直接转发

7，P/A 机制



当 SW1 和 SW2 刚通上电，如果 SW1 先向 SW2 发送 BPDU，并且 SW1 比 SW2 的 MAC 地址小，P=1 A=1 ， P=1 代表能进行 P/A 机制，这时端口状态为 DP discarding

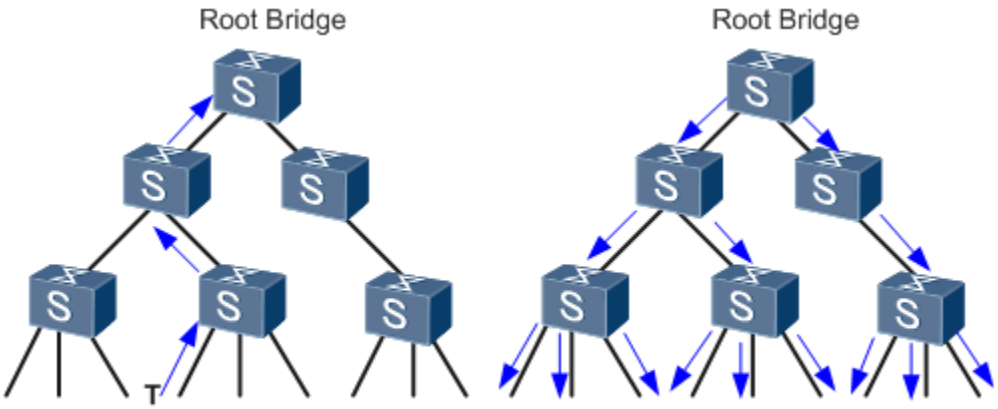
当 SW2 收到这份 BPDU，它会检查 BID，发现 SW1 发来的 BPDU 更优，SW2 的接口变为 RP discarding

在 SW2 向 SW1 发送 A=1 的 BPDU 之前，首先要进行同步（sync），除了 EP 端口和 AP/BP 端口外所有的接口变为 discarding 状态

SW2 向 SW1 发送 P=0 A=1 的 BPDU，这时端口状态变为 RP forwarding ，同步过程结束

SW1 收到 SW2 的 A=1 的 BPDU 报文，端口状态变为 DP forwarding

8，TC 处理机制



STP 拓扑变更到网络收敛一共 35 秒

络拓扑发生变化后，下游设备会不间断地向上游设备发送TCN BPDU报文。

上游设备收到下游设备发来的TCN BPDU报文后，只有指定端口处理TCN BPDU报文，其它端口也有可能收到TCN BPDU报文，但不会处理。

上游设备会把配置BPDU报文中的Flags的TCA位设置1，然后发送给下游设备，告知下游设备停止发送TCN BPDU

上游设备复制一份TCN BPDU报文，向根桥方向发送,直到根桥收到TCN BPDU

根桥把配置BPDU报文中的Flags的TC位置1后发送，通知下游设备直接删除桥MAC地址表项。 说明：TCN BPDU报文主要用来向上游设备乃至根桥通知拓扑变化。

RSTP 拓扑变更到网络收敛一共4秒

一旦检测到拓扑发生变化，为本交换设备的所有非边缘指定端口启动一个TC

为本交换设备的所有非边缘指定端口启动一个TC While Timer，该计时器值是Hello Time的两倍。

在这个时间内，清空状态发生变化的端口上学习到的MAC地址。 同时，由这些端口向外发送RST BPDU，其中TC置位。一旦TC While Timer超时，则停止发送RST BPDU。

其他交换设备接收到RST BPDU后，清空所有端口学习到MAC地址，除了收到RST BPDU的端口。然后也为自己所有的非边缘指定端口和根端口启动TC While Timer，重复上述过程。 如此，网络中就会产生RST BPDU的泛洪

9，收到次级 BPDU 会立即回复优质 BPDU

无论是 STP 还是 RSTP，收到次级 BPDU 会立即回复优质 BPDU

10，Timer

hello time:2s max age:20s (RSTP 的 max age:18s) message age:20 forwarding delay :15s

二，STP 与 RSTP 的基础

1，报文格式及类型

BPDU 消息字段	BPDU 消息说明
Protocol Identifier	总是为 0
Protocol Version Identifier	总是为 0
BPDU Type	BPDU 类型
Flags	标识位
Root Identifier	当前根桥的 BID
Root Path Cost	本端口累计到根桥的开销
Bridge Identifier	本交换设备的 BID
Port Identifier	发送该 BPDU 的端口 ID
Message Age	该 BPDU 的消息年龄
Max Age	消息老化年龄
Hello Time	发送两个相邻 BPDU 的时间间隔
Forward Delay	控制 Listening 和 Learning 状态的持续时间

STP 与 RSTP 的 BPDU 报文唯一的不同就是 Flag 置位不同

STP 的报文类型 0x00　配置 BPDU

0x80　TCN 的 BPDU

RSTP 的报文类型 0x02　RSTP 的 BPDU

2，STP 树的构建过程

A，RID

B，到根桥的 cost

C，发送者的 BID(BID 格式)

D，发送者的 PID（PID 格式）

E，自己的 PID

三，MSTP

1，为什么需要使用 MSTP

MSTP 的作用主要是用来负载，多域与多实例的负载，不会浪费链路

2，同一个域的条件

A，region name

B，region level

C，实例中的 vlan 映射

2，CST

公共生成树，在多域时域间的树

3， IST

内部生成树，在每个域内除了被阻塞的链路所构成的树

4， CIST

CIST 就是 CST+IST 的树

5，总根、域根

全网中 BID 最小的为总根，其他域中，离根最近的交换机为域根

四，Smart link

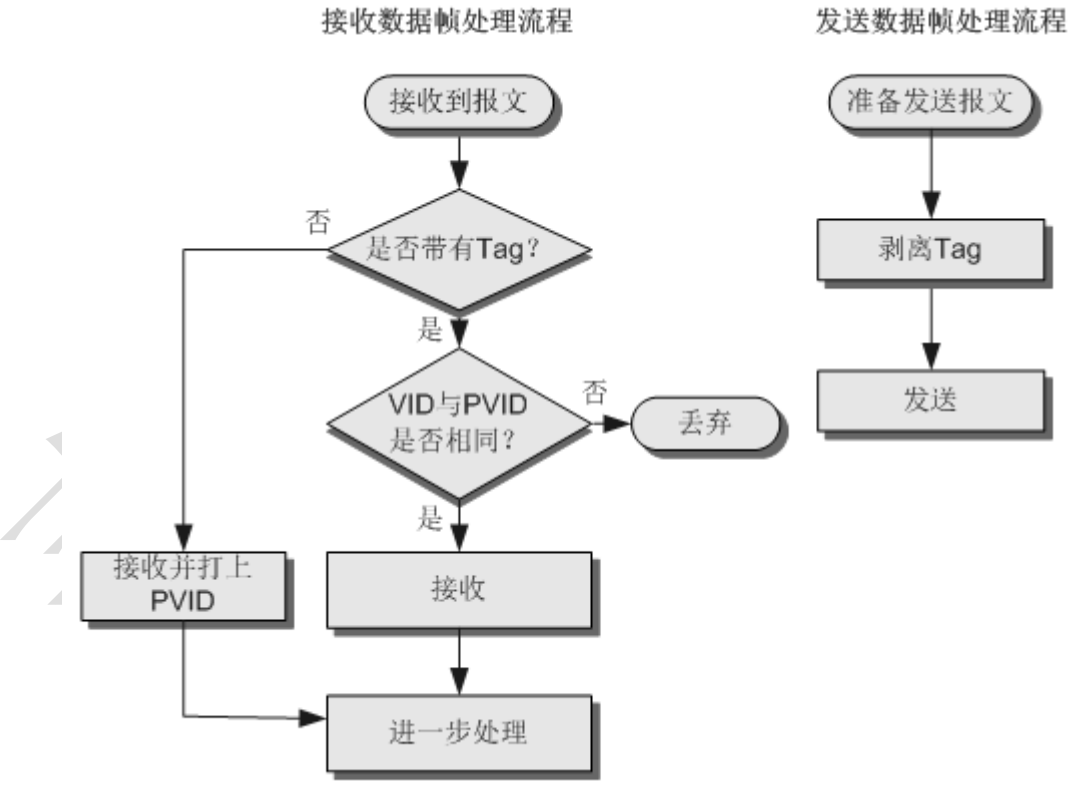
1，作用：华为私有，解决二层环路问题。

2， 原理：一台设备的两个端口启用一个 smart link 组，一个为 master 端口，一个为 slave 端口，两个端口同时只有一个端口在转发，一般为 master 端口在转发，slave 端口保持阻塞状态，当 master 端口 down 了之后，slave 端口变为转发状态

五，交换机的端口特性

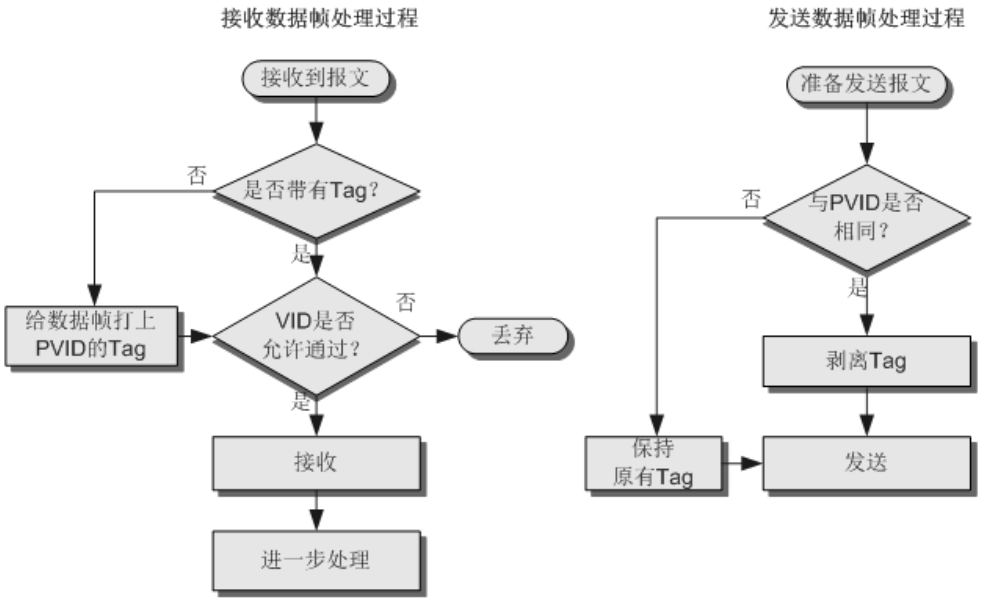
1， Access

Access 接口添加或剥除 VLAN 标签的处理过程



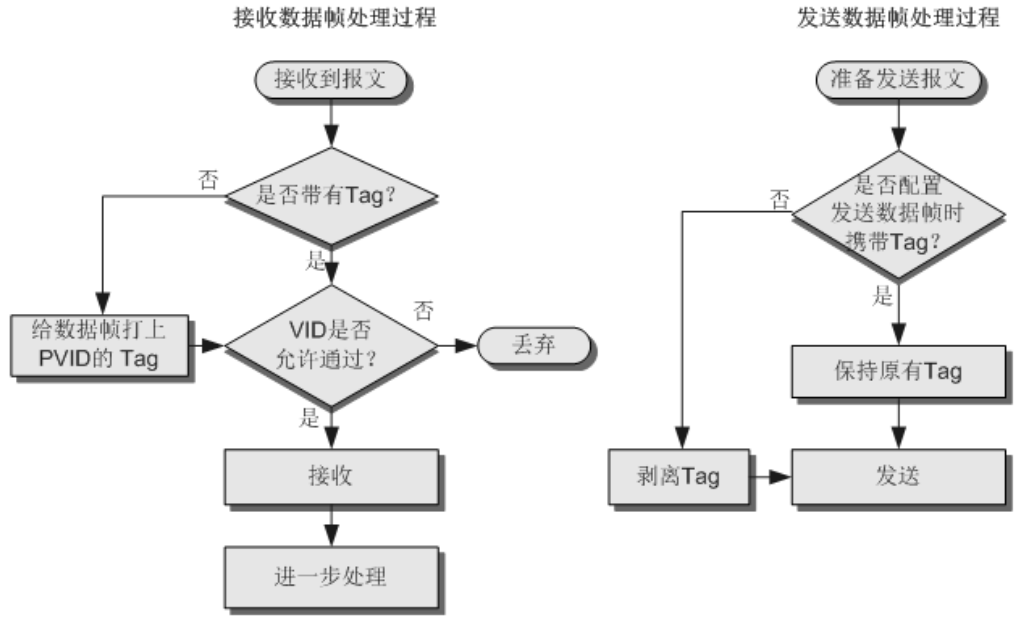
2， trunk

Trunk 接口添加或剥除 VLAN 标签的处理过程



3， hybrid

Hybrid 接口添加或剥除 VLAN 标签的处理过程



4，什么时候必须使用 Hybrid

当一个交换机端口属于不同 Vlan，一个为普通 VLAN，一个为 VOICE VLAN

基于 PC 端的 MAC 地址动态分配 VLAN

六，QinQ

1，作用：使私网 VLAN 透传公网，

A，扩展 VLAN，对用户进行隔离和标识不再受到限制。

B，QinQ 内外层标签可以代表不同的信息，如内层标签代表用户，外层标签代表业务，更利于业务的部署。

C，QinQ 封装、终结的方式很丰富，帮助运营商实现业务精细化运营。

2，原理 ：打上双层标签，使私网的 VLAN 透传到公网，外层放的是公网的标签，内层放的是私网的标签，在公网上剥离公网标签，两边设备就可以跨公网互访了

七，FR

1，工作过程：

2，LMI 的作用：获取 DLCI 号，维护 PVC 状态

3，InARP 的作用：把 DLCI 号解析成 IP 地址

八，Mux Vlan

1，作用：实现 Vlan 之间的隔离

2，原理 ：mux vlan 分为主 VLAN 和 从 VLAN，从 VLAN 又分为隔离 VLAN 和互通 VLAN，一台设备只能有一个隔离 VLAN，隔离 VLAN 不能和其他 VLAN 互访，通设备的互通 VLAN 间可以互访，但所有的隔离 VLAN 和互通 VLAN 都必须绑定一个主 VLAN。

九，Super Vlan（聚合 VLAN）

1， 作用：节省 IP 地址

2，原理：启用 super vlan 后，不同的 VLAN 可以共用一个网关 IP，正常的 VLAN，一个 VLAN 就需要一个网关 IP。

3，Super vlan 与 Mux vlan 的区别

Super VLAN 又称 VLAN aggregation，该技术涉及 Sub VLAN 和 super VLAN 的概念。Super VLAN 和通常意义上的 VLAN 不同，它是一种只能包含 Sub VLAN，不包含物理端口的 VLAN。Super VLAN 包含的所有 Sub VLAN 共用 Super VLAN 三层接口地址与上层通信。Sub VLAN 的类型可以是 smart VLAN 或 MUX VLAN，当这些 VLAN 加入 super VLAN 后就称为 Sub VLAN。Sub VLAN 只包含物理端口，不能建立三层 VLAN 虚接口。

MUX VLAN 是一种包含上行端口和业务虚端口的 VLAN。一个 MUX VLAN 可包含多个上行端口，但只包含一个业务虚端口，不同 MUX VLAN 间的业务流相互隔离。MUX VLAN 与接入用户存在一对一的映射关系，因此可根据 MUX VLAN 区分不同的接入用户。

从应用上来说：

MUX VLAN 一般用于 DSLAM 接入设备,每个用户一个 VLAN，并与上网 PVC 关联起来(不同于普通 VLAN 关联到以太网口)，到上层汇聚设备再加上外层 VLAN，可以唯一区分上网用户。

super vlan 用于 SR 设备，解决共用网关问题，即不同 VLAN 用户可以使用相同网关 ip，从而节省 IP 地址资源。

十， ARP（需要知道报文的封装）

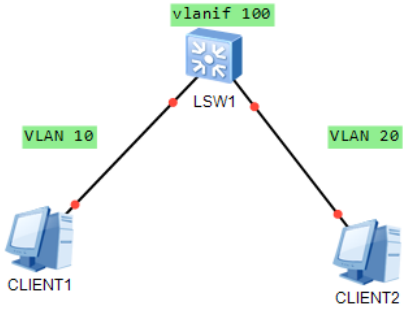
1， 正向 ARP：三层映射二层 二层广播封装，没有三层

```
⊞ Ethernet II, Src: Cisco_ea:b8:c1 (00:19:06:ea:b8:c1), Dst: Broadcast (ff:ff:ff:ff:ff:ff)
⊞ 802.1Q Virtual LAN, PRI: 0, CFI: 0, ID: 123
⊟ Address Resolution Protocol (request)
    Hardware type: Ethernet (0x0001)
    Protocol type: IP (0x0800)
    Hardware size: 6
    Protocol size: 4
    Opcode: request (0x0001)
    [Is gratuitous: False]
    Sender MAC address: Cisco_ea:b8:c1 (00:19:06:ea:b8:c1)
    Sender IP address: 192.168.123.1 (192.168.123.1)
    Target MAC address: 00:00:00_00:00:00 (00:00:00:00:00:00)
    Target IP address: 192.168.123.2 (192.168.123.2)
```

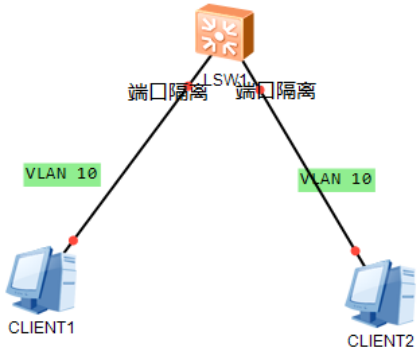
2，反向 ARP：把 MAC 地址解析成 IP 地址，无盘工作站

3，代理 ARP：当两台 PC 想要互相访问，两台 PC 首先要在同一个网段，但不在同一个广播域时就需要代理 ARP

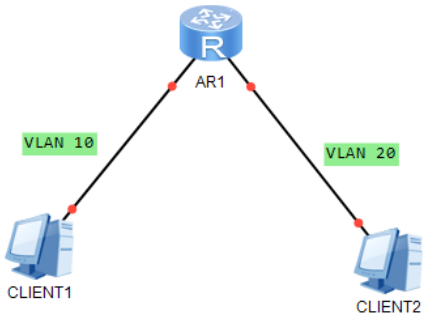
A，vlan 间代理：super vlan



B，vlan 内代理：同处于一个广播域，但配置了二层隔离，双方都需要通过网关进行访问，双方封装的目的 MAC 也是网关的 MAC 地址



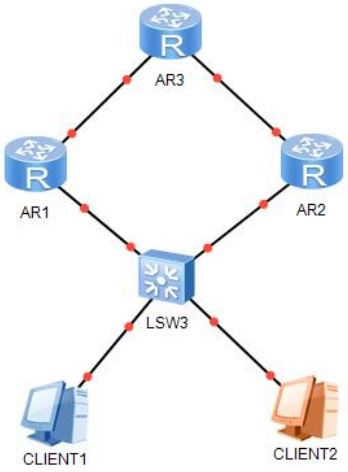
C，路由式代理：PC1 想要访问 PC2，PC1 首先会判断 PC2 的 IP 地址是否和自己在同一个广播域，如果在同一个广播域，那么 PC1 就会向中间的路由器发送 ARP 的请求，当路由器的接口开启了代理 ARP 的功能后，当收到一个 PC 请求的 ARP 后，会查自己的路由表中的路由是否可达，如果可达，路由器会把自己的 MAC 地址回复给发送 ARP 的 PC。如果 PC1 和 PC2 不在同一个广播域内，则无法通信



4，免费 ARP（DAD, vrrp 主备切换）

5，如何使用代理 ARP 来实现 VRRP 的功能

在 R1 和 R2 上都启用代理 ARP 的功能，R1 和 R2 都会回复 ARP 请求，如果 PC1 和 PC2 想要访问 R3，当 R1 设备 DOWN 了之后，也可以通过 R2 继续访问。



十一，MAC 地址漂移

MAC 地址漂移即设备上一个接口学习到的 MAC 地址在同一 VLAN 中另一个接口上也学习到，后学习到的 MAC 地
址表项覆盖原来的表项。正常情况下，网络中不会在短时间内出现大量 MAC 地址漂移的情况。出现这种现象一般
都意味着网络中出现环路，形成广播风暴。处于风暴影响中的每个交换机节点都有 MAC 地址漂移的现象。因此，
可以利用该现象来监控网络中是否成环。

场景：

①存在二层环路，解决方法：STP

②存在攻击者，解决方法：利用 DHCP Snooping 绑定表实现 IPSG 功能

③正常用户来回移动，解决方法与②相同

VRRP 准备频繁切换

相关配置：

①为端口配置静态 MAC

②为接口配置 MAC 地址学习优先级

③禁止相同的优先级的接口发生 MAC 地址漂移

IPSG 是 IP Source Guard 的简称。IPSG 可以防范针对源 IP 地址进行欺骗的攻击行为。随着网络规模越来越大，基于

源 IP 的攻击也逐渐增多。一些攻击者利用欺骗的手段获取到网络资源，取得合法使用网络资源的权限，甚至造成

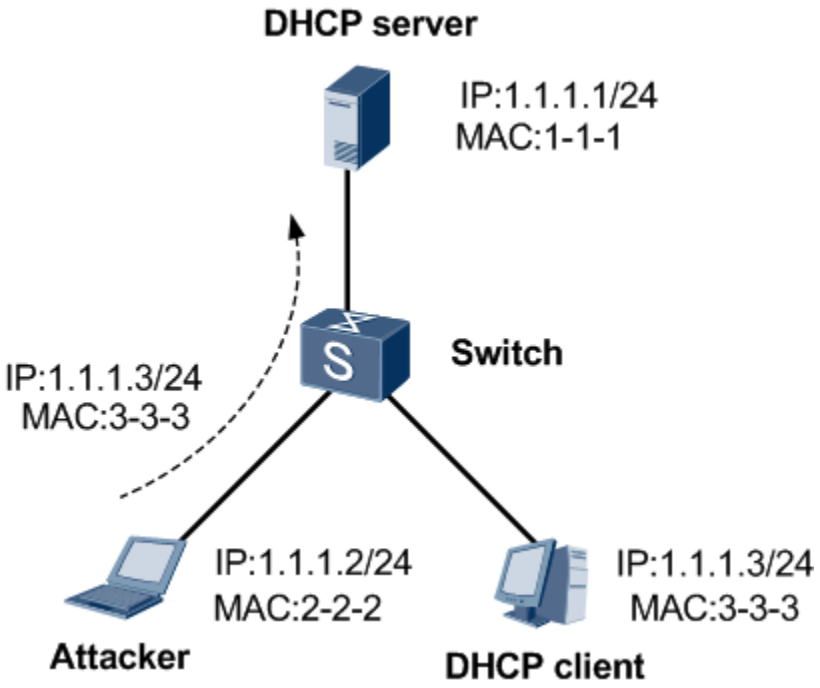
被欺骗者无法访问网络，或者信息泄露。IPSG 针对基于源 IP 的攻击提供了一种防御机制，可以有效的防止基于源

地址欺骗的网络攻击行为。

IPSG 功能是基于绑定表（DHCP 动态和静态绑定表）对 IP 报文进行匹配检查。当设备在转发 IP 报文时，将此 IP 报文中

的源 IP、源 MAC、接口、VLAN 信息和绑定表的信息进行比较，如果信息匹配，表明是合法用户，则允许此报文正

常转发，否则认为是攻击报文，并丢弃该 IP 报文。



如图所示，攻击者伪造合法用户报文，篡改了 Router 上 MAC 表的出接口信息，使服务器回复的报文被发送给攻击

者。

为了防止此类攻击，可以在 Router 上配置 IPSG 功能，对进入接口的 IP 报文进行绑定表匹配检查，合法用户发送报文

的信息和绑定表一致，允许其通过；攻击者伪造的报文信息和绑定表不一致，Router 将报文丢弃。

配置 MAC 地址漂移检测

执行命令 system- - view，进入系统视图。

执行命令 vlan vlan-id ，创建 VLAN 并进入 VLAN 视图。

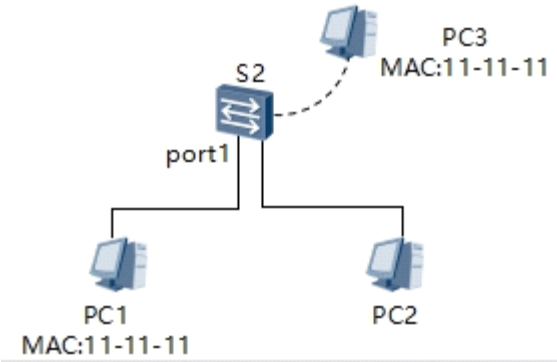
执行命令 loop- - detect eth- -p loop { [block- -c mac] block- - time block-time retry- - times retry-times |
alarm- -y only }，配置 MAC 地址漂移检测功能。

有的 MAC 地址是否发生了漂移。若发生漂移，设备会上报告警到网管系统。

当系统检测到 VLAN 内有 MAC 地址发生漂移时，可以进行以下处理动作：

①接口阻断或 MAC 地址阻断。当检测到 MAC 地址发生漂移则执行接口阻断或 MAC 地址阻断动作。当指定 block-mac
参数时，将不阻断整个接口，而是按照发生漂移的 MAC 地址进行流量阻断。

②发送告警。当检测到 MAC 地址发生漂移时只给网管发送告警。



拓扑描述：为防止非法用户 PC3 伪造 PC1 MAC 地址入侵 Switch，可以提高服务器侧接口 Port1 的 MAC 地址学习优先级。

十二，二层环路与三层环路的区别

环路的原因：二层环路是由于物理拓扑出现环路，如 3 台交换机 3 角形连接。三层环路一般物理拓扑有环路，并且设备之间路由表形成互指。(物理拓扑不成环，2 台设备使用静态路由互指也可
能成环，这种特殊情况除外)。

二层设备和三层设备的工作行为区别：

二层交换机工作行为：收到的数据帧查看 2 层头部，根据目的 MAC 地址转发，目的 MAC 分广播，组播，单播

广播：目的 MAC 为全 F。收到广播报文，除了接收的端口外，向其余所有端口转发(泛洪)

组播：目的 MAC 的第 8 位为 1。收到组播报文，首先判断目的 MAC 是否本机要接受，如收到 STP 的 BPDU，而自身也
运行 STP，此报文上送 CPU 处理，不做转发。假如此报文自身不需要接受，则处理方式为泛洪。
单播：目的 MAC 的第 8 位为 0。收到单播报文，如果目的 MAC 在自身 MAC 表中不存在，则称为未知单播，处理方式
为泛洪。假如目的 MAC 在自身 MAC 表中存在，则称为已知单播，把报文向 MAC 表中的接口转发(如该接口等于报文的
接收端口，则报文丢弃)

三层设备工作行为：收到数据包查看三层目的 IP，根据目的 IP 地址转发，分为广播，组播，单播。
广播：目的 IP 为全 1。收到广播包，上送 CPU 处理(注意不是丢弃报文)，三层设备是隔离广播域，不是丢弃广播报
文。
组播：目的 IP 为 224.0.0.0-239.0.0.0。开启组播路由协议则转发，否则丢弃。
单播：目的 IP 在路由表中存在则按出端口转发，目的 IP 在路由表中不存在则丢弃。

环路的影响：
二层环路：广播风暴和数据帧复制，MAC 地址震荡；假设交换机收到广播帧或者组播帧或者未知单播帧，会采用
泛洪形式处理，数据帧在转发时候产生了拷贝复制，数据帧无休止被转发，如此往复，最终导致整个网络带宽资源
被耗尽，设备负载过大，网络瘫痪不可用。(此现象极易产生)
三层环路：数据包会在设备之间有限的互相转发，因为在三层 IP 头部存在 TTL 字段所以报文不会无休止转发。

防环机制：
二层防环：STP，SMART-LINK 等技术，或使用 LACP 链路捆绑和设备堆叠等技术，使得物理拓扑上没有环路。
三层防环：只要依靠路由协议自身的防环机制。
静态路由，依靠人工预防
RIP：参见 RIP 防环机制，16 跳，水平分割，毒性逆转，触发更新。
OSPF：参见 OSPF 章节，区域内依靠 SPF 算法，区域间依靠区域结构设计和 ABR 的水平分割原则。
ISIS：区域内依靠 SPF 算法，区域间依靠路由泄漏的 DOWN 位。

BGP：AS 之间依靠 AS 号，AS 内部只传一跳，如使用路由反射器依靠簇 LIST 和起源 ID，使用联盟依靠联盟的私有 AS 号。
组播：参见组播章节，依靠 PRF 检查。
转发层面：二层环路无防环机制，三层环路有 TTL 机制。

总结:
二层环路较易产生，需要运行破环机制经过计算阻塞某些端口实现预防，且由于二层设备的处理行为导致了后果
特别严重。
三层环路不容易产生，由于三层设备的处理行为及 TTL 机制，所以后果并不十分严重。且每种路由协议都有比较完
善的防环机制，三层环路比较容易发生在特殊的场景下，如双点双向路由发布。具体见 LAB 题：双点双向。

十四，三层交换机与路由器的区别
八字真言
交换谋快，路由谋转。

三层交换机和路由器的区别
网络在规模和速度方面都在急剧发展，为满足传统路由器所不具备的高速数据传输和简化复杂网
络的需求，基于这种情况三层交换机应运而生。

1、硬件上的区别，三层交换机是通过交换芯片转发数据的，路由器则是通过 cpu 转发数据的。所
以三层交换机在网络收敛上慢于路由器并且抵抗网络震荡的能力也弱。

2、数据的处理方式的区别，三层交换机的首包通过 cpu 转发，一次路由多次交换，同时通过 arp 协议建立交换芯片硬件
表项（mac 地址与 ip 地址的映射表），后续报文通过交换芯片直接硬件转发，即；“一次路由，多
次交换”，交换机的硬件三层表项中只包含了目的地址、目的 ip(或下一跳 ip)对应的 mac 地址、出口
vlan 及端口。路由器通过路由表选择路由后，路由表将激活路由下发到 FIB（转发信息表）表中，数
据在到达路由器时，通过 FIB 表的最长匹配原则查表转发。 所以三层交换机在数据转发上的速度要

优于路由器。

3、功能的上的区别，路由器提供包括分组过滤，分组转发，优先级，复用，加密，压缩和防火墙
等功能，并且接口类型丰富，支持的三层功能强大、支持负载分担、链路备份、nat 转换、及其其
他网络进行路由信息的交换，路由器在大型网络中的协议计算，路由表大小，收敛时间等都优于
三层交换机。

三层交换机的优点在于可以加快局域网内数据的交换，并且加入路由功能也是为这个目的服务
的。由于局域网内大量的网际互访，单纯使用二层交换机不能实现不同网络间的互访，如单纯使
用路由器，由于接口限制和路由转发效率低，将限制网络速率和网络规模，采用具有路由功能的
三层交换机就成为首选。
三层交换机的主要用途是代替传统路由器作为网络的核心。所以在没有广域网连接的需求，同时
需要路由的地方，都可以运用三层交换机。在局域网中，一般将三层交换机运用于网络的核心层
和汇聚层。

路由技术-IGP

2016 年 7 月 19 日
10:24

一，RIP

RIP 为 DV 协议，工作在 UDP 上，端口号 520，没有邻居概念，只有两种报文：request 和 response

1，防环机制

A，计数到无穷大(16 跳)：RIP 为矢量路由协议，在路由的出方向增加 cost 值，最大为 16，大于等于 16 最丢弃

B，水平分割：当从一个接口收到的路由，不会从这个接口再发回去

C，触发更新(没有触发更新时出环路的场景)：当有新的路由或者路由消失时，路由器会立刻把路由更新发送出去

D，毒性逆转：毒性逆转优于水平分割，当收到路由时，会把这条路由置位 16 跳向对方回复，充当确认机制

E，路由毒化：当收到路由的跳数为 16 时，会认为这条路由不可达，立刻删除路由表中对应的条目

2，毒性逆转与水平分割的区别：毒性逆转有 16 跳作为回复，水平分割没有回复，但毒性逆转的开销更大

3，毒性逆转与路由毒化的区别：毒性逆转和毒化路由并没有可比性，一个是确认的回复，一个是路由撤销

4，RIP 的计时器

A，更新计时器 30s：RIP 每 30 秒把路由表中所有的路由向外发送，开销大

B，老化计时器 180s：当路由器收不到 30 秒的路由更新时，会最多等待 180 秒，180 秒内如果没有收到路由更新，则删除路由表中条目

C，垃圾收集计时器 120s（作用？）：当删除路由表中的条目时，会在数据库中存在 120 秒，向下游发送 16 跳的路由撤销，告诉其它路由器路由不可达。作用就是使 RIP 网络快速收敛

D，抑制计时器（华为没有这个计时器，作用是防止路由抖动）

5，RIP 收敛慢的场景（180s）：

，

非直连链路 DOWN，需要等待 180 秒才能收敛

6，RIP v1 与 V2 的区别

A，V1 有类（更新无掩码，边界自动汇总）；V2 无类

V1 不支持 VLSM，V2 支持 VLSM

B，V1 无认证；V2 有认证(如何携带认证，报文大小)：V2 的认证占用路由条目，明文占用一个路由条目，MD5 占用两个路由条目，RIPV2 最大字节数为 512，UDP 头占用 8 个字节，RIP 头占用 4 个字节，剩余 500 字节，一个路由条目占用 20 个字节，所有一个 RIP 报文最大存放 25 条路由

C，V1 广播更新；V2 组播更新(224.0.0.9)（什么时候会使用单播更新）：当接口被 silent，并且指 peer 时，才能进行单播更新

D，V1 无 tag；V2 携带 tag（作用）：对路由进行标记，实现更加精确的路由控制

E，V1 无 next-hop；V2 携带 nexp-hop（场景三个）：解决次优问题。携带 TAG 条件，当一个接口收到的这条路由又要从这个接口发出去时，一般会携带 tag

E， RIP V1 与 V2 收发报文的区别及互操作。（默认是哪个版本）

	收	发
V1	V1 广播	V1 广播
V2	V2 的组播和广播	V2 的组播
V2 广播	V1 和 V2 的组播，广播	V2 的广播
V1 兼容	V1 和 V2 的组播和广播	V1 的广播

F， RIP V1 与 V2 的路由收发规则（默认是哪个版本）

RIPv2 默认关闭路由汇总

由于 RIP ver 1 的路由条目中并不包含掩码长度，所以也就并不知道网络位是哪部分，主机位又是哪部分，因此，如果收到的路由与接收接口不属于同一主类，则一律使用主类地址来检测，但如果收到的路由与接收接口属于同一主类，则以该接口 IP 地址的掩码长度来检测，最后计算出是否是主机地址，如果是，就以 32 位的掩码存放在路由表中。

RIPv1 的收发规则

发送规则

1. 主网边界自动汇总
2. 同一主网，发送路由和出接口的掩码不一致，华为自动汇总后发送
3. 同一主网，发送路由和出接口的掩码一致，不做任何汇总发送

接收规则：

1. 接收到的路由和接收接口属于不同主网，直接匹配主网掩码。
2. 收到的路由和接口接口属于同一主网，依次匹配 1.主网掩码 2.接口掩码 3.位主机路由(32 位)

G，RIP V2 与 RIPng 的区别

为了实现在 IPv6 网络中应用，RIPng 对原有的 RIP 协议进行了修改：

1 RIPng 使用 UDP 的 521 端口（RIP 使用 520 端口）发送和接收路由信息。

2 RIPng 的目的地址使用 128 比特的前缀长度（掩码长度）。

3 RIPng 使用 128 比特的 IPv6 地址作为下一跳地址。

4 RIPng 使用链路本地地址 FE80::/10 作为源地址发送 RIPng 路由信息更新报文。

5 RIPng 使用组播方式周期性地发送路由信息，并使用 FF02::9 作为链路本地范围内的路由器组播地址。

RIPng 报文由头部（Header）和多个路由表项 RTEs（Route Table Entry）组成。在同一个 RIPng 报文中，RTE 的最大数目根据接口的 MTU 值来确定。

6 RIPng 不支持认证，使用 IPV6 的认证

二，OSPF

1， 详述 OSPF 邻接关系建立过程

OSPF 共有 8 种状态机，分别是：Down、Attempt、Init、2-way、Exstart、Exchange、Loading、Full。

Down： 邻居会话的初始阶段，表明没有在邻居失效时间间隔内收到来自邻居路由器的 Hello 数据包。

Attempt： 该状态仅发生在 NBMA 网络中，表明对端在邻居失效时间间隔（dead interval）超时后仍然没有回复 Hello 报文。此时路由器依然每发送轮询 Hello 报文的时间间隔（poll interval）向对端发送 Hello 报文。

Init： 收到 Hello 报文后状态为 Init。

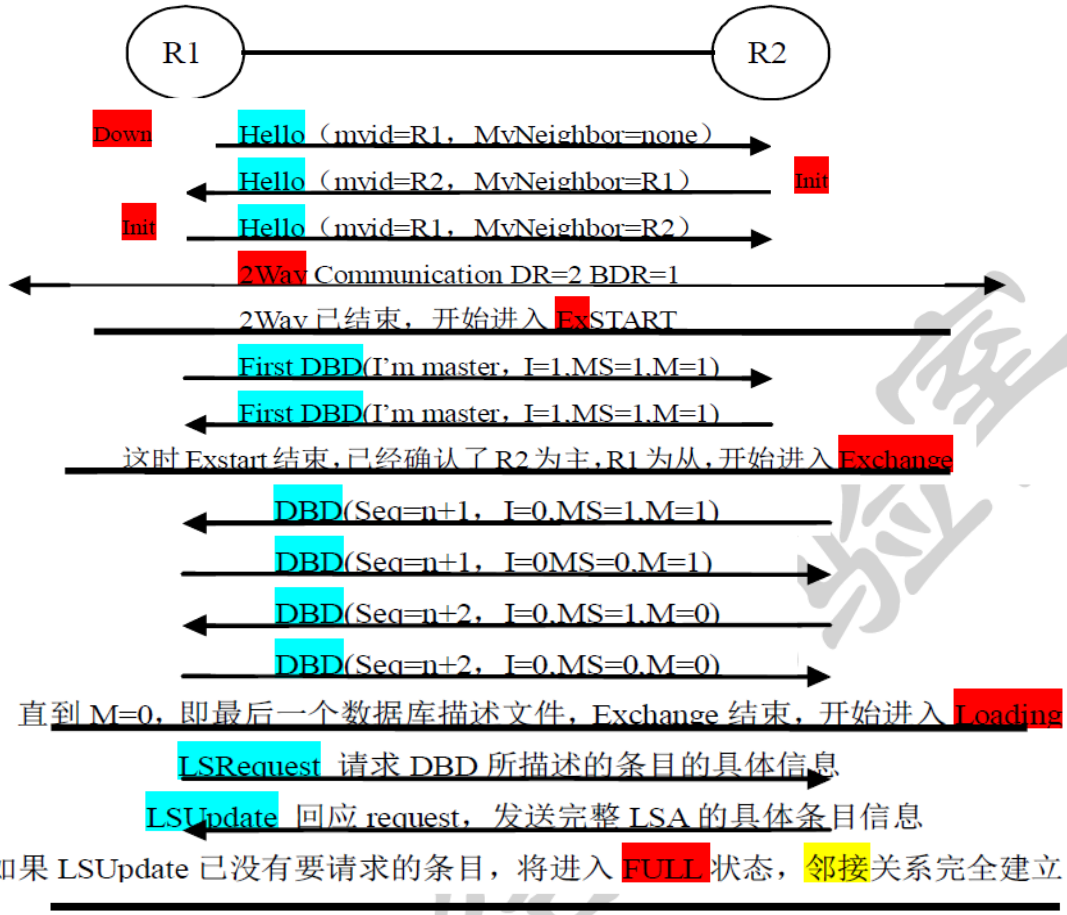
2-way： 收到的 Hello 报文中包含有自己的 Router ID，则状态为 2-way；如果不形成邻接关系则邻居状态机就停留在此状态，否则进入 Exstart 状态。

Exstart： 如果形成邻居关系，则从 Init 状态转到 Exstart 状态，开始协商主从关系，并确定 DD 的序列号。

Exchange： 主从关系协商完毕后开始交换 DD 报文，此时状态为 Exchange。

Loading： DD 报文交换完成即 Exchange done，此时状态为 Loading。

Full： LSR 重传列表为空，此时状态为 Full。



A， one-way 是什么状态，如何进入 two-way

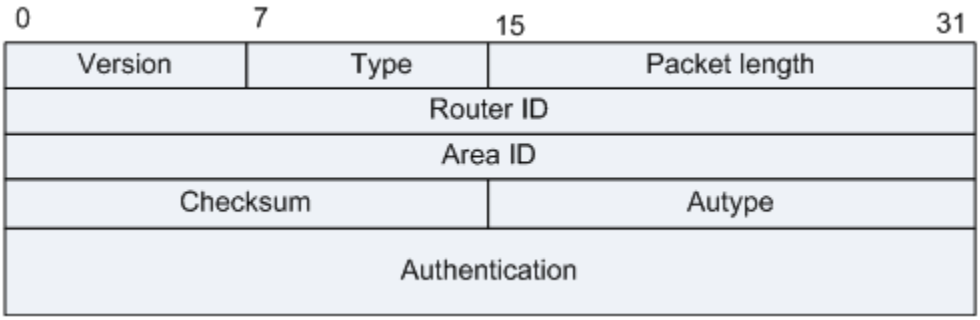
当收到一个 HELLO 包中，没有包含自己的 router id，这时为 one-way，
当收到的 HELLO 包中包含自己的 router id 则为 two-way

B，DR 的选举过程及时间：首先所有的路由器都会成为 DR others，先选举 BDR，BDR 发现没有 DR，自动成为 DR，DR others 发现没有 BDR，再选举出来一个 BDR。DR 的选举时间等于 Dead time，只有在广播网络中才会选举 DR。

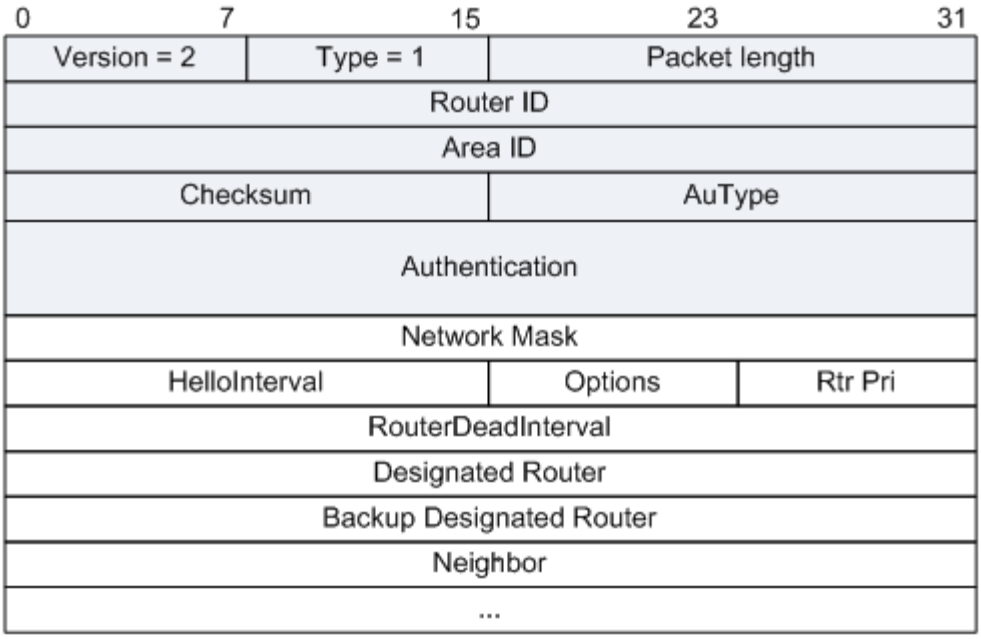
C，first DD 与 DD 的区别：首先 first DD 报文是空的，作用是选举出主从，并且统一一个序列号，保证接下来的同步过程有序可靠。DD 报文的作用是双方发送自己的 LSA 信息，并同步

D，如何选举主从，选举主从的作用：选择 router id 大的为主，接下来交互的 DD 报文统一序列号，保证同步数据库的有序与可靠性

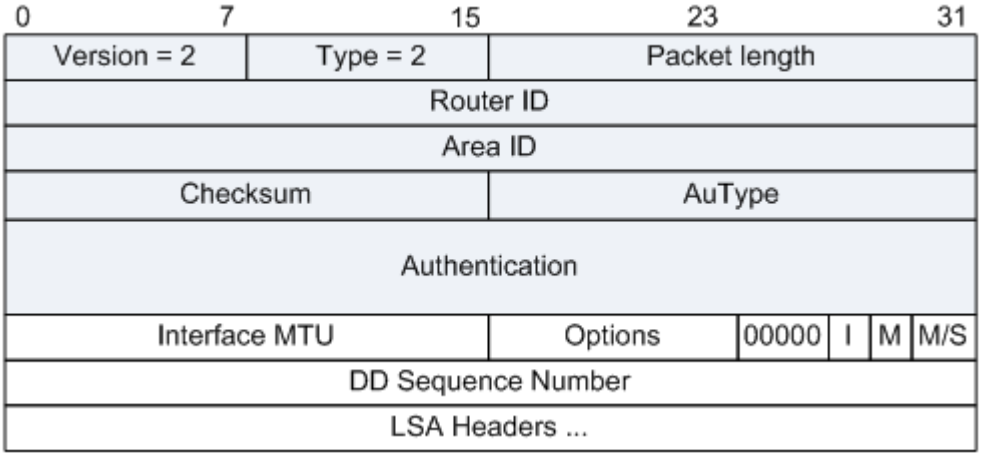
2，OSPF 的报文（需要记住头部格式）



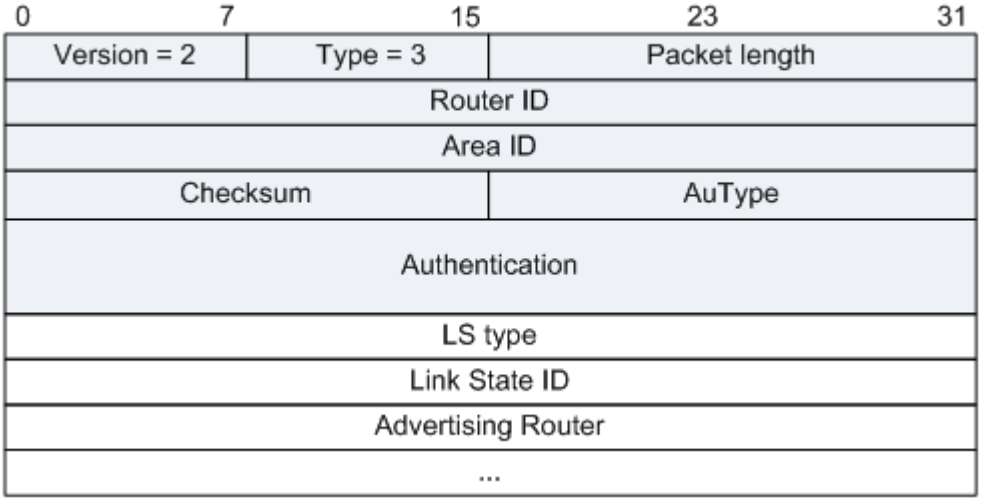
A，Hello（各种链接类型的发送间隔）：建立和维持邻居关系



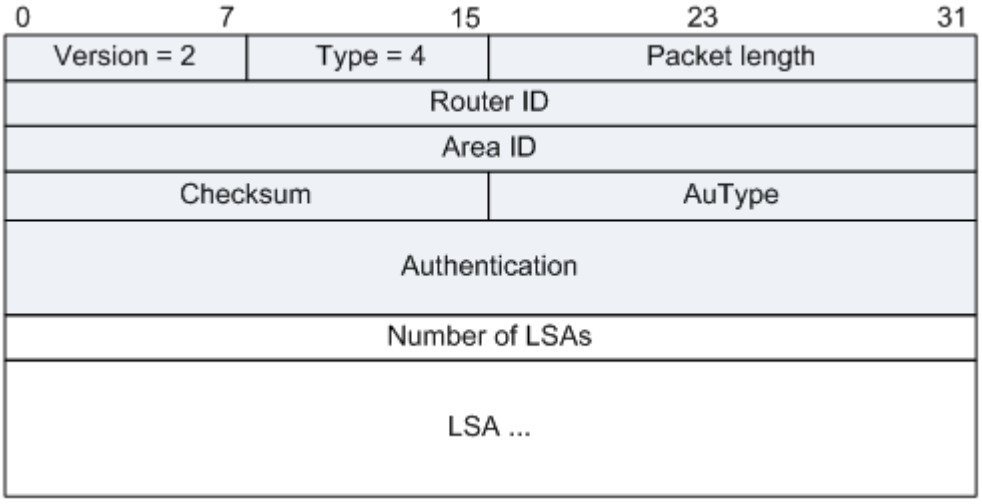
B，DBD（重传时间 5s）：DBD 分为 firstDBD 和 DBD <1>firstDBD 不携带 Lsa 头部信息。通过 firstDBD 确认主从关系。主的作用只是为了控制序列号的同步。Router-ID 高的将成为主。<2>DBD 只携带 LAS 的头部信息，没有携带 LAS 的具体信息。承载完整 LAS 是 LASUpdate 包。



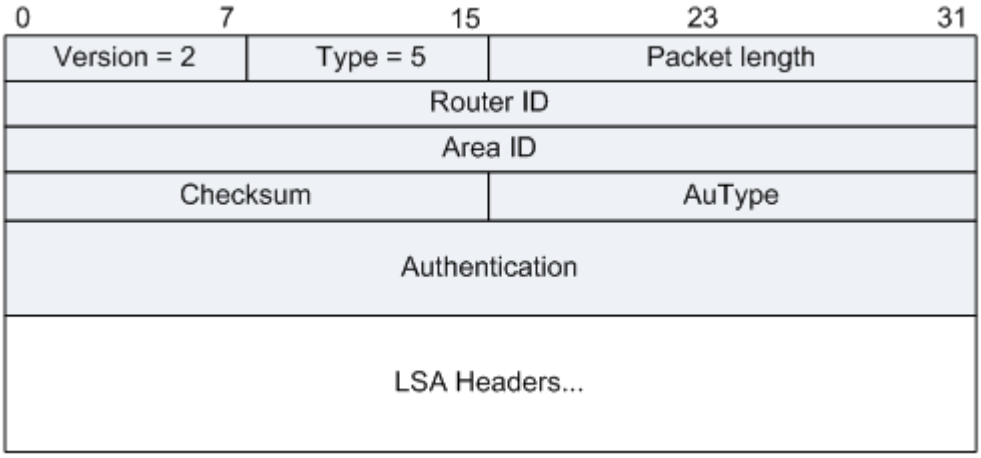
C， LSR ： 含有真正 LSA 完整信息的，用来回应 LSRequest。



D，LSU（重传时间 5S）：是不携带 LAS 头部的，只通过（公告 ID，LSA L 类型，linkID）来请求具体的条目。



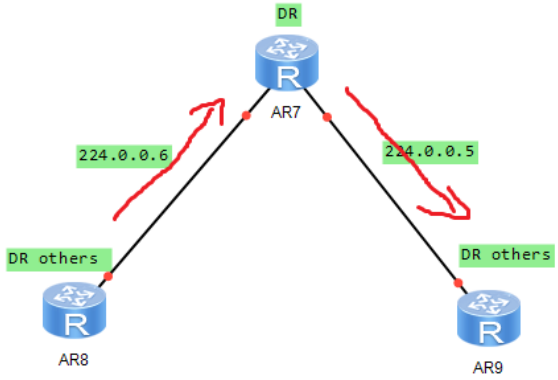
E， LSA ： 对 LSU 的确认



3，除了 LSA 的确认机制外，针对发送的 LSU 还有没有其它确认机制。

DR 可以进行隐式确认，当 DR 收到 DR others 的 LSU 时，不需要回复 LSACK，因为 DR 会向其他的 DR others 发送 LSU 的更新，这就进行了隐式确认

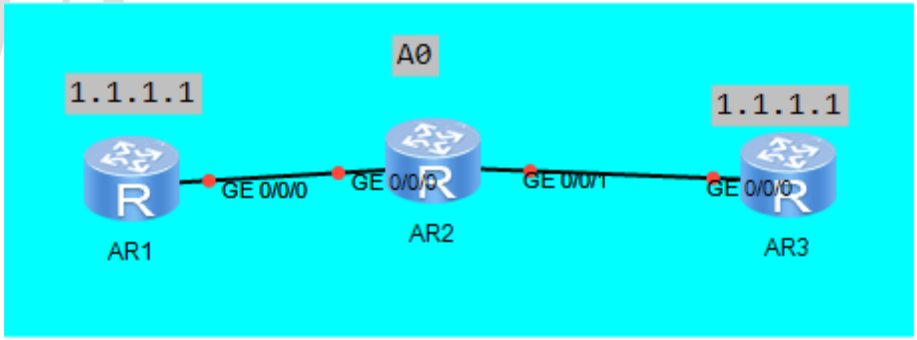
（DR others 向 DR 发送更新为 224.0.0.6，DR 向 DR others 发送的更新地址为 224.0.0.5）



4，影响 OSPF 邻接关系建立的因素(10 条)

A，Route-ID（Route-ID 冲突导致的问题）：

在同一区域内：

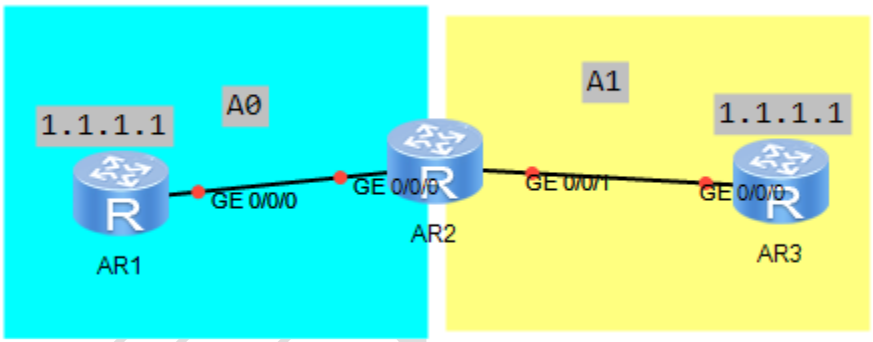


R1 和 R2 及 R2 和 R3 都可以正常建立邻居，同步数据库的时候就会出现问题，R2 的 lsdb 中，adv 为 1.1.1.1 的 lsa（LSA1 和 LSA2）只有一份。 路由计算会出现问题，假设 R1 宣告(network)一条路由 10.10.10.0/24，R1 会把这条 LSA（adv=1.1.1.1，type=1，LS

ID=1.1.1.1，seq=8001）发送给 R2，R2 收到后会发给他的邻居 R3，R3 收到发现通告者是 1.1.1.1，但是自己又没有这个网段，于是会给 R2 发送一个自己的 LSA1（age=1s，seq=8002），R2 收到后会与之前 adv=1.1.1.1 的 LSA1 进行比较，选择这条 seq 更大的 LSA1，然后也会转发给 R1，R1 收到后发现自己有这个网段，又会发送一条新的 LSA1（seq=8003），会一直出现这样重复的情况，而导致路由动荡 假设 R1 引入一条路由 10.10.10.0/24，R1 会把这条 LSA（adv=1.1.1.1，type=5，LS ID=1.1.1.1，seq=8001）发送给 R2，R2 收到后会发给他的邻居 R3，R3 收到发现通告者是 1.1.1.1，但是自己又没有这个网段，于是会给 R2 发送一个（age=3600s，seq=8001）的 LSA5，R2 收到后，会与之前收到的 LSA5 进行比较，因为 seq 和 check sum 与之前的一样，所以会优选 age=3600s 的，然后也会转发给 R1，R1 收到后发现自己有这个网段，又会发送一条新的 LSA5（seq=8002），会一直出现这样重复的情况，而导致路由动荡。

实验现象：R2 有时候有路由，有时候没路由，在一段时间后，有一台会自己修改 router-id

在不同区域：



邻居关系 ok，区域内及区域间路由能学到进路由表. 如果 R1 和 R3 不引入外部路由的话，是不会出现问题的。因为 ospf 在区域间使用 LSA3，LSA3 是由区域的 ABR 根据 LSA1、LSA2 产生的，adv 是 ABR 的 router-id，区域间路由只是被当成叶子挂在 ABR 上，本区域内的 spt 树上不会出现在有相同 router-id 的节点，也就不会出现问题。但是如果在相同 router-id 的设备上做引入的时候就会出现问题了，因为 asbr 的 router-id 是需要被 ospf 域内的所有路由器所知道的，如果发现 asbr 的 router-id 与本设备的 router-id 一样时，会出现问题 分析： 假设 R1 引入一条路由 10.10.10.0/24，R1 会把这条 LSA（adv=1.1.1.1，type=5，LS ID=1.1.1.1，seq=8001）发送给 R2，R2 收到后会发给他的邻居 R3，R3 收到发现通告者是 1.1.1.1，但是自己又没有这个网段，于是会给 R2 发送一个（age=3600s，seq=8001）的 LSA5，R2 收到后，会与之前收到的 LSA5 进行比较，因为 seq 和 check sum 与之前的一样，所以会优选 age=3600s 的，然后也会转发给 R1，R1 收到后发现自己有这个网段，又会发送一条新的 LSA5（seq=8002），会一直出现这样重复的情况，而导致路由动荡。

B，接口区域 ID：接口优于全局。不同区域无法接收 HELLO 包

C，认证：认证类型分为不认证（00），明文认证（01）和 MD5 认证（02），OSPF 的认证放在 OSPF 头中，所以 OSPF 一边接口认证，一边区域认证可以认证成功。

E，MA 网络掩码（为什么 p2p 中掩码可以不一致）：MA 网路中掩码必须一致，因为 MA 网络中所有路由器公用一个网段，只有一个 1LSA 的 transit 和一个 2L 的 Network 来描述当前的网络拓扑和网络号。P2P 网络中掩码之所以可以不一致是因为 P2P 中有 1LSA 的 stub 类型来描述每一个网络的掩码信息，并且华为默认不检查 P2P 的掩码信息，并且在 PPP 链路中 NCP 阶段，两台路由器会互推自己的 IP 地址，并且以 32 位主机路由的方式加入自己的路由表，所以 P2P 网络中建立邻居不需要掩码一致。

F， MA 网络中优先级不能为零，DR 选举不成功。双方停留在 Attempt 状态

G，区域类型（option 字段中的 E 位与 N 位）：E 位代表能传递 5LSA，N 位代表 NSSA 区域

E	N	
1	0	普通区域
0	1	NSSA 区域
0	0	STUB 区域

G，hello-dead 间隔（区别网络类型）

	HELLO TIME	DEAD TIME
P2P	10	40
MA	10	40
NBMA	30	120
P2mP	30	120

I，MTU（默认不检查，不一致时会停留 在哪种状态）：如果开启了 MTU 检查，如果双方 MTU 不一致，则一方停留在 Exstart 状态，另一方停留在 Exchange 阶段

L，网络类型（四种，当两边不一致是否一定建立不了邻居，如果能建立会不会有问题，哪种网络类型发送单播，哪种发送组播）

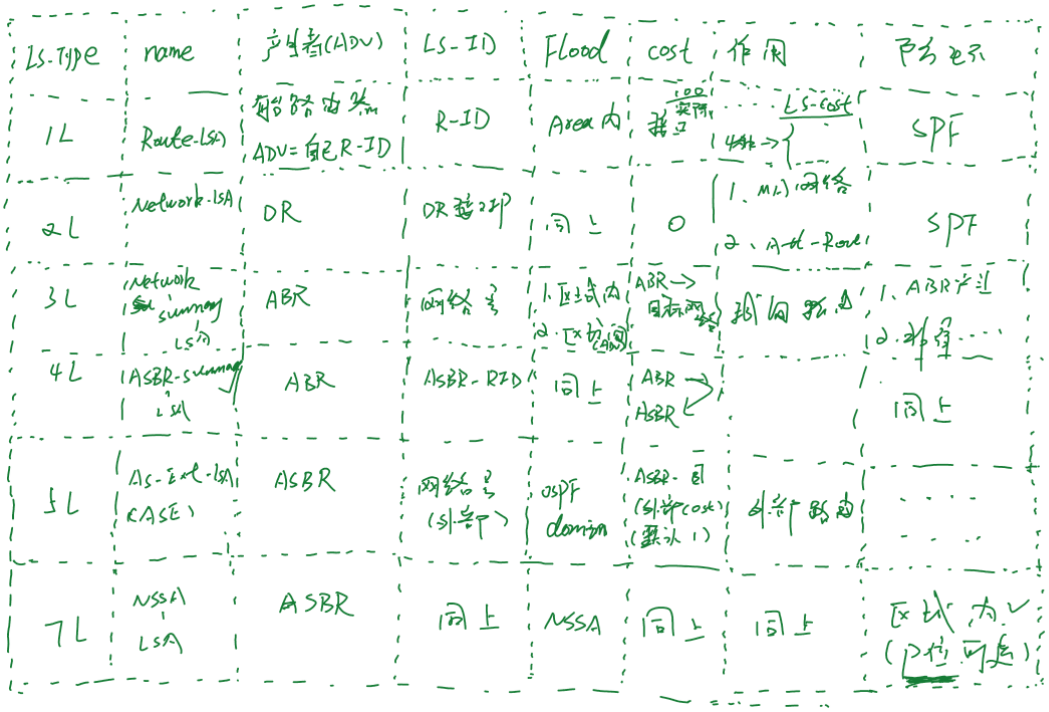
双方网络类型不一致，不能建立 FULL 的邻接关系，但如果修改 HELLO，DEAD 时间,可以建立邻居关系（除了 NBMA 这种网络类型，NBMA 即使修改时间也无法和其他网络类型建立邻居关系），P2P 和 P2MP 可以建立 FULL 的邻居关系，其他网络类型两两之间都无法建立 FULL 的邻居关系。

	HELLO	DD	LSR	LSU	LSACK
P2P	组播	组播	组播	组播	组播
MA	组播	单播	单播	组播	组播
NBMA	单播	单播	单播	单播	单播
P2MP	组播	单播	单播	单播	单播
Vlink	单播	单播	单播	单播	单播

M，silence（特点）：OSPF 的 silence 接口，不收不发

5，LSA 的类型

A，列举各种 LSA 的 link-state ID；ADV；泛洪范围（1LSA 的四种类型）



1LSA 详细内容

	Link ID	Link Data
P2P	邻居的 router id	自己的接口 IP
Stub	自己的接口网段	当前网段的掩码信息
Transit	伪节点（DR）的接口 IP	自己的接口 IP
Vlink	邻居的 router id	自己的接口 IP

B，描述 1，2 类 LSA 的作用

首先 1LSA，运行 OSPF 的每台路由器都会产生并且只产生一份 1LSA，根据不同的网络类型会产生不同的 1LSA，其中分为四种类型，P2P 描述的是 P2P 网络中的拓扑信息，Stub 描述的是 P2P 网络中的路由信息，Transit 描述的是 MA 网络中的路由信息，vlink 描述的是做了虚链路的路由器的拓扑信息。2LSA 中描述的就是 MA 网络中的拓扑信息，主要表示的就是 MA 网络中的伪节点连接着哪几台路由器（attched router 字段）。1LSA 和 2LSA 只在区域内泛洪。所以 OSPF 中 1LSA 和 2LSA 的作用就是描述当前网络中的拓扑信息及网络号以及开销。

C，LSA 的泛洪机制，泛洪周期，LSA 的标识方法，LSA 的新旧如何判断

LSA 的泛洪就是向水一样流出去，除了接收端口外向其他所有运行了 OSPF 接口泛洪。

泛洪周期为 1800 秒，老化时间为 3600 秒，所有的 OSPF 路由器每 1800 秒把自己数据库中所有的 LSA 向外泛洪。当收到 3600 秒的 LSA，则直接删除数据库中对应的 LSA。

OSPF 中通过三要素标识一条 LSA：LSA 的 Type，Link state ID，ADV router

OSPF 通过序列号，checksum，age time 来判断 LSA 的新旧。序列号最小为 80000001，最大为 7fffffff，序列号越大代表 LSA 越新，checksum 一般都是相同的，主要判断 checksum 是否出错，出错则不收这条 LSA，age time 一般来说经过一台路由器则加 1 秒，age time 越小越新，但当一台路由器收到两条 LSA，两条 LSA 的 age time 的相差时间小于 900 秒（15 分钟），则认为两份 LSA 是相同的，则会保留先收到的 LSA，不收后收到的 LSA，主要是为了保证网络的稳定性，如果收到两条 LSA 的相差时间大于 900 秒则会选择 age time 小的那份 LSA。

D，特殊区域中的默认路由是以几类 LSA 存在

	3L	7L	默认 3L	默认 7L
Stub	√	×	√	×
NSSA	√	√	×	√
Totally stub	×	×	√	×
Totally NSSA	×	√	√	√

Totally NSSA 中存在默认 3LSA 和 7LSA，但 3LSA 优于 7LSA，一般会用默认 3L 来访问外部。

E， FA 的作用，及产生条件，5 类 LSA 携带 FA 与不携带 FA 的区别。

FA 为 forwarding address，FA 的作用：解决次优和放环

5L 的 FA 产生的条件：（1）下一跳非 P2P 和 P2MP

（2）下一跳接口必须使能 OSPF

（3）下一跳接口不能被 silent

5L 中如果携带 FA 地址，则直接选择通过 FA 的地址去往目标网段

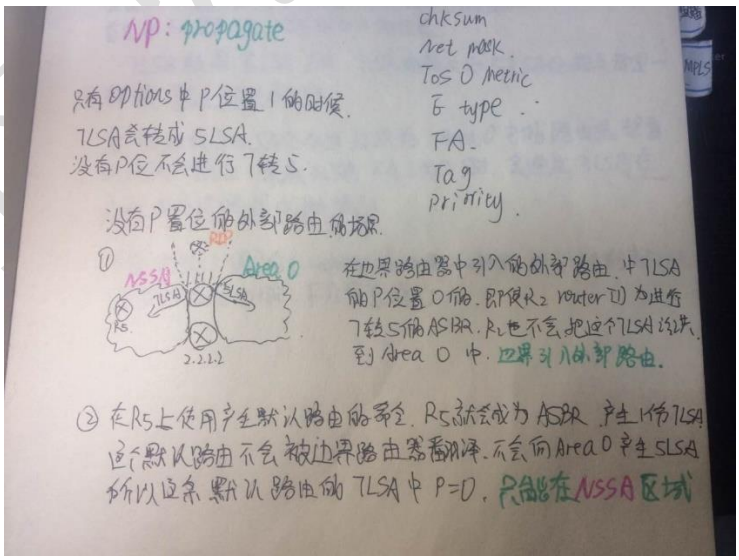
如果没有携带 FA 地址，则选择通过 ASBR 去往目标网段

F，7 类 LSA 中的 P 位的作用

7LSA 中的 P 位作用就是是否能进行 7 转 5，P=1 则表示能进行 7 转 5. P=0 的场景

（1） NSSA 区域中的默认路由不会进行 7 转 5

（2）



6，如何减小 OSPF LSDB 的大小

A，分区域设计：因为 1,2LSA 只在本区域泛洪，分区域设计可以减少每个区域 1 2 类 LSA 的数量

B，特殊区域：特殊区域无法传递 5LSA，可以减少 OSPF domain 中 5LSA 的数量

C，过滤(方法有几种)：过滤的方式有三种：（1）filter 命令过滤 3LSA，在区域下使用，可以在 import 和 export 方向 （2）filter lsa out 命令过滤 3LSA,5LSA,7LSA,ALL（3）filter-policy，在 OSPF 执行了 SPF 计算后，进行路由过滤。过滤只能在产生者上进行过滤，3LSA 在 ABR 上过滤，5LSA 在 ASBR 上进行过滤

D，汇总：summary+no-advertise，汇总也可以执行过滤。需要注意做了虚链路的区域不能针对 area0 的路由进行汇总，否则可能会产生环路

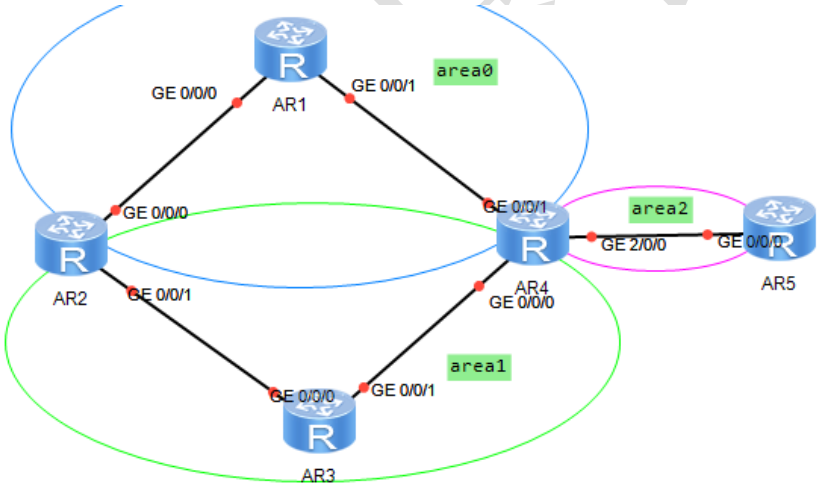
E，另外 2 类 LSA 也可以减小 OSPF LSDB 大小的能力，但这不属于我们控制范围

7，OSPF 路由选路的原则，及在什么情况下会负载

A，选路原则：区域内的>区域间的>TYPE 1>TYPE2

B，负载条件：1 cost 一致，2 区域一致

C，如下图：R4 与 R5 上分别引入外部路由，问 R2 如何去往这两条外部路由。



R2 如果想要访问 R5，R2 会通过 area 0 进行访问，因为非骨干区域传来的 4LSA，ABR 不参与计算，R2 会选择 R1 作为他的下一跳

R2 如果想要访问 R4，R2 会通过 area 1 进行访问，因为 R2 会通过 area0 和 area1 都收到 R4 的 1LSA，R2 上去往 R4 会有两个下一跳，但两个下一跳属于不同的区域，所以 R2 去往 R4 不能负载，如果 R2 通过两条路的 cost 值相同，R2 则会选择区域号大的作为 R2 的下一跳，R2 会选择 R3 作为下一跳

8，OSPF V2 与 V3 的区别

A，V2 有认证，V3 无认证（通过 ipv6 实现）

B，V2 基于 IP，V3 基于链路

C，V3 实现了拓扑与路由的分离(1，2LSA 中不再有网络信息)

D，V3 头部增加了实例号字段，可以实现一个接口配置多个进程

E，报文发送的目的地址不同

F，V3 必须手工指定 router-id

F， 增加了两类 LSA，Type8: Link-LSA；Type9: Intra-Area-Prefix-LSA (需要明白这两条 LSA 的作用)

OSPFv3 和 OSPFv2 协议比较如下：

相同点：

1> 网络类型和接口类型。

2> 接口状态机和邻居状态机。

3> 链路状态数据库（LSDB）。

4> 洪泛机制（Flooding mechanism）。

5> 相同类型的报文：Hello 报文、DD 报文、LSR 报文、LSU 报文和 LSAck 报文。

6> 路由计算基本相同。

不同点：

1> OSPFv3 基于链路，而不是网段。

OSPFv3 运行在 IPv6 协议上，IPv6 是基于链路而不是基于网段的。

在配置 OSPFv3 时，不需要考虑是否配置在同一网段，只要在同一链路，就可以不配置 IPv6

全局地址而直接建立联系。

2> OSPFv3 上移除了 IP 地址的意义。

这样做的目的是为了使“拓扑与地址分离”。OSPFv3 可以不依赖 IPv6 全局地址的配置来计算出 OSPFv3 的拓扑结构。IPv6 全局地址仅用于 Vlink 接口。

3> OSPFv3 的报文及 LSA 格式发生改变。

OSPFv3 报文不包含 IP 地址。

OSPFv3 的 Router LSA 和 Network LSA 里不包含 IP 地址。IP 地址部分由新增的两类 LSA（Link

LSA 和 Intra Area Prefix LSA）宣告。

OSPFv3 的 Router ID、Area ID 和 LSA Link State ID 不再表示 IP 地址，但仍保留 IPv4 地址格式。

广播、NBMA 及 P2MP 网络中，邻居不再由 IP 地址标识，只由 Router ID 标识。

4> OSPFv3 的 LSA 报文里添加 LSA 的洪泛范围。

OSPFv3 在 LSA 报文头的 LSA Type 里，添加 LSA 的洪泛范围，这使得 OSPFv3 的路由器更加灵

活，可以处理不能识别的 LSA：

OSPFv3 可以存储或洪泛不识别的报文，而 OSPFv2 只简单丢弃掉不识别的报文。

OSPFv3 允许洪泛范围为区域或链路本地，并且设置 U 位（报文可按洪泛范围为链路本地
来处理）的不识别报文存储或通过 stub 区域。
例如：R1 和 R2 都可识别某类 LSA，它们之间通过 R3 连接，但 R3 不识别该类 LSA。这样，当
R1 洪泛此类 LSA 时，R3 虽然不识别，但还是可以洪泛给 R2，R2 收到后继续处理。

5> OSPFv3 支持一个链路上多个进程。
一个 OSPFv2 物理接口，只能和一个 OSPFv2 实例绑定。
但是一个 OSPFv3 的物理接口，可以和多个 OSPFv3 实例绑定，并用不同的 Instance ID 区分。
这些运行在同一条物理链路上的多个 OSPFv3 实例，分别与链路对端设备建立邻居及发送报文，
且互不干扰。这样可以充分共享同一链路资源。

6> OSPFv3 利用 IPv6 链路本地地址。
IPv6 使用链路本地地址在同一链路上发现邻居及自动配置等。运行 IPv6 的路由器不转发
目的地址为链路本地地址的 IPv6 报文，此类报文只在同一链路有效。链路本地单播地址从
FE80/10 开始。
OSPFv3 是运行在 IPv6 上的路由协议，同样适用链路本地地址来维持邻居，同步 LSA 数据库。
除 Vlink 外的所有 OSPFv3 接口都使用链路本地地址作为源地址及下一跳来发送 OSPFv3 报文。
这样做的好处是：
不需要配置 IPv6 全局地址，就可以得到 OSPFv3 拓扑，实现拓扑与地址分离。
通过在链路上泛洪的报文不会传到其他链路上，来减少报文不必要的泛洪来节省带宽。

7> OSPFv3 移除所有认证字段。
OSPFv3 的认证直接使用 IPv6 的认证及安全处理，不再需要其自身来完成认证，使用协议
时只需关注协议本身即可。

8> 新增两种 LSA。
Link LSA：用于路由器宣告各个链路上对应的链路本地地址及其所配置的 IPv6 全局地址，

仅在链路内洪泛。

Intra Area Prefix LSA：用于向其他路由器宣告本路由器或本网络（广播网络及 NBMA）的

IPv6 全局地址信息，在区域内洪泛。

9> OSPFv3 只通过路由器 ID 来标识邻居。

OSPFv2 在广播网络、NBMA 及 P2MP 网络中是通过 IPv4 接口地址来标识的。

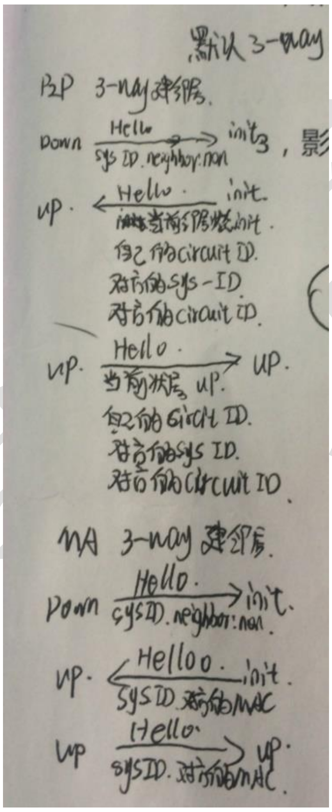
OSPFv3 只通过 Router ID 来标识邻居，这样即使没有配置 IPv6 全局地址，或者 IPv6 全局地址

配置都不在同一网段，OSPFv3 的邻居还是可以建立并维护的，以达到"拓扑与地址分离"的

目的

三，ISIS

1， 邻居关系建立过程(2-way,3-way)



A， p2p 网络 (TLV 240，对端的 system-id)

当收到对方发来的 HELLO 包中有自己的 system-id 则 UP

B， MA 网络 (TLV 6，对端的 MAC)

当收到对方发来的 HELLO 包中有自己的 MAC 地址则 UP

3，影响 ISIS 邻居关系建立的因素（8 条）

A，level：level-1 不能和 level-2 建立邻居

B，system-id：标识一台路由器，具有唯一性

C，认证(接口认证，区域认证，路由域认证 三者的区别)

接口认证是针对 HELLO 包的认证，区域认证和路由域的认证是针对除了 HELLO 包之外的认证

D， MTU (通过 padding 来填充，MA 与 p2p 的 padding 情况有什么差别？默认情况下 MTU 的范围 ma 不能小于 1500,p2p 不能小于 1497)

通过 PDU 的长度就知道了邻居 MTU 的大小，MA 网路中所有的 HELLO 都会填充 Padding 填充，P2P 只有在建立邻居时 HELLO 包才会被 Padding 填充，其他的 hello 包不会被填充

E，网络接口类型

P2P，MA，两边网络类型不一致无法建立邻居，所需 TLV 不同（TLV240 和 TLV6）

F，同一子网，掩码可以不一致（hello 包不携带掩码，携带 ip，路由器用收到的 IP 与本地接口的掩码进行与运算，必须处于同一个网段，如 192.168.1.1/24 和 192.168.1.200/25 之间建立邻居，结果.1 侧显示对端为 init 状态，.200 显示无邻居。P2P 可以配置忽略网段检查，命令：接口下 isis peer-ip-ignore）

G，3-way only 的情况：2-way 可以和 3-way 建立邻居 3-way 和 3-way only 可以建立邻居，但是 2-way 和 3-way only 无法建立邻居

G，cost-style（影响路由计算，narrow 与 wide 的区别（三点））

1：接口 cost 值不同 narrow 为 0-64 wide 为 2 的 24 次方

2：wide 支持 sub-tlv，可以用来打 TAG

3：TLV 不同 narrow 的 TLV 为 2,128,130 wide 的 TLV 为 22,1356

4，ISIS 报文（九种）

P2P 的 hello MA 网络中 level-1，level-2 的 hello，level-1，level-2 的 LSP，level-1，level-2 的 PSNP，level-1，level-2 的 CSNP

5，ISIS 数据库的同步过程(手册中有详细描述)

A，P2P(需要注意，手册里面讲的过程与实际抓包过程(直接推送 LSP)不一致，考试时按手册讲即可。)

此处应该有图

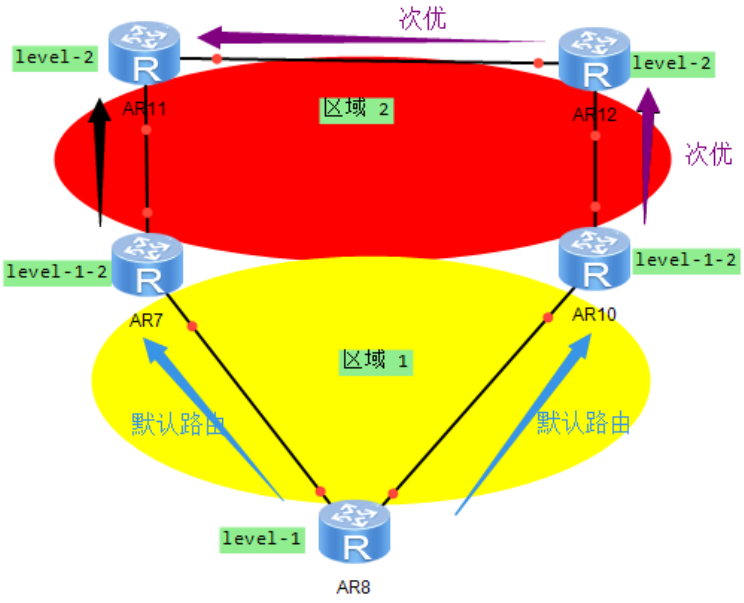
B，MA（报文发送的目的地址）

此处应该有图

6，路由泄漏

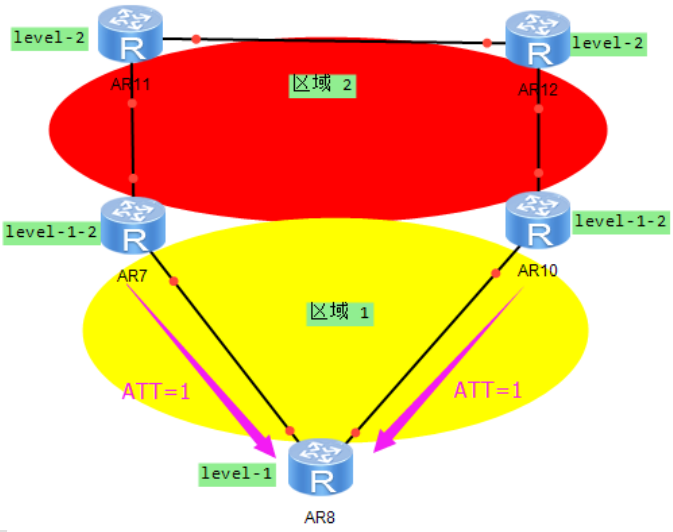
A，为什么要进行路由泄漏

解决次优问题，level-2 的 LSP 不会进入 level-1 中，level-1 自动进入 level-2 中，当 level-1 的路由器想要访问 level-2 的路由器时，可能就会产生次优路径。



B， ATT 置位的条件：level-1-2 的路由器有本区域 level-1 的邻居，有一个其他区域的 level-2 的邻居，就会 ATT 位置 1

作用：产生默认路由



C， 泄漏后如何防环(up/down 位)（ narrow 不支持 ）：level-1-2 的路由器在向 level-1 泄露路由时，这些 level-2 的 LSP 会自动置一个 UP/DOWN 位，当这些泄露的路由再次想传入 level-2 时，level-1-2 的路由器不会再向 level-2 的路由器传这些泄露的 LSP，为了防环

ISIS 的路由选择：level-2 > level-1 > leaked

7，DR 与 DIS 的区别

A， 选举条件，选举时间

DR 先比优先级再比 router id　　DIS 先比优先级再比 MAC 地址

B， DIS 支持抢占，DR 不支持

因为 OSPF 需要 DR 同步数据库，所有路由器都需要和 DR 同步数据库，如果 DR 可以被抢占可能会造成网络的震荡，DR 的选举有不确定性
DIS 设备之所以可以被抢占是因为 ISIS 中，只需要 DIS 周期性的发送 CSNP，每一台路由器都可以实现这个功能

C，DIS 优先级可以为零，DR 为 0 时不参与选举

D，DR 有 BDR 备份

E，DIS 3s hello，DR 10s/30s hello

F，邻接关系建立方式：OSPF 都和 DR 和 BDR 建立邻接关系，DR others 和 DR others 之间都是邻居关系，ISIS 的邻居关系为全邻接

8，ISIS 的区域与 OSPF 的 area 有什么区别

ISIS 只要是连续的 level-2 的路由器组成的就是骨干区域

A，OSPF 区域一致才能建立邻居，ISIS 只有在 level-1 时才要求一致

B，OSPF 区域类型更加丰富

C，表现形式不一样(OSPF 点分十进程，ISIS xx.xxxx)

D，在 OSPF 中，一条链路只能属于一个区域，而在 ISIS 中，一条链路可以属于不同的区域

H，OSPF 与 ISIS 的区别（参考题库）

从 5 个方面介绍 OSPF 和 ISIS 的区别：

<1>基本点比较

OSPF 只支持 IP 环境；ISIS 支持 IP 环境和 CLNP 环境。

OSPF 报文封装在 IP 报文中，协议号 89；ISIS 报文直接封装在链路层数据帧中。所以安全性相对高些.

OSPF 基于接口划分区域,多区域设计,层次设计,area0 为中心；ISIS 基于路由器划分区域。

OSPF 支持 P2P、BMA、NBMA、P2MP、虚链路网络类型；ISIS 支持广播和 P2P 网络类型。

<2>邻接关系比较

OSPF 邻接关系只有一种；ISIS 邻接关系分成 level-1 和 level-2 邻接关系

OSPF 的 DR 和 ISIS 的 DIS（ISIS 支持抢占、优先级 0 也可以成为 DIS、没有备份 DIS）

OSPF 的 MA 网络中普通路由器之间不能形成邻接关系；ISIS 的 MA 网络中所有路由器之间都能形成邻接关系

<3>链路状态数据库同步过程比较

OSPF 的 LSA 种类很多；ISIS 的 LSP 只有路由器 LSP 和伪节点 LSP。

OSPF 的 LSA 的生存周期从 0 递增；ISIS 从最大值递减。

Note:

OSPF 的 LSA 种类很多，数据库结构复杂，定位故障困难；ISIS 的 LSP 只有路由器 LSP 和伪节点 LSP，
数据库结构简单，定位故障容易
OSPF 的 LSA 生存周期是从 0 增加（maxage=3600,refresh 周期为 1800,不可调）；ISIS 从最大值减小（maxage 1200s，refreshment 周期为 900s,可调）

<4>路由计算过程比较
OSPF 将前缀作为 SPF 的节点；ISIS 将前缀作为叶子（叶子发生变化时可以用 PRC 来更新叶子而不需要进行 SPF 计算）。
OSPF 的接口开销根据接口带宽变化(0-65535)；ISIS 的接口开销值缺省相同(所有接口默认为 10,最大可达 4Byte,即 2^32-1)
Note:
Full SPF (Dijkstra)计算仅发生一次,初次; 之后任何变化都只计算受影响的节点周边拓扑,这就是 iSPF(增量 SPF);至于 iSPF tree 上 node 的叶子路由的变化,则需要 PRC 计算,即只对那片叶子(路由)做计算即可,大大减少 CPU 的计算负荷.
但 ospf 在 area 内任何路由变化(LSA1/2 中路由条目)触发 iSPF 计算,仅 LSA3/4/5/7 的变化才 PRC, 而 isis 任何路由变化都是 PRC.

<5>性能及扩展能力比较
OSPF 支持按需拨号链路；ISIS 不支持。
ISIS 采用 TLV 结构，扩展性更好。

11，常见 TLV 类型
TLV 1：Area ID
TLV 2：描述邻居信息 narrow
TLV 6：MA 网络中描述邻居 MAC
TLV 8：padding
TLV 10：认证
TLV 22：描述邻居信息 wide
TLV 128：内部路由 narrow
TLV 130：外部路由 narrow
TLV 132：接口 IP
TLV 135：路由信息 wide
TLV 240：p2p 网络中描述邻居状态

路由技术-BGP

2016 年 7 月 19 日
14:52

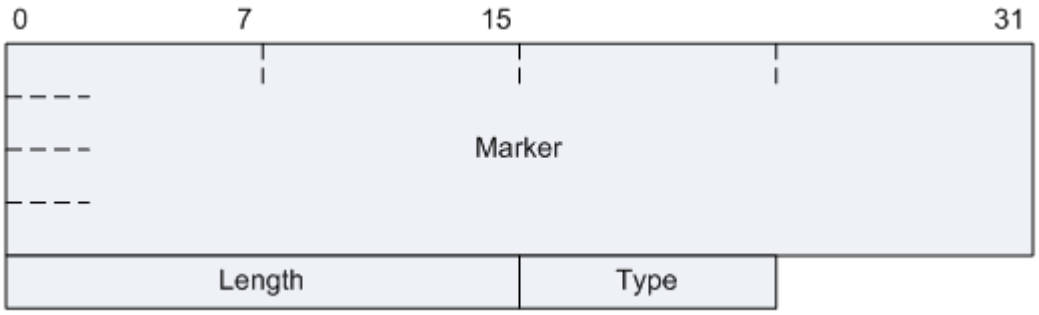
一，BGP 的作用，及应用场景

在 AS 间传递路由

1，AS 的概念：首先 AS 是一个虚拟的概念，不真实存在，可以把一个公司，一个个人定义为一个 AS，可以当做一个逻辑的集合体，可以为每个 AS 定义一个 AS 号来进行管理

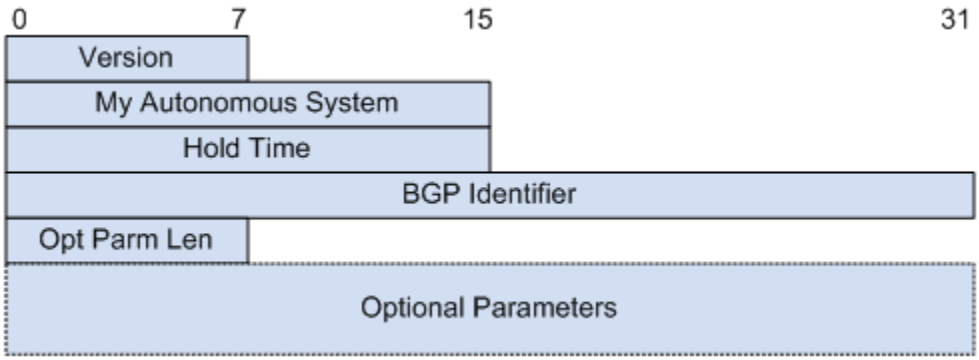
二，BGP 的邻居状态机及报文

BGP 的认证在 BGP 的头部中

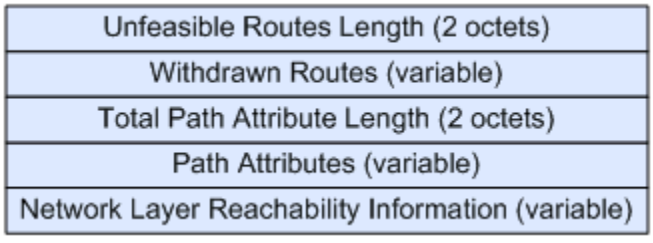


BGP 的报文：

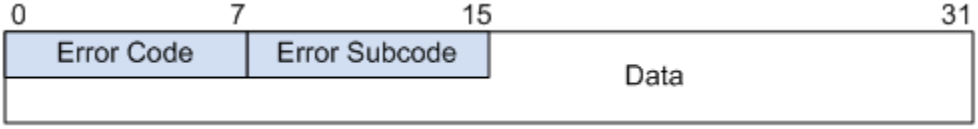
1. OPEN



2.Update

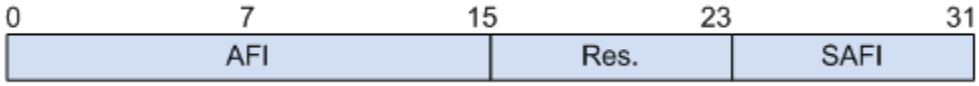


3. NOTIFICATION

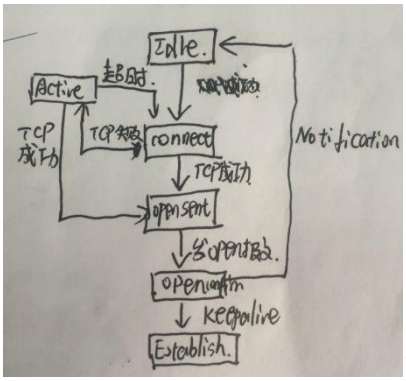


4. keepalive

5. refresh



1，详细描述邻居关系的建立过程



2，Idle 状态做什么事：什么也不做，等待

3，connect 与 active 的区别：connect 是主动建立 TCP 连接，active 是被动等待 TCP 连接

4，open 报文里面主要包含什么内容

A，version（默认版本 4）；AS；Router ID；Hold time （180）；

B，option 中默认通告了哪几种能力

1：扩展能力 AFI SAFI

可以判断 BGP 可以运行在哪些协议下

AFI	SAFI
1 单播	1 IPV4
2 组播	2 IPV6
196 二层	128 VPNV4

2：路由刷新

（1） 请求：refresh bgp all import

（2） ORF：出方向的路由过滤 只允许通过需要的路由

3：4 字节 AS 号

OPEN 报文中原本的 AS 号为 2 字节，最大为 65535，但如果需要的 AS 号超过了 65535，则在 OPEN 报文中显示 AS 号为 23456，在 option 中查看真实的 AS 号，用于扩展

C，open 报文里 AS 号 23456 代表什么：同上

5，什么时候会产生 refresh 报文

请求：refresh bgp all import

ORF：出方向的路由过滤 只允许通过需要的路由

6，什么时候会产生 notification 报文

详情见文档

三，影响 BGP 邻居建立的因素

1，版本：默认为 4

2，AS 号：IBGP 邻居要求一致，EBGP 邻居一定不一致

3，Route-ID：不能相同

4，179 端口被禁止：如果被禁止，则无法建立 TCP 连接，也就无法建立邻居

5 peer 可达性：peer 地址必须可达

7，认证：

8， Connect interface ：收到报文的源地址必须和 peer 的地址相同

1， BGP 属性(需要知道每条属性的作用)

1，公认属性：ORIGIN

AS_PATH

NEXT_HOP

所有 BGP 路由器都可以识别，且必须存在于 Update 消息中

如果缺少这种属性，路由信息就会出错

2， 公认可选：LOCAL-PREF

ATOMIC-AGGREGATE

所有 BGP 路由器都可以识别，但不要求必须存在于 Update 消息 中即就算缺少这类属性，路由信息也不会出错

3， 可选传递：AGGREGATOR

COMMUNITY

在 AS 之间具有可传递性的属性

BGP 路由器可以不支持此属性，但它仍然会接收这类属性，并传递给其他对等体

4，可选非传递：MULTI_EXIT_DISC（ MED ）

CLUSTER_LIST

ORIGINATOR_ID

如果 BGP 路由器不支持此属性，则相应的这类属性会被忽略，且不会传递给其他对等体

表 5-2 路径属性类型码和属性值

属性类型	属性值
1：Origin	• IGP • EGP • Incomplete
2：As_Path	• AS_SET • AS_SEQUENCE • AS_CONFED_SET • AS_CONFED_SEQUENCE
3：Next_Hop	下一跳的 IP 地址
4：Multi_Exit_Disc	MED 用于判断流量进入 AS 时的最佳路由
5：Local_Pref	Local_Pref 用于判断流量离开 AS 时的最佳路由
6：Atomic_Aggregate	BGP Speaker 选择聚合后的路由，而非具体的路由
7：Aggregator	发起聚合的路由器 ID 和 AS 号
8：Community	团体属性
9：Originator_ID	反射路由发起者的 Router ID
10：Cluster_List	反射路由经过的反射器列表

MED 和 local-prefer 的区别

		MED	LP
EBGP	Export	√	×
	Import	√	√
IBGP	Export	√	√
	Import	√	√

五，BGP 的选路规则（需要根据每条规则画出相应的场景）

首先，路由的下一跳必须可达，然后 BGP 按照下面顺序选路：

<1>prefer-value (越大越好)

<2>local-pref (越大越好)

<3>本地始发

<4>as-path (越短越好)

<5>origin（i>e>?）

<6>med (越小越好)

<7>ebgp>ibgp

<8>igp costfornext-hop (越小越好)

<9>是否支持负载均衡（maximumload-balance）

<10>cluster-list (越短越好)

<11>originator-id (越小越好)

<12>router-id(越小越好)

<13>next-hop ip address#neighbor‘s ip address (越小越好)

以上 13 条规则的内容及顺序务必牢记;

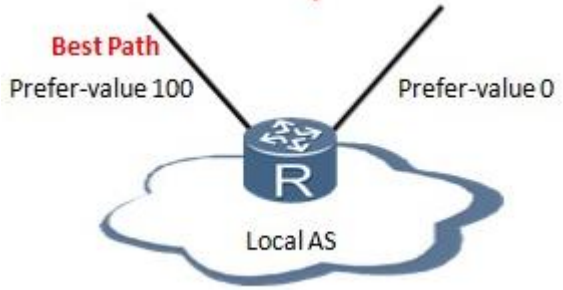
另外在解释每条规则的时候,要准备场景.

BGP 的选路规则,举例说明每条规则的具体使用

首先，路由的下一跳必须可达：

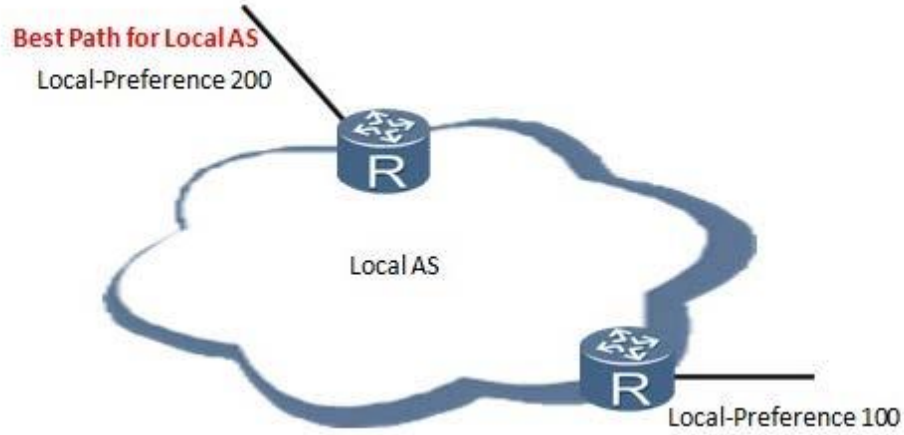
(1) prefer-value

首选值，数值越大越优先，本地有效



(2)local-pre

本地优先级，数值越大越优先，可传递给 IBGP 邻居，如果没有配置默认为 100



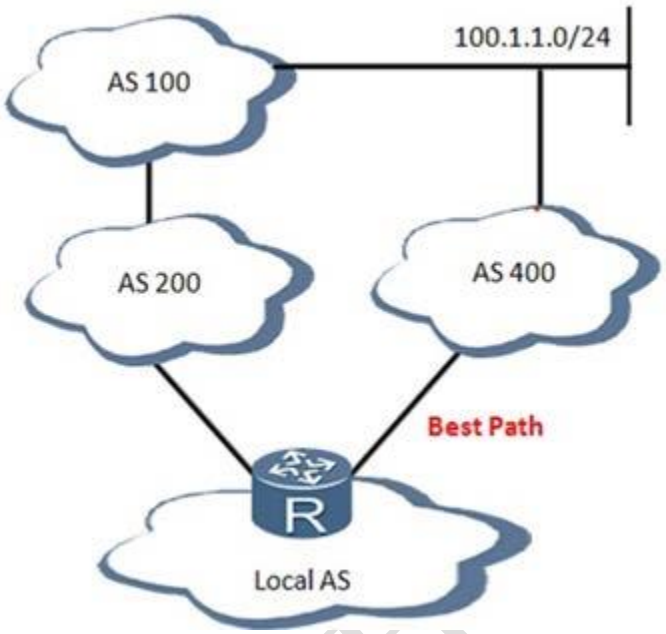
(3)本地始发

本地生成路由优先，aggregate 手工生成聚合路由>summary automatic 自动聚合路由>network 命令宣告路由>import-route 引入的路由。

如上图，如果 R1，R2 都将 10.1.12.0/24 宣告进 BGP 的话，R2 能收到 R1 发过来的 10.1.12.0 的路由，

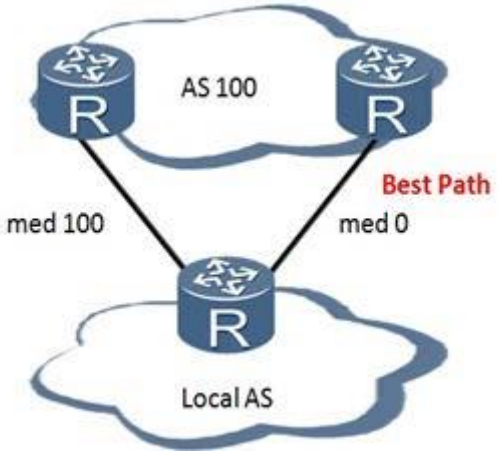
prefer-value 和 local-preference 都一样，但是本地发起的优先，所以 bgp 表中自己宣告的路由为最优路径。

(4)as-path
as-path 最短的路由（单个 AS 计数为 1）。AS_CONFED_SEQUENCE 和 AS_CONFED_SET（联盟内部 AS 号）不计入 as-path 长度。AS_SET 长度计为 1。（此条选路法则可以用命令忽略: bestrouteas-path-ignore）



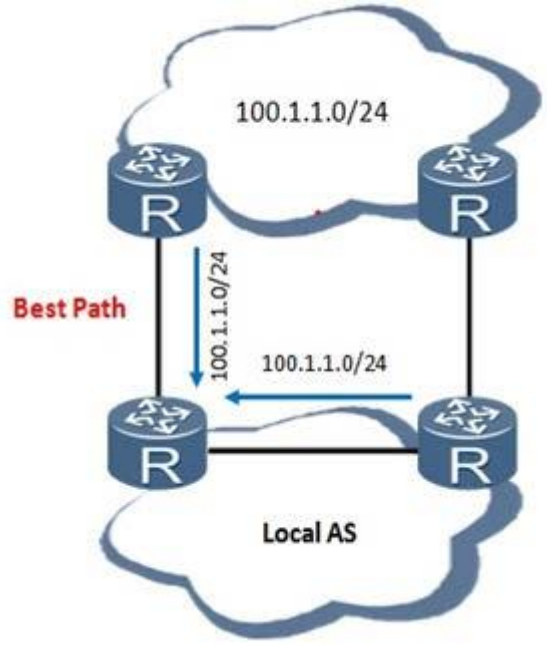
上图左侧路径路由传递过来 as-path 为 200, 100 长度为 2，右侧传递过来 as-path 为 400, 长度为 1。
优选右侧传递过来的路由为最优路径。
(5)origin(i>e>?)
IGP>EGP>Incomplete

(6)med
数值越小越优先，默认为 0。(bestroute med-none-as-maximum 可以将 med 默认值改到最大 4294967295）默认只比较 as-path 中最近一个 as 号相同的路由，否则忽略此条。

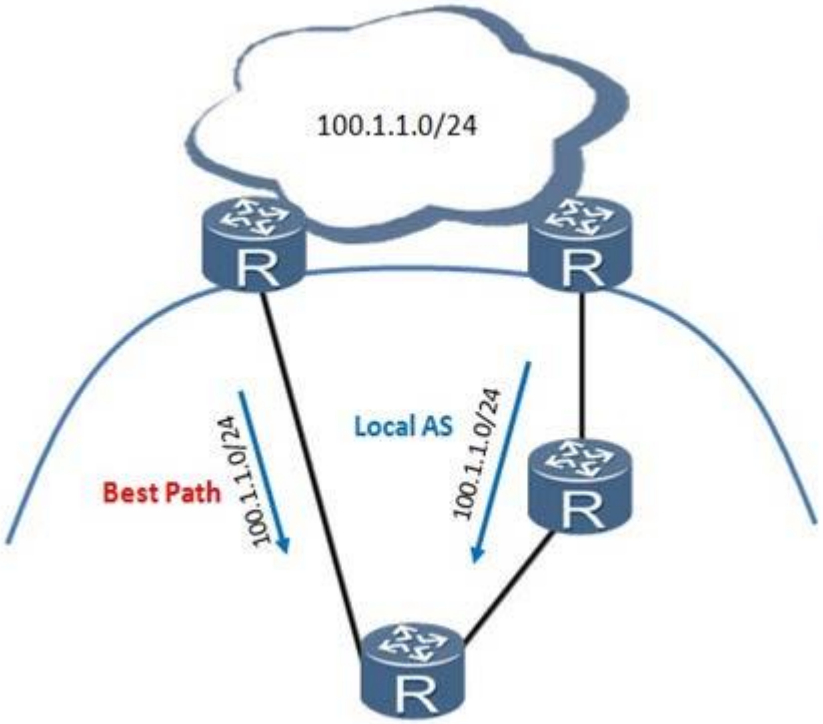


compare-different-as-med 命令后，强制比较不同 as 的路由 med。
bestroute med-confederation，只比较 as-path 只包含联盟内部 as 且最近一个联盟内部 as 号相同的路由的 med。

deterministic-med，按相同最近 AS 号的先比，消除按接收顺序两两比较 med 对比较结果的影响。
e.g.
as-path med type(第 7 条选路法则）router id
200 100 0 IBGP 1.1.1.1 （配了之后）Best
200 300 100 EBGP 5.5.5.5
200 100 100 EBGP 2.2.2.2 没配 deterministic-med 之前最优
(7)ebgp>ibgp
ebgp>ibgp>localcross 路由>remotecross 路由



(8)igp cost for next-hop
下一跳 igp 的 metric 最小的路由

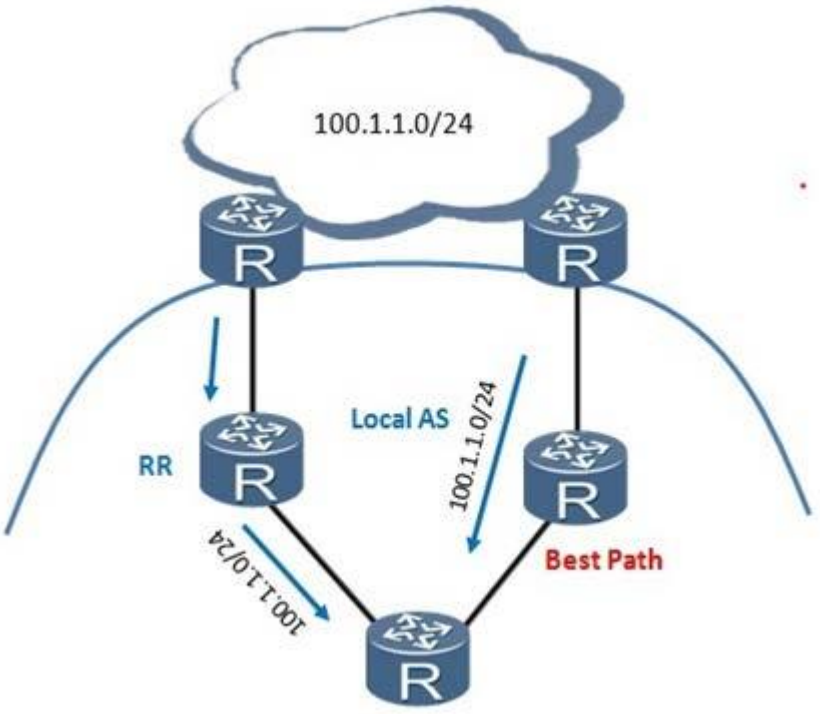


(9)是否支持负载均衡<maxmum load-balance>

前 8 条一样，且 as-path 完全相同（都是聚合路由或都不是），如果配置了多路径负载均衡的话，进行负载均衡（这里默认 IBGP 和 EBGP 路由都参与负载均衡）

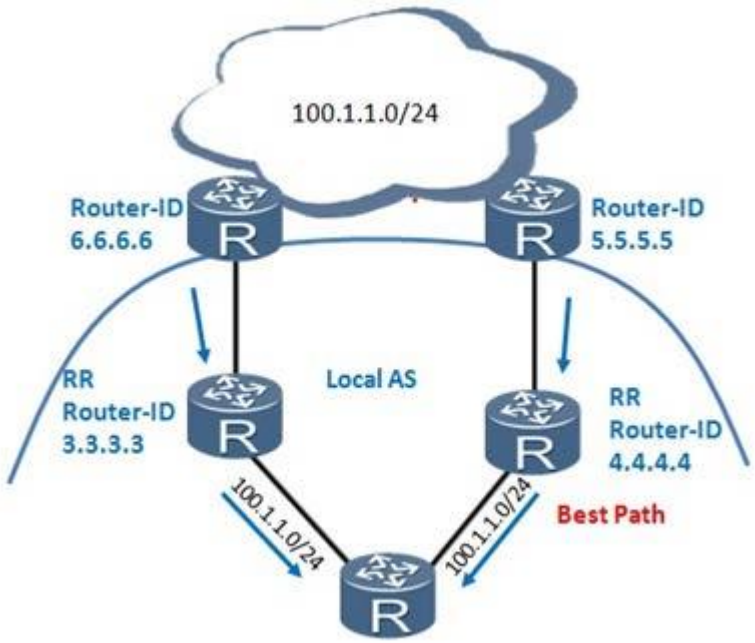
(10)cluster-list

每一个 cluster-id 计数为 1，长度最小的优先。



(11)orinator-id

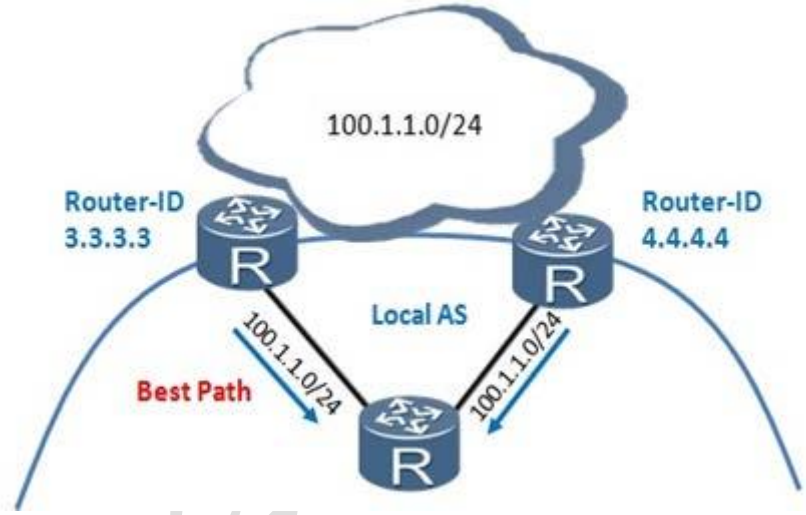
越小越优先



as 外部路由，originator-id 就是边界路由器 5 和 6，这里虽然邻居 R3 路由器 ID 小，但是由于 originator-id 是 4 那边小，所以选择右侧过来的路由为最优。

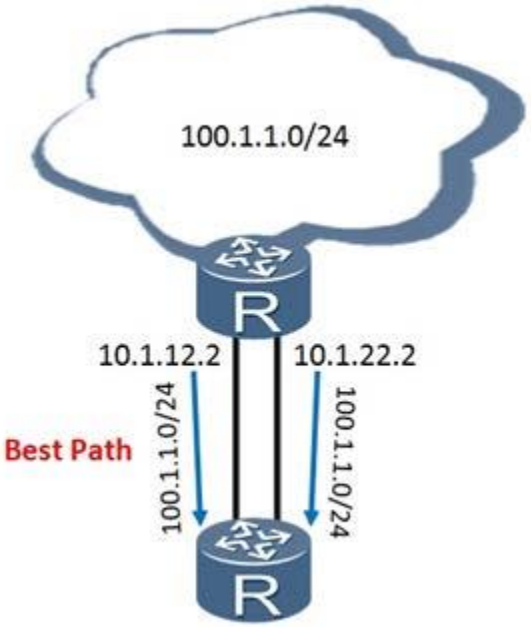
(12)router-id

越小越优先。



(13)peer ip address

peer 命令后的地址，地址低的发来路由优先。



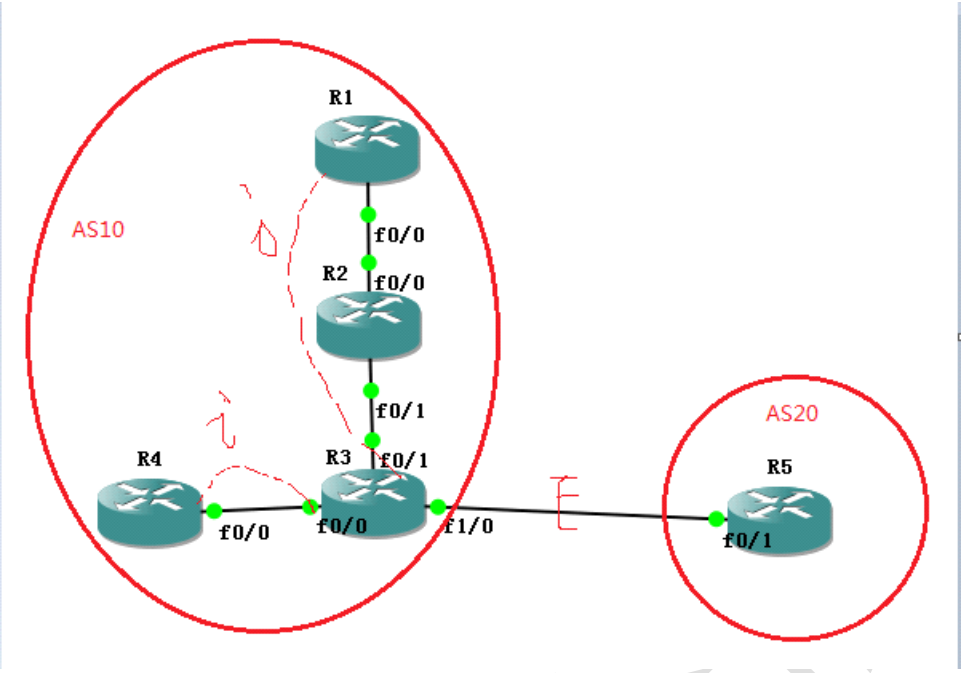
如图，下方路由器与上面路由器的 2 个地址 10.1.12.2 和 10.1.22.2 分别建立 2 个邻居，2 个链路 igp 开销也相同，由于这 2 个邻居其实是同一个路由器，所以路由器 id 一样，至此前面所有的法则都无法打破僵局，最终只能根据建邻居地址越低越优先，选择了 10.1.12.2 的邻居关系发来的 BGP 路由。

六，BGP 同步

1， 什么是 BGP 同步，同步能解决什么问题，为什么需要关闭同步

BGP 同步规则：开启同步下，从 IBGP 收到一条路由不会传给任何 EBGP 邻居，除非从自身的 IGP 中也学到这条路由。目的是防止 AS 内部出现路由黑洞，向外部通告了一个本 AS 不可达的虚假的路由。同步规则只影响 IBGP 邻居之间的路由传递，不影响 EBGP 邻居之间的路由传递。在开启同步的情况下，从 IGP 中没有收到这条路由，在自身 BGP 表中不是 best 的路由，也不会装进自己的路由表。即使配置为路由反射器后，也不会反射给任何客户端(反射器只反射最优路由)

实验场景说明：R1，R2，R3，R4 属于同一个 AS10，R5 属于 AS20，R3 分别和 R1，R4 建立 IBGP 邻居，并且 R3 是路由反射器。R3 和 R5 建立 EBGP 邻居。R2 不运行 BGP。



R1 上发布一条 1.1.1.1/32 的路由进 BGP 中，在 R3 的 BGP 进程中开启同步，R3 收到的 1.1.1.1/32 这条路由不是 best 路由，也不会装进 R3 的路由表中，不会传给 R4 和 R5。如果关闭同步后，R3 传递给 R5，在 R5 上数据包访问 1.1.1.1 的方向: R5 给 R3，R3 给 R2，R2 收到数据包，由于目的地址不可达丢弃数据包。此现象称为路由黑洞。

R3 上显示信息如下：

```
R3#show ip bgp  1.1.1.1
BGP routing table entry for 1.1.1.1/32, version 6
Paths: (1 available, no best path)
Flag: 0x820
  Not advertised to any peer
  Local, (Received from a RR-client)
    10.1.1.1 (metric 2) from 10.1.1.1 (10.1.1.1)
      Origin IGP, metric 0, localpref 100, valid, internal, not synchronized
```

假设从 R5 发布一条 5.5.5.5/32 进 BGP 中，由于 R3 和 R5 之间是 EBGP 邻居，所以 R3 收到后不会被同步规则影响，具体 BGP 表如下：

```
   Network          Next Hop            Metric LocPrf Weight Path
* i1.1.1.1/32      10.1.1.1                 0    100      0 i
*> 5.5.5.5/32      10.1.35.5                0             0 20 i
R3#
```

1.1.1.1/32 由于没有满足同步规则，没有打>号，而 5.5.5.5/32 仍然是 best 的路由，R3 会把 5.5.5.5/32 正常传递给 R4 和 R1。

但是,BGP 路由主要来自于互联网,IGP 不能承受几十万条的外部路由，所以我们一般不建议将 BGP 路由注入 IGP 中。

bgp 同步的解决方案

1.full mesh ibgp 解决方案

AS 内部的所有路由器都运行 full mesh ibgp,就可以关闭所有路由器的同步而不影响路由的通告和连通性.

不足：当 as 内部路由器数量很多时,需要建立 N*(N-1)/2 个 ibgp 会话,带来过度的系统开销,扩展性不好.

2.路由反射器解决方案

AS 内部的所有路由器都运行 bgp,在 AS 内部部署路由反射器,构建 hub and spoke 的 ibgp(会话数为 N-1), 然后关闭所有 bgp 路由器的同步.

3.bgp 联盟解决方案:

联盟将一个 AS 划分为若干个子 AS。每个子 AS 内部建立 IBGP 全连接关系，子 AS 之间建立联盟 EBGP 连接关系，但联盟外部 AS 仍认为联盟是一个 AS。配置联盟后，原 AS 号将作为每个路由器的联盟 ID。

七，BGP 有哪些防环技术

1，AS-Path(AS-Set)：如果收到一条路由的 AS 号中有自己所在 AS，那么就不会收这条路由，可以配置命令 peer allow-as loop {number}，number 代表可以重复出现几次 AS 号。

2，IBGP 转发规则（通过 IBGP 学到的路由不能转发给其他 IBGP 邻居）

3，路由反射器中的 Originator-ID 和 Cluster-list

4，路由聚合时，聚合路由器会自动产生的指向 null0 的路由（自动和手动聚合都会产生）

5，IBGP 默认不能重发布给 IGP：默认把 BGP 的路由引入到 IGP 中只会引入 EBGP 的路由，不会引入 IBGP 的路由，如果想要引入 IBGP 的路由需要在 import bgp 后面再加上 IBGP

八，BGP 路由条目在什么情况下不加表(全局路由表与 BGP 路由表)

BGP 表中有路由，但是此路由是不可用路由，不装进 IP 路由表。

①如果此路由的下一跳不可达，忽略此路由（本地优化）

②开启同步并且不满足同步的条件

③如果是 VPNv4 的路由，无法迭代到下一跳对应的隧道，或者隧道不是/32 位掩码，则此路由不进客户的实例路由或者不传给 EBGP 的邻居

④在 BGP 进程中开启了 dampening 命令使能了路由振荡抑制（默认未使能），BGP 的震荡抑制使用惩罚值来衡量一条路由的稳定性，惩罚值越高则说明路由越不稳定。路由每发生一次振荡，即路由器收到该路由的 withdraw 报文或者收到该路由的属性更新的 update 报文时，BGP 便会给次路由增加一定的惩罚值（1000）.当惩罚值超过抑制阈值（默认 2000）时，此路由被抑制，不加入到 IP 路由表中，路由器也不再向其他 BGP 对等体发布更新报文。BGP 会将该路由的 best 标志去掉。

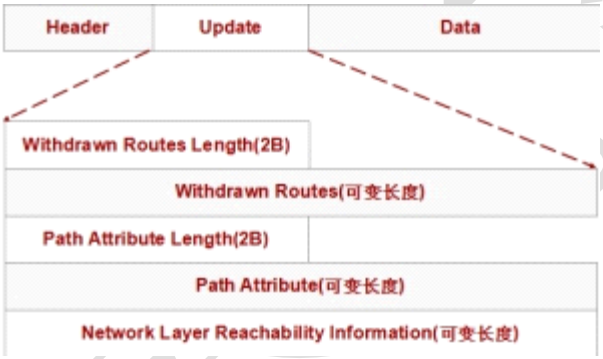
⑤bgp-rib-only 命令用来禁止 BGP 路由下发到 IP 路由表。（只在 BGP 路由表中有，不进自身路由表）

⑥active-route-advertise 命令用来配置 BGP 仅发布在 IP 路由表中被优选的路由。（应用场景：缺省情况下路由只需在 BGP 中优选即可向邻居发布。配置了此命令之后，路由必须同时满足在 BGP 协议层面优选与路由管理层面活跃两个条件，才能向邻居发布。）如果一条路由 BGP 是有效的，但如果从其他协议学到了这条路由，由于 BGP 路由协议优先级低，所以不进路由表。如开启此命令则不向邻居通告这条 BGP 路由。

而以下场景是设备不接收 BGP 路由，BGP 中也没有路由。
①当从 EBGP 的邻居收到一条路由中 AS-Path 包含自己的 AS 则不收这条路由；
②从 IBGP 的邻居收到一条路由中起源 ID 是自己的 router-id 也不收这条路由；
③RR 反射器收到路由的 cluster-list 包含自己的 cluster-id 也不收这条路由

九，BGP 在传递 IPV4,IPV6,VPNV4 路由时为什么更加简单

1.BGP 是外部路由协议，用来在 AS 之间传递路由信息，是一种增强的距离矢量路由协议，丰富的 Metric 度量方法，支持 CIDR（无类别域间选路）。
2.Update：该报文则是邻居之间用于交换路由信息的报文，其中包括撤销路由信息和可达路由信息及其各种路由属性。是 BGP 五个报文中最重要的报文。
Update 报文



UPDATE 消息被用作在 BGP 对等体之间传递路由信息。多条可达路由信息可以被通告到相应的对等体上，或者多条不可达路由信息被撤消。
Network Layer Reachability Information ：（变长）网络可达信息。包括一系列的 IP 地址前缀。格式与撤消路由字段一样<length, prefix>。
最小 UPDATE 消息的长度为 23 个字节(19 字节的报文头+2 字节的撤消路由长度+2 字节的路径属性长度)。这样的 UPDATE 消息被称之为 End-of-RIB，用于 BGP GR。
一条 UPDATE 消息可以发布多条具有相同路由属性的可达路由，这些路由可共享一组路由属性。所有包含在一个给定的 Update 消息里的路由属性适用于该 Update 消息中的 NLRI 字段里的所有目的地（用 IP 前缀表示）。
一条 UPDATE 消息可以撤销多条不可达路由。每一个路由通过目的地（用 IP 前缀表示），清楚的定义了 BGP Speaker 之间先前通告过的路由。一条 UPDATE 消息可以只用

于撤销路由，这样就不需要包括路径属性或者网络可达信息。相反，也可以只用于通告可达路由，就不需要携带 WithdrawnRoutes 了。

3.OPEN 报文中的 Optional Parameters 字段：是一个可选参数用于 BGP 验证或多协议扩展（Multiprotocol Extensions）等功能。每一个参数为一个（Parameter Type-Parameter Length-Parameter Value）三元组。
AFI：1 代表 IPv4，2 代表 IPv6；SAFI：1 代表单播，2 代表组播，128 代表 VPN

4.属性值
一套属性放入一个 Update，路由放入属性中传递
Mp-reachable-NLRI 属性号 14
Mp-unreachable-NLRI 属性号 15

十，IBGP 与 EBGP 传递路由有什么区别

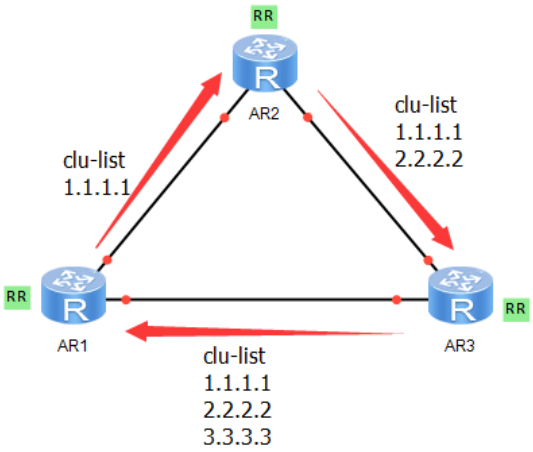
IBGP 和 EBGP 传路由的区别

1.从 IBGP 邻居学到的路由向 EBGP 传递路由的时候，下一跳改为自己，从 EBGP 邻居学到的路由向 IBGP 传递路由的时候，下一跳不改变。
2.传递给 EBGP 邻居的时候会添加自己的 AS 号，传递给 IBGP 邻居的时候不加自己的 AS 号。
3.EBGP 传递路由的时候剥离 LP 属性，IBGP 传递路由时携带 LP 属性
4.从 IBGP 邻居收到的路由传递给 EBGP 路由时候 MED 默认剥离，从 EBGP 邻居收到路由传递给 IBGP 邻居的时候 MED 不变。
5.向 IBGP 反射路由时候，保留 cluster-list 和起源 ID 属性，向 EBGP 传递路由时，剥离这两个属性。
6.引入防环保护，EBGP 通过 AS-path 防环，IBGP 通过水平分割防环，然后可以引入 RR

十一，RR 的原理

非非不传
打破 IBGP 的水平分割
利用 cluster-list 和 originator-id 防环

Cluster-list 防环场景



组播

一，组播基础

1， 为什么需要使用组播，带来哪些好处？

可以降低网络流量，减轻服务器负担，减少冗余流量，节约带宽，可以点对多点

2， 组播地址划分

永久组地址：224.0.0.0~224.0.0.255

SSM 地址：232.0.0.0~232.255.255.255

ASM 地址：224.0.1.0~231.255.255.255

233.0.0.0~238.255.255.255

本地管理地址：239.0.0.0~239.255.255.255

3， 常见的永久组地址

224.0.0.1 所有节点

224.0.0.2 所有路由器

224.0.0.5 OSPF

224.0.0.6 OSPF

224.0.0.9 RIP

224.0.0.13 PIM

224.0.0.18 VRRP

224.0.0.22 IGMPV3

4， 组播 IP 与组播 MAC 的映射关系（IPV4,IPV6），为什么需要这样映射？

无法根据组播地址得到组播 MAC，只能通过人为的方式规定一个地址

组播地址 01005E0 /25，32 个 IP 地址对应一个 MAC 地址，一般不会影响，因为即使 MAC 地址冲突了，只要检查二层是否能通过就可以，然后可以通过三层的 IP 地址进行转发。

二，IGMP

1， IGMP 的作用

IGMP 通过在接收者主机和组播路由器之间交互 IGMP 报文实现组成员管理功能

项目	IGMPv1	IGMPv2	IGMPv3
查询器选举方式	依靠组播路由协议PIM选举	同网段组播路由器之间竞争选举	同网段组播路由器之间竞争选举
普遍组查询报文	支持	支持	支持
成员报告报文	支持	支持	支持
特定组查询报文	不支持	支持	支持
成员离开报文	不支持	支持	没有定义专门的成员离开报文，成员离开通过特定类型的报告报文来传达
特定源组查询报文	不支持	不支持	支持
指定组播源	不支持	不支持	支持
可识别报文协议版本	IGMPv1	IGMPv1、IGMPv2	IGMPv1、IGMPv2、IGMPv3
ASM模型	支持	支持	支持
SSM模型	需要IGMP SSM Mapping技术支持	需要IGMP SSM Mapping技术支持	支持

2，详细讲述各版本的工作过程及区别

V1 普遍组查询 报告报文 普遍组查询组地址为 0.0.0.0 目的地址为 224.0.0.1 成员抑制功能

V2 普遍组查询 特定组查询 报告报文 leave 报文 特定组查询组地址为想要查询的组的地址，目的地址为想要查询的组地址

V3 查询报文 报告报文 删除了成员抑制功能 include exclude 六种

3， IGMP v2 中的 last reporter 作用

详情见理论视频

4， IGMP snooping 的工作过程及作用

IGMP snooping 的作用是侦听路由器与接收者之间的报文，通过组播 MAC，组播 IP 和 PC 与路由器的端口形成二层组播转发表，可以减少二层组播流量的泛洪，只传给对应的接受者，可以节省开销，减少无用的组播流量

5， IGMP snooping proxy

当交换机开启了 snooping 功能后，就没有了抑制报告报文的作用，所有的成员都会发送 report 报文，路由器处理了很多相同的 report 报文，为了减少路由器处理报文的数量，所以在交换机上开启 IGMP snooping proxy，可以减少交换机与路由器之间的报文数量

针对路由器来说是成员，针对成员来说是查询者

三，PIM

1， PIM 的作用，为什么称为协议无关组播；分哪几种模式

PIM 运行在路由器之间，生成组播的路由表象，指导组播流量的转发

协议无关组播：只是需要 IGP 的路由来进行 RPF 检测，并不关心当先运行的是什么协议

DM 密集模式 SPT 树越多，就认为接收者越多，也就越密集

SM 稀疏模式

2，PIM DM 的工作机制

A，邻居发现(Holdtime)

B，泛洪

C，剪枝/加入

D，状态刷新（作用）

E，嫁接

F，断言

3，PIM SM 的工作机制

1， RPT 树的形成

2， SPT 树的形成

3， 注册消息的作用，及什么时候会发送注册停止。

4，切换的过程，切换后的剪枝

4， PIM SSM mapping

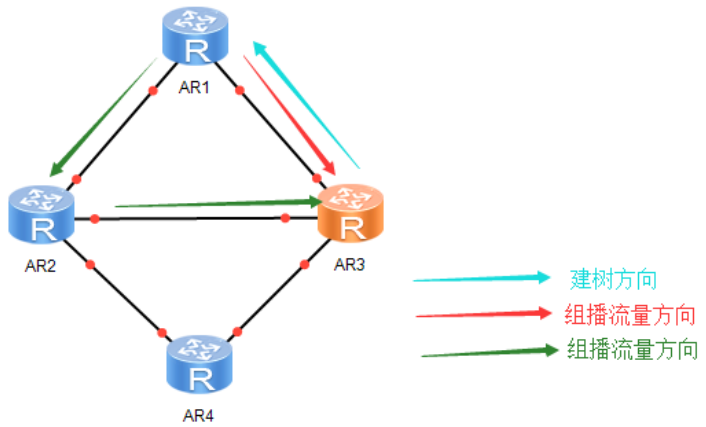
运行在 IGMPV1 IGMPV2 中，由于 V1 和 V2 加组的时候没有指定一个源的信息，也就只能建立 RPT 是，无法建立 SPT 树启动了 mapping 后，要手动绑定一个源信息，让路由器知道源的信息，从而建成 SPT 树。

5， PIM 的报文（10 种,哪些是单播哪些是组播）

单播	组播
Graft	Hello
Graft ack	Assert
Resigtor	State-refresh
Resigtor stop	Join/prune
RP-advertisement	Bootstrap

6，RPF 的作用及使用场景

RPF 的作用：可以解决次优路径，防环，确定建树的出接口



IPv6

2016 年 7 月 21 日
10:17

一，IPv6 的特点

1，地址空间巨大

2，精简报文结构

3，实现自动配置和重新编址

4，支持层次化网络结构

5，支持端对端安全

6，更好的支持 QoS

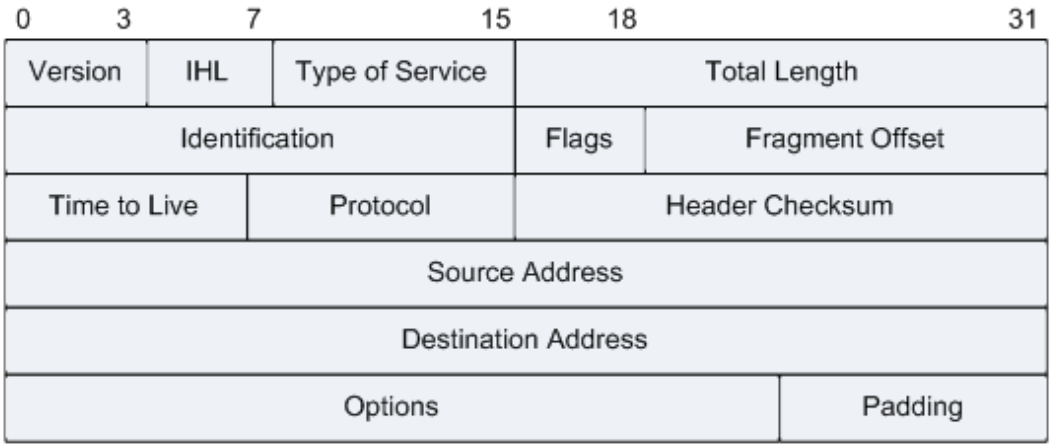
7，支持移动特性

二，IPV6 的报文格式，明白其中主要字段的含义

1， IPV4 报文格式及大小

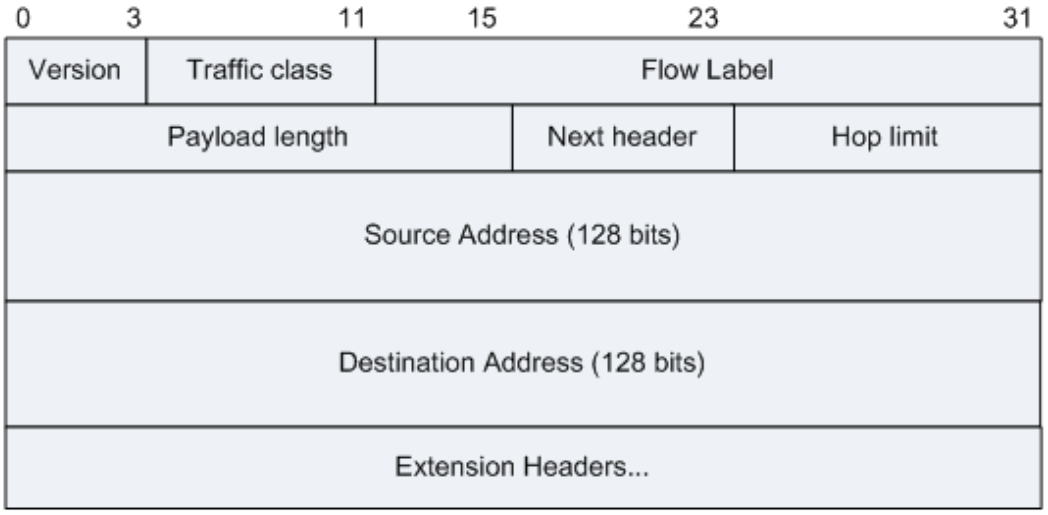
IPV4 报文大小 20 字节~60 字节

图1 IP头格式



IPV6

图1 IPv6报文头格式



2， Ipv4 QOS 与 IPV6 qos 的区别

IPV6 的 qos 比 IPV4 的 qos 多了四字节的 flow label 流标签

3， Next-header 的作用及常见的 next-header

Next-header 为扩展报头，可以表示上层协议

ICMPv6 58

表1 IPv6扩展报头

报头类型	代表该类报头的Next Header字段值	描述
逐跳选项报头	0	该选项主要用于为在传送路径上的每跳转发指定发送参数，传送路径上的每台中间节点都要读取并处理该字段。逐跳选项报头目前的主要应用有以下三种： • 用于巨型载荷（载荷长度超过65535字节）。 • 用于设备提示，使设备检查该选项的信息，而不是简单的转发出去。 • 用于资源预留（RSVP）。
目的选项报头	60	目的选项报头携带了一些只有目的节点才会处理的信息。目前，目的选项报文头主要应用于移动IPv6。
路由报头	43	路由报头和IPv4的Loose Source and Record Route选项类似，该报头能够被IPv6源节点用来强制数据包经过特定的设备。
分段报头	44	同IPv4一样，IPv6报文发送也受到MTU的限制。当报文长度超过MTU时就需要将报文分段发送，而在IPv6中，分段发送使用的是分段报头。
认证报头	51	该报头由IPsec使用，提供认证、数据完整性以及重放保护。它还对IPv6基本报头中的一些字段进行保护。
封装安全净载报头	50	该报头由IPsec使用，提供认证、数据完整性以及重放保护和IPv6数据报的保密，类似于认证报头。

扩展报头。IPv6取消了IPv4报头中的选项字段，并引入了多种扩展报文头，在提高处理效率的同时还增强了IPv6的灵活性，为IP协议提供了良好的扩展能力。当超过一种扩展报头被用在同一个分组里时，报头必须按照下列顺序出现：

- IPv6基本报头
- 逐跳选项扩展报头
- 目的选项扩展报头
- 路由扩展报头
- 分片扩展报头
- 授权扩展报头
- 封装安全有效载荷扩展报头
- 目的选项扩展报头（指那些将被分组报文的最终目的地处理的选项。）
- 上层扩展报头

不是所有的扩展报头都需要被转发路由设备查看和处理的。路由设备转发时根据基本报头中Next Header值来决定是否要处理扩展头。

除了目的选项扩展报头出现两次（一次在路由扩展报头之前，另一次在上层扩展报头之前），其余扩展报头只出现一次。

4， Ipv6 对移动技术的支持用到了哪些扩展报头

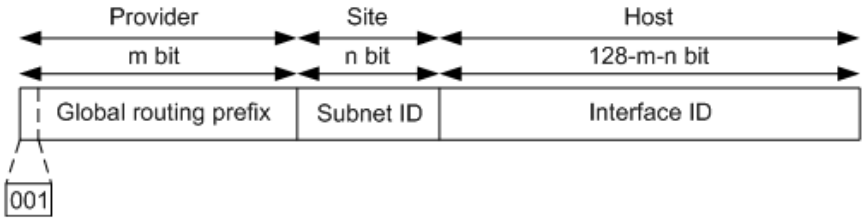
扩展报头 60 目的选项扩展头

三，IPv6 地址

1，单播地址

A，全球单播

图2 全球单播地址格式



B，Link-local(作用；如何产生；EUI-64 方法）

Link-local 地址的作用就是可以充当 IPV6 地址中的下一跳作用，只在本链路上有效。

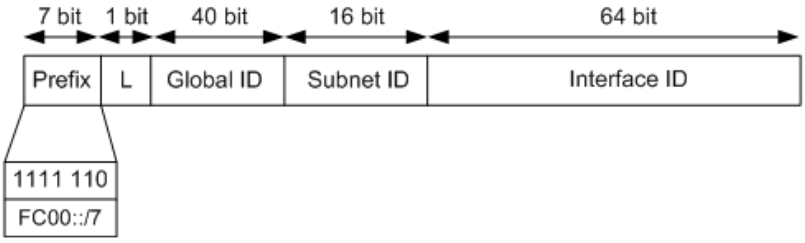
Link-local 地址可以自动生成，也可以手动配置，如果需要自动生成的话，可以通过全球单播地址，就会自动生成 link-local 地址，或者把 48 位的 MAC 地址从中间分开，插入 FFFE，将第七位的 0 改为 1，就形成

64 位的接口标识，这个地址就是 EUI-64，再加上 64 位的网络位就形成了 link-local 地址

C， 唯一本地：FC00：：/7

相当于 IPV4 中的私有地址 127.0.0.1

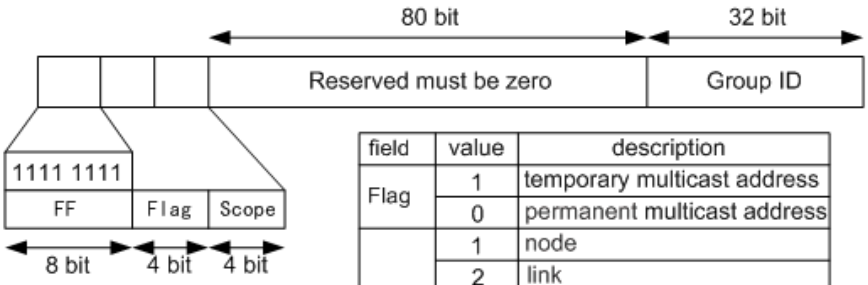
图4 唯一本地地址格式



2，组播地址

FF00：：/8 IPV6 组播地址前缀

图5 IPv6组播地址格式



field	value	description
Flag	1	temporary multicast address
	0	permanent multicast address
Scope	1	node
	2	link
	4	management
	5	site
	8	organization
	E	global
	the rest	unsigned or reserved

A，常见的永久组地址

预定义组播地址

- Node-local
  - 所有节点的组播地址： FF01:0:0:0:0:0:0:1
  - 所有路由器的组播地址： FF01:0:0:0:0:0:0:2
- Link-local
  - 所有节点的组播地址： FF02:0:0:0:0:0:0:1
  - 所有路由器的组播地址： FF02:0:0:0:0:0:0:2
  - Solicited-Node组播地址： FF02:0:0:0:0:1:FFXX:XXXX
  - 所有OSPF路由器组播地址： FF02:0:0:0:0:0:0:5
  - 所有OSPF的DR路由器组播地址： FF02:0:0:0:0:0:0:6
  - 所有RIP路由器组播地址： FF02:0:0:0:0:0:0:9
  - 所有PIM路由器组播地址： FF02:0:0:0:0:0:0:13

B，被请求节点组播地址(作用)

被请求节点组播地址由前缀 FF02::1:FF00:0/104 和单播地址的最后 24 位组成。

会根据单播地址生成，全球单播地址，链路本地地址

用于地址解析和重复地址检测

3，任播地址（和单播地址取值范围相同）

任播地址设计用来在给多个主机或者节点提供相同服务时提供冗余功能和负载分担功能。将一个单播地址分配给多个节点或者主机，这样在网络中如果存在多条该地址路由，当发送者发送以任播地址为目的 IP 的数据报文时，发送者无法控制哪台设备能够收到，这取决于整个网络中路由协议计算的结果。这种方式可以适用于一些无状态的应用，例如 DNS 等。目前 IPv6 中任播主要应用于移动 IPv6。

A，不能作为源地址

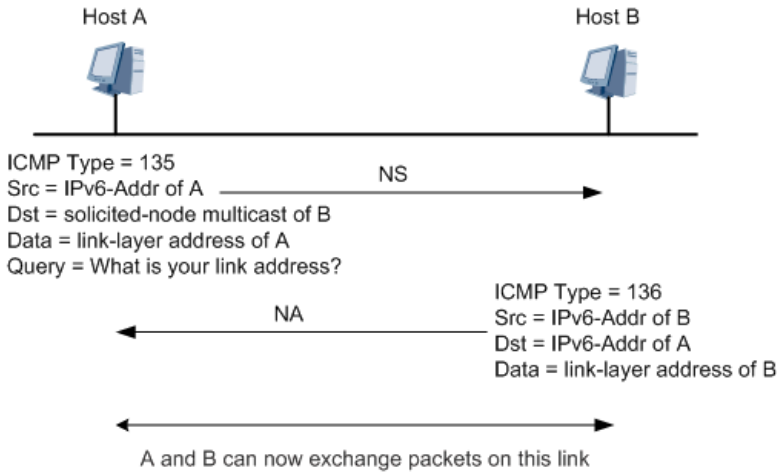
B，不能配置在主机上

四，ICMPv6（NDP 协议）通过 ICMPv6 来实现

RS：133　RA：134　NS：135　NA：136　重定向：137

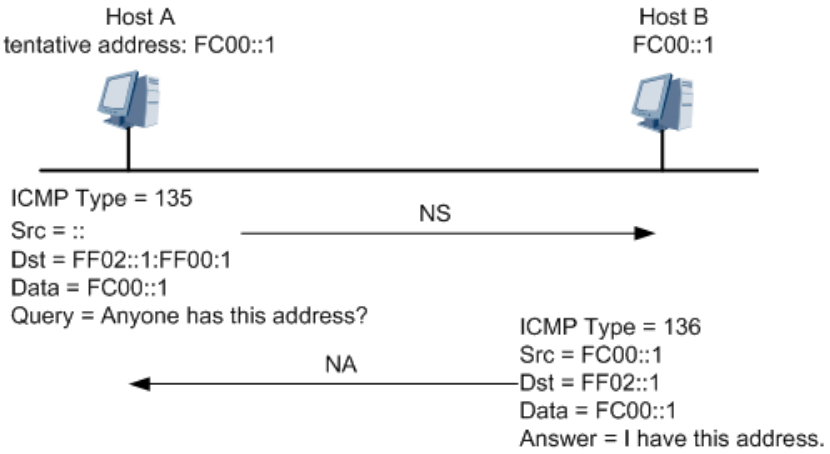
1，地址解析

图1 IPv6地址解析过程



2，重复地址检测

图3 重复地址检测示例

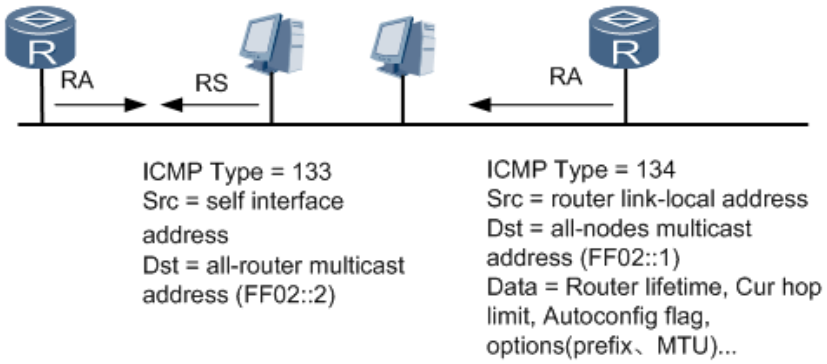


- 如果 Host B 发现 FC00::1 是自身的一个试验地址，则 Host B 放弃使用这个地址作为接口地址，并且不会发送 NA 报文。
- 如果 Host B 发现 FC00::1 是一个已经正常使用的地址，Host B 会向 FF02::1 发送一个 NA 报文，该消息中会包含 FC00::1。这样，Host A 收

到这个消息后就会发现自身的试验地址是重复的。Host A 上该试验地址不生效，被标识为 duplicated 状态

3，路由器发现

图4 路由器发现示例

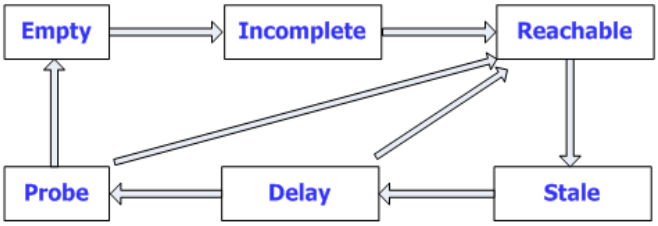


4，跟踪邻居状态

RFC2461中定义了5种邻居状态，分别是：未完成（Incomplete）、可达（Reachable）、陈旧（Stale）、延迟（Delay）、探查（Probe）。

邻居状态之间具体迁移过程如图2所示，其中Empty表示邻居表项为空。

图2 邻居状态迁移示例

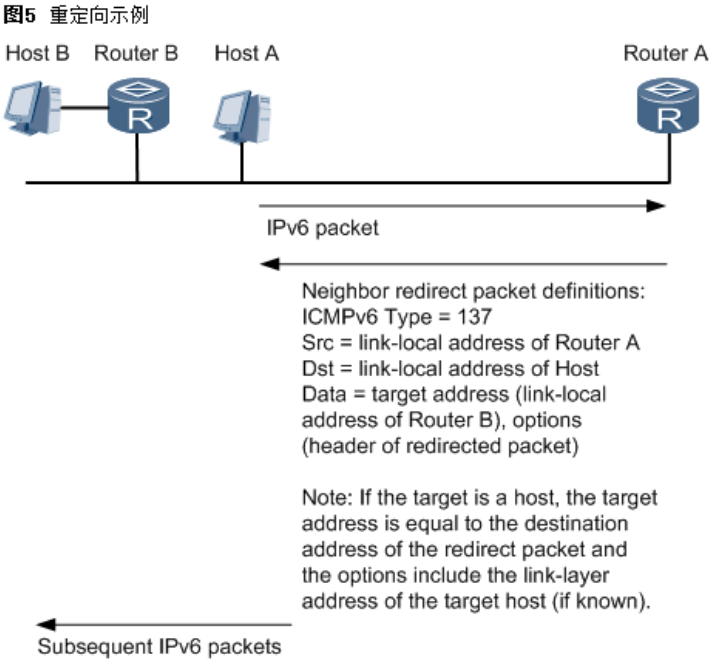


下面以A、B两个邻居节点之间相互通信过程中A节点的邻居状态变化为例（假设A、B之前从未通信），说明邻居状态迁移的过程。

1. A先发送NS报文，并生成缓存条目，此时，邻居状态为Incomplete。
2. 若B回复NA报文，则邻居状态由Incomplete变为Reachable，否则固定时间后邻居状态由Incomplete变为Empty，即删除表项。
3. 经过邻居可达时间，邻居状态由Reachable变为Stale，即未知是否可达。
4. 如果在Reachable状态，A收到B的非请求NA报文，且报文中携带的B的链路层地址和表项中不同，则邻居状态马上变为Stale。
5. 在Stale状态若A要向B发送数据，则邻居状态由Stale变为Delay，并发送NS请求。
6. 在经过一段固定时间后，邻居状态由Delay变为Probe，其间若有NA应答，则邻居状态由Delay变为Reachable。
7. 在Probe状态，A每隔一定时间间隔发送单播NS，发送固定次数后，有应答则邻居状态变为Reachable，否则邻居状态变为Empty，即删除表项。

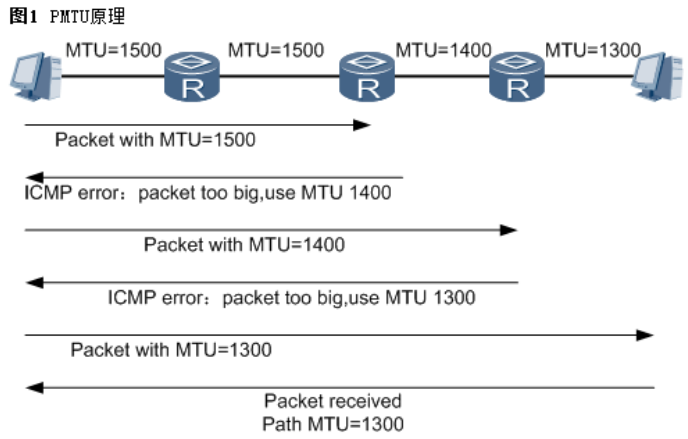
5，ICMP 的重定向

当网关路由器指导更好的转发路径时，会以重定向的方式告知主机



6，PMTU 的工作过程

ICMPv6 中间节点不支持分片，只能在源端进行分片，提高转发效率



7，阐述两台 ipv6 客户端的互访过程

就是地址解析中 NS NA 的过程

五，IPv6 地址获得有哪些方式

1，手动配置：手动配置 IPv6 地址/前缀及其他网络配置参数（DNS、NIS、SNTP 服务器地址等参数）。

2，无状态自动地址分配：由接口 ID 生成链路本地地址，再根据路由通告报文 RA（Router Advertisement） 包含的前缀信息自动配置本机地址。

3，有状态自动地址分配：即 DHCPv6 方式。DHCPv6 又分为如下两种:

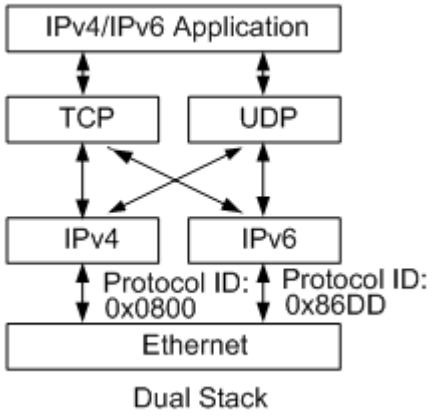
A，DHCPv6 有状态自动分配：DHCPv6 服务器自动分配 IPv6 地址/PD 前缀及其他网络配置参数（DNS、NIS、SNTP 服务器地址等参数）。

B，DHCPv6 无状态自动分配：主机 IPv6 地址仍然通过路由通告方式自动生成，DHCPv6 服务器只分配除 IPv6 地址以外的配置参数，包括 DNS、NIS、SNTP 服务器等参数。

六，ipv4 向 ipv6 的过渡技术

1，双栈技术

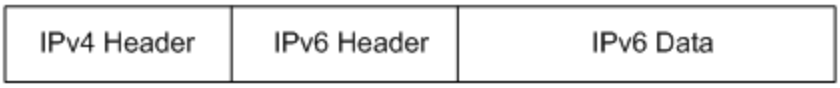
节点同时支持 IPv4 和 IPv6 协议栈



2，IPv6 over IPv4 隧道

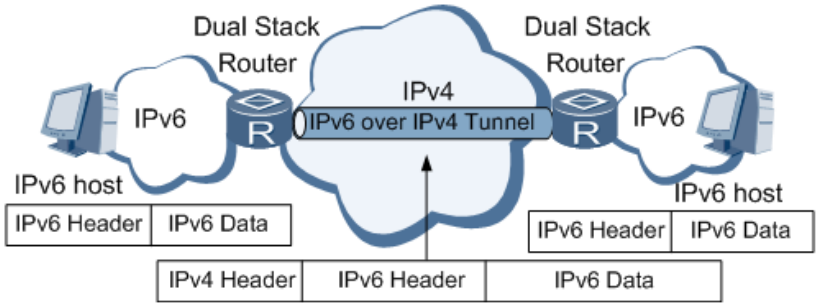
过渡初期使用

图2 IPv6 over IPv4手动隧道封装格式



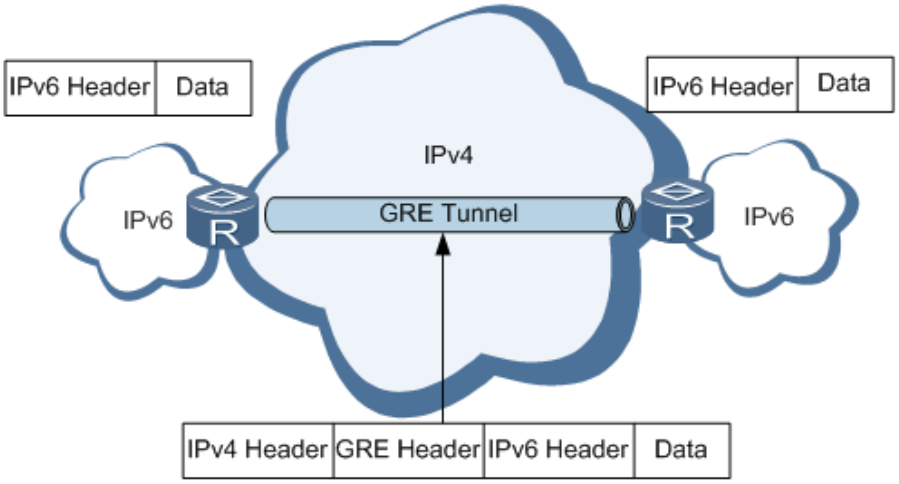
通过隧道技术，使 IPv6 报文在 IPv4 网络中传输

图1 IPv6 over IPv4 隧道原理



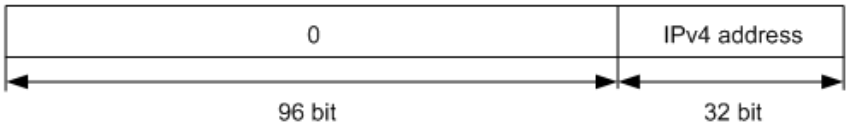
手动隧道包括 IPv6 over IPv4 手动隧道和 IPv6 over IPv4 GRE 隧道

图3 IPv6 over IPv4 GRE隧道



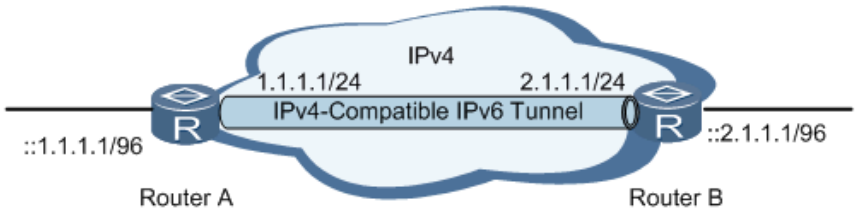
自动隧道包括 IPv4 兼容 IPv6 自动隧道、6to4 隧道和 ISATAP 隧道

图4 IPv4兼容IPv6地址



下面以图5为例说明IPv4兼容IPv6自动隧道的转发机制：

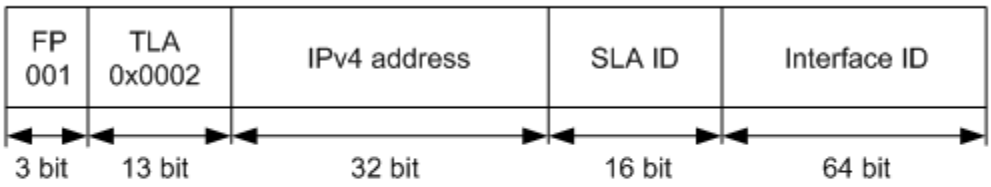
图5 IPv4兼容IPv6隧道



3，IPv4 over IPv6 隧道

过渡后期使用

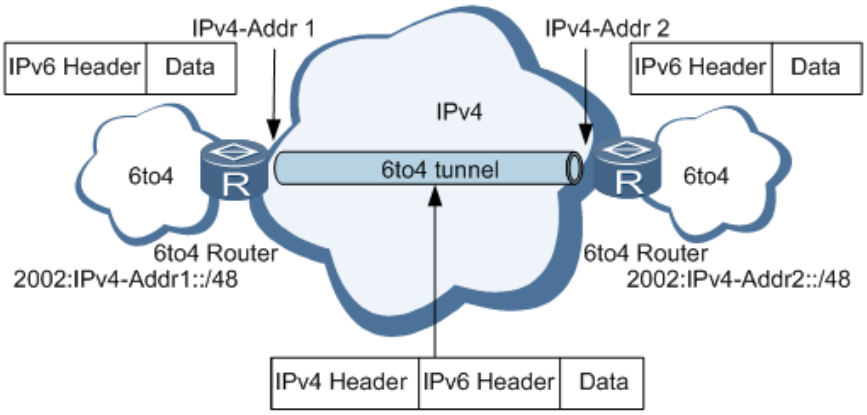
图6 6to4地址



- FP：可聚合全球单播地址的格式前缀（Format Prefix），其值为001。
- TLA：顶级聚合标识符（Top Level Aggregator），其值为0x0002。
- SLA：站点级聚合标识符（Site Level Aggregator）。

通过隧道技术，使 IPv4 报文在 IPv6 网络中传输

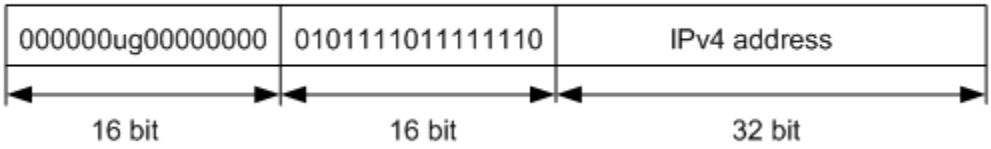
图7 6to4隧道示例一



4， ISATAP 隧道

ISATAP（Intra-Site Automatic Tunnel Addressing Protocol）是另外一种自动隧道技术。ISATAP 隧道同样使用了内嵌 IPv4 地址的特殊 IPv6 地址形式，只是和 6to4 不同的是，6to4 是使用 IPv4 地址做为网络前缀，而 ISATAP 用 IPv4 地址做为接口标识。其接口标识符格式如图 10 所示：

图10 ISATAP地址接口标识格式



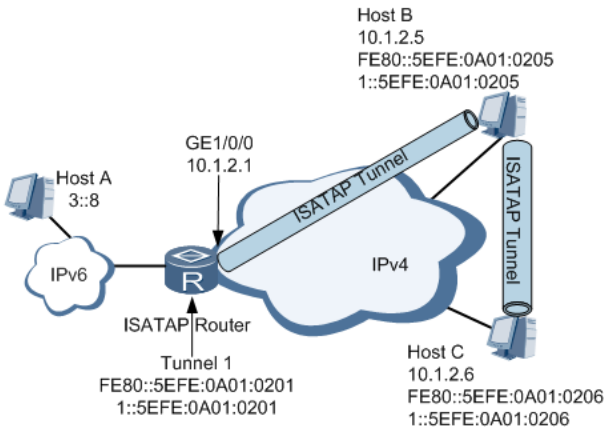
如果 IPv4 地址是全局唯一的，则 u 位为 1，否则 u 位为 0。g 位是 IEEE 群体/个体标志。由于 ISATAP 是通过接口标识来表现的，所以，ISATAP 地址有全局单播地址、链路本地地址、ULA 地址、组播地址等形式。ISATAP 地址的前 64 位是通过向 ISATAP 路由器发送请求来得到的，它可以进行地址自动配置。在 ISATAP 隧道的两端设备之间可以运行 ND 协议。ISATAP 隧道将 IPv4 网络看作一个非广播的点到多点的链路（NBMA）。

ISATAP 过渡机制允许在现有的 IPv4 网络内部署 IPv6，该技术简单而且扩展性很好，可以用于本地站点的过渡。ISATAP 支持 IPv6 站点本地路由和全局 IPv6 路由域，以及自动 IPv6 隧道。ISATAP 同时还可以与 NAT 结合，从而可以使用站点内部非全局唯一的 IPv4 地址。典型的 ISATAP 隧道应用是在站点内部，所以，其内嵌的 IPv4 地址不需要是全

局唯一的。

图11为ISATAP隧道一个典型应用场景：

图11 ISATAP隧道示例



如上图所示，在IPv4网络内部有两个双栈主机Host B和Host C，它们分别有一个私网IPv4地址。要使其具有ISATAP功能，需要进行如下操作：

1. 首先配置ISATAP隧道接口，这时会根据IPv4地址生成ISATAP类型的接口ID。
2. 根据接口ID生成一个ISATAP链路本地IPv6地址，生成链路本地地址以后，主机就有了在本地链路上进行IPv6通信的能力。
3. 进行自动配置，主机获得IPv6全球单播地址、ULA地址等。
4. 当主机与其它IPv6主机进行通讯时，从隧道接口转发，将从报文的下一跳IPv6地址中取出IPv4地址作为IPv4封装的目的地址。如果目的主机在本站点内，则下一跳就是目的主机本身，如果目的主机不在本站点内，则下一跳为ISATAP路由器的地址。

MPLS VPN

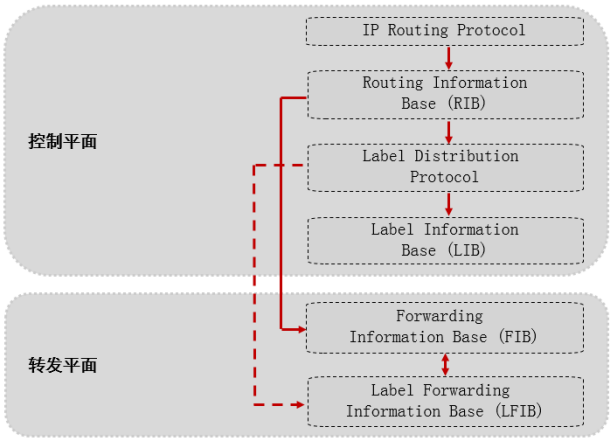
2016 年 7 月 21 日
11:40

一，MPLS：多协议标签交换

1，为什么需要使用 MPLS

由于路由器的转发效率较慢，开发 MPLS 加快数据的转发
封装在二层与三层之间，在路径转发时，不需要查三层的 IP，只需要查找到 2.5 层，所以可以提高了转发效率
路由的传递方向就是标签的分配方向，

2，MPLS 中的转发表项



1，RIB：路由表，控制层面 RIB 中有递归过程

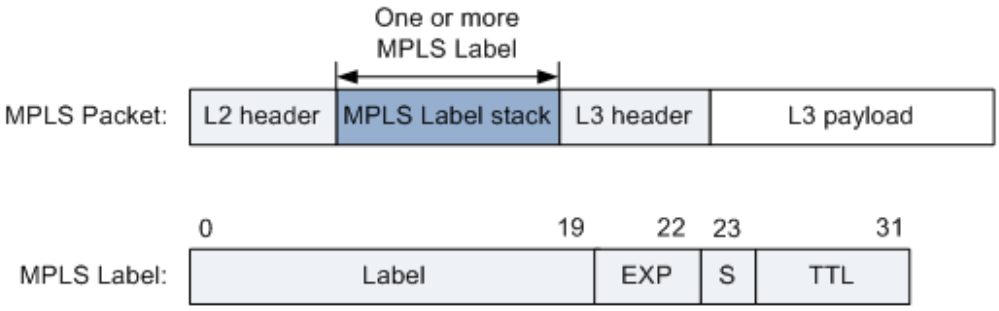
2，LIB：标签信息表，控制层面 标签与 FEC 的对应关系 有所有的标签，无论是优的还是次优的

3，FIB：转发信息数据库，转发层面 FIB 表中有递归后的下一跳

4，LFIB：标签转发数据库，转发层面 标签与 FEC 的对应关系 只有最最优的标签，指导数据包转发

3， MPLS 的报文封装位置及格式

图1 MPLS报文格式



，

1，标签的范围

0~15 为特殊标签

16~1023 为静态分配标签的范围

1024 以上为动态分配标签，动态分配标签的方式有三种：LDP，MP-BGP，RSVP-TE

3 号标签标示隐式空标签

0 号标签标示显示空标签

2，TTL 处理的两种方式

Uniform：统一方式 保持 IP 和 MPLS 的 TTL 值相同，可以统计 MPLS 中有多少个路由器，每经过一台路由器 TTL 值减 1，无论是 MPLS 还是 IP

Pipe：管道方式 只会在 MPLS 的 ingress 和 egress 路由器 TTL 减 1，无法了解 MPLS 中有多少个节点。

4，MPLS 中的术语

1， LER、LSR

LER：标签边界路由器
LSR：标签交换路由器

2，push、swap、pop

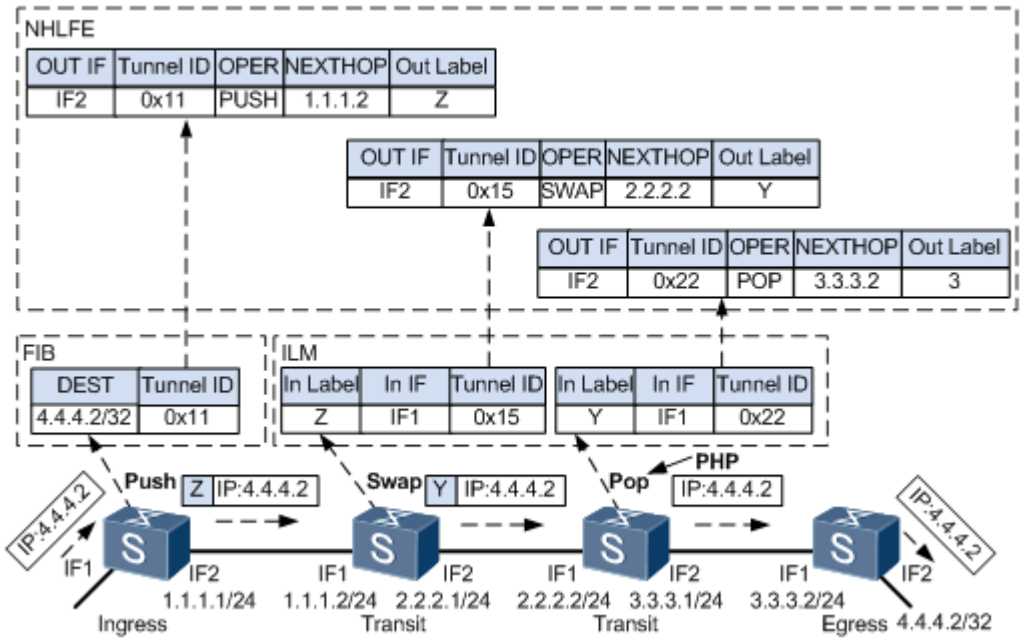
push：ingress 标签压入
transit：swap 标签交换
pop：egress 标签弹出

3，Ingress、transit、Egress

Ingress：第一个封装标签的路由器，就是数据转发的方向
Transit：只负责传输的作用
Egress：标签弹出的路由器，路由和标签传递的方向

4，详述 MPLS 数据包的转发过程

图2 MPLS详细转发过程



控制层面：下游向上游分配标签
转发层面：ingress（R1）先通过查找 FIB，查找 tunnel id，再看 tunnel id 对应的 NHLFE，执行 push 动作，压入标签，确定出接口，下一跳和出标签。数据包传给 R2（transit），先查找 ILM，找到刚才收到的标签和对应的 tunnel id，再通过 tunnel id 查找 NHLFE，执行 swap 动作，确定出接口，下一跳和出标签。数据包传给 R3（transit），先查找 ILM，找到刚才收到的标签和对应的 tunnel id，再通过 tunnel id 查找 NHLFE，执行 pop 动作，次末跳弹出，执行 IPV4 转发。R4（egress）收到数据包，拆封装，拆掉二层之后是三层，查找 FIB 进行转发。
Tunnel ID：为 0 则进行 IPV4 转发，如果非 0，则通过 NHLFE 进行 MPLS 转发

1，LIB：标签信息数据库

2，NHLFE：下一跳标签转发条目

3，ILM：入接口的标签映射

二，MPLS LDP：标签分发协议，自动构建 LSP（标签交换隧道）

LDP 只会为 IGP 分配标签，而且只为 32 位主机路由分配标签

1， LDP 的消息类型

A，发现（Discovery）消息：用于通告和维护网络中 LSR 的存在。

B，会话（Session）消息：用于建立、维护和终止 LDP 对等体之间的会话。

C， 通告（Advertisement）消息：用于创建、改变和删除 FEC 的标签映射。

E， 通知（Notification）消息：用于提供建议性的消息和差错通知。

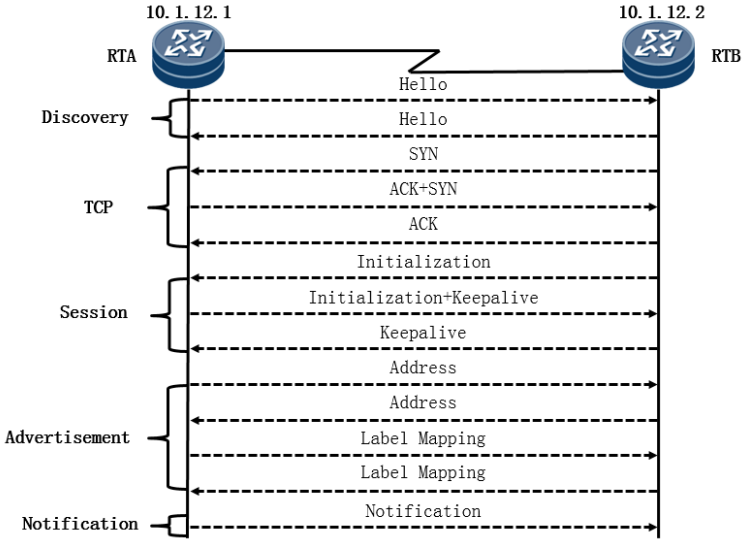
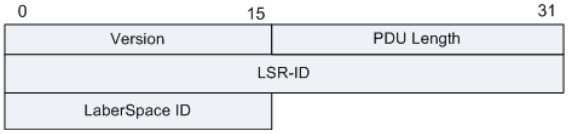
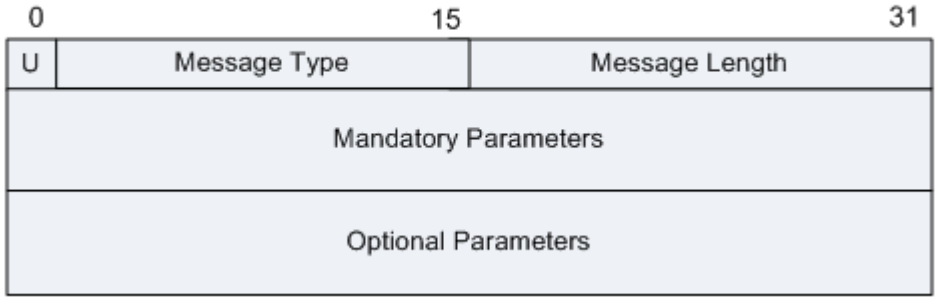


图1 LDP协议消息头部格式



字段	长度	说明
Version	2字节	表示版本号。目前LDP的版本号始终为1。
PDU Length	2字节	表示PDU的总长度，包括LDP ID和整组LDP消息，不包括Version和PDU Length字段。 例如某个LDP报文中包含3个Hello消息，则该报文的PDU length = 3 * Message length。
LSR-ID	4字节	LDR-ID标识一台LSR，必须全局唯一。
LaberSpace ID	2字节	标识了LSR内的标签空间。对于平台范围标签空间，这些数值都应当为0。
Bunch of messages	变长	是一组LDP消息的集合，可以是一个或者多个LDP消息。 • 当LDP报文以UDP方式传输时，“Bunch of messages”只能是Hello消息的集合。 • 当LDP报文以TCP方式传输时，“Bunch of messages”可以是除Hello消息外任意类型的LDP消息的集合。

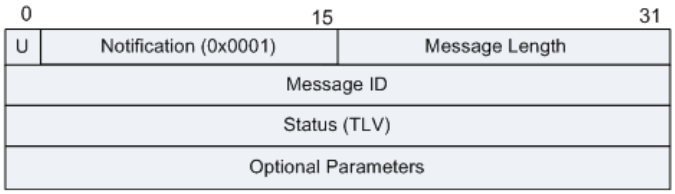
图2 LDP消息格式



通告（Notification）消息

LSR发送通告消息来通知重要事件到LDP对等体。通告消息通知致命错误或提供咨询信息，如处理LDP消息的结果或LDP会话的状态。

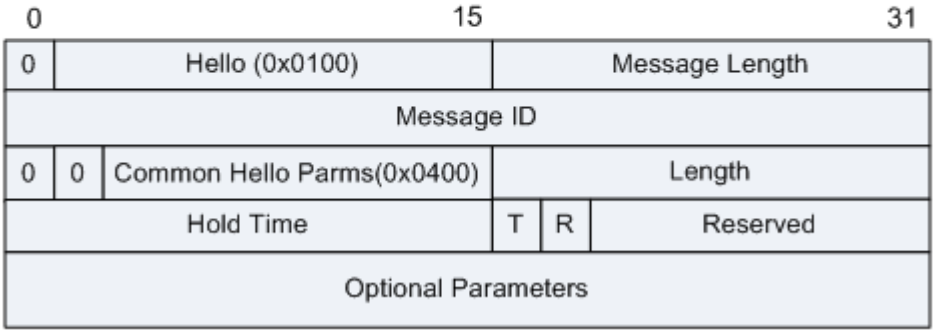
图3 Notification消息格式



Hello消息

用于通告和维护网络中LSR的存在。

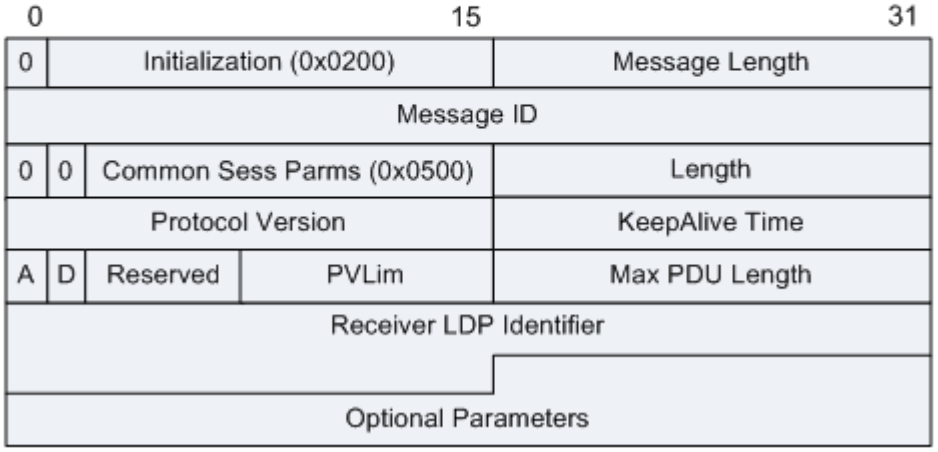
图4 Hello消息格式



Initialization消息

LDP的Initialization消息在LDP回家建立阶段发送，格式如下：

图5 Initialization消息格式

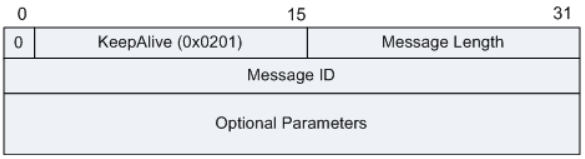


Keepalive 消息每 5 秒发送一次，保活。

KeepAlive消息

Keepalive消息无Mandatory Parameters字段及后面的字段，用于维护SESSION的状态，所以这里不需要什么特别的内容，只要对方知道自己还存在就好。

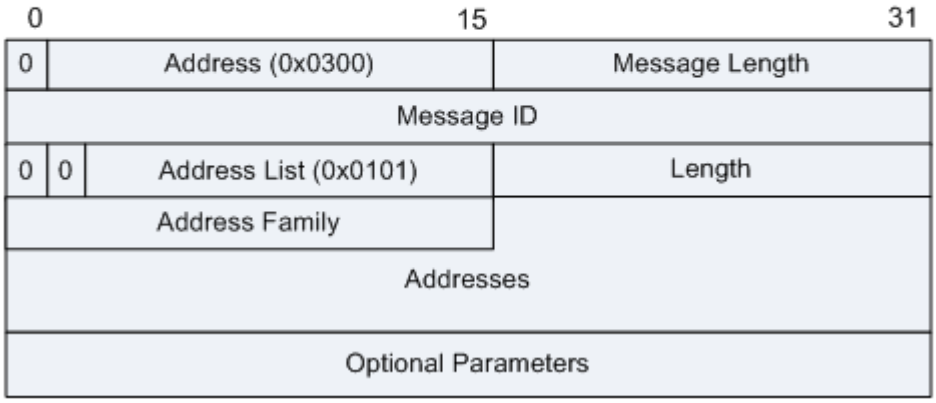
图6 KeepAlive消息格式



地址（Address）消息

Address消息用于LSR发送地址消息到LDP邻居，以公告其接口地址。

图7 Address消息格式



地址撤销（Address Withdraw）消息

LSR发送Address Withdraw消息到LDP对等体，以撤销之前公告的接口地址。当接口地址被删除或接口down后，就会发送Address Withdraw消息。

图8 Address Withdraw消息格式

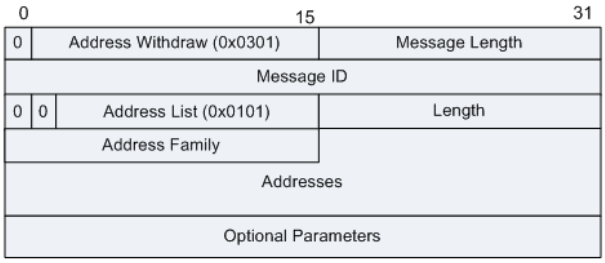


图9 Label Mapping消息格式

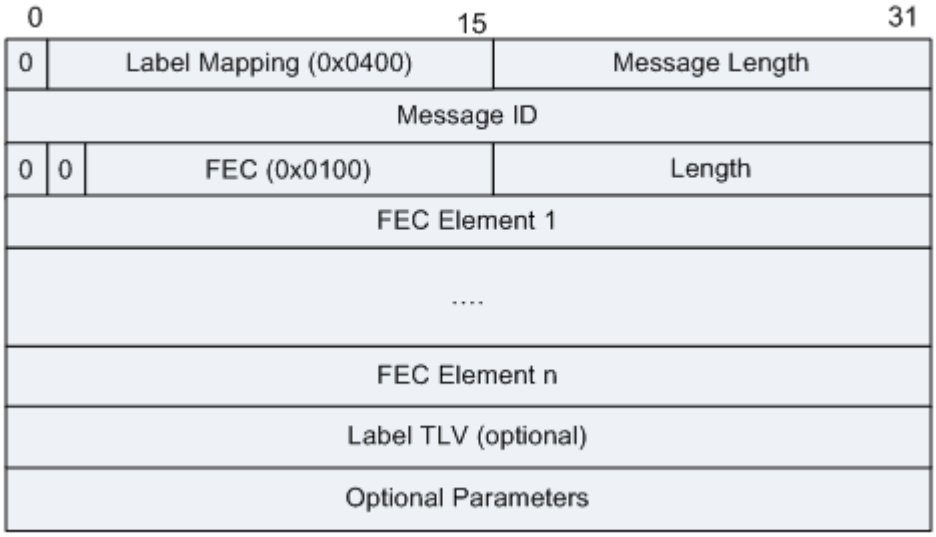
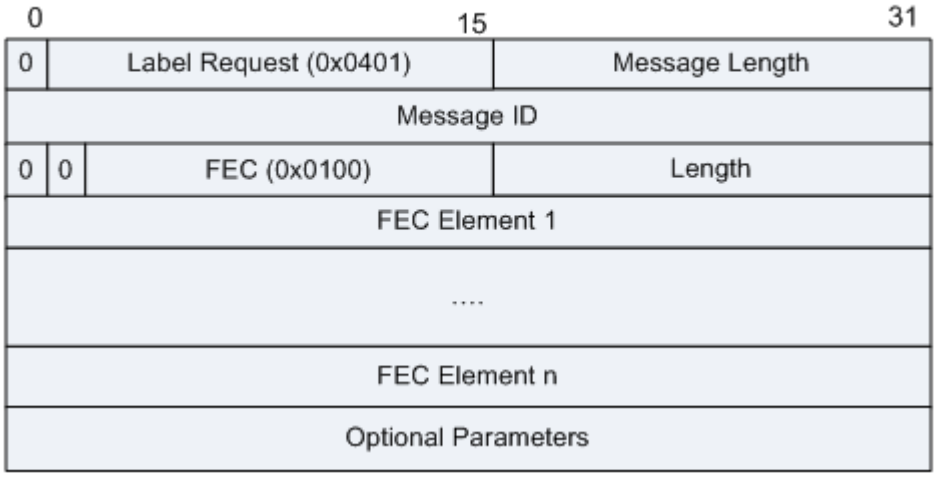


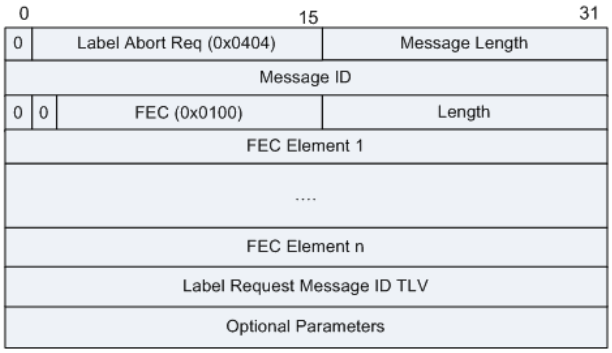
图12 Label Request消息格式



Label Abort Request消息

上游LSR发送了Label Request消息后但还没有收到Label Mapping消息前，发现FEC对应的下一跳变化了或者其他可能的原因需要发送新的Label Request消息时，上游会向下游发送Label Abort Request消息。

图14 Label Abort Request消息格式



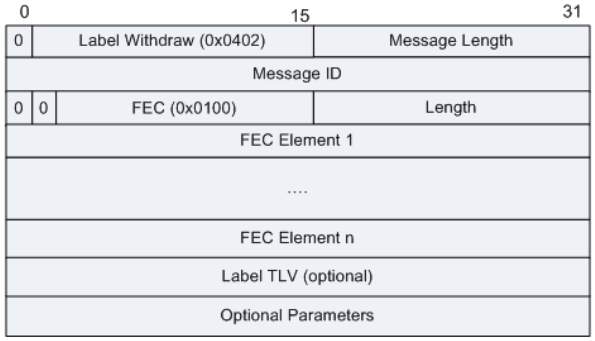
Label Withdraw消息

Label Withdraw消息一般由下游LSR发往上游LSR，通知上游LSR之前通告的与某FEC对应的Label不再使用，上游LSR需要解除Label和FEC的映射关系。

下列情况下会发送Label Withdraw消息：

- 下游节点不再有某条FEC，如果已经为该FEC发送了Label Mapping消息，则发送Label Withdraw消息；
- 下游单方面的决定不再使用标签转发时也会发送Label Withdraw消息。

图16 Label Withdraw消息格式



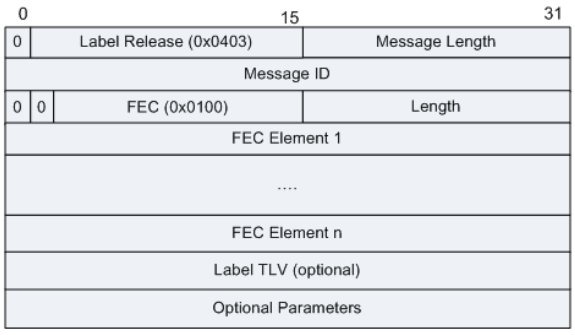
Label Release消息

Label Release消息一般由上游发往下游，通知撤销Label和FEC的邦定，该消息相当于Label Request消息的逆过程。

在下列情况下会发送Label Release消息：

- 上游LSR的标签保持方式是保守方式，发送Label Mapping消息的LSR不再是FEC的下一跳时，上游LSR需要发送Label Release消息来撤销Label和FEC的映射关系；
- 上游LSR的标签保持方式是保守方式，从不是FEC的下一跳收到Label Mapping消息后，上游LSR需要发送Label Release消息；
- LSR收到Label Withdraw消息后需要发送Label Release消息。

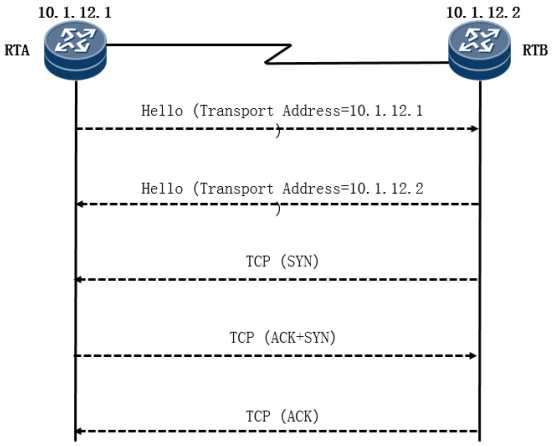
图19 Label Release消息格式



2，LDP 邻居建立的过程

A，发现阶段（UDP；646；hello，目的地址为 224.0.0.2）：发现对端的 LSR-id

B， 会话建立阶段（TCP；646；大向小）:TCP 三次握手，地址大的向地址小的建立 TCP 连接

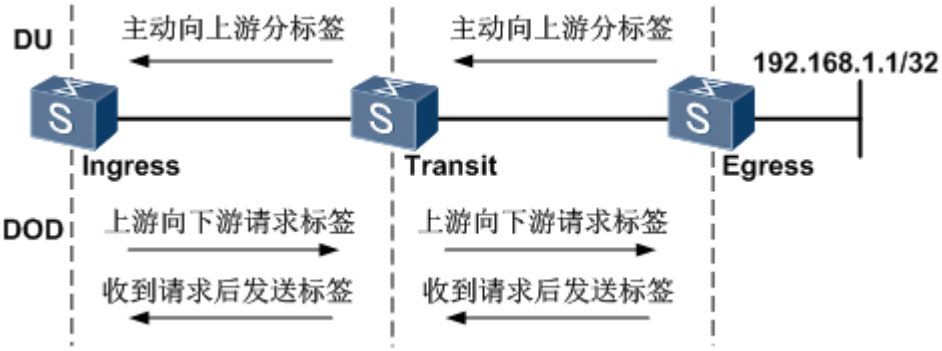


3，LDP 标签的发布与管理方式(画图说明)

A，标签发布方式

下游自主 DU：主动给上游为每一个 FEC 分配一个标签，开销大

下游按需 DOD（什么时候会用到此种方式）：如果下游路由器上的路由太多，就需要 DOD 方式，来节省开销，只有上游路由器发送请求时才会为对应的 FEC 分配标签。



B，标签分配控制方式

独立 Independent：自主的给上游分配标签，无论是否下游给自己分配了标签，都会给上游分配标签。但可能会造成标签的断裂

有序 Ordered：只有收到下游给自己分配的标签，才会给自己的上游分配标签。

C，标签的保持方式

自由 Liberal：当收到对应一个 FEC 的两个标签时，会保留下来，留作备份。

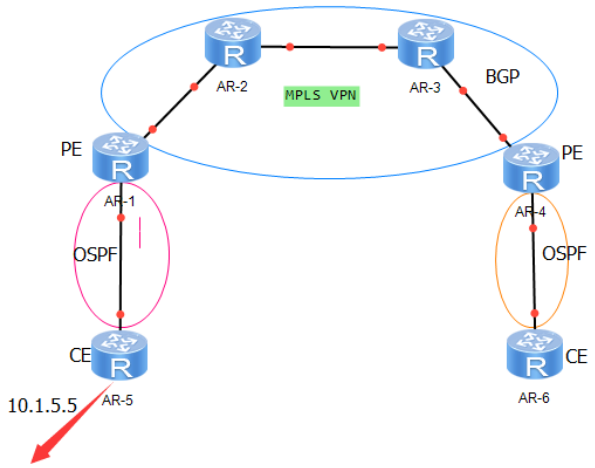
保守 Conservative：当收到对应一个 FEC 的两个标签时，只会保留最优的标签。

目前设备支持如下组合方式：

- 下游自主方式（DU）＋ 有序标签分配控制方式（Ordered）＋ 自由标签保持方式（Liberal），该方式为缺省方式。
- 下游按需方式（DoD）＋ 有序标签分配控制方式（Ordered）＋ 保守标签保持方式（Conservative）。

三，MPLS VPN

1，画图说明 MPLS VPN 的工作过程

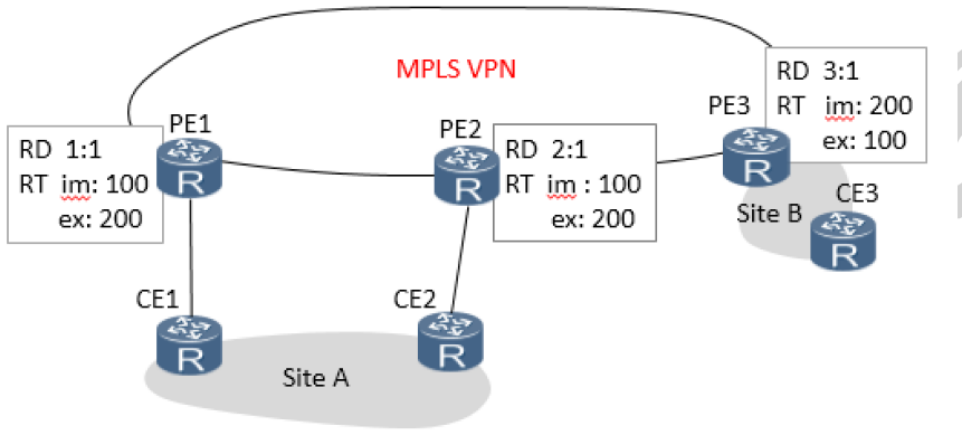


在 PE 上创建实例 RD： RT：
PE 和 CE 上实例启用路由协议【PE—CE 间路由协议】
PE 与 PE 上创建 VPNV4 邻居
R1 和 R4 上把实例的路由引入 BGP 中，同样把 BGP 的路由引入实例
如果 PE 与 CE 间运行的是 BGP，那么就不需要互相引入
R1 的 BGP VPNV4 routing 表中有 96 位的 VPN 路由
R4 上检查 RT 值，确定对方的 export 方向的 RT 值要和自己的 import 方向的 RT 值要相同，才能收 R1 的路由

2，RT 与 RD 的作用

RD：区分实例，标记路由，只在本地有效，区分不同站点的相同路由
RT：对路由进行控制，控制路由的导入与导出

场景：分析RD及RT的对应关系



3， 什么是 VPNV4 路由

通过把 64bit 的 RD 值+32bit 的 IPV4 的路由就形成了 96bit 的 VPNV4 路由

4， 常见的扩展团体属性有哪些

RT　SoO　cost-community

5， 命令 policy-vpn-target 的作用

收到一条 VPNV4 路由，要查看 RT 值是否和自己的 import 方向的 RD 值是否相同，如果不同则不收，如果关闭了这个功能，那么收到了一条 VPNV4 的路由首先会收入自己的 VPNV4 路由表，再查看是否对应自己的 RT 值，如果不匹配则只存在于自己的 VPNV4 路由表中，不会放入实例路由表。

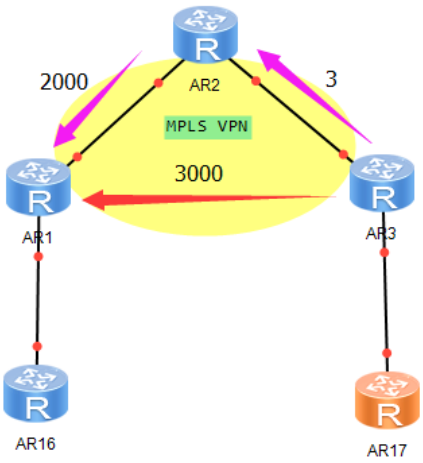
6， 如何使用 RT 值来搭建 Hub-Spork 的网络环境

7， 双标签的作用，由哪种协议分配

公网标签的作用：负责数据包在公网的传输

私网标签的作用：指导数据包进入相应的实例

公网标签（外层标签）由 LDP 分配，私网标签（内层标签）由 MP-BGP 分配，如果只有外层标签的话，由于 LDP 分配的 3 号标签会进行次末跳弹出，进行 IPV4 转发，但是在 MPLS VPN 中并不识别 IPV4 的路由，就无法转发数据包而丢弃，所以这时需要通过内层标签，进行转发，通过标签与实例的对应关系，就知道转发到哪个实例。

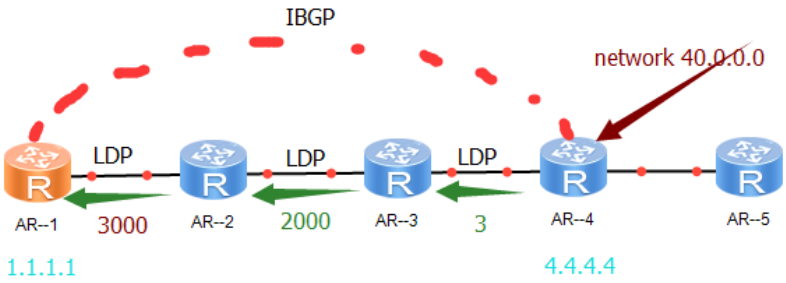


8，隧道迭代是什么意思，以及作用

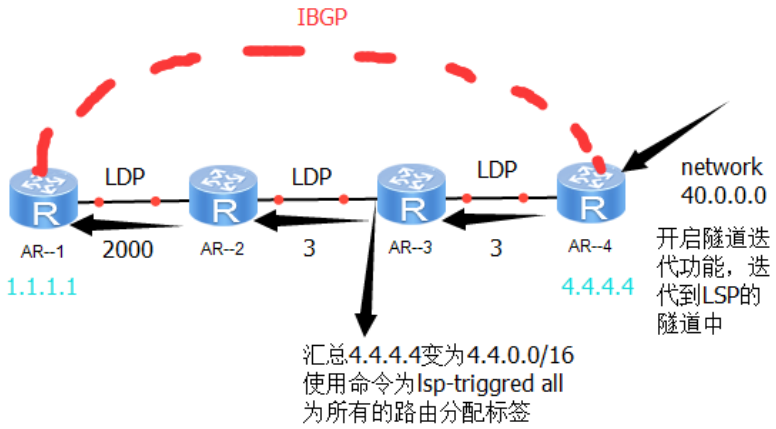
非标签的公网路由和静态路由不会进行隧道迭代，迭代到相应的 LSP

可以解决路由黑洞

40 的路由 network 到 BGP 中，40 的路由是没有标签的，所以 40 的路由无法通过 LSP 传递到 R1 上，需要使用一条命令 `route recursive-lookup tunnel`，把所有的路由都迭代到 LSP 中，那么 40 的路由就可以通过 4.4.4.4 的 LSP 进行转发



9，汇总会对 MPLS VPN 产生什么影响



如果对 MPLS VPN 进行汇总，那就会造成 LSP 的断裂，形成黑洞

在 R4 上 network40.0.0.0，并开启隧道迭代的功能，让 40.0.0.0 的路由放入 LSP 中来传，R4 给 R3 分配的标签为 3 号标签，在 R3 的接口上汇总 4.4.4.4 的路由为 4.4.0.0/16，默认 LDP 只为 32 位路由分配标签，可以使用命令 lsp-triggred all 为所有的路由都分配一个标签，由于 4.4.0.0/16 是重新产生的路由，又会产生一个 3 号标签，R2 给 R1 分配的为 2000 的标签，当 R1 收到了 40.0.0.0 的路由时，想要访问 40.0.0.0，通过查找标签隧道通过 MPLS VPN 进行转发，R1 传给 R2，R2 上次末跳弹出，传给 R3，R3 上需要查找 FIB 表，40.0.0.0 的下一跳是谁，但 R3 并没有 40.0.0.0 的路由，就会产生丢包

10，PHP 的作用及没有 PHP 会有什么影响；显式空与隐式空

POP 次末跳弹出，提前一跳弹出，始发路由器给上游分配一个 3 号标签，如果没有 pop 机制需要先查 LFIB 再查 FIB 表，减少一次查表次数，节省开销。

四，MPLS 中 LSP 的备份方式

（详见视频）

1， FRR

2，先讲未使用 FRR 之前当链路发生切换时的情况，再讲使用 FRR 的好处，最后讲 FRR 的配置。

3，FRR 断链回切，同步

Feature

2016 年 7 月 21 日

12:19

一，SNMP

1，SNMP 的作用

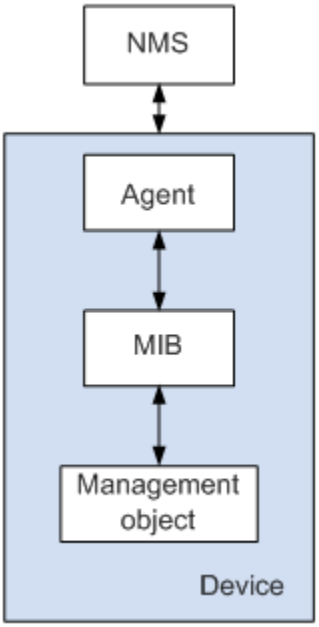
简单网络管理协议 SNMP（Simple Network Management Protocol）是广泛应用于 TCP/IP 网络的网络管理标准协议。SNMP 提供了一种通过运行网络管理软件的中心计算机（即网络管理工作站）来管理设备的方法。SNMP 的特点如下：

- 简单：SNMP 采用轮询机制，提供最基本的功能集，适合小型、快速、低价格的环境使用，而且 SNMP 以 UDP 报文为承载，因而受到绝大多数设备的支持。
- 强大：SNMP 的目标是保证管理信息在任意两点传送，以便于管理员在网络上的任何节点检索信息，进行故障排查。

5，SNMP 的组件有哪些，各自的作用

SNMP 系统包括网络管理系统 NMS（Network Management System）、代理进程 Agent、被管对象 Management object 和管理信息库 MIB（Management Information Base）四部分组成。

图1 SNMP管理模型



下面介绍网络管理系统中各主要元素：

- NMS

  NMS 在网络中扮演管理者角色，是一个采用 SNMP 协议对网络设备进行管理/监视的系统，运行在 NMS 服务器上。

  - NMS 可以向设备上的 Agent 发出请求，查询或修改一个或多个具体的参数值。
  - NMS 可以接收设备上的 Agent 主动发送的 Trap 信息，以获知被管理设备当前的状态。
- Agent

Agent 是被管理设备中的一个代理进程，用于维护被管理设备的信息数据并响应来自 NMS 的请求，把管理数据汇报给发送请求的 NMS。

- Agent 接收到 NMS 的请求信息后，通过 MIB 表完成相应指令后，并把操作结果响应给 NMS。
- 当设备发生故障或者其它事件时，设备会通过 Agent 主动发送信息给 NMS，向 NMS 报告设备当前的状态变化。

- **Management object**

Management object 指被管理对象。每一个设备可能包含多个被管理对象，被管理对象可以是设备中的某个硬件，也可以是在硬件、软件（如路由选择协议）上配置的参数集合。

- **MIB**

MIB 是一个数据库，指明了被管理设备所维护的变量（即能够被 Agent 查询和设置的信息）。MIB 在数据库中定义了被管理设备的一系列属性：对象的名称、对象的状态、对象的访问权限和对象的数据类型等。

通过 MIB，可以完成以下功能：

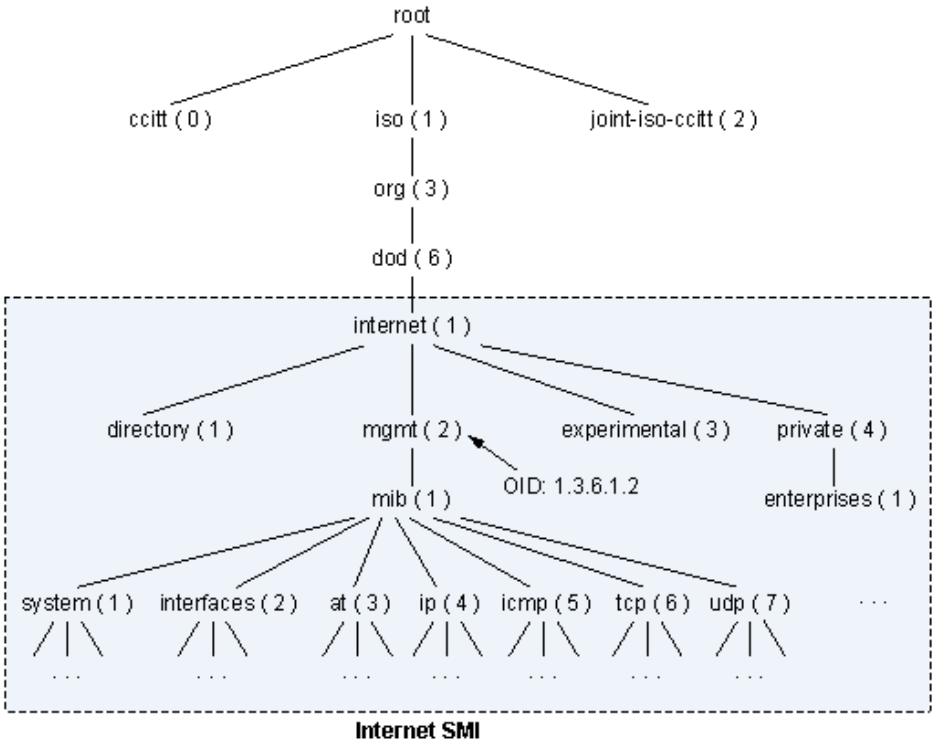
- Agent 通过查询 MIB，可以获知设备当前的状态信息。
- Agent 通过修改 MIB，可以设置设备的状态参数。

SNMP 的管理信息库采用和域名系统 DNS 相似的树型结构，它的根在最上面，根没有名字。如图 2 所示的是管理信息库的一部分，它又称为对象命名树。每个 OID（object identifier，对象标识符）对应于树中的一个管理对象，如 system 的 OID 为 1.3.6.1.2.1.1，interfaces 的 OID 为 1.3.6.1.2.1.2。

通过 OID 树，可以高效且方便地管理其中所存储的管理信息，同时也方便了对其中的信息进行批量查询。

特别地，当用户在配置 Agent 时，可以通过 MIB 视图来限制 NMS 能够访问的 MIB 对象。MIB 视图实际上是 MIB 的子集合。

图2 OID树结构



6，各版本的区别

SNMPv1/SNMPv2c报文结构

如图1所示，SNMPv1/SNMPv2c报文主要由版本、团体名、SNMP PDU三部分构成。

图1 SNMPv1/SNMPv2c报文结构

报文中的主要字段定义如下：

- 版本：表示SNMP的版本，如果是SNMPv1报文则对应字段值为0，SNMPv2c则为1。
- 团体名：用于在Agent与NMS之间完成认证，字符串形式，用户可自行定义。团体名包括“可读”和“可写”两种，执行GetRequest、GetNextRequest操作时，采用“可读团体名”进行认证；执行Set操作时，则采用“可写团体名”认证。
- SNMPv1/SNMPv2c PDU：包含PDU类型、请求标识符、变量绑定列表等信息。其中SNMPv1 PDU包括GetRequest PDU、GetNextRequest PDU、SetRequest PDU、Response PDU和Trap PDU几种类型，SNMPv2c PDU在SNMPv1的基础上新增了GetBulkRequest PDU。

  为了简化起见，SNMP操作今后叫做Get、GetNext、Set、Response、Trap和GetBulk操作。

SNMPv1/SNMPv2c操作类型

如表1所示，SNMPv1/SNMPv2c规定了6种操作类型，用来完成NMS和Agent之间的信息交换。

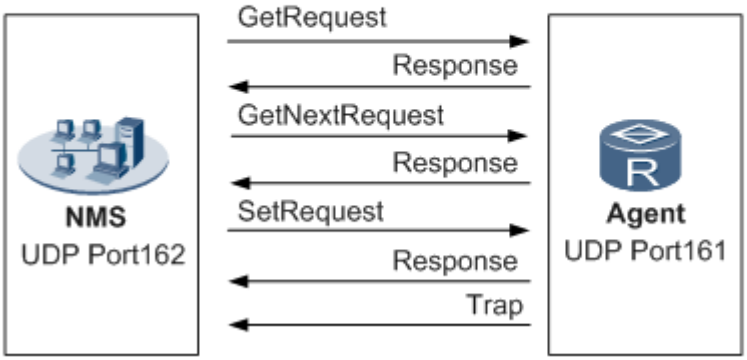
表1 SNMPv1/SNMPv2c中涉及的操作

操作	描述
Get	Get操作可以从Agent中提取一个或多个参数值。
GetNext	GetNext操作可以从Agent中按照字典序提取下一个参数值。
Set	Set操作可以设置Agent的一个或多个参数值。
Response	Response操作可以返回一个或多个参数值。这个操作是由Agent发出的，它是GetRequest、GetNextRequest、SetRequest和GetBulkRequest四种操作的响应操作。Agent接收到来自NMS的Get/Set指令后，通过MIB完成相应的查询/修改操作，然后利用Response操作将信息回应给NMS。
Trap	Trap信息是Agent主动向NMS发出的信息，告知管理进程设备端出现的情况。
GetBulk	GetBulk操作实现了NMS对被管理设备的信息群查询。

SNMPv1/SNMPv2c工作原理

SNMPv1和SNMPv2c的工作原理基本一致。SNMPv1/SNMPv2c的工作原理如图2所示。

图2 基本操作类型



- Get 操作

  假定 NMS 想要获取被管理设备 MIB 节点 sysContact 的值，使用可读团体名为 public，过程如下所示：

  1. NMS：向 Agent 发送 Get 请求报文。报文中各字段的设置如下：版本号为所使用的 SNMP 版本；团体名为 public；PDU 中 PDU 类型为 Get 类型，绑定变量填入 MIB 节点名 sysContact。
  2. Agent：首先对报文中携带版本号和团体名进行认证，认证成功后，Agent 根据请求查询 MIB 中的 sysContact 节点，得到 sysContact 的值并将其封装到 Response 报文中的 PDU，向 NMS 发送响应；如果查询不成功，Agent 会向 NMS 发送出错响应。

- GetNext 操作

  假定 NMS 想要获取被管理设备 MIB 节点 sysContact 的下一个节点 sysName 值，使用可读团体名为 public，过程如下所示：

1. NMS：向 Agent 发送 GetNext 请求报文。报文中各字段的设置如下：版本号为所使用的 SNMP 版本；团体名为 public；PDU 中 PDU 类型为 GetNext 类型，绑定变量填入 MIB 节点名 sysContact。
2. Agent：首先对报文中携带版本号和团体名进行认证，认证成功后，Agent 根据请求查询 MIB 中的 sysContact 的下一个节点 sysName，得到 sysName 的值并将其封装到 Response 报文中的 PDU，向 NMS 发送响应；如果查询不成功，Agent 会向 NMS 发送出错响应。

- Set 操作

  假定 NMS 想要设置被管理设备 MIB 节点 sysName 的值为 HUAWEI，使用可写团体名为 private，过程如下所示：

  1. NMS：向 Agent 发送 Set 请求报文。报文中各字段的设置如下：版本号为所使用的 SNMP 版本；团体名为 private；PDU 中 PDU 类型为 Set 类型，绑定变量填入 MIB 节点名 sysContact 和需要设置的值 HUAWEI。
  2. Agent：首先对报文中携带版本号和团体名进行认证，认证成功后，Agent 根据请求设置管理变量在管理信息库 MIB 中对应的节点，设置成功后向 NMS 发送响应；如果设置不成功，Agent 会向 NMS 发送出错响应。

- Trap 操作

  Trap 不属于 NMS 对被管理设备的基本操作，它是被管理设备的自发行为。当被管理设备达到告警的触发条件时，会通过 Agent 向 NMS 发送 Trap 消息，告知设备侧出现的异常情况，便于网络管理人员及时处理。例如被管理设备热启动后，Agent 会向 NMS 发送 warmStart 的 Trap。

  这种 Trap 信息是受限制的。只有在设备端的模块达到模块预定义的告警触发条件时，Agent 才会向管理进程报告。这种方法有其好处是仅在严重事件发生时才发送 Trap 信息，减少报文交互产生的流量。

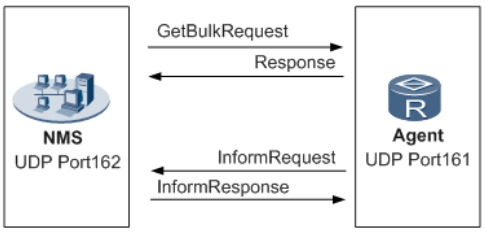
SNMPv3报文结构

SNMPv3定义了新的报文格式，其报文结构如图1所示。

图1 SNMPv3报文结构

SNMPv2c新增的操作如图3所示。

图3 SNMPv2c新增操作



- GetBulk操作

  基于GetNext实现，相当于连续执行多次GetNext操作。在NMS上可以设置被管理设备在一次GetBulk报文交互时，执行GetNext操作的次数。

SNMP 报文中的主要字段定义如下：

- 版本：表示 SNMP 的版本，SNMPv3 报文则对应字段值为 2。
- 报头数据：主要包含消息发送者所能支持的最大消息尺寸、消息采用的安全模式等描述内容。
- 安全参数：包含 SNMP 实体引擎的相关信息、用户名、认证参数、加密参数等安全信息。
- Context EgineID：SNMP 唯一标识符，和 PDU 类型一起决定应该发往哪个应用程序。
- Context Name：用于确定 Context EgineID 对被管理设备的 MIB 视图。
- SNMPv3 PDU：包含 PDU 类型、请求标识符、变量绑定列表等信息。其中 SNMPv3 PDU 包括 GetRequest PDU、GetNextRequest PDU、SetRequest PDU、Response PDU、Trap PDU 和 GetBulkRequest PDU。

SNMPv3 的体系结构

SNMPv3 提出了一个新的 SNMP 体系结构，这个体系结构为各种基于 SNMP 的 NMS 提供了一个通用的实现模型，即 SNMPv3 实体。SNMPv3 实体可以分为 SNMPv3 引擎（SNMPv3 Engine）和 SNMPv3 应用程序（SNMPv3 Application），引擎与应用程序均由多个小模块组成。

SNMPv3 实体这种模块化的结构有以下优点：

- 适应性强：适用于多种操作环境，既可以管理最简单的网络，又能够满足复杂网络的管理需求。

- 方便管理：SNMP 框架体系由多个功能相对独立的子系统或应用程序集合而成，因而可以很方便地对其进行管理。例如，若系统发生了故障，可以根据发生故障的功能类型，定位到相应的子系统。
- 扩展性好：通过 SNMP 实体，可以很方便地进行系统扩展。比如，为了应用新的安全协议，就可以在安全子系统中为其定义单独的模块，从而在 SNMP 中支持该协议。

SNMPv3 由于采用了用户安全模块 USM（User Security Model）和基于视图的访问控制模块 VACM（View-based Access Control Model），在安全性上得到了提升。

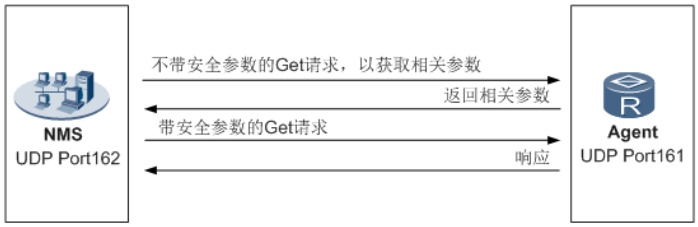
- USM：提供身份验证和数据加密服务。实现这个功能要求 NMS 和 Agent 必须共享同一密钥。
  - 身份验证：身份验证是指 Agent 或 NMS 接到信息时首先必须确认信息是否来自有权限的 NMS 或 Agent 并且信息在传输过程中未被改变。RFC2104 中定义了 HMAC，这是一种使用安全哈希函数和密钥来产生信息验证码的有效工具，在互联网中得到了广泛的应用。SNMP 使用的 HMAC 可以分为两种：HMAC-MD5-96 和 HMAC-SHA-96。前者的哈希函数是 MD5，使用 128 位 authKey 作为输入。后者的哈希函数是 SHA-1，使用 160 位 authKey 作为输入。
  - 加密：加密算法实现主要通过对称密钥系统，它使用相同的密钥对数据进行加密和解密。加密的过程与身份验证类似，也需要管理站和代理共享同一密钥来实现信息的加密和解密。SNMP 使用以下二种加密算法：
    - DES：使用 56bit 的密钥对一个 64bit 的明文块进行加密。
    - AES：使用 128bit、192bit 或 256bit 密钥长度的 AES 算法对明文进行加密。
- VACM：对用户组或者团体名实现基于视图的访问控制。用户必须首先配置一个视图，并指明权限。用户可以在配置用户或者用户组或者团体名的时候，加载这个视图达到限制读写操作或 Trap 的目的。

SNMPv3的工作原理

SNMPv3的实现原理和SNMPv1/SNMPv2c基本一致，唯一的区别是SNMPv3增加了身份验证和加密处理。下面以Get操作为例介绍下SNMPv3的工作原理。

假定NMS想要获取被管理设备MIB节点sysContact的值，使用认证加密方式，过程如图2所示：

图2 SNMPv3的Get操作



1. NMS：向 Agent 发送不带安全参数的 Get 请求报文，向 Agent 获取 Context EgineID、Context Name 和安全参数（SNMP 实体引擎的相关信息）。

2. Agent：响应 NMS 的请求，并向 NMS 反馈请求的参数。
3. NMS：再次向 Agent 发送 Get 请求报文，报文中各字段的设置如下：
   - 版本：SNMPv3 版本。
   - 报头数据：指明采用认证、加密方式。
   - 安全参数：NMS 通过配置的算法计算出认证参数和加密参数。将这些参数和获取的安全参数填入相应字段。
   - PDU：将获取的 Context EgineID 和 Context Name 填入相应字段，PDU 类型设置为 Get，绑定变量填入 MIB 节点名 sysContact，并使用已配置的加密算法对 PDU 进行加密。
4. Agent：首先对消息进行认证，认证通过后对 PDU 进行解密。解密成功后，Agent 根据请求查询 MIB 中的 sysContact 节点，得到 sysContact 的值并将其封装到 Response 报文中的 PDU，并对 PDU 进行加密，向 NMS 发送响应。如果查询不成功或认证、解密失败，Agent 会向 NMS 发送出错响应。

表2 SNMP各版本支持的特性概况

特性	SNMPv1	SNMPv2c	SNMPv3
访问控制	基于团体名和MIB View进行访问控制	基于团体名和MIB View进行访问控制	基于用户、用户组和MIB view进行访问控制
认证加密	基于团体名的认证	基于团体名的认证	支持认证和加密，认证和加密的方式如下： 认证： • MD5 • SHA 加密： • DES56 • AES128
错误码	支持6个错误码	支持16个错误码	支持16个错误码
Trap告警	支持	支持	支持
GetBulk	不支持	支持	支持

二，NTP

1， NTP 的作用

网络时间协议 NTP（Network Time Protocol）是 TCP/IP 协议族里面的一个应用层协议。NTP 用于在一系列分布式时间服务器与客户端之间同步时钟。NTP 的实现基于 IP 和 UDP。NTP 报文通过 UDP 传输，端口号是 123。

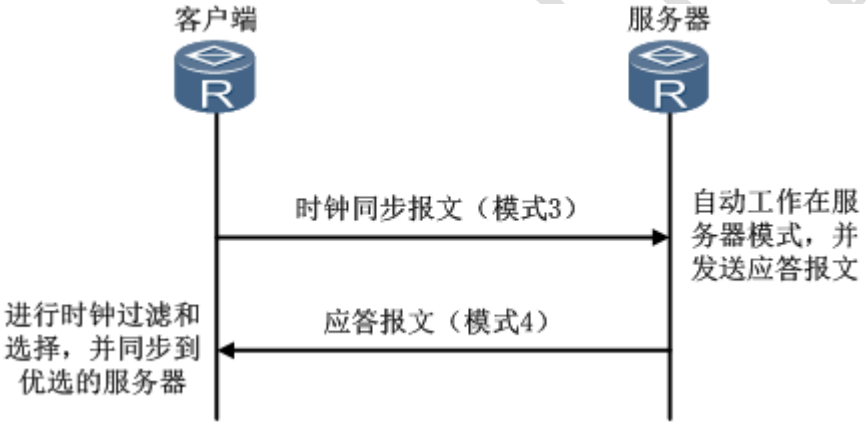
2， NTP 的工作模式(五种)

单播服务器/客户端模式

单播服务器/客户端模式运行在同步子网中层数较高层上。这种模式下，需要预先知道服务器的 IP 地址。

- 客户端：运行在客户端模式的主机（简称客户端）定期向服务器端发送报文，报文中的 Mode 字段设置为 3（客户端模式）。当客户端接收到应答报文时，客户端会进行时钟过滤和选择，并同步到时钟优选的服务器。客户端不管服务器端是否可达及服务器端的层数。运行在这种模式的主机，通常是网络内部的工作站，它可以依照对方的时钟进行同步，但不会修改对方的时钟。
- 服务器：运行在服务器模式（简称服务器）的主机接收并回应报文，报文中的 Mode 字段设置为 4（服务器模式）。运行在服务器模式的主机，通常是网络内部的时间服务器，它可以向客户端提供同步信息，但不会修改自己的时钟。

图 1 单播客户端/服务器模式



运行在客户端模式的主机在重新启动时和重新启动后定期向运行在服务器模式的主机发送 NTP 报文。服务器收到客户端的报文后，首先将报文的目的 IP 地址和目的端口号分别与其源 IP 地址和源端口号相交换，再填写所需的信息，然后把报文发送给客户端。服务器无需保留任何状态信息，客户端根据本地情况自由管理发送报文的时间间隔。

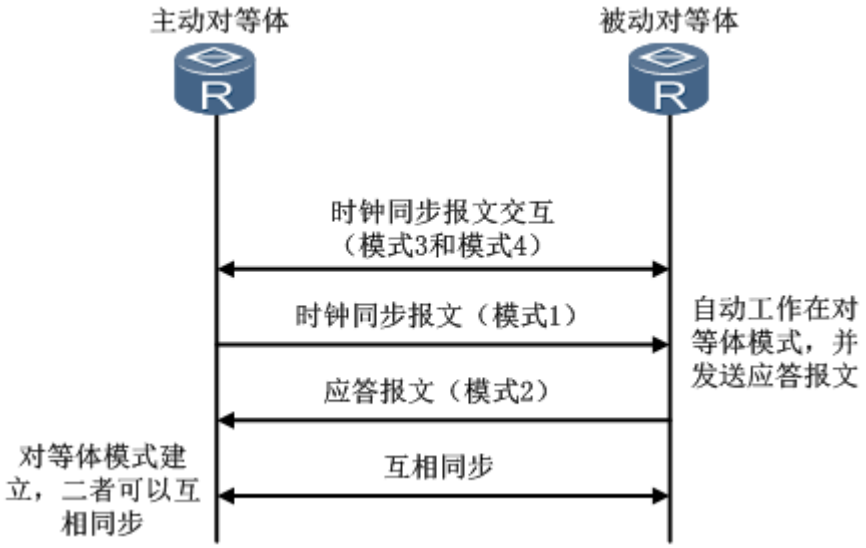
对等体模式

对等体模式运行在同步子网中层数较低处。这种模式下，主动对等体和被动对等体可以互相同步，等级低（层数大）的对等体向等级高（层数小）的对等体同步。

对等体模式下，主动对等体会发起 Mode 字段为 3（客户端模式）NTP 报文，由被动对等体响应 4（服务器模式）的 NTP 报文。这一交互过程主要是为了获得网络延迟，使两端设备进入对等体模式。

- 主动对等体：运行在这一模式下的主机定期发送报文，报文中的 Mode 字段设置为 1（主动对等体）。不考虑它的对等体是否可达以及对等体的层数。运行在这一模式下的主机可以向对方提供同步信息，也可以依照对方的时间信息同步本地时钟。
- 被动对等体：运行在这一模式的主机接收并回应报文，报文中的 Mode 字段设置为 2（被动对等体）。运行在被动对等体模式的主机可以向对方提供同步信息，也可以依照对方的时间信息同步本地时钟。

图 2 对等体模式



说明：

被动对等体不需要用户配置，只有当本机收到 NTP 报文时才建立连接及相关的状态变量。

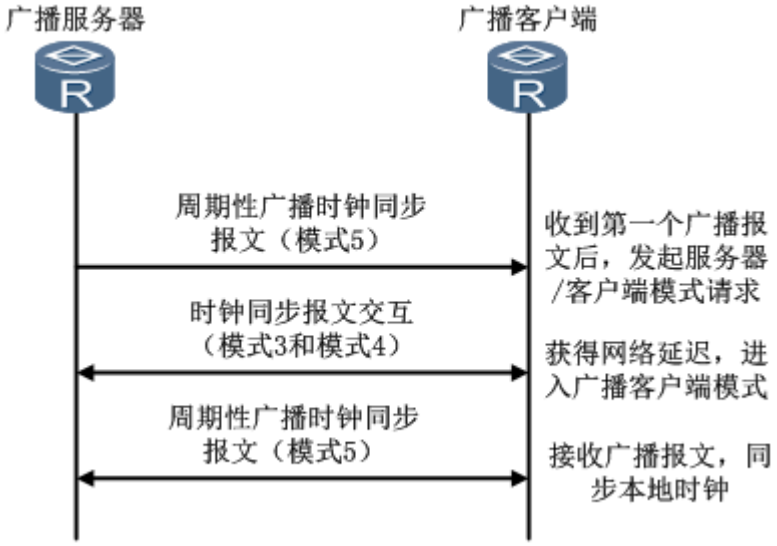
广播模式

广播模式应用在有多台工作站、不需要很高的准确度的高速网络。典型的情况是网络中的一台或多台时间服务器定期向工作站发送广播报文，广播报文在毫秒级的延迟基础上确定时间。

- 广播服务器：运行在广播模式下，周期性向广播地址 255.255.255.255 发送时钟同步报文，报文中的 Mode 字段设置为 5（广播模式或组播模式）。不管它的对等体是否可达或层数为多少。运行在广播模式的主机通常是网络内运行高速广播介质的时间服务器，向所有对等体提供同步信息，但不会修改自己的时钟。
- 广播客户端：客户端侦听来自服务器的时钟同步报文。当接收到第一个时钟同步报文，客户端与服务器交互 Mode 字段为 3（客户端模式）和 4（服务器模式）的 NTP 报文，即客户端先启用一个短暂的服务器/客户端模式与远程服务器交换消息，以获得客户端与服务器间的网络延迟。之

后恢复广播模式，继续侦听时钟同步报文的到来，根据到来的时钟同步报文对本地时钟再次进行同步。

图 3 广播模式

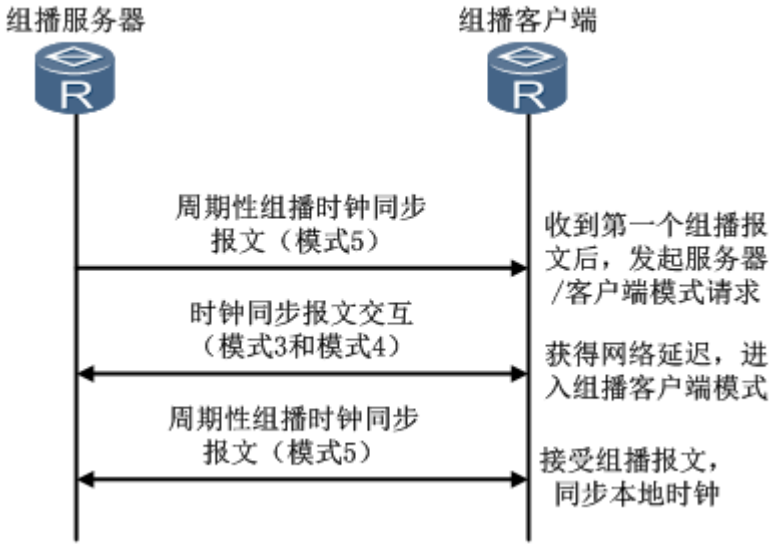


组播模式

组播模式适用于有大量客户端分布在网络中的情况。通过在网络中使用 NTP 组播模式，NTP 服务器发送的组播消息包可以到达网络中所有的客户端，从而降低由于 NTP 报文过多而给网络造成的压力。

- 组播服务器：服务器端周期性向组播地址发送时钟同步报文，报文中的 Mode 字段设置为 5（广播模式或组播模式）。运行在组播模式的主机通常是网络内运行高速广播介质的时间服务器，向所有对等体提供同步信息，但不会修改自己的时钟。
- 组播客户端：客户端侦听来自服务器的组播消息包。当客户端接收到第一个组播报文后，客户端与服务器交互 Mode 字段为 3（客户端模式）和 4（服务器模式）的 NTP 报文，即客户端先启用一个短暂的服务器/客户端模式与远程服务器交换消息，以获得客户端与服务器间的网络延迟。之后，客户端恢复组播模式，继续侦听组播消息包的到来，根据到来的组播消息包对本地时钟进行同步。

图 4 组播模式



多播模式

多播模式适用于服务器分布分散的网络中。客户端可以发现与之最近的多播服务器，并进行同步。多播模式适用于服务器不稳定的组网环境中，服务器的变动不会导致整网中的客户端重新进行配置。

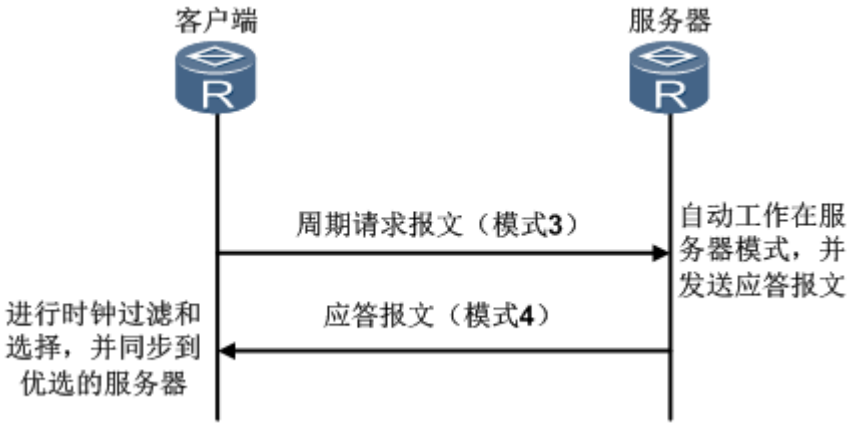
- 多播服务器：多播服务器持续侦听报文。若某个服务器可以被同步，则服务器将使用客户端的单播地址返回报文（Mode 字段设置为 4）。
- 多播客户端：多播模式下的客户端周期性地向 IPv4/IPv6 组播地址发送请求报文（Mode 字段设置为 3）。当客户端接收到应答报文时，客户端会进行时钟过滤和选择，并同步到时钟优选的服务器。

为了防止多播模式下，客户端不断的向多播服务器发送 NTP 请求报文增加设备的负担，协议规定了最小连接数的概念。多播模式下，客户端每次和服务器时钟同步后，都会记录下此次同步过程中建立的连接数，将调用最少连接的数量称为最小连接数。以后当客户端调动的连接数达到了最小连接数且完成了同步，客户端就认为同步完成；同步完成后每过一个超时周期，客户端都会传送一个报文，用于保持连接。同时，为了防止客户端无法同步到服务器，协议规定客户端每发送一个 NTP 报文，都会将报文的生存时间 TTL（Time To Live）进行累加（初始为 1），直到达到最小连接数，或者 TTL 值达到上限（上限值为 255）。若 TTL 达到上限，或者达到最小连接数，而客户端调动的连接数仍不能完成同步过程，则客户端将停止一个超时周期的数据传输以清除所有连接，然后重复上述过程。

说明：

在 NTP 模块实现中，对每一个同步源都建立了一个 PEER 结构，并把这些 PEER 结构以 Hash 的形式存储成链状。每一个 PEER 结构对应于一个连接。

图 5 多播模式



三，Netstream

1， Netstream 的作用

NetStream 是一种基于网络流信息的统计技术，可以对网络中的业务流量情况进行统计和分析。

2， Netstream 的原理：

表1 传统的流量统计的实现方法和局限性

名称	实现方法	局限性
基于IP报文计数	在路由表中存放计数器索引，对通过设备的字节和包分别计数。	统计的信息简单，无法针对多种信息进行统计。
使用ACL	通过ACL精确的匹配流，匹配后进行计数。	要求ACL的容量很大，对于ACL规则以外的流没有办法统计。
SNMP协议	使用网管协议，能够实现一些简单的统计功能，比如接口计数、IP报文计数、ACL匹配计数等。	功能不强。要不断的通过轮询向网管查询，浪费CPU和网络资源。
端口镜像	通过端口镜像，把流量复制一份，发送至专用的服务器进行统计分析。	成本高，需要购买专用的服务器进行统计，同时消耗设备的一个接口，对于无法镜像的端口无能为力。
物理层复制	在物理层通过分光器或者其他设备复制流量，发送至专用的服务器进行统计。	成本高，需要购买专用的服务器进行统计，同时还需要购买专用的硬件设备。

一个典型的 NetStream 系统由网络流数据输出器 NDE（NetStream Data Exporter）、网络流数据收集器 NSC（NetStream Collector）和网络流数据分析器 NDA（NetStream Data Analyzer）三部分组成

- NDE

  NDE 负责对网络流进行分析处理，提取符合条件的流进行统计，并将统计信息输出给 NSC。 输出前也可对数据进行一些处理，比如聚合。配置了 NetStream 功能的设备在 NetStream 系统中担当 NDE 角色。

- NSC

  NSC 通常为运行于 Unix 或者 Windows 上的一个应用程序，负责解析来自 NDE 的报文，把统计数据收集到数据库中，可供 NDA 进行解析。NSC 可以采集多个 NDE 设备输出的数据，对数据进行进一步的过滤和聚合。

- NDA

NDA 是一个网络流量分析工具，它从 NSC 中提取统计数据，进行进一步的加工处理后生成报表，为各种业务提供依据（比如流量计费、网络规划、攻击监测）。通常，NDA 具有图形化用户界面，使用户可以方便地获取、显示和分析收集到的数据。

NetStream 系统的工作过程如下：

1. 配置了 NetStream 功能的设备（即 NDE）把采集到的关于流的详细统计信息定期发送给 NSC；
2. 信息由 NSC 初步处理后发送给 NDA；
3. NDA 对数据进行分析，以用于计费、网络规划等应用。

NetStream 采样

对接口出/入方向的流量使用 NetStream 采样的方法。通过设定适当的采样间隔，只针对样本报文进行流信息统计分析，收集到的统计信息也可以基本正确地反映整个网络流的状况，同时也能降低使能 NetStream 功能对设备性能的影响。

NetStream 采样有四种方式：

- 随机报文间隔采样：

  在此模式下，报文在配置数目间隔内被随机采样。即，如果报文间隔数配置为 100，则每 100 个报文随机采样 1 个报文。适用于有规律的流量。

- 固定报文间隔采样：

  在此模式下，报文在配置数目间隔内被周期采样。即，如果报文间隔配置数为 100，假设在第 5 个报文被采样后，则每隔 100 个报文都会再次采样，如第 105 个报文会再采集一次，以此类推采样下去。适用于网络流量统计计费。

- 随机时间间隔采样：

  在此模式下，报文在配置时间间隔内被随机采样。即，如果报文间隔时间配置为 100，则每 100 毫秒随机采样 1 个报文。适用于有规律的流量。

- 固定时间间隔采样：

在此模式下，报文在配置时间间隔内被周期采样。即，如果报文间隔时间配置为 100，假设在第 5 毫秒进行第一次采样后，则每隔 100 毫秒都会再次采样，如第 105 毫秒时会再采集一次，以此类推采样下去。适用于网络流量较大的情况。

NetStream 采样

对接口出/入方向的流量使用 NetStream 采样的方法。通过设定适当的采样间隔，只针对样本报文进行流信息统计分析，收集到的统计信息也可以基本正确地反映整个网络流的状况，同时也能降低使能 NetStream 功能对设备性能的影响。

NetStream 采样有四种方式：

- 随机报文间隔采样：

  在此模式下，报文在配置数目间隔内被随机采样。即，如果报文间隔数配置为 100，则每 100 个报文随机采样 1 个报文。适用于有规律的流量。

- 固定报文间隔采样：

  在此模式下，报文在配置数目间隔内被周期采样。即，如果报文间隔配置数为 100，假设在第 5 个报文被采样后，则每隔 100 个报文都会再次采样，如第 105 个报文会再采集一次，以此类推采样下去。适用于网络流量统计计费。

- 随机时间间隔采样：

  在此模式下，报文在配置时间间隔内被随机采样。即，如果报文间隔时间配置为 100，则每 100 毫秒随机采样 1 个报文。适用于有规律的流量。

- 固定时间间隔采样：

  在此模式下，报文在配置时间间隔内被周期采样。即，如果报文间隔时间配置为 100，假设在第 5 毫秒进行第一次采样后，则每隔 100 毫秒都会再次采样，如第 105 毫秒时会再采集一次，以此类推采样下去。适用于网络流量较大的情况。

NetStream 流老化

NetStream 流老化是设备向 NSC 输出流统计信息的前提。设备启用 NetStream 功能后，流统计信息首先会被存储在设备的 NetStream 缓存区中。当存储在设备上的 NetStream 流信息老化后，设备会把缓存区中的流统计信息通过指定版本的 NetStream 输出报文发送给 NSC。

NetStream 流老化的分类：

- 按时老化
    - 活跃流的老化

      从第一个报文开始，一条流在指定的时间内一直能被采集到。流活跃时间超过设定的时长后，需要输出该流的统计信息，这种老化称为活跃流的老化。该种老化方式主要用于持续时间较长的流量，定期输出统计信息。

    - 非活跃流的老化

      从最后一个报文开始，一条流在指定的时间内没有被采集到（即在设定时长内统计到的报文数目没有增加），设备会向 NetStream 服务器输出该流的统计信息，这种老化称为非活跃的流老化。通过这种老化，可以清除设备上 NetStream 缓存区中的无用表项，充分利用统计表项资源。该种老化方式主用于短时流量，流量停止则立即输出统计信息，节省内存空间。

- 由 TCP 连接的 FIN 和 RST 报文触发老化

  对于 TCP 连接，当有标志为 FIN 或 RST 的报文发送时，表示一次会话结束。因此当一条已经存在的 TCP 协议 NetStream 流中流过一条标志为 FIN 或 RST 的报文时，可以立即把相应的 NetStream 流老化掉。

- 统计字节超过限制时老化

  NetStream 缓存区中的流需要记录流过的报文字节数，当字节数量超过定义的变量上限时，该流就会溢出。所以系统在检测到某条流的字节统计超过限制（硬件的字节计数器是 32 比特，最大计数值为 4294967295，约为 3.9G 字节）时，为了避免计数错误，系统会立即自动把该流老化掉。

- 强制老化

  用户可以通过执行命令强制将 NetStream 缓存区中所有流老化。

该功能主要用于老化条件尚未满足，但又需要最新的统计信息。或者 NetStream 业务发生异常，导致流缓存区中某些流始终不老化。

NetStream 流输出

NetStream 流输出是指储存缓存区里面的流老化后把流统计信息输出到指定的 NSC，以便进行后续更为详尽的分析。

流输出方式

原始流输出方式

原始流输出是指所有流的统计信息都要被统计。在流老化时间超时后，每条流的统计信息都要输出到 NetStream 服务器。

原始流的优点是：NetStream 服务器可以得到每条流的详细统计信息。正因为这样，其缺点是：增加了网络带宽和设备的 CPU 占有率，而且为了存储这些信息，需要占用大量的存储介质空间，增加了设备的开销。

聚合流输出方式

聚合流输出是指采用聚合流输出功能后，设备对与聚合关键项完全相同的流统计信息进行汇总，从而得到对应的聚合流统计信息，并且将该聚合统计信息发送到相应的接收聚合统计信息的 NetStream 服务器。通过对原始流进行聚合后输出，可以明显减少网络带宽。支持如表 1 所示的聚合方式。

例如：现有四条 TCP 原始流，其目的地址相同、源地址不同，源端口、目的端口均相同，选择表 1 中的“protocol-port（协议-端口聚合）”方式，该聚合方式依据“协议号、源端口、目的端口”的聚合关键项进行聚合。因为这四条 TCP 流的源端口、目的端口和协议号相同，所以在聚合流统计表项中只会记录一条聚合流统计信息。设备只将聚合统计信息发送给相应的接收聚合统计信息的 NetStream 服务器。

表1 聚合方式列表

聚合方式	聚合关键项
as（自治系统聚合）	源自治系统号、目的自治系统号、输入接口索引、输出接口索引
as-tos（自治系统-ToS聚合）	源自治系统号、目的自治系统号、输入接口索引、输出接口索引、ToS
protocol-port（协议-端口聚合）	协议号、源端口、目的端口
protocol-port-tos（协议-端口-ToS聚合）	协议号、源端口、目的端口、ToS、输入接口索引、输出接口索引
source-prefix（源前缀聚合）	源自治系统号、源掩码长度、源前缀、输入接口索引
source-prefix-tos（源前缀-ToS聚合）	源自治系统号、源掩码长度、源前缀、ToS、输入接口索引
destination-prefix（目的前缀聚合）	目的自治系统号、目的掩码长度、目的前缀、输出接口索引
destination-prefix-tos（目的前缀-Tos聚合）	目的自治系统号、目的掩码长度、目的前缀、ToS、输出接口索引
prefix（前缀聚合）	源自治系统号、目的自治系统号、源掩码长度、目的掩码长度、源前缀、目的前缀、输入接口索引、输出接口索引
prefix-tos（前缀-Tos聚合）	源自治系统号、目的自治系统号、源掩码长度、目的掩码长度、源前缀、目的前缀、ToS、输入接口索引、输出接口索引

灵活流输出方式

对于灵活流输出，其流的建立条件是按照自定义的条件设置。根据自身需要，用户可以对报文按照协议类型、DSCP 优先级、源 IP 地址、目的 IP 地址、源端口号、目的端口号或流标签进行分类统计，从而将分类统计信息发送给 NetStream 服务器。灵活流方式相比原始流方式可减少流量的占用。可以为用户提供一种自由的 NetStream 流量统计方式。

输出报文的版本格式

NetStream 输出的报文主要有 V5、V8、V9 和 V10 四个版本。所有版本的报文都是通过 UDP 协议传递统计信息的。

- 版本 5：根据七元组产生原始的数据流，但报文格式固定，不易扩展。
- 版本 8：支持聚合输出格式，但报文格式固定，不易扩展。
- 版本 9：基于模板方式，使统计信息的输出更为灵活，可以灵活输出各种组合格式的数据。版本 9 支持对 BGP 下一跳、MPLS 等统计输出。
- 版本 10：基于模板方式，根据数据流特征输出统计信息。具有很强的可扩展性，对于不同的需求输出不同格式的数据。

表1 NetStream的缺省配置

参数	缺省值
NetStream采样	固定报文间隔采样，采样比值为100
活跃流的老化时间	30分钟
不活跃流的老化时间	30秒
由TCP连接的FIN和RST报文触发老化	未使能
统计字节超过限制时老化	使能
IPv4单播原始流统计信息输出报文版本	V5
IPv4组播原始流统计信息输出报文版本	V5
IPv4聚合流统计信息输出报文版本	V8
IPv4灵活流统计信息输出报文版本	V9
RPF检查失败的异常流量统计信息输出报文版本	V5

四，VRRP（详见题库）

1，使用及原理

2，优先级为 0 与 255 发生在什么情况，及作用

3，VRRP 中免费 ARP 的作用

4，VRRP 与 MSTP 的组合使用

5，VRRP 与 BFD 的组合使用

6，如果确保 VRRP 的往返路径一致

五，FTP（详见视频）

1，主动模式

2，被动模式

3，ALG

六，BFD 与 NQA

1，知道 BFD 与 NQA 的作用，及区别

NQA 可以检测应用层的信息，BFD 检测的是底层的信息。

BFD

双向转发检测 BFD（Bidirectional Forwarding Detection）是一种全网统一的检测机制，用于快速检测、监控网络中链路或者 IP 路由的转发连通状况。

目的

为了减小设备故障对业务的影响，提高网络的可靠性，网络设备需要能够尽快检测到与相邻设备间的通信故障，以便及时采取措施，保证业务继续进行。在现有网络中，有些链路通常通过硬件检测信号，如 SDH 告警，检测链路故障，但并不是所有的介质都能够提供硬件检测。此时，应用就要依靠上层协议自身的 Hello 报文机制来进行故障检测。上层协议的检测时间都在 1 秒以上，这样

的故障检测时间对某些应用来说是不能容忍的。同时，在一些小型三层网络中，如果没有部署路由协议，则无法使用路由协议的 Hello 报文机制来检测故障。

BFD 协议就是在这种背景下产生的，BFD 提供了一个通用的标准化的介质无关和协议无关的快速故障检测机制。具有以下优点：

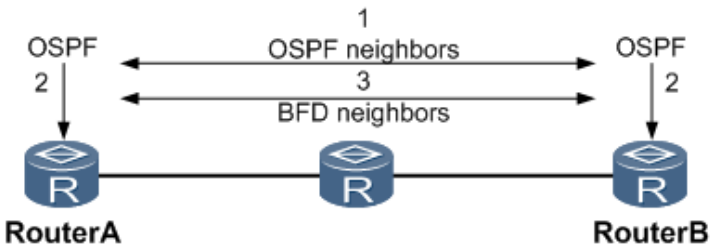
- 对相邻转发引擎之间的通道提供轻负荷、快速故障检测。这些故障包括接口、数据链路，甚至有可能是转发引擎本身。
- 用单一的机制对任何介质、任何协议层进行实时检测。

受益

BFD 可以实现快速检测并监控网络中链路或 IP 路由的转发连通状态，改善网络性能。相邻系统之间通过快速检测发现通信故障，可以更快地帮助用户建立起备份通道以便恢复通信，保证网络可靠性。

BFD 在两台网络设备上建立会话，用来检测网络设备间的双向转发路径，为上层应用服务。BFD 本身并没有邻居发现机制，而是靠被服务的上层应用通知其邻居信息以建立会话。会话建立后会周期性地快速发送 BFD 报文，如果在检测时间内没有收到 BFD 报文则认为该双向转发路径发生了故障，通知被服务的上层应用进行相应的处理。下面以 OSPF 与 BFD 联动为例，简单介绍会话工作流程。

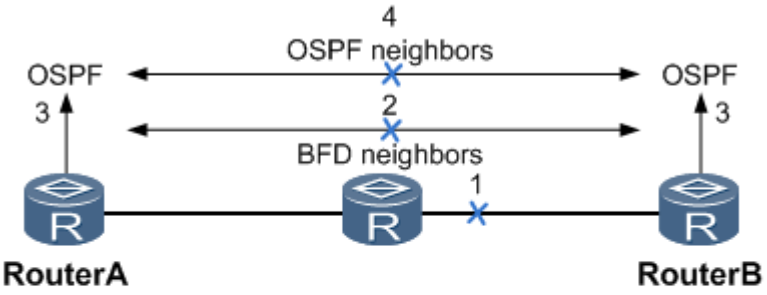
图1 BFD会话建立流程图



上图所示是一个简单的网络组网，两台设备上同时配置了OSPF与BFD，BFD会话建立过程如下所示：

1. OSPF通过自己的Hello机制发现邻居并建立连接。
2. OSPF在建立了新的邻居关系后，将邻居信息（包括目的地址和源地址等）通告给BFD。
3. BFD根据收到的邻居信息建立会话。
4. 会话建立以后，BFD开始检测链路故障，并做出快速反应。

图2 BFD故障发现处理流程图



如上图所示：

1. 被检测链路出现故障。
2. BFD快速检测到链路故障，BFD会话状态变为Down。
3. BFD通知本地OSPF进程BFD邻居不可达。
4. 本地OSPF进程中断OSPF邻居关系。

BFD 检测机制

BFD 的检测机制是两个系统建立 BFD 会话，并沿它们之间的路径周期性发送 BFD 控制报文，如果一方在既定的时间内没有收到 BFD 控制报文，则认为路径上发生了故障。

BFD 提供异步检测模式。在这种模式下，系统之间相互周期性地发送 BFD 控制报文，如果某个系统连续几个报文都没有接收到，就认为此 BFD 会话的状态是 Down。

BFD 会话建立方式

BFD 会话的建立有两种方式，即静态建立 BFD 会话和动态建立 BFD 会话。静态和动态创建 BFD 会话的主要区别在于本地标识符（Local Discriminator）和远端标识符（Remote Discriminator）的配置方式不同。BFD 通过控制报文中的 Local Discriminator 和 Remote Discriminator 区分不同的会话。

- 静态建立 BFD 会话

  静态建立 BFD 会话是指通过命令行手工配置 BFD 会话参数，包括配置本地标识符和远端标识符等，然后手工下发 BFD 会话建立请求。

- 动态建立 BFD 会话

  动态建立 BFD 会话时，系统对本地标识符和远端标识符的处理方式如下：

- 动态分配本地标识符

  当应用程序触发动态创建 BFD 会话时，系统分配属于动态会话标识符区域的值作为 BFD 会话的本地标识符。然后向对端发送 Remote Discriminator 的值为 0 的 BFD 控制报文，进行会话协商。

- 自学习远端标识符

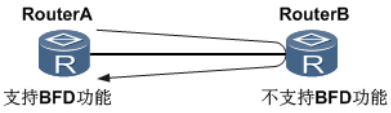
  当 BFD 会话的一端收到 Remote Discriminator 的值为 0 的 BFD 控制报文时，判断该报文是否与本地 BFD 会话匹配，如果匹配，则学习接收到的 BFD 报文中 Local Discriminator 的值，获取远端标识符。

表1 BFD参数缺省值

参数	缺省值
全局BFD功能	未使能
发送间隔	1000毫秒
接收间隔	1000毫秒
本地检测倍数	3
等待恢复时间	0分钟
会话延迟Up时间	0秒钟
BFD报文优先级	7

BFD 的单臂回声功能

图1 BFD单臂回声功能



如图1所示，RouterA支持BFD功能，RouterB不支持BFD功能。在RouterA上配置单臂回声功能的BFD会话，检测RouterA到RouterB之间的单跳路径。RouterB接收到RouterA发送的BFD报文后，直接在网络层将该报文环回，从而快速检测RouterA和RouterB之间的直连链路的连通性。

BFD 与 VRRP 联动

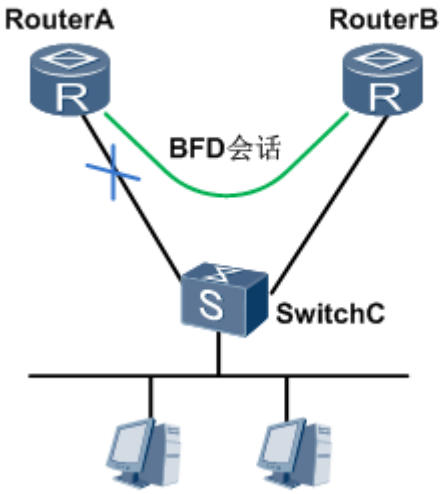
BFD 与 VRRP 联动

VRRP 的协议关键点是当 Master 出现故障时，Backup 能够快速接替 Master 的转发工作，保证数据流的中断时间尽量短。

当 Master 出现故障时，VRRP 依靠 Backup 设置的超时时间来判断是否应该抢占，切换速度在 1 秒以上。将 BFD 应用于 Backup 对 Master 的检测，可以实现对 Master 故障的快速检测，缩短用户流量中断时间。BFD 对 Backup 和 Master

之间的实际地址通信情况进行检测，如果通信不正常，Backup 就认为 Master 已经不可用，升级成 Master。VRRP 通过监视 BFD 会话状态实现主备快速切换，切换时间控制在 50 毫秒以内。

图1 BFD与VRRP联动



如图 1 所示，RouterA 和 RouterB 之间配置 VRRP 备份组建立主备关系，RouterA 为主用设备，RouterB 为备用设备，用户过来的流量从 RouterA 出去。在 RouterA 和 RouterB 之间建立 BFD 会话，VRRP 备份组监视该 BFD 会话，当 BFD 会话状态变化时，通过修改备份组优先级实现主备快速切换。

当 BFD 检测到 RouterA 和 SwitchC 之间的链路故障时，上报给 VRRP 一个 BFD 检测 Down 事件，RouterB 上 VRRP 备份组的优先级增加，增加后的优先级大于 RouterA 上的 VRRP 备份组的优先级，于是 RouterB 立刻升为 Master，后继的用户流量就会通过 RouterB 转发，从而实现 VRRP 的主备快速切换。

NQA

网络质量分析 NQA（Network Quality Analysis）是一种实时的网络性能探测和统计技术，可以对响应时间、网络抖动、丢包率等网络信息进行统计。NQA 能够实时监视网络 QoS，在网络发生故障时进行有效的故障诊断和定位。

同时，NQA 也是网络故障诊断和定位的有效工具。

目的

为了使网络服务质量可见，使用户能够自行检查网络服务质量是否达到要求，需要采取以下措施：

- 在设备上提供能够说明网络服务质量的数据。
- 在网络中部署探针设备能对网络服务质量进行监控。

部署上述措施时，需要在设备侧提供时延、抖动、丢包率等相关统计参数和使用专用的探针设备，增加了设备和资金的投入。

当设备提供 NQA 时，就不用部署专门的探针设备，可以有效的节约成本。NQA 可以实现对网络运行状况的准确测试，输出统计信息。

NQA 监测网络上运行的多种协议的性能，使用户能够实时采集到各种网络运行指标，例如：HTTP 的总时延、TCP 连接时延、DNS 解析时延、文件传输速率、FTP 连接时延、DNS 解析错误率等。

原理描述

构造测试例

NQA 测试中，把测试两端称为客户端和服务器端（或者称为源端和目的端），NQA 的测试是由客户端（源端）发起。在客户端通过命令行配置测试例或由网管端发送相应测试例操作后，NQA 把相应的测试例放入到测试例队列中进行调度。

启动测试例

启动 NQA 测试例，可以选择立即启动、延迟启动、定时启动。在定时器的时间到达后，则根据测试例的测试类型，构造符合相应协议的报文。但配置的测试报文的大小如果无法满足发送本协议报文的最小尺寸，则按照本协议规定的最小报文尺寸来构造报文发送。

测试例处理

测试例启动后，根据返回的报文，可以对相关协议的运行状态提供数据信息。发送报文时的系统时间作为测试报文的发送时间，给报文打上时间戳，再发送给服务器端。服务器端接收报文后，返回给客户端相应的回应信息，客户端在接收到报文时，再一次读取系统时间，给报文打上时间戳。根据报文的发送和接收时间，计算出报文的往返时间。

2， NQA 用于 HTTP 探测时的工作过程

HTTP 测试

目前 NQA 支持九种测试类型：ICMP-echo、DHCP、FTP、HTTP、UDP-jitter、SNMP、TCP、UDP-echo 和 DLSw 测试

NQA 的 HTTP 测试主要是测试客户端是否可以与指定的 HTTP 服务器建立连接，从而判断该设备是否提供了 HTTP 服务以及建立连接的时间。

如图 1 所示，NQA 的 HTTP 测试提供三个阶段的响应速度：

- DNS 解析时间：客户端（RouterA）发送 DNS 报文给 DNS 服务器，DNS 服务器将 HTTP 服务器域名解析为 IP 地址，DNS 解析报文返回到客户端所花费的总时间。
- TCP 建立连接时间：客户端（RouterA）与 HTTP 服务器通过 TCP“三次握手”建立连接所用的时间。
- 交易时间：客户端（RouterA）发送 Get 或 Post 报文给 HTTP 服务器，响应报文到达 HTTP 服务器的时间。

联动功能的实现

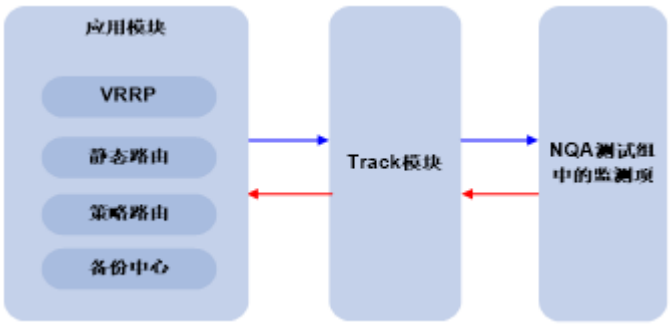
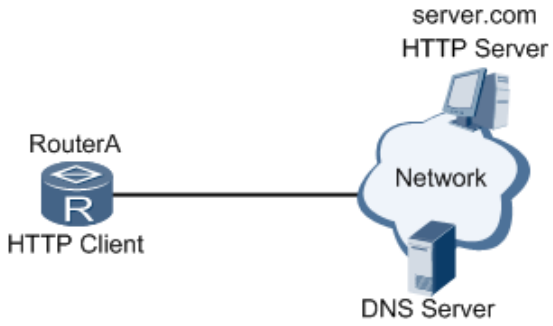


图 1 HTTP 测试场景



通过HTTP测试，从客户端接收到的信息中可以计算出：

- 最小DNS查询时间、最大DNS查询时间及DNS查询时间总和。
- 最小TCP连接建立时间、最大TCP连接建立时间及TCP连接建立时间总和。
- 最小HTTP交易时间、最大HTTP交易时间及HTTP交易时间总和。

HTTP测试的结果和历史记录将记录在测试例中，可以通过命令行来查看探测结果和历史记录。

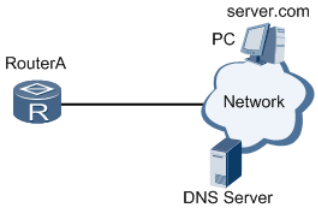
DNS测试

NQA的DNS测试以UDP报文为承载，通过模拟DNS Client向指定的DNS服务器发送域名解析请求，根据域名解析是否成功及域名解析需要的时间，来判断DNS服务器是否可用，及域名解析速度。

如图1所示，DNS测试的过程如下：

1. 客户端（RouterA）向DNS Server发送要求解析给定的DNS名称的Query报文。
2. DNS Server收到报文后，通过解析构造Response报文，然后再把这个数据包发回到客户端。
3. 客户端收到数据包后通过计算客户端接收报文的时间和客户端发送报文的时间的差，计算出DNS域名解析时间。从而清晰的反映出网络DNS协议的性能状况。

图1 DNS测试场景



DNS测试只是模拟域名解析的过程，不会保存要解析的域名与IP地址的对应关系。

DNS测试的结果和历史记录将记录在测试例中，可以通过命令行来查看探测结果和历史记录。

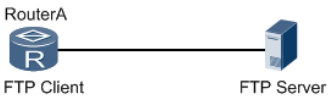
FTP测试

NQA的FTP测试以TCP报文为承载，用于检测是否可以与指定的FTP服务器建立连接，以及从FTP服务器下载指定文件或向FTP服务器上载指定文件的速度。

如图1所示，FTP测试提供两个阶段的响应速度：

- 控制连接时间：客户端（RouterA）与FTP Server通过TCP“三次握手”建立控制连接的时间以及通过控制连接交互信令的时间。
- 数据连接时间：客户端（RouterA）通过数据连接从FTP服务器下载指定文件或向FTP服务器上载指定文件的时间。

图1 FTP测试场景



通过FTP测试，从客户端接收到的信息中可以计算出：

- 最小控制连接时间、最大控制连接时间及平均控制连接时间。
- 最小数据连接时间、最大数据连接时间及平均数据连接时间。

FTP测试支持文件下载和文件上载操作。文件下载操作并不会把文件放到本地的文件系统，只是计算下载该文件所需要的时间，取得数据后随即自动释放占用的内存；文件上载操作并不是将本地文件放到服务器上，而是上传固定大小及内容的文件（文件名由用户配置，数据为系统内部指定的固定数据；如果配置的文件名和服务器上已有的文件重名，则覆盖原来的文件），测试完成后该文件并不被删除。因此，FTP测试与本地文件系统无关。

FTP测试的结果和历史记录将记录在测试例中，可以通过命令行来查看探测结果和历史记录。

DHCP测试

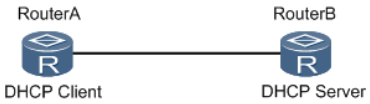
NQA的DHCP测试以UDP报文为承载，模拟DHCP Client在指定的接口上发起DHCP请求，根据是否申请到地址，确定接口所在的网络中是否有DHCP Server服以及测试申请到地址的时间。

如图1所示，DHCP测试过程如下：

1. 源端（RouterA）从需要获得地址的接口，向接口所在网段广播查询DHCP Server的Discovery报文。
2. DHCP Server（RouterB）收到报文后，向源端回送Offer报文，报文中包含了DHCP Server的IP地址。
3. 源端向接口所在网段广播要求获取IP地址的Request报文，报文中包含了DHCP Server的IP地址信息。
4. DHCP Server收到报文后，向源端回送ACK报文，报文中包含了DHCP Server分配给相应接口的IP地址。

   源端收到数据包后通过计算源端接收报文的时间和源端最初发送Discovery报文的时间的差，计算出从DHCP服务器获取IP地址的时间。

图1 DHCP测试场景



DHCP测试只是借用操作接口发送DHCP报文，申请到地址后立即释放DHCP租约，不会为接口真正申请地址，因此不会占用DHCP Server的地址资源。进行DHCP测试的操作接口必须处于Up状态。

DHCP测试的结果和历史记录将记录在测试例中，可以通过命令行来查看探测结果和历史记录。

ICMP测试

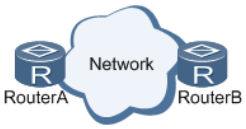
NQA的ICMP测试例用于检测源端到目的端的路由是否可达。ICMP测试提供类似于命令行下的Ping命令功能，但输出信息更为丰富：

- 默认情况下能够保存最近5次的测试结果。
- 结果中能够显示平均时延、丢包率，最后一个报文正确接收的时间等信息。

如图1所示，ICMP测试的过程如下：

1. 源端（RouterA）向目的端（RouterB）发送构造的ICMP Echo Request报文。
2. 目的端（RouterB）收到报文后，直接回应ICMP Echo Reply报文给源端（RouterA）。

图1 ICMP测试场景



源端收到报文后，通过计算源端接收时间和源端发送时间之差，计算出源端到目的端的通信时间，从而清晰的反应出网络性能及网络畅通情况。

ICMP测试的结果和历史记录将记录在测试例中，可以通过命令行来查看探测结果和历史记录。