Cisco CCIE Routing and Switching

---- kaka's Note

Author: 房 智 勇

-----上海交通大学网络信息管理部

CCIE Routing and Switching--kaka's Note

bibliography

Part 1. Routing

<Routing TCP/IP,Volume I>

<Routing TCP/IP,Volume II>

<Cisco OSPF Command and Configuration Handbook>

<Internet Routing Architectures,2nd edition>

<Cisco BGP design and implemention>

<Cisco BGP-4 Command and Configuration Handbook>

<Cisco Press - IS-IS Network Design Solutions[2002]>

<Redhair-Routing Protocols Illstrated Vol.1>

<Cisco Press - OSPF Network Design Solutions>

Part2.Switching

<Cisco LAN Switching>

<Cisco Self-Study: BCMSN(642-811)>

<Catalyst 3750 Multilayer Switch Software Configuration Guide>

Qos

<KKblue Qos Handbook>

<Huawei-3Com Qos introduction>

<Cisco IOS Quality of Service Solutions Configuration Guide,Release 12.4>:

<Catalyst QoS-Quality of Service in Campus Networks>

Part3.Multicast

<Climber Cisco multicast Note>

<Cisco Multicast-PPT>

<IP 组播网络设计开发（第 1 卷）>

<Interdomain Mulitcast Solutions guide>

Part4.WAN Connections

<Cisco IOS Dial Technologies Configuration Guide,Release 12.4>

<Cisco IOS Wide-Area Networking Configuration Guide,Release 12.4>

<Cisco Troubshooting Remote Network Access>

Part5.IPv6

<Cisco IOS IP Configuration Guide,Release 12.4>

<Implementing Cisco IPv6 Networks>

Part6 VPN

IPSec VPN

MPLS VPN

Part7.Feature

<Cisco Documentation http://www.cisco.com/univercd/home/home.htm>

OSI 7-Layers Model

Basic Introduction

把一个大的网络分成几个小点的网络称之为网络分段(network segment),这些工作由routers,switches 和bridges 来完成

引起LAN 拥塞的可能的原因是:

1. 太多的主机存在于1 个广播域(broadcast domain)
2. 广播风暴
3. 多播(multicast)
4. 带宽过低

在网络中使用routers 的优点:

1. 它们默认是不会转发广播的
2. 它们可以基于layer-3(Network layer)的信息来对网络进行过滤

switches 的主要目的:提高LAN 的性能,提供给用户更多的带宽

冲突域(collision domain):Ethernet 术语之1,处于冲突域里的某个设备在某个网段发送数据包,强迫该网段的其他所有设备注意到这个包.而在某1 个相同时间里,不同设备尝试同时发送包,那么将在这个网段导致冲突的发生,降低网络性能

bridges 在某种意义上等同与switches,不同的地方是bridges 只包括2 到4 个端口(port),而switches 可以包括多达上百端口.但是相同的地方是它们都可以分割大的冲突域为数个小冲突域,因为1个端口即为1个冲突域,但是它们仍然处在1个大的广播域中.分割广播域的任务,可以由routers 来完成

Notice：交换机的每个端口构成一个冲突域， hub 只有一个冲突域， router 每个接口都是一个广播域和冲突域

Internetworking Models

早期各个网络厂商拥有私有网络,不便于同其他厂商的网络进行通讯.于是,在20 世纪70 年代末期,ISO 组织创建了OSI(Open System Interconnection)参考模型.

OSI 参考模型,用于帮助不同厂家创建可与对方进行协同工作的网络设备和软件等等,最大的特点是分层.但是它仍然只是个参考模型而非物理模型

OSI 参考模型分层化的优点:

1. 允许多厂家共同发展网络标准化组件
2. 允许不同类型的网络硬件和软件相互通信
3. 防止其中某层的变化影响到其他层,避免牵制到整个模型

OSI model

OSI 参考模型分为7层,高3层定义了端用户如何进行互相通信;底部4层定义了数据是如何端到端的传输.最高3层,也称之为上层(upper layer),它们不关心网络的具体情况,这些工作是又下4 层来完成

OSI 参考模型共有7 层

7, Application layer
6, Presentation layer
5, Session layer
4, Transport layer
3, Network layer
2, Data Link layer
1, Physical layer

在整个OSI 参考模型上运行的网络设备有:

1. 网络管理工作站(NMS)
2. 网页和应用程序服务器
3. 网关(gateways)
4. 网络上的主机(hosts)

OSI 参考模型每层的任务:

1. Application 层:提供用户接口

2. Presentation 层:表述数据;对数据的操作诸如加密,压缩等等
3. Session 层:建立会话,分隔不同应用程序的数据
4. Transport 层:提供可靠和不可靠的数据投递;在错误数据重新传输前对其进行更正
5. Network 层:提供逻辑地址,用于routers 的路径选择
6. Data Link 层:把字节性质的包组成帧;根据MAC地址提供对传输介质的访问;实行错误检测,不实行错误更正
7. Physical 层:在设备之间传输比特(bit);定义电压,线速,针脚等物理规范

OSI 参考模型每层的功能:

1. Application 层:提供文件,打印,数据库,和其他应用程序等服务
2. Presentation 层:数据加密,压缩和翻译等等
3. Session 层:会话控制
4. Transport 层:提供端到端的连接
5. Network 层:路由(routing)
6. Data Link 层:组成帧
7. Physical 层:定义物理拓扑结构

Application layer:

Application layer 作为实际应用程序和presentation layer 的接口通过某种方式把应用程序的有关信息送达到协议栈的下面各层,它只是应用程序的一个接口,需要处理远程资源时才会起作用,应用层还负责识别并建立想要通信的计算机一方的可用性。

Presentation layer:

表示层因为它的用途而得名,它为应用层提供数据,并负责数据转换和代码的格式化。主要提供数据压缩加密转换服务,例如tiff jpeg midi rtf 等…

Session layer:

负责建立,管理和终止表示层实体之间的session 连接,他在系统之间协调通信过程,并提供3 种不同的方式来组织他们之间的通信 全双工,半双工,单工(full duplex ,half duplex simplex)通信,总之,会话层基本上用来使不同应用程序的数据与其它应用程序的数据保持隔离.

一些Session layer 协议和接口的例子:

1. Network File System(NFS)
2. Structured Query Language(SQL)
3. Remote Procedure Call(RPC)
4. X Window
5. AppleTalk Session Protocol
6. Digital Network Architecture Session Control Protocol(DNA SCP)

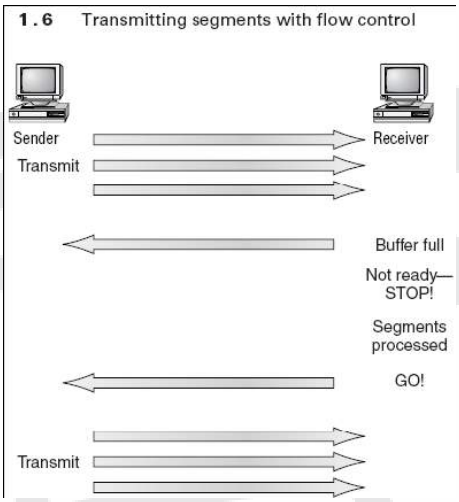
Transport layer:

将数据分段并重组为数据流(data stream)。TCP UDP 都工作在传输层,当采用TCP/IP 协议时,程序开发者可以在这2 者之间做出选择。传输层负责为实现上层应用程序的多路复用,建立会话连接和断开虚电路提供机制.通过提供透明的数据传输,他也对高层隐藏了任何与网络有关的细节信息。

流量控制:(flow control)

在传输层通过进行流量控制,以及通过在系统之间允许用户请求可靠的数据传输, 就可以保证数据的完整性。流量控制可以防止在连接的一侧的发送主机使接受主机的缓冲区产生溢出。缓冲区溢出将导致数据的不完整.如果数据发送方传输数据过快,接受方将数据报(datagrams)暂时存储在缓冲区(buffer)可靠的数据传输采用了面向连接(connection-oriented)通信方式,保证:

1. 接受方接受到被传输的段(segment)以后将发回确认(acknowledge)给发送方
2. 任何没有经过确认的段将被重新传输
3. 段在达到接受方之前应按照适当的顺序
4. 可以进行管理的流控制技术用于避免拥塞,超载(overloading)和数据的丢失



面向连接的通信:(connection-oriented communication)

在可靠的传输层操作中，一个想要传送的设备同过创建会话与远程设备建立连接，通常称为3 次握手协议

3握手协议：(three-way handshake)；

1. 第一个”同意连接”数据段用来请求同步，
2. 第二个，第三个数据段用来确认请求，并在主机之间建立连接参数。 这里，接受方的排序也要求进行同步，以便建立双向连接.
3. 最后一个数据段也用来确认。她通知目的主机已同意建立连接，并且已经建立了实际的连接。可以开始数据传输了.传输数据量过大时，会出现一些问题，某一台机器收到大量的数据包，数度太快，造成缓冲区溢出，最后不得不丢弃.随后到来的所有数据包。但不用担心，网络中有流量控制系统，出现这种情况，接收方会发出一个not ready 的信号， 待处理完毕后，又发出 ready，go on 的信号，继续传输。 流量控制类型： 窗口机制， 缓冲和拥塞避免.

a. 窗口机制(windowing)

发送方在没有收到确认是，别允许发送的数据段的数量，称为窗口
窗口的尺寸大小控制了有多少信息从一端传向另一端，虽然有些协议以数据包的数量来量化信息，但tcp/ip通过计算字节数来量化信息

b. 确认

为了保证数据传送的不重复性和不被丢失，可以同过“带重传的肯定确认”来实现，方法是要求接收方在收到数据是，发给发送方一个确认信息，来与发送方机器保持通信。 当发送一个数据包时，发送方及其启动一个计时器，在规定时间内，未收到对方确认是， 显示request time out, 重新发送一次

Network layer:

负责设备的寻址，跟踪网络中设备的位置，并决定传送数据的最佳路径。 路由器和3 层交换机工作在这个层上。

路由工作原理：

首先，接受到一个包，然后检查其目的ip，查询路由表，选择最佳路径，选择一个interface，包就被送到那个端口，并被封装成帧，送出本地网络，如果在路由表里找不到相应的目的网络的表项，则自动丢弃该包。

网络层有2 种类型的包(packets)：数据包(data)和路由更新(router update)包，前者，很显然，用来传送用户数据。后者，用来向相邻路由器通知连接到网络的所有路由器的更新信息。这种协议为主动路由协议，如RIP EiGRP OSPF.

路由表：

包含如下信息：

1. Network Address：他们与特定的协议有关的网络地址。
2. Interface ： 当数据包被发送到特定的网络时，数据包将选择一个外出接口
3. Metric ： 指到远程网络的距离。不同的协议度量方式不同。

路由器特点：

1. 屏蔽广播包和组播包(multicast)
2. 使用logic address, 它存在于网络层的报头中，用来决定下一跳(hop)的路由地址
3. 可管理，创建访问列表
4. 可提供第2 层的桥接功能，并通过同一个接口传送。
5. 可提供vlan 间的连接
6. 提供quality of service(Qos)

Data link layer:

the Data Link layer 负责数据的物理传输,错误检测,网络拓扑和流控制.这个意味着在数据LAN 上将根据硬件地址来进行投递,还要把Network layer 的包翻译成比特用于在Physicallayer 上传输.

IEEE 以太网(Ehernet)的Data Link layer 有2 个子层:

1. Media Access Control(MAC)802.3:
   这层定义了物理地址和拓扑结构,错误检测,流控制等.共享带宽,先到先服务原则(first come/first served)
2. Logical Link Control(LLC)802.2:
   负责识别Network layer 协议然后封装(encapsulate)数据.LLC 头部信息告诉Data Link layer 如何处理接受到的帧,LLC 也提供流控制和控制比特的编号

Switches and Bridges at the Data Link Layer

第二层的设备switches 被认为是基于硬件的bridges, 因为采用的是1 种叫做application-specific integrated circuit(ASIC)的特殊硬件.ASICs 可以在很低的延时(latency)里达到gigabit 的速度;而bridges 是基于软件性质的.

延时:1 个帧从进去的端口到达出去的端口所耗费的时间

透明桥接(transparent bridging):如果目标设备和帧是在同1个网段,那么2层 设备将堵塞端口防止该帧被传送到其他网段;如果是和目标设备处于不同网段,则该帧将只会被传送到那个目标设备所在的网段每个和switches 相连的网段必须是相同类型的设备,比如你不能把令牌环(Token Ring)上的主机和以太网上的主机用switches 混合相连,这种方式叫做media translation,不过你可以用routers 来连接这样不同类型的网络.

在LAN 内使用switches 比使用hubs 的好处:

1.插入switches 的设备可以同时传输数据,而hubs 不可以

2.在switches 中,每个端口处于1 个单独的冲突域里,而hubs 的所有端口处于1 个大的冲突域里,可想而知,前者在LAN 内可以有效的增加带宽.但是这2 种设备的所有端口仍然处于1 个大的广播域里

Physical layer:

the Physical layer 负责发送和接受比特.比特由1 或者0 组成.这层也用于识别数据终端装备(data terminal equipment,DTE)和数据通信装备(data communication equipment,DCE)的接口

DCE 一般位于服务商(sevice provider)而DTE 一般是附属设备.可用的DTE 服务通常是经由modem 或者channel service unit/data sevice unit(CSU/DSU)来访问

hubs:其实是多端口的repeaters,重新放大信号用,解决线路过长,信号衰减等问题.

1 个物理星形(star)拓扑结构,实际在逻辑上是逻辑总线(bus)拓扑结构

Ethernet Networking

以太网采用1 种争夺(contention) 介质访问方法,这个机制使得在1 个网络上所有主机共享带宽.采用了Physical layer 和Data Link layer 的规范.它采用1 种带冲突检测的载波监听多路访问的(Carrier Sense Multiple Access with Collision Detection,CSMA/CD)机制CSMA/CD:帮助共享带宽的设备避免同时发送数据,产生冲突的协议.补偿算法(Backoffalgorithms)

用于决定产生冲突的2 台设备何时重新传输数据

CSMA/CD 网络带来的问题:

1.延迟(delay)

2.低吞吐量(throughput)

3.拥塞

Half- and Full-Duplex Ethernet

half-duplex(半双工)以太网:它只采用1 对线缆.如果hubs 与switches 相连,那么必须以半双工的模式操作,因为端工作站必须能够检测冲突.半双工以太网带宽的利用率只为上限的30%-40%full-duplex(全双工)以太网:采用2 对线缆,点对点point-to-point)的连接,没有冲突,双倍带宽利用率全双工以太网可以使用在以下的3 种形式里:

1.switch 和host 相连

2.switch 和switch 相连

3.用交叉线缆(crossover cable)相连的host 和host

自动检测机制(auto-detection mechanism):当全双工以太网端口电源启动时,它先与远端相连,并且与之进行协商.看是以10Mbps 的速度还是以100Mbps 的速度运行;再检查是否可以采用全双工模式,如果不行,则切换到半双工模式

Ethernet at the Data Link Layer

4 种类型的以太网帧:

1.Ethernet II

2.IEEE 802.2

3.IEEE 802.3

4.SNAP

Ethernet Addressing

MAC 地址是烧录在Network Interface Card(网卡,NIC)里的.MAC 地址,也叫硬件地址,是由48比特长(6 字节),16 进制的数字组成.0-24 位是由厂家自己分配.25-47 位,叫做组织唯一标志符(organizationally unique identifier,OUI).OUI是由IEEE分配给每个组织.组织按高到低的顺序分配1 个唯一的全局地址给每个网卡以保证不会有重复的编号.第47 位为individual/Group(I/G)位,当I/G 位为0 的时候,我们可以设想这个地址是MAC 地址的实际地址可以出现在MAC 头部信息;当I/G 位为1 的时候,我们可以设想它为广播或多播.第46 位叫做G/L 位,也叫U/L 位.当这个位为0 的时候代表它是由IEEE 分配的全局地址;当这个位为1 的时候,代表本地管理地址(例如在DECnet 当中)

Ethernet Frames

第二层用于把第一层的比特连接成字节,再组成帧(frames)

3 种介质访问方法的类型:

1.争夺(contention),用于在以太网中

2.令牌传递(token passing),用于在FDDI 和Token Ring 里

3.投票(polling),用于在IBM Mainframes 和100VG-AnyLAN 中

循环冗余校验(cyclic redundancy check,CRC):用于错误检测,而非错误更正

隧道(tunneling):把不同类型的帧封装在1 个帧里

Ethernet II 帧:

1.前导(preamble)字段:交替的1 和0 组成.5Mhz 的时钟频率,8 字节,包含7 字节的起始帧分界符(start frame delimiter,SFD),SFD 是10101011,最后1 个字节同步(sync)

2.目标地址(destination address,DA):6 字节

3.源地址(source address,SA):6 字节

4.类型(type)字段:用于辨别上层协议,2 字节

5.数据(data):64 到1500 字节

6.帧校验序列(frame check sequence,FCS):4 字节,存储CRC 值

802.3 Ethernet 帧:

1.前导(preamble)字段:交替的1和0组成.5Mhz 的时钟频率,8 字节,包含7 字节的起始帧分界

符(start frame delimiter,SFD),SFD 是10101011,最后1 个字节同步(sync)

2.目标地址(destination address,DA):6 字节

3.源地址(source address,SA):6 字节

4.长度(length)字段:不能辨别上层协议,2 字节

5.数据(data):64 到1500 字节

6.帧校验序列(frame check sequence,FCS):4 字节,存储CRC 值

802.2 and SNAP

因为802.3 Ethernet 帧没有鉴别上层协议的能力(使用的是length 字段),所以,它需要IEEE 定义的802.2 LLC 标准来帮它实现这个功能

802.2 帧(SAP):

1.目标服务访问点(dest SAP)字段: 1 个字节

2.源服务访问点(source SAP)字段: 1 个字节

3.控制字段:1 或2 个字节

4.数据:大小可变

1 个802.2 帧是由802.3Ethernet 帧加上LLC 信息组成,这样它就可以辨别上层协议

802.2 帧(SNAP):它有自己的协议来辨别上层协议

1.目标服务访问点(dest SAP)字段: 1 个字节,总为AA

2.源服务访问点(source SAP)字段: 1 个字节,总为AA

3.控制字段:1 或2 个字节,值总为3

4.OUI ID:3 字节

5.类型(type)字段:2 字节,辨别上层协议

6.数据:大小可变

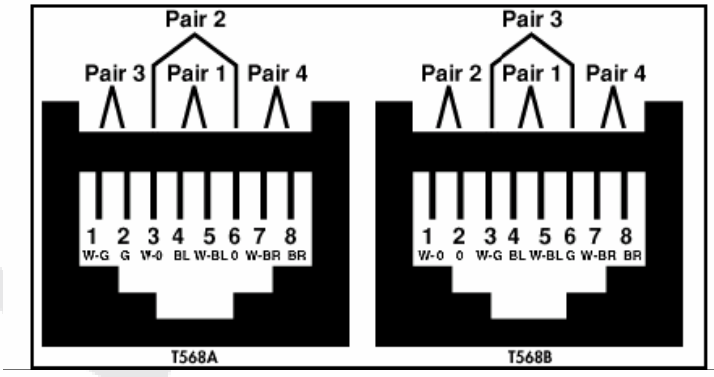
Ethernet @ Physical Layer

一些原始的和扩展的IEEE 802.3 的标准:

1. 1.10Base2:Base 是指基带传输技术,2 指最大距离接近200 米,实际为185 米,10 指10Mbps 的速度,采用的是物理和逻辑总线拓扑结构,AUI 连接器2.10Base5:5 指最大距离500 米,10 指10Mbps 的速度,采用的是物理和逻辑总线拓扑结构,AUI连接器
2. 3.10BaseT:10 指10Mbps 的速度,采用的是物理星形和逻辑总线拓扑结构, 3 类UTP 双绞线,RJ-45 连接器,每个设备必须与hub 或者switch 相连,所以1 个网段只能有1 台主机
3. 4.100BaseT:100 指100Mbps 的速度,采用的是物理星形和逻辑总线拓扑结构, 5,6 或者7 类UTP2 对双绞线,RJ-45 连接器, 1个网段1 台主机
4. 5.100BaseFX:100 指100Mbps 的速度,光纤技术,点对点拓扑结构,最大距离412 米, ST 或者SC连接器
5. 6.1000BaseT:1000 指1000Mbps 的速度,光纤技术,点对点拓扑结构,最大距离412 米, 5 类UTP4对双绞线,最大距离100 米

Ethernet 电缆的连接

Diagram showing both T568-A and T568-B cabling wire colors



Straight-Through Cable

直通线用于连接:

1.主机和switch/hub

2.router 和switch/hub

直通线只使用1,2,3,6 针脚,2 端的连法是一一对应

Crossover Cable

交叉线用于连接:

1switch 和switch

2.主机和主机

3.hub 和hub

4.hub 和switch

5.主机与router 直连

交叉线只使用1,2,3,6 针脚,2 端的连法是1 连3,2 连6,3 连1,6 连2。即一端使用T568A一端使用T568B

Rolled Cable

反转线不是用来连接以太网连接的,它是用来连接主机与router 的com 口(console serial port)的,它采用1 到8 跟针脚,2 端全部相反对应

当主机与router的console口用反转线连好后,启动Window系统里的超级终端程序即可对router 进行连接:

1.Bps:9600

2.Data bits:8

3.Parity:None

4.Stop bits:1

5.Flow control:none

Data Encapsulation

封装(encapsulation):把OSI 参考模型每层自己的协议信息加进数据信息的过程,反之叫做解封装

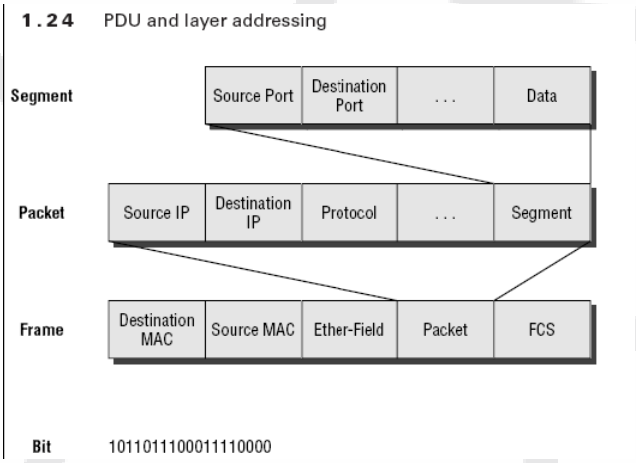
协议数据单元(protocol data units,PDU):数据包括封装进去的信息在OSI 参考模型每层的叫法:

1.Transport layer:segment

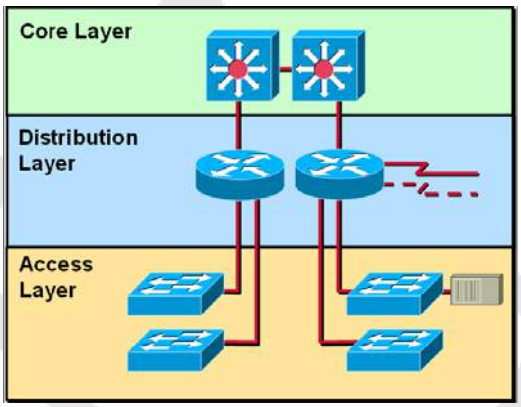
2.Network layer:packet 或者datagram

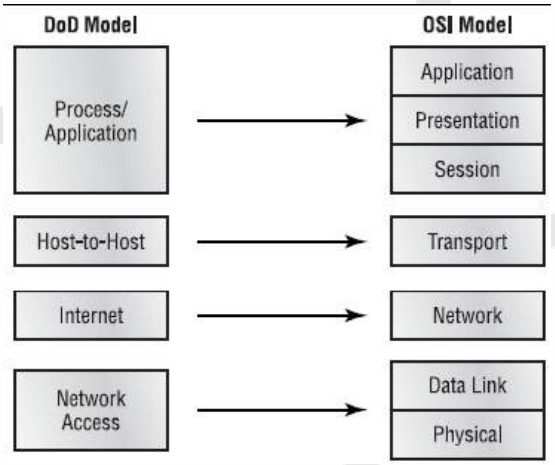
3.Data Link layer:frame

4.Physical layer:bits



Cisco 3-Layers model





TCP/IP Protocols

TCP/IP and DoD model:

DoD 模型分为4 层,从上到下是:

1. Process/Application layer
2. Host-to-Host layer
3. Internet layer
4. Network Access layer

在功能上和OSI 参考模型互相对应的话,那么:

1. DoD 模型的Process/Application 层对应OSI 参考模型的最高3 层
2. DoD 模型的Host-to-Host 层对应OSI 参考模型的Transport 层
3. DoD 模型的Internet 层对应OSI 参考模型的Network 层
4. DoD 模型的Network Access 层对应OSI 参考模型的最底2 层

The Process/Application Layer	telnet	FTP	LPD	SNMP
	TFTP	SMTP	NFS	X Window
Host-to-Host Layer	TCP		UDP	
Internet layer	ICMP	ARP	RARP	
	IP			
Network Access Layer	Ethernet	FrameRelay	Token Ring	FDDI

Dynamic Host Configuration Protocol (DHCP)/BootP(Bootstrap Protocol)

动态主机配置协议(DHCP)服务器可以提供的信息有:

1. IP 地址
2. 子网掩码(subnet mask)
3. 域名(domain name)
4. 默认网关(default gateway)
5. DNS
6. WINS 信息

一个DHCP 服务器可以给我们提供比这个更多的信息,为了收到一个ip 地址,发送DHCP 发现信息的客户机发送出2,3 层上的广播, 2 层mac 全为FF,3 层ip 为 255.255.255.255 在Transport Layer 采用UDP 发送

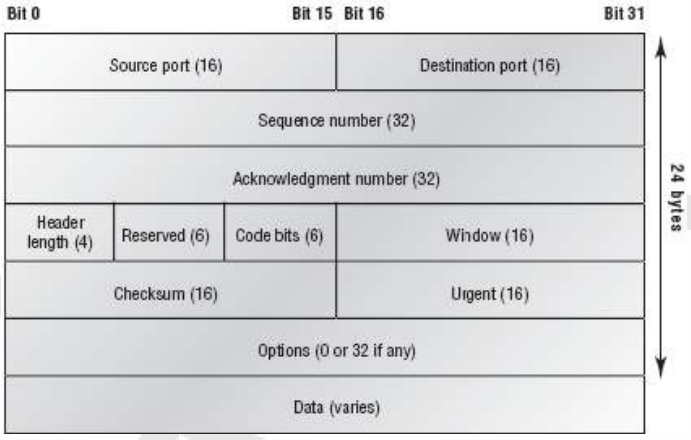
The Host-to-Host Layer Protocols

Host-to-Host 层描述了2 种协议:

1. 传输控制协议(Transmission Control Protocol,TCP)
2. 用户数据报协议(User Datagram Protocol,UDP)

TCP

当1个主机开始发送数据段(segment)的时候,发送方的TCP 协议要与接受方的TCP 协议进行协商并连接,连接后即所谓的虚电路(virtual circuit),这样的通信方式就叫做面向连接(connection-oriented).面向连接的最大优点是可靠,但是它却增加了额外的网络负担(overhead)



TCP segment copied from a network analyzer:

TCP - Transport Control Protocol

Source Port: 5973

Destination Port: 23

Sequence Number: 1456389907

Ack Number: 1242056456

Offset: 5

Reserved: %000000

Code: %011000

Ack is valid

Push Request

Window: 61320

Checksum: 0x61a6

Urgent Pointer: 0

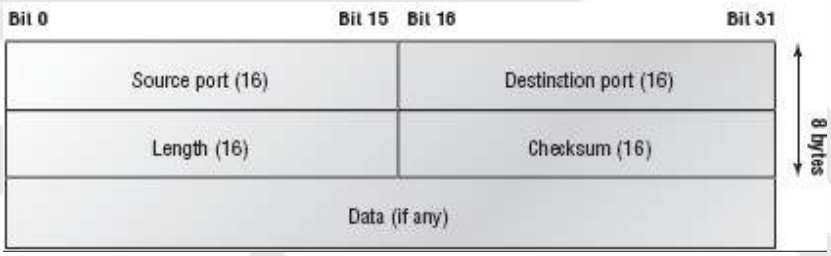
No TCP Options

TCP Data Area:

vL.5.+.5.+.5.+.5 76 4c 19 35 11 2b 19 35 11 2b 19 35 11

2b 19 35 +. 11 2b 19

Frame Check Sequence: 0x0d00000f



UDP

User Datagram Protocol(UDP)

UDP 协议的最他特点是无连接(connectionless),即不可靠,因为它不与对方进行协商并连接,它也不会给数据段标号,也不关心数据段是否到达接受方

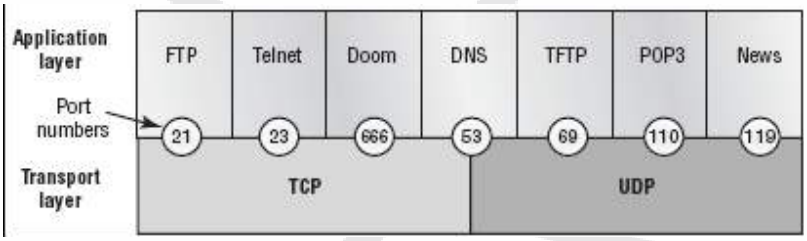
Key Concepts of Host-to-Host Protocols

现在把TCP 协议和UDP 协议的一些特性做个比较:

1. TCP.协议在传送数据段的时候要给段标号;UDP 协议不
2. TCP 协议可靠;UDP 协议不可靠
3. TCP 协议是面向连接;UDP 协议采用无连接
4. TCP 协议负载较高,采用虚电路;UDP 协议低负载
5. TCP 协议的发送方要确认接受方是否收到数据段(3次握手协议);UDP 反之
6. TCP 协议采用窗口技术和流控制;UDP 协议反之

Port Numbers

TCP 和UDP 协议必须使用端口号(port number)来与上层进行通信,因为不同的端口号代表了不同的服务或应用程序.1 到1023 号端口叫做知名端口号(well-known port numbers).源端口一般是1024 号以上随机分配



The Internet Layer Protocol

在DoD的模型中，设置internet-layer有2个主要理由，1路由，2为上层提供一个简单的网络接口。

Protocol @ internet-layer

1. internet protocol
2. ICMP
3. ARP
4. RARP

5. 代理ARP

Internet Protocol(IP)

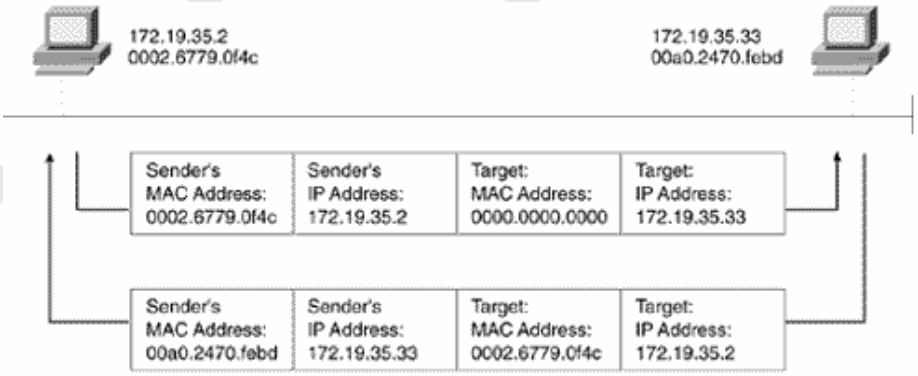
IP 协议查找每个数据包(packets)的地址,然后,根据路由表决定该数据包下1 段路径该如何走,寻找最佳路径

Internet Control Message Protocol(ICMP)

ICMP 协议一样是工作在DoD 模型的Internet 层,IP 协议使用ICMP 协议来提供某些不同的服务,ICMP 协议是一种管理协议一些ICMP 协议相关信息和事件:

1. 目标不可达(destination unreachable):假如1 个routers 不能把IP 协议数据报发送到更远的地方去,于是router 将发送ICMP 协议信息给数据报的发送方,告诉它说目标网络不可达
2. 缓冲区已满(buffer full):假如router 的缓冲区已经存满发送方发来的IP 协议数据报了,它将发送ICMP 协议信息
3. 息给发送方并告诉它缓冲区已满,如果再继续接受的话将导致缓冲区溢出,造成数据丢失
4. 跳(hops):IP 协议数据报经过1 个router,称为经过1 跳
5. Ping(Packet Internet Groper):采用ICMP协议信息来检查网络的物理连接和逻辑连接是否完好
6. Traceroute:根据ICMP 协议信息来跟踪数据在网络上的路径,经过哪些跳

Address Resolution Protocol(ARP)



地址解析协议，用于将IP地址解析为MAC地址，在cisco路由器上可以使用**debug arp**查看arp消息

```
Aretha#debug arp
IP ARP: rcvd req src 172.19.35.2 0002.6779.0f4c, dst 172.21.5.1 Ethernet0
IP ARP: sent rep src 172.21.5.1 0000.0c0a.2aa9,
            dst 172.19.35.2 0002.6779.0f4c Ethernet0

Aretha#
```

同时可以使用show arp查看ARP表的内容.

Cisco路由器保存Arp表的时间为4小时(14400s)

Martha(config)# interface ethernet 0
Martha(config-if)# arp timeout 1800

Arp 表也可以实现静态绑定

Martha(config)# arp 172.21.5.131 0000.00a4.b74c snap

使用**clear arp-cache** 可以清空ARP缓存区

Proxy-ARP

代理 arp 即 arp 欺骗，路由器通过发送代理 arp 信息，让主机认为路由器即为目的主机。
通过这种方式，路由器可以转发数据流

在 cisco 的路由器上，代理 arp 是默认打开的，可以在接口上 **no ip proxy-arp** 关闭代理 arp
典型的使用代理 arp 的 arp 表

```
C:\WINDOWS>arp -a

Interface: 192.168.20.66
  Internet Address      Physical Address      Type
  192.168.20.17         00-00-0c-0a-2a-a9     dynamic
  192.168.20.20         00-00-0c-0a-2a-a9     dynamic
  192.168.20.25         00-00-0c-0a-2a-a9     dynamic
  192.168.20.65         00-00-0c-0a-2c-51     dynamic
  192.168.20.70         00-02-67-79-0f-4c     dynamic
```

Gratuitous -ARP

主机偶尔会发送一个以自己 ip 作为目的地址的 arp 请求，防止 ip 冲突，但这种 arp 在很多 ip 中都没有实现

R -ARP

反向 arp 可以实现 ip 地址到已知硬件地址的映射，但很大程度上被 BOOTP 和 DHCP 所替代

Static Routing

The Route Table

当frame到达路由器的接口以后,路由器检查frame中的目标地址,如果目标地址为路由器 的接口的地址或广播地址的时候,路由器把、packet从frame中剥离出来,传递给Network Layer.然后packet中的目标地址将被检查,接下来还要检查protocol字段.最后再发送给合适的进程 如果packet是可路由的,路由器会查找自己的路由表寻找相应的路由条目.路由条目至少包 含以下2个要素:

1. 目标地址,这个地址是路由器能够到达的地址,
2. 到达目标地址的指向,这个指向也就是所谓的next hop(下一跳)

路由器在地址匹配过程中按最大程度进行匹配,地址按精确程度递减的排列,如下:

1. 主机地址
2. 子网
3. 汇总(summary)地址
4. 主网络号
5. 超网(supernet)
6. 默认(default)地址

如果在地址匹配过程中,不能和路由表中任何条目所匹配,packet将被丢弃,
然后,一个名为 Destination Unreachable(目标不可达)的ICMP信息将发回给源地址

查看IP路由表,使用show ip route,如下:

```
Lewis#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP,
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area,
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2,
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP,
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default,
       U - per-user static route, o - ODR

Gateway of last resort is not set

     10.0.0.0/24 is subnetted, 7 subnets
S       10.1.3.0 [1/0] via 10.1.4.1
S       10.1.2.0 [1/0] via 10.1.4.1
S       10.1.1.0 [1/0] via 10.1.4.1
S       10.1.7.0 [1/0] via 10.1.6.2
C       10.1.6.0 is directly connected, Serial1
C       10.1.5.0 is directly connected, Ethernet0
C       10.1.4.0 is directly connected, Serial0
Lewis#
```

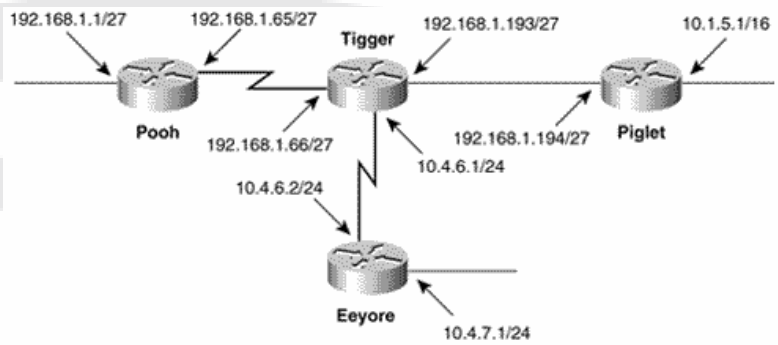
Configuration static routes

一般配置静态路由的步骤如下:

1. 为每条链路确定地址(包括子网地址和网络地址)
2. 为每个路由器,标识非直连的链路地址
3. 为每个路由器写出未直连的地址的路由语句(写出直连地址的语句是没必要的)

比如如上拓扑,写出所有链路的地址,如下:

10.1.0.0/16
10.4.6.0/24
10.4.7.0/24
192.168.1.192/27
192.168.1.64/27
192.168.1.0/27



以路由器 Piglet 为例,非直连的地址,如下:

10.4.6.0/24
10.4.7.0/24
192.168.1.64/27
192.168.1.0/27

最后把这些没有直连的语句写出来,如下:

```
Piglet(config)#ip route 192.168.1.0 255.255.255.224 192.168.1.193
Piglet(config)#ip route 192.168.1.64 255.255.255.224 192.168.1.193
Piglet(config)#ip route 10.4.6.0 255.255.255.0 192.168.1.193
Piglet(config)#ip route 10.4.7.0 255.255.255.0 192.168.1.193
Piglet(config)#ip classless
Piglet(config)#ip subnet-zero
```

上面的 192.168.1.193 是 next hop 地址.还有种方法就是使用出口接口(exit interface)来代替下一跳地址,假设 192.168.1.1 是路由器 Tigger 的 e0 口,上面的其中一条语句就可以写成:

```
Piglet(config)#ip route 10.4.7.0 255.255.255.0 e0
```

这两种方式是存在区别的,如下,先在使用下一跳地址的配置上查看路由表信息:

```
Piglet#sh ip route
S 10.4.7.0 255.255.255.0 [1/0] via 192.168.1.193
```

再在使用 exit interface 代替下一跳地址的配置上查看路由表信息,如下:

```
Piglet#sh ip route
S 10.4.7.0 255.255.255.0 is directly connected, Ethernet0
```

Floating Static Routes

设置一条管理距离稍大于正常使用的一条静态路由，如下，将一条路由的管理距离设置为 50，这样正常使用的这条链路 down 掉后，被设置为浮动路由的备份链路启用

```
Piglet(config)#ip route 10.4.7.0 255.255.255.0 192.168.1.193 50
```

Load Sharing

```
Rabbit#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default,
       U - per-user static route

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 4 subnets, 2 masks
C       10.1.10.0/24 is directly connected, Serial0
S       10.1.5.0/24  [1/0] via 10.1.10.1
                     [1/0] via 10.1.20.1
S       10.4.0.0/16  [1/0] via 10.1.10.1
                     [1/0] via 10.1.20.1
C       10.1.20.0/24 is directly connected, Serial1
S    192.168.0.0/16  [1/0] via 10.1.10.1
                     [1/0] via 10.1.20.1

Rabbit#
```

均衡负载可以是基于目标地址或者是基于每个packet的所谓机遇目标地址的均衡负载,是说假如有2条到达目标地址的路径,那么第一个packet将通过第一条链路到达第一个目标设备,第二个packet将通过第二条链路到达第二个目标设备,第三个packet又将通过第一条链路到达第三个目标设备等等,以次类推.

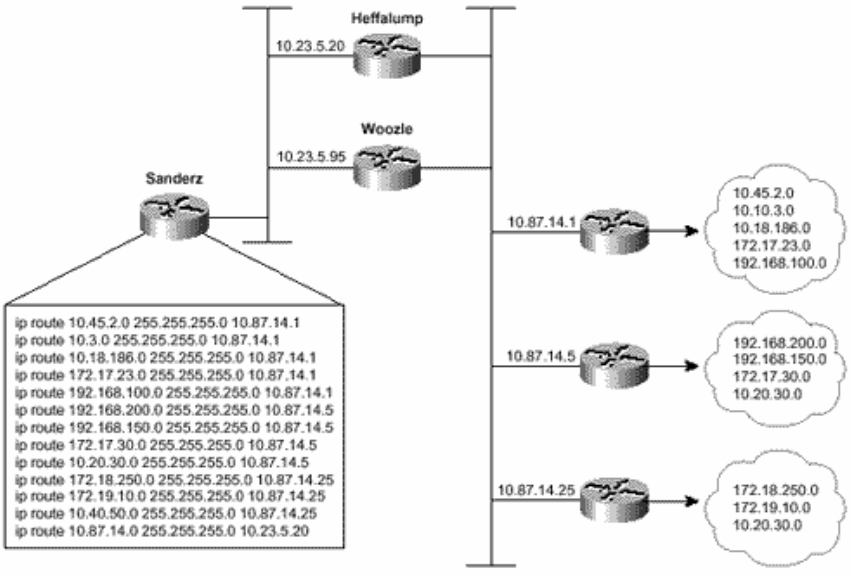
当Cisco路由器工作在默认的交换模式,Fast Switching(快速交换)模式下,就使用这种类型的均衡负载

Fast Switching的工作原理是:

当路由器对第一个packet进行发往目标地址的处理的时候,先查看路由表和选择出口接口,然后获取组成frame的信息(比如ARP表的查询)并进行封装,然后传输.之前获取的这些路由和数据链路信息将被保存在快速交换的 cache 中.接下来,当有要到达和第一个包相同的目标地址的包的时候,就可以不进行路由表和ARP表的查询,直接对 packet 进行交换。快速交换降低了 CPU 的占用和处理时间,并意味着去往某个目标地址的packet都从相同的路由器接口被路由出去.当有到达同一网络不同主机的packet,路由器可能会吧这些 packet 通过另外一条链路进行路由.因此,路由器能做的最好的就是给予目标地址的均衡负载所谓基于基于packet的均衡负载,是说假如有2条到达目标地址的路径,那么第一个packet将通过第一条链路到达目标设备,第二个packet将通过第二条链路到达目标设备,第三个packet又将通过第一条链路到达目标设备等等,以次类推.(这里考虑的是等价的均衡负载)

Cisco路由器工作在Process Switching(进程交换)模式的时候就采用基于 packet 的均衡负载进程交换,是指每次对 packet 的交换,都要查询路由表,选择出口接口和查询数据链路信息,因为每次的路由决策都是独立的.要在某个接口打开进程交换模式,使用 **no ip route-cache** 命令

Recursive Table Lookups

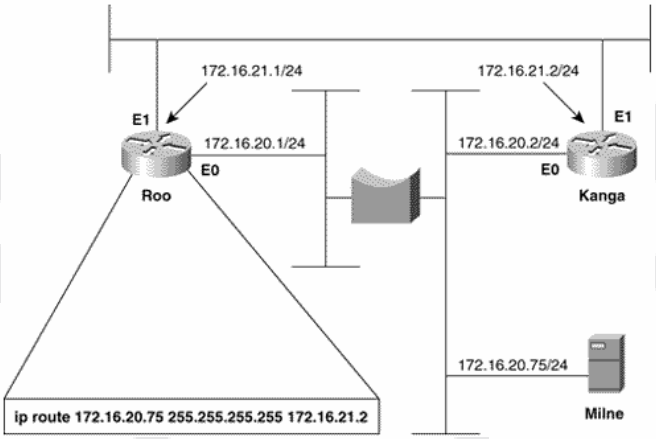


使用递归查询路由，在网络出现变更时配置将节省很多时间，Sanderz 采用了递归路由的方式构成路由表

如果不希望流量经过 Heffalump 而走 Woozle

```
Sanderz(config)# ip route 10.87.14.0 255.255.255.0 10.23.5.95
Sanderz(config)# no ip route 10.87.14.0 255.255.255.0 10.23.5.20
```

Troubleshooting Static Routes



如图,管理员可能会怕网桥的流量过大,使得服务器 Milne 的流量会被延误.于是管理员在路由器上添加一条到达服务器 Milne 的静态路由来避开网桥,如下:

```
Roo(config)#ip route 172.16.20.75 255.255.255.255 172.16.21.2
```

这样的方案看似合理,但是实际上,在路由器 Roo 上增加了到达服务器 Milne 的静态路由以后,packet 不但不能被路由器 Roo 路由,也不能被路由器 Kanga 路由

Traceroute 发现产生路由环路

```
Roo#trace 172.16.20.75
1 172.16.21.2 0 msec 0 msec 0 msec
2 172.16.20.1 4 msec 0 msec 0 msec
3 172.16.21.2 4 msec 0 msec 0 msec
4 172.16.20.1 0 msec 0 msec 4 msec
5 172.16.21.2 0 msec 0 msec 4 msec
6 172.16.20.1 0 msec 0 msec 4 msec
7 172.16.21.2 0 msec 0 msec 4 msec
8 172.16.20.1 0 msec 0 msec 4 msec
9 172.16.21.2 4 msec 0 msec 4 msec
10 172.16.20.1 4 msec 0 msec 4 msec
11 172.16.20.2 4 msec
```

Packet 本不应该被路由器 Kanga 路由,它应该意识到目标设备 Milne 位于和它的 E0 口直连的网络 172.16.20.0 上,然后经过数据链路来把 packet 传输给服务器 Milne.因此,问题可能出在数据链路上,如何确定数据链路是否正确?当要确定到达某个网络的逻辑链路信息是否正确的时候,就要查看路由表;要查看到达某一设备的物理路径信息是否正确的时候,就要查看 ARP 表.如下,使用 show arp 命令查看路由器 Kanga 的 ARP 表:

```
Kanga#sh arp
Protocol Address Age (Min) Hardware Addr Type Interface
Internet 172.16.20.75 2 00e0.1e58.dc39 ARPA Ethernet0
Internet 172.16.21.2 - 00e0.1e58.dcb4 ARPA Ethernet1
Kanga#
```

如图,服务器 Milne 的 MAC 地址实际为 0002.6779.0f4c,和这里的 ARP 表里的条目不符,由此可以判定问题是出在路由器 Kanga 上.再查看路由器 Roo 的 ARP 表,如下:

```
Roo#sh arp
Protocol Address Age (Min) Hardware Addr Type Interface
Internet 172.16.20.75 2 00e0.1e58.dc39 ARPA Ethernet0
```

为什么这里路由器 Roo 的 E0 口的 MAC 地址和服务器 Milne 的 MAC 地址一样?也就是说路由器 Kanga 错误的认为

路由器 Roo 的 E0 口就是服务器 Milne 的接口,于是把 MAC 地址 00e0.1e58.dc39 作为目标 Milne 的目标 MAC 地址进行封装,实际的却是路由器 Roo 的 E0 口接收到这个 frame,然后又把 frame 发回给路由器 Kanga

问题是,路由器 Kanga 是如何得到这个错误的 ARP 信息呢?答案是 proxy(代理) ARP,当路由器 Kanga 初次接收到发往服务器 M ilne 的 packet 的时候,它将发送 ARP 请求服务器 Milne 和路由器 Roo 的 E0 口都会对这个请求进行响应.因为路由器 Roo 有到达服务器 Milne 的路径,但是这条路径所在的网络不是它收到 ARP 请求的网络,于是它发送一条代理 ARP 的应答给路由器 Kanga.之前服务器 Milne 响应 ARP 请求,该 ARP 应答信息保存在路由器 Kanga 的 ARP 表中;由于网桥的延时,路由器 Roo 响应的那条代理 ARP 应答随后才到达路由器 Kanga 并保存,同时也覆盖了原先正常的 ARP 条目

有 2 种方法可以解决上述问题,其中一种是关闭路由器 Roo 的 E0 接口的代理 ARP 功能,如下:

```
Roo(config)#int e0
Roo(config-if)#no ip-proxy
```

还有种办法就是用在路由器 Kanga 上用静态 ARP 条目代替动态获取的,如下:

```
Kanga(config)#arp 172.16.20.75 0002.6779.0f4c arpa
```

Routing Information Protocol (RIP)

Operation of RIP

RIP 是通过 UDP 端口 520 来进行操作的,RIP 信息包是封装在 UDP segment 中的.

RIP 定义了 2 种信息类型:Request message(请求信息)和 Response message(应答信息).

请求信息

用来向邻居请求发送一个 update(更新),

应答信息

运载着这个被请求的 update.

RIP 的 metric 是基于 hop count(跳数)的,metric 为 16 代表不可达。

在刚启动的时候,RIP 从启用了 RIP 的接口上向外广播请求信息,接下来 RIP 进程进入一个循环状态:监听来自其他路由器的请求信息和应答信息.当邻居收到请求信息以后,就发送应答信息给这个发出请求信息的路由器,这个应答信息就包罗了它们的路由表。当发出请求信息的这个路由器收到了应答信息以后,如果这个 update 里包含的路由条目比它本身所拥有的更新,或者本身的路由表里没有这个条目,那就把它放进自己的路由表中.如果本身的路由表有这个条目,并且 update 里的路由条目的 metric,也就是跳数大于它自己的条目的跳数,而且这个 update 是源自它本身的条目指向的下一跳路由器,那么该路由在一段 holddown 周期里将标记为不可达.如果这个 holddown 周期超出,那个邻居仍然发送拥有较高 metric(较多跳数)的 update 作为应答,那么发出请求信息的路由器将接受这一新的 metric

RIP Timers and Stability Features

Update timer

在 RIP 启动之后,平均每 30 秒,启用了 RIP 的接口会发送应答信息(也就是 update),这个 update 包含了路由器除了被 split horizon(水平分割)抑制的完整的路由表.update 周期发送的时间间隔(即 update timer)为 25.5 秒到 30 秒之间(随机),并且 update 的目标地址为 255.255.255.255

invalid timer

当有新的路由条目被加进路由表以后,这个 invalid timer 就初始化为 180 秒(即 6 个 update 周期),如果一条路由 update 在这 180 秒里没有被收到的话,那么这条路由的跳数(即 metric)就被设置为 16,即不可达

flush timer

Cisco 一般把这个时间设置为 240 秒(RFC 1058 中规定的是 300 秒).如果超出这 240 秒,路由将被标记为不可达,并从路由表中移除掉.如下就是一条被标记为不可达,但是还没有被移除的路由:

```
Mayberry#sh ip route
10.0.0.0 255.255.0.0 is subnetted, 4 subnets
C 10.2.0.0 is directly connected, Serial0
R 10.3.0.0 255.255.0.0 is possibly down,
Routing via 10.1.1.1, Ethernet0
C 10.1.0.0 is directly connected, Ethernet0
```

R 10.4.0.0 [120/1] via 10.2.2.2, 00:00:00, Serial0
Mayberry#

holddown timer

为 180 秒. 如果新收到的路由条目的跳数大于本身条目的跳数,那么该路由将进入长为 180 秒的 holddown 状态

4 种 timer 的操作命令如下:

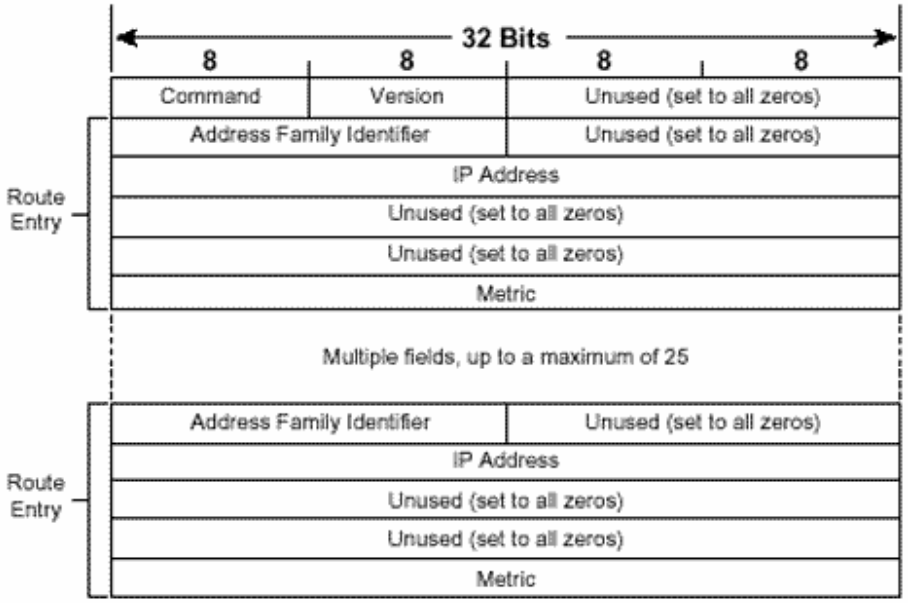
timers basic update invalid holddown flush

如果修改了某个路由器的 timer,那么其他参与 RIP 进程的所有的路由器也必须做出相应的 timer 调整

RIP 还使用 split horizon(水平分割),poison reverse(毒性反转)和 triggered update(触发更新). 所谓触发更新,是指路由的 metric 一旦被更改,这样的 update 就会被发送,而且这种 update 不会使接收到这种 update 的路由器重置自己的 timer

RIP Message Format

路由条目最多为 25 条,如果某个路由器要发送多余 25 条路由条目的 RIP 信息,那么就要产生并发送多个 RIP 信息. RIP 信息最大为 504 字节,加上 8 字节的 UDP 头部信息,最大为 512 字节 2



Command
1代表是请求信息
2代表是应答信息

AFI为Address Family Identifier,对于IP设置值为2

Metric 1~16

Version V1 或者V2

Request Message Types

RIP 请求信息可以向别的路由器请求完整的路由表信息或某些特定的路由条目信息. 在前者中,请求信息包含一个 AFI 设置为 0,地址为 0.0.0.0,并且 metric 为 16 的路由条目,收到这样的请求的设备以 unicast 的方式应答,并把整个路由表发给请求方(遵循水平分割和边界汇总原则)

如果要想知道某些特定路由条目的信息,那么可以把需要知道的上述特定的路由条目的地址附加在请求信息中发送即可,收到这样的请求信息的设备将一个个的处理这些路由条目并建立应答信息. 如果收到这个请求信息的设备本身就有这些请求中的路由条目,它就把条目的 metric 的值设置为它自己的 metric 的值;如果没有,metric 就设置为 16

Classful Routing

当一个 packet 进入运行 RIP 的路由器后,路由器将查询自己的路由表. 首先读取网络号(比如 A,B 或 C 类)部分,如果没有匹配条目,packet 被丢弃并发送一个 ICMP 目标不可达信息给源设备,依次类推,直到找到能够匹配的子网的条目,然后转发这个 packet;找不到的话丢弃这个 packet 并发送一个 ICMP 目标不可达信息给源设备

Classful Routing: Directly Connected Subnets

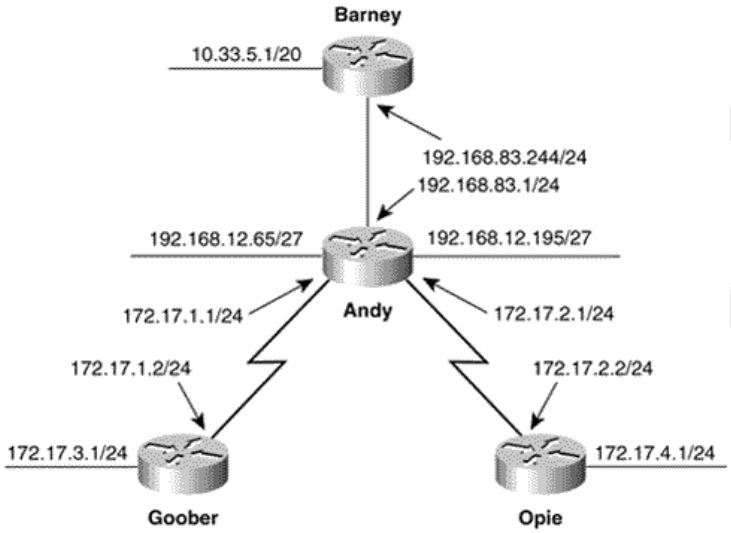
MtPilate#sh ip route

```
172.25.0.0 255.255.255.0 is subnetted, 3 subnets
R 172.25.153.0 [120/1] via 172.25.15.2, 00:00:03, Serial0
R 172.25.131.0 [120/1] via 172.25.15.2, 00:00:03, Serial0
C 172.25.15.0 is directly connected, Serial0
```

1. 假如有 1 个目标地址为 192.168.35.3 的 packet 进入这个路由器,但是在路由表中没有找到匹配的网络 192.168.35.0,所以这个 packet 将被丢弃
2. 如果有的目标地址为 172.25.33.89 的 packet 进入这个路由器,在路由表中找到匹配的主网络号 172.25.0.0/24,进一步找匹配的子网条目,却没有找到 172.21.33.0/24,所以仍然将被丢弃
3. 假如有一个 packet 的目标地址为 172.25.153.220 进入这个路由器,找到匹配的主网络号 172.25.0.0/24,子网条目 172.25.153.0 也能够匹配,所以 packet 将被转发给下一跳 172.25.15.2

注意:RIP 的信息格式里没有包含子网信息,所以在配置 RIP 的时候,应该只采用主网络号和默认的掩码

Configuring RIP




```
Goober(config)#router rip
Goober(config-router)#network 172.17.0.0

Opie(config)#router rip
Opie(config-router)#network 172.17.0.0

Barney(config)#router rip
Barney(config-router)#network 10.0.0.0
Barney(config-router)#network 192.168.83.0

Andy(config)#router rip
Andy(config-router)#network 172.17.0.0
Andy(config-router)#network 192.168.12.0
Andy(config-router)#network 192.168.83.0
```

在路由器 Andy 使用 **debug ip rip** 命令看看这些 update 的发送情况,如下:

```
Andy#debug ip rip
RIP protocol debugging is on
Andy#
RIP: sending update to 255.255.255.255 via Ethernet0 (192.168.12.65)
subnet 192.168.12.192, metric 1
network 10.0.0.0, metric 2
network 192.168.83.0, metric 1
network 172.17.0.0, metric 1
RIP: sending update to 255.255.255.255 via Ethernet1 (192.168.83.1)
network 192.168.12.0, metric 1
network 172.17.0.0, metric 1
RIP: sending update to 255.255.255.255 via Ethernet2 (192.168.12.195)
subnet 192.168.12.64, metric 1
network 10.0.0.0, metric 2
network 192.168.83.0, metric 1
network 172.17.0.0, metric 1
RIP: sending update to 255.255.255.255 via Serial0 (172.17.1.1)
subnet 172.17.4.0, metric 2
subnet 172.17.2.0, metric 1
```

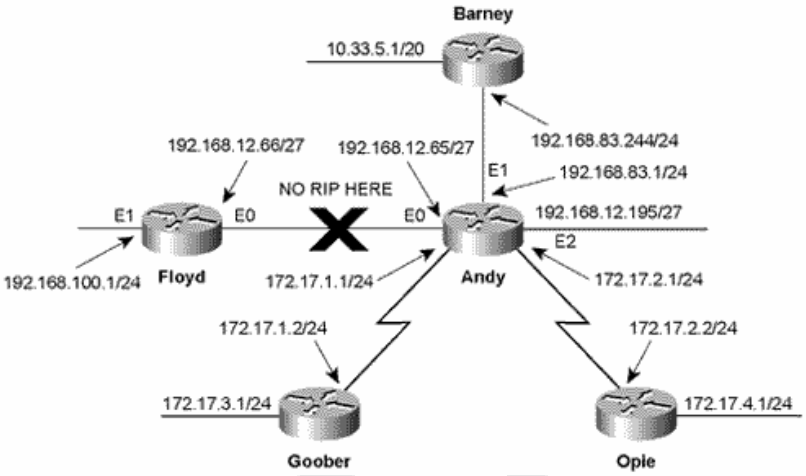
```
network 10.0.0.0, metric 2
network 192.168.83.0, metric 1
network 192.168.12.0, metric 1
RIP: sending update to 255.255.255.255 via Serial1 (172.17.2.1)
subnet 172.17.1.0, metric 1
subnet 172.17.3.0, metric 2
network 10.0.0.0, metric 2
network 192.168.83.0, metric 1
network 192.168.12.0, metric 1
RIP: received update from 172.17.2.1 on Serial0
172.17.3.0 in 1 hops
RIP: received update from 192.168.83.244 on Ethernet1
10.0.3.0 in 1 hops
RIP: received update from 172.17.2.2 on Serial1
172.17.4.0 in 1 hops
```

可以看到水平分割的规则的体现,从 E1 口宣告给路由器 Barney 的路由条目不包括 10.0.0.0 或 192.168.83.0

还有要注意的是,由于 E0 口和 E2 口都是和网络 192.168.12.0 相连,并且采用的都是/27 的掩码,而连接其他不同网络的 E1,S0 和 S1 口,子网要经过汇总以后才能被宣告出去.同样的,网络 192.168.83.0 和网络 172.17.0.0 也要经过汇总以后才会被宣告出去

Passive Interface

假如你新增一台路由器Floyd,你不希望它和路由器Andy之间交换RIP信息,就可以用到passive-interface命令,这个命令的作用在这里是避免 RIP update 从该接口被发送出去.所以可以在路由器 Andy 上做如下配置:




```
Andy(config)#router rip
Andy(config-router)#passive-interface Ethernet0
Andy(config-router)#network 172.17.0.0
Andy(config-router)#network 192.168.12.0
Andy(config-router)#network 192.168.83.0
```

对于路由器 Floyd 的配置,如下:

```
Floyd(config)#router rip
Floyd(config-router)#network 192.168.100.0
```

Configuring Unicast Updates

在使用 FR 等非广播型链路时需要使用 Rip 的单播更新，例如上图在 andy 和 Goober 建立单播更新

```
Andy(config)#router rip
Andy(config-router)#passive-interface s0
```

```
Andy(config-router)#neighbor 172.17.1.2
Andy(config-router)#network 172.17.0.0

Goober(config)#router rip
Goober (config-router)#passive-interface s0
Goober (config-router)#neighbor 172.17.1.1
```

Manipulating RIP Metrics

offset-list {access-list-number | name} {in| out} offset [type number]

通过如上命令可以手工修改 RIP 协议的 Metric 值，in|out 分别定义了入站和出站 2 个方向 offset 即偏移量

Troubleshooting RIP

如果一台高速路由器向低速路由器发送RIP信息量过大,这样有可能造成低速路由器处理不及,造成路由条目丢失,可以使用如下命令来定义一个发送update的时间延迟(默认为0ms,可以设置为8到50ms,output-delay {delay}

RIP v2

RIPv2 和 RIPv1(RIP)相比,增加的特性有:

1. 路由条目里包含了子网掩码的信息
2. 路由条目包含了下一跳的地址信息
3. 路由 update 包的验证
4. 外部路由标签(tag)
5. 通过 multicast 的方式发送 update 包 组播地址 224.0.0.9
6. RIPv2 是一种基于无类(classless)的路由协议
7. 支持认证功能

Case Study: A Basic RIPv2 Configuration

Cisco路由器默认只发送RIPv1信息,但是可以同时接收RIPv1和RIPv2的信息,如果你想只发送和接收RIPv2信息,使用version 2命令,如下:

```
RTA(config)#router rip
RTA(config-router)#version 2
RTA(config-router)#network 172.16.0.0
```

同理,如果你想只发送和RIPv1信息,如下:

```
RTA(config)#router rip
RTA(config-router)#version 1
RTA(config-router)#network 172.16.0.0
```

如果要恢复成原来的状态,在路由配置模式下使用no version即可

RIP v1在网络边界自动汇总，且不能关闭

RIP v2在网络边界自动汇总，但可以关闭，路由进程中用no auto-summary

可以在接口上定义 RIP 消息版本

```
interface Ethernet0
ip address 192.168.50.129 255.255.255.192
ip rip send version 1
ip rip receive version 1
!
interface Ethernet1
ip address 172.25.150.193 255.255.255.240
ip rip send version 1 2
```

Case Study: split-horizon

默认在 FR 环境中，RIP 的水平分割时自动关闭的，关闭水平分割，可以在接口模式下使用 no ip split-horizon

Case Study: Authentication

```
(config)#key chain kaka (指定一个钥匙链名字)
(config-keychain)#key 1 (定义一把钥匙)
```

(config-keychain-key)#key-string mike (定义该钥匙的口令)
(config)#int s0 (进入需要认证的接口)
(config-if)#ip rip authentication key-chain kaka (使用钥匙链)
(config-if)#ip rip authentication mode md5 (使用MD5的认证，该命令无则用明文)
钥匙链的名字只有本地意义

钥匙管理 -- 从一个钥匙到另一个钥匙的迁移（钥匙号从低到高检查）

```
key chain kaka
  key 1
    key-string mike
    accept-lifetime 16:30:00 Nov 28 2004 duration 43200 (持续43200秒)
    send-lifetime 16:30:00 Nov 28 2004 duration 43200
  key 2
    key-string mmike
    accept-lifetime 04:00:00 Nov 29 2004 13:00:00 Apr 15 2005 （到期时间）
    send-lifetime 04:00:00 Nov 29 2004 13:00:00 Apr 15 2005
  key 3
    key-string mmmike
    accept-lifetime 12:30:00 Apr 15 2005 infinite （永远）
    send-lifetime 12:30:00 Apr 15 2005 infinite
```

系统允许 30min 的时间重叠来在不同的系统始终之间校正

EIGRP

Operation of EIGRP

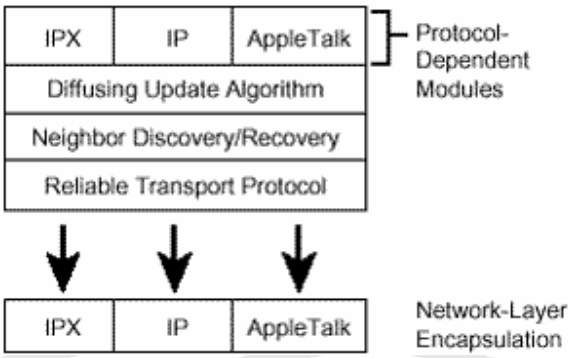
EIGRP 是一个距离向量路由协议,但是它还具有链路状态路由协议的一些特征.距离向量的路由协议一般都是基于 Bellman-Ford(或 Ford-Fulkerson)算法的.这样的算法容易引起路由循环(loop)和计数无穷大(counting to infinity),所以就必须采取些减少和避免上述问题的措施比如水平分割,holddown timer,毒性反转等.

由于路由器在给别的邻居(neighbor)传输它所接收到的路由之前,先要进行路由的计算,所以在大型网络里,收敛会比较慢.并且一旦链路变化,又会引起大量的路由被宣告和距离向量路由协议相比,链路状态路由协议受上述问题的影响就小的多。

1. 链路状态包(Link-State Packet,LSP)的转发是不依靠路由计算的,所以大型网络可以较为快速的进行收敛.
2. 它只宣告链路和链路状态,而不宣告路由,所以即使链路发生了变化,不会引起该链路的路由被宣告.但是链路状态路由协议使用的是 Dijkstra 算法,该算法比较复杂,并且较占 CPU 和内存资源和其他路由协议单独计算路由相比,链路状态路由协议采用种扩散计算(diffusingcomputations),通过多个路由器并行的记性路由计算,这样就可以在无环路产生的情况下快速的收敛

EIGRP 的路由 update(更新)的发送周期的不固定的,它只在网络链路发生变化以后才被发送,并且更新中可以只包含发生变化了的路由条目,而且只发给受到影响的路由器.这样就对链路带宽进行了节约。在 WAN 低速链路上,EIGRP 可能会占用大量带宽,默认只占用链路带宽 50%,之后发布的 IOS 允许使用命令 ip bandwidth-percent eigrp 来修改这一默认值

EIGRP 是一种基于无类的路由协议, EIGRP packet 只能通过 MD5 加密的方式进行验证
EIGRP 支持 IP,IPX 和 AppleTalk
EIGRP 使用和 IGRP 相同的公式来计算 metric,然而,这个 metric 要在 IGRP 算出来的 metric 之上乘以 1 个 256



EIGRP 的 4 个组件如上图:

1. Protocol-Dependent Module(PDM)
2. 可靠传输协议(Reliable Transport Protocol,RTP)
3. 邻居的发现/恢复
4. 扩散更新算法(Diffusing Update Algorithm,DUAL)

Reliable Transport Protocol

RTP 负责 EIGRP packet 的按顺序(可靠)的发送和接收,这个可靠的保障是通过 Cisco 私有的一个算法,reliable multicast 实现的,使用组播地址 224.0.0.10,每个邻居接收到这个可靠的组播包的时候就会以一个 unicast 作为确认

按顺序的发送是通过 packet 里的 2 个序列号实现的,每个 packet 都包含发送方分配的 1 个序列号,发送方没发送 1 个 packet,这个序列号就递增 1.另外,发送方也会把最近从目标路由器接收到的 packet 的序列号放在这个要发送的 packet 里

在某些情况下,RTP 也可以使用无需确认的不可靠的发送,并且使用这种不可靠发送的 packet 中不包含序列号

EIGRP 使用多种类型的 packet,这些 packet 通过 IP 头部信息里的协议号 88 来标识:

1. Hello packet:用来发现和恢复邻居,通过组播的方式发送,使用不可靠的发送
2. ACK(acknowledgement) packet:不包含数据(data)的 Hello 包,使用 unicast 的方式,不可靠的发送
3. Update packet:传播路由更新信息,不定期的,通过可靠的方式发送(比如网络链路发生变化).当只有一台路由器需要路由更新时,update 通过 unicast 的方式发送;当有多个路由器需要路由更新的时候,通过组播的方式发送
4. Query(查询) & Reply(应答) packet:是 DUAL finite state machine 用来管理扩散计算用的,查询包可以是组播或 unicast;应答包是通过 unicast 的方式发送,并且方式都是可靠的
5. Request(请求) packet:最初是打算提供给路由服务器(server)使用的,但是从来没实现过

如果 packet 通过可靠的组播方式发送出去,并且没有收到邻居反馈的 ACK 包,那么这个 packet 会再次以 unicast 的方式发送给那个未响应的邻居,虽然经过 16 次的重传 unicast,仍然没有收到 ACK 包的话,那么这个邻居就宣告为无效

在从组播切换到 unicast 之前,等待 ACK 包的时间可以由 multicast flow timer(MFT)指定,后续的重新传 unicast 的发送间隔可以由 RTO(retransmission timeout)指定.每个邻居的 MFT 和 RTO 都可以通过 SRTT(smooth round-trip time)来计算,SRTT 的单位是(毫秒)ms,用来衡量路由器从发送 EIGRP packet 到某个邻居并接收到这个邻居的 ACK 包所花费的平均时间

Neighbor Discovery/Recovery

EIGRP 的 Update 包是非周期性发送的,Hello 包在一般的网络中(比如点到点,point-to-point)是每 5 秒组播 1 次(要随机减去 1 个很小的时间防止同步);在多点(multipoint)X.25,帧中继(Frame Relay,FR)和 ATM 接口(比如 ATM SVC)和 ISDN PRI 接口上,Hello 包的发送间隔是 60 秒.在所有的情况中,Hello 包是不需要确认的.可以在接口配置模式下修改该接口的 Hello 包默认的发送间隔,命令为 **ip hello-interval eigrp**

当一个路由器收到从邻居发来的 Hello 包的时候,这个 Hello 包包含了一个 hold time,这个 hold time 告诉这个路由器等待后续 Hello 包的最大时间.如果在超出这个 hold time 之前没有收到后续 Hello 包,那么这个邻居就会被宣告为不可达,并通知 DUAL 这个邻居已丢失.默认 hold time 是 3 倍于 Hello 包发送间隔的,

更高链路 -- 默认Hello间隔和保持时间是5s和15s
T1或低于T1链路 -- 分别是60s和180s

可以在接口配置模式下修改这个默认的 hold time,命令为 **ip hold-time eigrp**.

EIGRP 邻居信息都记录在邻居表(neighbor table)中,使用 **show ip eigrp neighbors** 命令查看 IP EIGRP 的邻居,如下:

```
Wright#show ip eigrp neighbors
IP-EIGRP neighbors for process 1
H   Address     Interface   Hold Uptime    SRTT   RTO    Q      Seq
                            (sec)          (ms)       Cnt    Num
3   10.1.1.2    Et0           10 09:01:27    12   200    0      5
2   10.1.4.2    Se1           13 09:02:11    23   200    0      11
1   10.1.2.2    Et1           14 09:02:12     8   200    0      15
0   10.1.3.2    Se0           12 09:02:12    21   200    0      13
Wright#
```

Q Count

表示在因 RTO 超时而等待重传 unicast 的 packet 的数量.

从邻居那里接收到的最近的 update 包,查询包和应答包的序列号(Seq Num)也记录在邻居表中,RTP 跟踪这些序列号确保这些 packet 的收发是有秩序的

H

代表了这台路由器所学到的邻居的顺序号

The Diffusing Update Algorithm

DUAL: Preliminary Concepts

为了能够让 DUAL 正确的操作,低层协议必须满足以下几个条件:

1. 一个节点要在有限的时间里检测到新邻居的存在或和一个邻居的连接的丢失
2. 在链路上传输的所有信息必须在有限的时间里按正确的顺序收到
3. 所有的消息,包括链路 cost 的更改,链路故障,和新邻居的发现,都应该是在有限时间里,一个一个的依次处理

Cisco 的 EIGRP 使用邻居的发现/恢复和 RTP 来确保上述前提条件

adjacency(邻接):

在刚启动的时候,路由器使用 Hello 包来发现邻居并标识自己用于邻居的识别.当邻居被发现以后,EIGRP 会在它们之间形成一种邻接关系.邻接是指在这 2 个邻居之间形成一条交换路由信息的虚链路(virtual link).当邻接关系形成以后,它们之间就可以相互发送路由 update,这些 update 包括路由器它所知道的所有的路及其 metric.对于每个路由,路由器都会基于它邻居宣告的距离(distance)和到达那个邻居的链路的 cost 来计算出一个距离

Feasible Distance(FD,可行距离):

到达每个目标网络的最小的 metric 将作为那个目标网络的 FD.比如,路由器可能有 3 条到达网络 172.16.5.0 的路由,metric 分别为 380672,12381440 和 660868,那么 380672 就成了 FD

Feasible Condition(FC,可行条件):

邻居宣告到达目标网络的的距离小于本地路由器到达目标网络的 FD

Feasible Successor(FS,可行后继路由器):

如果一个邻居宣告到达目标网络的距离满足 FC,那么这个邻居就成为 FS.比如,路由器到达目标网络 172.16.5.0 的 FD 为 380672,而他邻居所宣告到达目标网络的距离为 355072,这个邻居路由器满足 FC,它就成为 FS;如果邻居路由器宣告到达目标网络的距离为 380928,即不满足 FC,那么这个邻居路由器就不能成为 FS

FS 和 FC 是避免环路的核心技术,FS 也是 downstream router(下游路由器),因为从 FS 到达目标网络的距离比本地路由器到达目标网络的 FD 要小

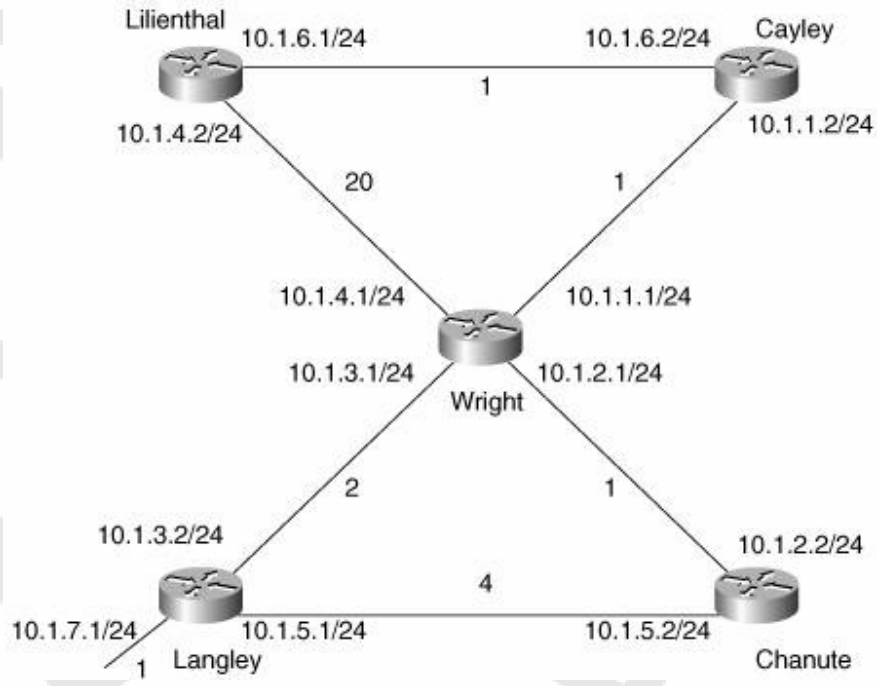
存在一个或多个 FS 的目标网络被记录在拓扑表(Topological Table)中,拓扑表包括以下内容:

1. 目标网络的 FD
2. 所有的 FD
3. 每一个 FS 所宣告的到达目标网络的距离
4. 本地路由器计算出的,经过每个 FS 到达目标网络的距离,即基于 FS 所宣告到达目标网络的距离和本地路由器到达那个 FS

   的链路的 cost
5. 发现 FS 的网络相连的接口

Successor(后继路由器):

对于列举在拓扑表里的每个目标网络,将选取 metric 最小的的路由放置在路由表里,宣告这条路由的邻居就成为 Successor,或者是下一跳路由器



metric weights 0 0 0 1 0 0 命令来定义只使用延迟(DLY)来参与 metric 的计算,DLY 已在图上做出标记.使用 show ip eigrp tology 查看路由器 Langley 的拓扑表,如下:

```
Langley#show ip eigrp topology
IP-EIGRP Topology Table for process 1

Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status

P 10.1.3.0/24, 1 successors, FD is 512
         via Connected, Serial0
P 10.1.2.0/24, 1 successors, FD is 768
         via 10.1.3.1 (768/256), Serial0
         via 10.1.5.2 (1280/256), Serial1
P 10.1.1.0/24, 1 successors, FD is 768
         via 10.1.3.1 (768/256), Serial0
         via 10.1.5.2 (1536/512), Serial1
P 10.1.7.0/24, 1 successors, FD is 256
```

```
            via Connected, Ethernet0
        P 10.1.6.0/24, 1 successors, FD is 1024
            via 10.1.3.1 (1024/512), Serial0
            via 10.1.5.2 (1792/768), Serial1
        P 10.1.5.0/24, 1 successors, FD is 1024
            via Connected, Serial1
        P 10.1.4.0/24, 1 successors, FD is 5632
            via 10.1.3.1 (5632/5120), Serial0
            via 10.1.5.2 (6400/5376), Serial1
```

路由器 Langley 到达目标网络 10.1.6.0 的 FS 分别为路由器 Wright(10.1.3.1)和路由器 Chanute(10.1.5.2)

(1024/512)中的 1024 为本地路由器 Langley 到达目标网络的 metric,即 256*(2+1+1)=1024;512 为邻居路由器(Wright)宣告的 metric,即 256*(1+1)=512.通过路由器 Chanute 到达目标网络的 metric 为 256*(4+1+1+1)=1792;路由器 Chanute 宣告的 metric 为 256*(1+1+1)=768.所以从路由器 Langley 到达目标网络 10.1.6.0 最小的 metric 为 1024,这个 metric 就成为 FD.并且路由器 Langley 的每条路由都只有 1 个 Successor,

查看下路由器 Langley 的路由表,如下:

```
Langley#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route
Gateway of last resort is not set
     10.0.0.0/8 is subnetted, 7 subnets
C       10.1.3.0 is directly connected, Serial0
D       10.1.2.0 [90/768] via 10.1.3.1, 00:32:06, Serial0
D       10.1.1.0 [90/768] via 10.1.3.1, 00:32:07, Serial0
C       10.1.7.0 is directly connected, Ethernet0
D       10.1.6.0 [90/1024] via 10.1.3.1, 00:32:07, Serial0
C       10.1.5.0 is directly connected, Serial1
D       10.1.4.0 [90/5632] via 10.1.3.1, 00:32:07, Serial0
```

D 代表 EIGRP,并且管理距离 AD 为 90

```
Cayley#show ip eigrp topology
IP-EIGRP Topology Table for process 1
Codes: P - Passive, A - Active, U - Update, Q - Query, R - Reply,
       r - Reply status
P 10.1.3.0/24, 1 successors, FD is 768
          via 10.1.1.1 (768/512), Ethernet0
P 10.1.2.0/24, 1 successors, FD is 512
          via 10.1.1.1 (512/256), Ethernet0
P 10.1.1.0/24, 1 successors, FD is 256
          via Connected, Ethernet0
P 10.1.7.0/24, 1 successors, FD is 1024
          via 10.1.1.1 (1024/768), Ethernet0
P 10.1.6.0/24, 1 successors, FD is 256
          via Connected, Serial0
P 10.1.5.0/24, 1 successors, FD is 1536
          via 10.1.1.1 (1536/1280), Ethernet0
P 10.1.4.0/24, 2 successors, FD is 5376
          via 10.1.6.1 (5376/5120), Serial0
          via 10.1.1.1 (5376/5120), Ethernet0
```

路由器 Cayley 到达网络 10.1.4.0 有 2 个 Successor,因为 2 条路径的 FD 都为 256*(20+1)=5376.因此路由器 Cayley 会在这 2 条路径上做等价的负载均衡.验证如下:

```
Cayley#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR
Gateway of last resort is not set
     10.0.0.0/24 is subnetted, 7 subnets
D       10.1.3.0 [90/768] via 10.1.1.1, 00:01:19, Ethernet0
D       10.1.2.0 [90/512] via 10.1.1.1, 00:01:19, Ethernet0
C       10.1.1.0 is directly connected, Ethernet0
D       10.1.7.0 [90/1024] via 10.1.1.1, 00:01:19, Ethernet0
C       10.1.6.0 is directly connected, Serial0
D       10.1.5.0 [90/1536] via 10.1.1.1, 00:01:19, Ethernet0
D       10.1.4.0 [90/5376] via 10.1.1.1, 00:01:19, Ethernet0
                 [90/5376] via 10.1.6.1, 00:01:19, Serial0
```

如果 FS 宣告的路由的 metric 比当前 Successor 的 metric 小的话,那么这个 FS 就将成为新的 Successor,如下几种情况可能会引起这个现象的发生:

1. 发现一条新的路由
2. 现有 Successor 的 metric 增加,超过了 FS 的 metric,或现有 FS 的 metric 减小到小于现有 Successor 的 metric

FS 减少了扩散计算的次数,提高了网络性能,同时也降低了收敛的次数.如果到达

Successor 的链路出故障,或者链路的 cost 增加,并超过了 FD,那么路由器会先在它的拓扑表中查找 FS,如果有 FS 的话,这个 FS 将成为新的 Successor;如果找不到可用的 FS,它才重新进行扩散计算

The DUAL Finite Machine

当 EIGRP 路由器不进行扩散计算的时候,所有的路由都处于被动(passive)状态.当发生输入事件(input event)的时候,路由器会对路由的 FS 列表进行重新评估,输入事件的源:

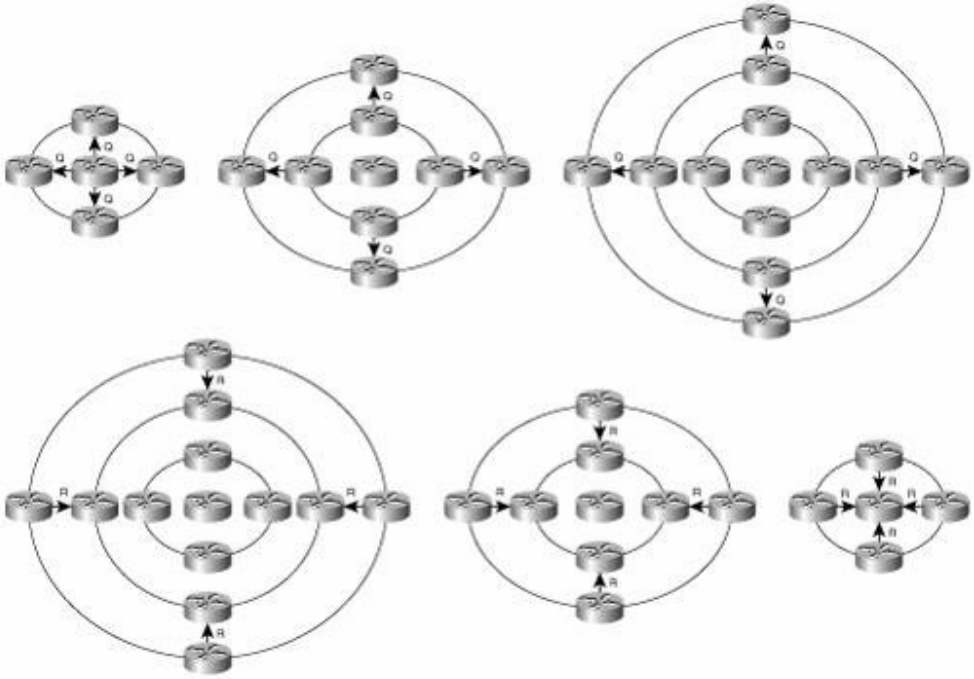
1. 直连链路的 cost 发生变化
2. 直连链路状态的变化(比如 up 或 down)
3. 收到更新包
4. 收到查询包
5. 收到应答包

路由器重新评估的第一步是在本地路由器上执行本地计算(local computation),可能的结果如下:

1. 如果 FS 的 metric 最小,并且和现有 Successor 不同的话,那这个 FS 就成为新的 Successor
2. 如果新的距离(distance)小于 FD,FD 将被更新
3. 如果新的距离和已经存在的距离不同,将向所有邻居发送更新包

当路由器执行本地计算的时候,路由仍然处于被动状态,如果本地路由器在它的拓扑表里发现可用的 FS,更新包将发送给它所有的邻居,但是路由状态不会改变;如果没有发现可用的 FS,路由器将进行扩散计算并且路由会进入活跃(active)状态.在路由器扩散计算完成和路由状态返回为被动状态之间,路由器:

1. 不能更改路由的 Successor
2. 不能更改正在宣告的路由的距离
3. 不能更改路由的 FD]
4. 不能开始进行路由的另一个扩散计算



查询包包含了本地路由器计算出来的到达目标地址的新的距离,当每个邻居收到这个查询包的时候,就开始进行自己的本地计算:

1. 如果邻居有到达目标地址的一个或多个 FS,它就反馈应答包给产生这个查询包的路由器.这个应答包包含了这个邻居所计算出的它到达目标地址的最小的距离
2. 如果邻居没有 FS,它就把路由更改为活跃状态并记性扩散计算

扩散计算在查询的时候,范围是扩大,在应答的时候,范围是缩小

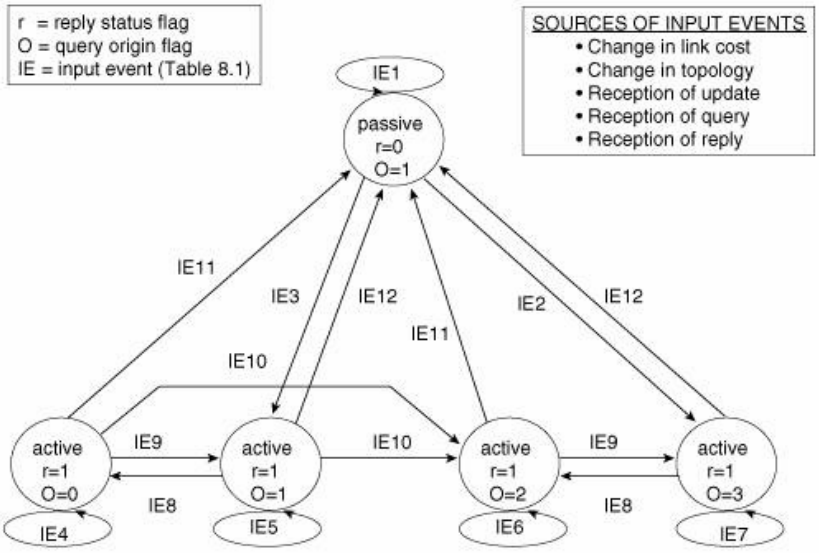
对于每一台接收到查询包的邻居路由器,本地路由器会设置一个答复状态标记(reply status flag,r)来跟踪这些没有未处理的查询包.当本地路由器接收到所有发出去的查询包的应答的时候,扩散计算就完成了

在某些情况中,路由器并不是能够收到所有发出去的查询包的应答.比如有可能发生在拥有许多低速或低质量链路的大型网络中

在扩散计算的开始,active timer 设置为 3 分钟(早期 IOS 版本设置为 1 分钟),如果在这个 active timer 超时之前仍然没有收到发出去的查询包的所反馈的所有的应答包的时候,那么这些没有收到应答包的路由就被标记为 stuck-in-active(SIA),这些没有反馈应答的邻居将从邻居表中被删除,而且扩散计算会认为是这个邻居反馈了一个为无穷大的 **metric.active timer** 的修改命令为 **timers active-time**

在扩散计算完成后,最初发送查询包的路由器会把 FD 设置为无穷大,这样确保了任何反馈应答包的路由器都满足 FC 并且能够成为 FS,对于这些应答包,metric 的计算是基于应答包中宣告的距离加上到达这个反馈应答包的邻居路由器的链路 cost.metric 最小的成为 Successor,并且 FD 就设置为这个最低的 metric.任何不满足 FC 的 FS 将从拓扑表中被删除.注意在收到所有的应答之前是不会选择 Successor 的

输入事件能够引起路由状态的变化,DUAL 定义了多种活跃状态.查询源标记(query origin flag,0)用来标记当前状态.如下是 DUAL finite machine 的输入事件(IE):



IE1:满足 FC 或目标不可达
IE2:从 Successor 收到了应答包;不满足 FC
IE3:除了来自 Successor 的查询的其他的输入事件;不满足 FC
IE4:除了最近的应答或来自 Successor 的查询的其他的输入事件
IE5:除了最近的应答或来自 Successor 的查询,或增加到达目标地址的距离的其他的输入事件
IE6:除了最近的应答的输入事件
IE7:除了最近的应答,或增加到达目标地址的距离的其他的输入事件
IE8:到达目标地址是距离增加
IE9:接收到了最近的应答包;当前 FD 不满足 FC
IE10:从 Successor 收到了查询包
IE11:收到了最近的应答包;当前 FD 满足 FC
IE12:收到了最近的应答包;当前 FD 设置为无穷大

EIGRP packet 的活动可以通过命令 debug eigrp packets 来跟踪,默认情况下会显示所有的 EIGRP packet,由于 Hello 包和 ACK 包的数量可能过大导致不便于跟踪,所以可以使用关键字来指定跟踪的 EIGRP packet 类型:debug eigrp packet query reply update,如下:

```
Carley#debug eigrp packet update query reply
EIGRP Packets debugging is on
(UPDATE, QUERY, REPLY)
Carley#
%LINEPROTO-5-UPDOWN: Line protocol on Interface Ethernet0, change state to down
EIGRP: Enqueueing QUERY on Serial0 iidbQ un/rely 0/1 serno 45-49
EIGRP: Enqueueing QUERY on Serial0 nbr 10.1.6.1 iidbQ un/rely 0/0 peerQ un/rely 0/0 serno 45-49
EIGRP: Sending Query on Serial0 nbr 10.1.6.1
AS 1, Flags 0x0, Seq 45/64 idbQ 0/0 iidbQ un/rely 0/0 peerQ un/rely 0/1 serno 45-49
```

Flags:
标记,0x0 表示标记位没有设置;0x1 表示设置了初始化(initialization)位;0x2 表示设置了条件接收(conditional receive)位.这个标记用在私有的 Reliable Multicasting 算法中

Seq:
序列号

idbQ:

表示在接口上的输入队列包和输出队列包的数目

iidbQ:

接口上等待传输的不可靠的组播包和等待传输的可靠的组播包的数目

peerQ:

接口上等待传输的不可靠的 unicast packet 和等待传输的可靠的 unicast packet 的数目

serno:

路由的双重连接的序列号的指针,用在内部(或私有),用于在一个快速变化的拓扑中跟踪正确的路由信息

扩散计算的核心思想如下:

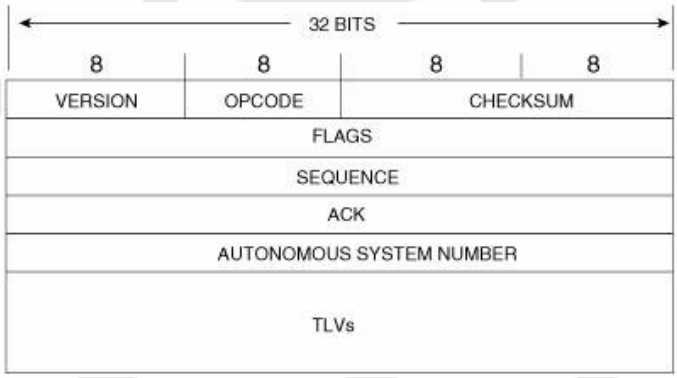
1. 任何时间发生了 IE 的话,将执行本地计算
2. 如果在拓扑表里发现了多个 FS,metric 最低的作为新的 Successor
3. 如果没有找到可用的 FS,路由进入活跃状态,并向邻居发出查询包,寻找新的 FS
4. 当所有发出去的查询包都被应答包响应,或者在 active timer 超出之前,路由会一直处于活跃状态
5. 如果扩散计算结果没有找到新的可用 FS 的话,那么将宣告目标不可达

EIGRP Packet Formats

EIGRP 包的 IP 头部指定了协议号为 88,EIGRP 包的最大长度为 1500 字节,(IP 的 MTU)

The EIGRP Packet Header

下图是 EIGRP 包的头部格式:



版本号(Version):定义产生 EIGRP 进程的版本,目前有 2 种软件版本可用,不过建议使用较后发布的版本

操作码(Opcode):定义 EIGRP 包的类型,如下:

1:Update
3:Query
4:Reply
5:Hello
6:IPX SAP

校验和(Checksum):标准的 IP 校验和,基于除了 IP 头部的整个 EIGRP 包计算的

标记(Flags):目前有 2 个标记,最右边的位是初始化(Init)位,即 0x00000001,表示路由条目是新的邻居关系的开始;

接收(Conditional Received)位,用于私有的可靠组播算法中,设置为 0x00000002

序列号(Sequence):用于 RTP 中,32 位长

确认(ACK):32 位长,从邻居接收到的.ACK 字段非 0 的 Hello 包被当作是 ACK 包而不是 Hello 包.另,如果 packet 本身 unicast,那 ACK 字段只能为非 0 因为 ACK 包从来都不是组播

自治系统号(AS Number):EIGRP 域的标识

EIGRP 头部之后的就是 TLV(Type/Length/Value)字段,如下是常见的 TLV 字段类型:

一般 TLV 类型:

0x0001:EIGRP 参数

0x0003:序列号

0x0004:软件版本,指老 IOS 版本(软件版本 0)还是较新的 IOS 版本(软件版本 1)IOS10.3(11),11.0(8)和 11.1(3)

0x0005:下一组播序列号

IP 特有 TLV 类型:

0x0102:IP 内部路由

0x0103:IP 外部路由

AppleTalk 特有 TLV 类型:

0x0202:AppleTalk 内部路由

0x0203:AppleTalk 外部路由

0x0204:AppleTalk 线缆(Cable)配置

IPX 特有 TLV 类型:

0x0302:IPX 内部路由

0x0303:IPX 外部路由

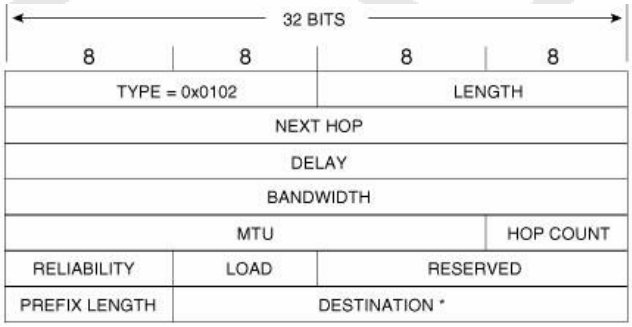
General TLV Fields

一般 TLV 运载了 EIGRP 管理信息并且无需指定可路由协议(routed protocol),如下图:

IP-Specific TLV Fileds

每个内部路由和外部路由的 TLV 都包含一个路由条目,每个 Update 包,查询包,

应答包至少包含一个路由 TLV,路由 TLV 包括了路由的 metric.内部路由 TLV 的格式如下图:



延迟(Delay):

单位是 10 微秒的延迟总和,IGRP 中这个字段为 24 位,EIGRP 中为 32 位.如果这里为 0xFFFFFFFF,则表示不可达

带宽(Bandwidth):

为 256*BW(IGRP),其中 BW(IGRP)取最小值,或者用 2560000000 除以链路带宽最小的那个所得的值

MTU:到目标地址 MTU 最小的,不参与 metric 的计算

跳数(Hop Count):

0x01 到 0xFF 之间的一个数,表示到达目标地址所经的跳数

可靠性(Reliability):

每 5 分钟计算 1 次的,是 0x01 到 0xFF 之间的一个数,表示沿途经过出口接口的错误率的和

负载(Load):

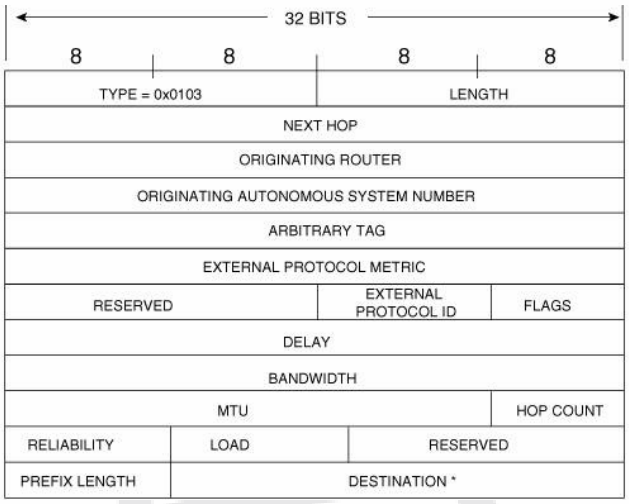
每 5 分钟计算 1 次的,是 0x01 到 0xFF 之间的一个数,表示沿途经过出口接口的负载的和

保留(Reserved)字段:设置为 0x0000

目标(Destination)地址:

长度可变,类似 IGRP 信息格式中的目标地址字段,不足的用 0 填充,直到达到 4 个 8 位长

外部路由 TLV 的格式如下图:



Arbitrary Tag:Route Map tag

外部协议 ID(External Protocol ID):标识外部路由协议,如下:

0x01:IGRP

0x02:EIGRP

0x03:静态路由

0x04:RIP

0x05:Hello

0x06:OSPF

0x07:IS-IS

0x08:EGP

0x09:BGP

0x0A:IDRP

0x0B:直连

Metric = [k1*Bw(min)+(k2*Bw(min))/(256-LOAD)+k3*DLY(sum)]*[k5/(Reliablity+k4)]

Configuring EIGRP

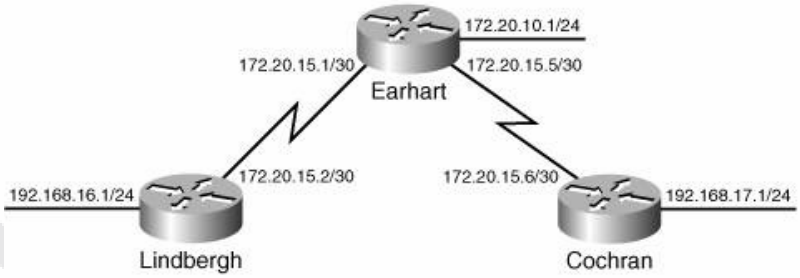
Case Study: A Basic EIGRP Configuration

EIGRP 配置的 2 个步骤:

1. 使用 router eigrp <process-id>启动 EIGRP

2. 使用 network 命令指定参与 EIGRP 进程的主网络号

process-id 范围是 1 到 65535;也可以使用 InterNIC 分配的 AS 号.例子如下图:



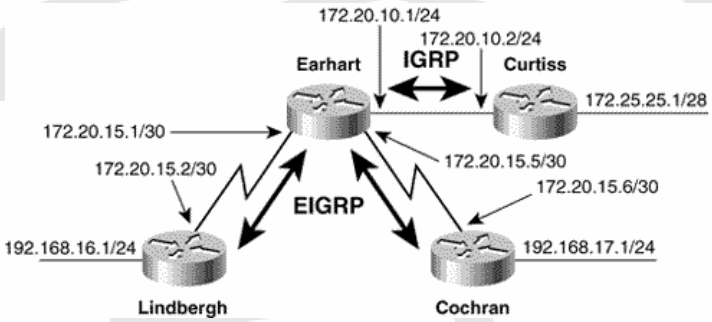
配置分别如下:

```
Earhart(config)#router eigrp 15
Earhart(config-router)#network 172.20.0.0
```

```
Lindbergh(config)#router eigrp 15
Lindbergh(config-router)#network 172.20.0.0
Lindbergh(config-router)#network 192.168.16.0

Cochran(config)#router eigrp 15
Cochran(config-router)#network 172.20.0.0
Cochran(config-router)#network 192.168.17.0
```

Case Study: Redistribution with IGRP

如果一个 IGRP 和一个 EIGRP 进程拥有相同 process-id,那么它们将自动进行路由的重分布



配置如下:

```
Earhart(config)#router igrp 15
Earhart(config-router)#network 172.20.0.0
Earhart(config-router)#passive-interface s0
Earhart(config-router)#passive-interface s1
Earhart(config)#router eigrp 15
Earhart(config-router)#network 172.20.0.0
Earhart(config-router)#passive-interface e0

Curtiss(config)#router igrp 15
Curtiss(config-router)#network 172.20.0.0
Curtiss(config-router)#network 172.25.0.0
```

配置完成以后,我们来查看下路由器 Curtiss 的路由表,如下:

```
Curtiss#sh ip route
I 192.168.16.0/24 [100/8676] via 172.20.10.1 00:00:06, Ethernet0
I 192.168.17.0/24 [100/8676] via 172.20.10.1 00:00:06, Ethernet0
  172.25.0.0/28 is subnetted, 1 subnets
C 172.25.25.0/24 is directly connected, Ethernet1
  172.20.0.0/24 is subnetted, 1 subnets
C 172.20.10.0/24 is directly connected, Ethernet0
Curtiss#
```

可以看出路由器 Curtiss 没有学到到达网络 172.20.15.0/30 和 172.20.15.4/30 的路由,因为 IGRP 是基于类的路由协议,它不接收经过变长子网划分的路由.

可以在路由器 Earhart 上使用命令 ip summary-address eigrp AS-number address mask 创建一条汇总路由

```
Earhart(config)#router igrp 15
Earhart(config-router)#network 172.20.0.0
Earhart(config-router)#passive-interface s0
Earhart(config-router)#passive-interface s1
Earhart(config)#router eigrp 15
```

```
Earhart(config-router)#network 172.20.0.0
Earhart(config-router)#passive-interface e0
Earhart(config)#int e0
Earhart(config-if)#ip addr 172.20.10.1 255.255.255.0
Earhart(config-if)#ip summary-address eigrp 15 172.20.15.0 255.255.255.0
```

再次查看路由器 Curtiss 的路由表:

```
Curtiss#sh ip route
I 192.168.16.0/24 [100/8676] via 172.20.10.1 00:00:06, Ethernet0
I 192.168.17.0/24 [100/8676] via 172.20.10.1 00:00:06, Ethernet0
172.25.0.0/28 is subnetted, 1 subnets
C 172.25.25.0/24 is directly connected, Ethernet1
172.20.0.0/24 is subnetted, 1 subnets
C 172.20.10.0/24 is directly connected, Ethernet0
I 172.20.15.0/24 [100/8576] via 172.20.10.1 00:00:18, Ethernet0
Curtiss#
```

Case Study: Disabling Automatic Summarization

EIGRP 一样会进行自动汇总,可以手动的关闭它.在路由进程中使用 no auto-summary

Case Study: Authentication

EIGRP packet 的认证是在 Cisco IOS Release 11.3 及其之后才出现的,只支持 MD5 加密方式.步骤如下:

1. 定义钥匙链(key chain)的名字
2. 定义钥匙链的钥匙(key)(1 个或 1 组)
3. 在接口配置模式启用认证,并定义要使用的钥匙链
4. 配置管理钥匙(可选)

现要在路由器 Cochran 到 Earhart 之间启用认证,如下图:

```
Cochran(config)#key chain kaka
Cochran(config-keychain)#key 1
Cochran(config-keychain)#key-string cisco
Cochran(config-keychain)#int s0
Cochran(config)#int s0
Cochran(config-if)#ip addr 172.20.15.6 255.255.255.252
Cochran(config-if)#ip authentication key-chain eigrp 15 kaka
Cochran(config-if)#ip authentication mode eigrp 15 md5
```

EIGRP 的认证的配置方法和 RIPv2 的认证配置方法类似

如上配置中，如果不使用 **ip authentication mode eigrp 15 md5**

无论 key-chain 是否启用，认证始终处于关闭状态。所以 EIGRP 仅支持 MD5 认证方式

Troubleshooting EIGRP

EIGRP over Frame-Relay

EIGRP 在帧中继环境中，默认水平分割是关闭的，但实质是打开的，这是一个 IOS 的 bug

需要在接口模式下使用 **no ip split-horizon eigrp 15** 来关闭

EIGRP-stub Area

EIGRP 可以定义 stub 区域，在路由进程模式下输入 **eigrp stub {receive-only|connected|static|summary}**

Receive-only:	禁止路由器同 EIGRP-AS 内任何路由器共享其路由，禁止发送任何类型路由
Connected:	允许 EIGRP-stub 区域发送直连路由,默认打开
Static:	允许 EIGRP-stub 区域发送静态路由
Summary:	允许 EIGRP-stub 区域发送汇总路由,默认打开

Stub 降低了中央路由器查询路由时的查询范围，从而减轻了 SIA 产生的几率

EIGRP loading balance

路由器配置 **maximum-path**，最多 6 条，默认 4 条。然后 **variance 4** 设置非等价均衡负载倍数

Open Shotest Path First

The Hello Protocol

Hello协议的目的:

1. 用于发现邻居
2. 在成为邻居之前,必须对Hello包里的一些参数协商成功
3. Hello包在邻居之间扮演着keepalive的角色
4. 允许邻居之间的双向通信
5. 它在NBMA(Nonbroadcast Multi-access)网络上选举DR和BDR

Cisco路由器上Hello包默认的发送间隔(HelloInterval)是10秒;NBMA网络是30秒,

通过ip ospf hello-interval <seconds>来修改;

如果在4倍于这个时间间隔里(40秒和120秒)内仍然没有收到来自邻居的新的Hello包,这个邻居将被宣告为无效(dead)

通过命令ips ospf dead-interval <seconds>来修改

Hello Packet包含以下信息:

1. 源路由器的RID
2. 源路由器的Area ID
3. 源路由器接口的掩码
4. 源路由器接口的认证类型和认证信息
5. 源路由器接口的Hello包发送的时间间隔
6. 源路由器接口的无效时间间隔
7. 优先级
8. DR/BDR
9. 五个标记位(flag bit)
10. 源路由器的所有邻居的RID

Network Type

OSPF定义的5种网络类型:

1. 点到点网络
2. 广播型网络
3. NBMA网络
4. 点到多点网络
5. 虚链接(virtual link)

点到点网络,

比如T1线路,是连接单独的一对路由器的网络,点到点网络上的有效邻居总是可以形成邻接关系的,在这种网络上,OSPF包的目标地址使用的是224.0.0.5,这个组播地址称为AllSPFRouters

广播型网络,

比如以太网,Token Ring和FDDI,这样的网络上会选举一个DR和BDR,DR/BDR的发送的OSPF包的目标地址为224.0.0.5,运载这些OSPF包的帧的目标MAC地址为0100.5E00.0005;而除了DR/BDR以外的OSPF包的目标地址为224.0.0.6,这个地址叫AllDRouters

NBMA网络,

比如X.25,Frame Relay,和ATM,不具备广播的能力,因此邻居要人工来指定,在这样的网络上要选举DR和BDR,OSPF包采用unicast的方式

点到多点网络

是NBMA网络的一个特殊配置,可以看成是点到点链路的集合.在这样的网络上不选举DR和BDR

虚链接:

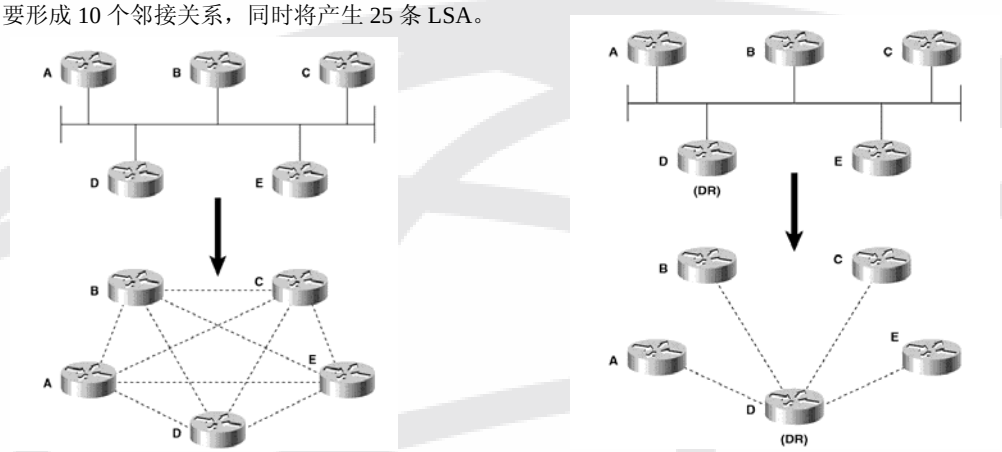
OSPF包是以unicast的方式发送

所有的网络也可以归纳成2种网络类型:

1. 传输网络(Transit Network)
2. 末梢网络(Stub Network)

DRs and BDRs

在 DR 和 BDR 出现之前，每一台路由器和他的邻居之间成为完全网状的 OSPF 邻接关系，这样 5 台路由器之间将需要形成 10 个邻接关系，同时将产生 25 条 LSA。



而且在多址网络中，还存在自己发出的 LSA 从邻居的邻居发回来，导致网络上产生很多 LSA 的拷贝，所以基于这种考虑，产生了 DR 和 BDR。

DR 将完成如下工作

1. 描述这个多址网络和该网络上剩下的其他相关路由器
2. 管理这个多址网络上的 flooding 过程

同时为了冗余性，还会选取一个 BDR，作为双备份之用

DR BDR 选取规则:

DR BDR 选取实在接口状态机的方式触发的。

1. 路由器的每个多路访问(multi-access)接口都有个路由器优先级(Router Priority),8 位长的一个整数,范围是 0 到 255,Cisco 路由器默认的优先级是 1 优先级为 0 的话将不能选举为 DR/BDR.优先级可以通过命令 ip ospf priority 进行修改
2. Hello 包里包含了优先级的字段,还包括了可能成为 DR/BDR 的相关接口的 IP 地址
3. 当接口在多路访问网络上初次启动的时候,它把 DR/BDR 地址设置为 0.0.0.0,同时设置等待计时器(wait timer)的值等于路由器无效间隔(Router Dead Interval)

DR--BDR 选举过程:

1. 在和邻居建立双向(2-Way)通信之后,检查邻居的Hello包中Priority,DR和BDR字段,列出所有可以参与DR/BDR选举的邻居.所有的路由器声明它们自己就是DR/BDR(Hello包中DR字段的值就是它们自己的接口地址;BDR字段的值就是它们自己的接口地址)
2. 从这个有参与选举DR/BDR权的列表中,创建一组没有声明自己就是DR的路由器的子集(声明自己是DR的路由器将不会被选举为BDR)
3. 如果在这个子集里,不管有没有宣称自己就是BDR,只要在Hello包中BDR字段就等于自己接口的地址,优先级最高的就被选举为BDR;如果优先级都一样,RID最高的选举为BDR
4. 如果在Hello包中DR字段就等于自己接口的地址,优先级最高的就被选举为DR;如果优先级都一样,RID最高的选举为DR;如果没有路由器宣称自己就是DR,那么新选举的BDR就成为DR
5. 要注意的是,当网络中已经选举了DR/BDR后,又出现了1台新的优先级更高的路由器,DR/BDR是不会重新选举的
6. DR/BDR选举完成后,DRother只和DR/BDR形成邻接关系.所有的路由器将组播Hello包到AllSPFRouters地址224.0.0.5以便它们能跟踪其他邻居的信息,即DR将洪泛update packet到224.0.0.5;DRother只组播update packet到AllDRouter地址224.0.0.6,只有DR/BDR监听这个地址

OSPF Interface

```
Renoir#show ip ospf interface Serial1.738
Serial1.738 is up, line protocol is up
   Internet Address 192.168.21.21/30, Area 7
   Process ID 1, Router ID 192.168.30.70, Network Type POINT_TO_POINT, Cost: 781
   Transmit Delay is 1 sec, State POINT_TO_POINT,
   Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
     Hello due in 00:00:07
   Neighbor Count is 1, Adjacent neighbor count is 1
     Adjacent with neighbor 192.168.30.77
   Message digest authentication enabled
     Youngest key id is 10
```

使用show ip ospf interface
命令查看接口
Process ID: 进程号，cisco 特有

RID: 路由器的 router-id

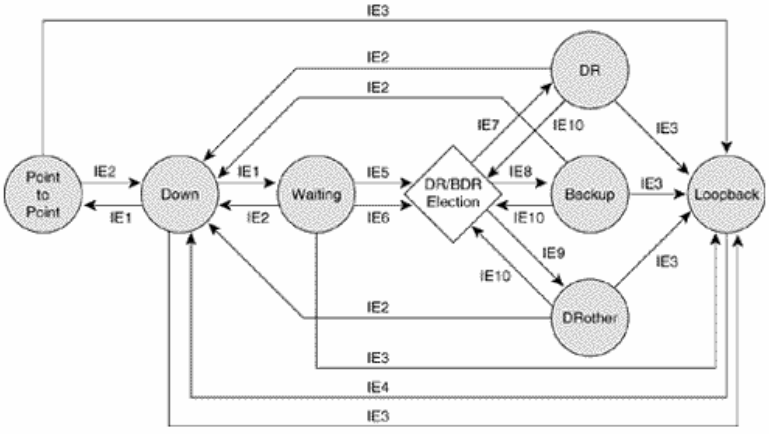
Network Type : P2P,P2MP,...
InfTransdelay:LSA 通告从路由器接口发送后经历的时间，以 Transmit Delay 显示，缺省值为 1sec
ip ospf transmit-delay 可以修改

Router Priority:缺省值为 1
可以通过 ip ospf priority 修改，0 为不参与选举
Cost:1 到 65535 之间的一个整数,Cisco 默认 cost 的算法是 100000000/BW,BW 为接口带宽.如果带宽大于 100M 的话,将产生 1 个小于 1 的小数,这是不允许的.因此从 IOS 版本 11.2 之后,可以使用命令 ospf auto-cost reference-bandwidth 来修正这个问题,允许管理者更改缺省的参考带宽

Wait Timer:选举 DR/BDR 之前,等待邻居路由器宣告 DR/BDR 的 Hello 包的时间长度,这个时间长度等于无效时间(RouterDeadInterval)

RxmtInterval:没有得到确认的情况下,重传 OSPF packet 所等待的时间长度,默认为 5 秒,可以使用命令 ip ospf retransmit-interval <seconds>修改
Hello Timer:由 HelloInterval 设置的,当它超时后,将从接口发送一个 Hello 包,上图显示 Hello Timer 将在 3 秒后超时

AuType:认证类型,OSPF 的认证类型可以是 null(无认证),简单口令或者 MD 加密.如果使用 null 认证方式,show ip ospf interface 将不会显示认证类型和钥匙(key)信息



OSPF state Machine

IE1:低层协议指明接口可操作
IE2:低层协议指明接口不可操作
IE3:网络管理系统或低层协议指明接口 loop up
IE4:网络管理系统或低层协议指明接口 loop down
IE5:收到 Hello 包

IE6:Wait Timer 超时

IE7:选举为 DR

IE8:被选举为 BDR

IE9:没有被选举为 DR/BDR,成为 DRothers

IE10:邻居路由器发生了变化

OSPF Neighbors

邻接关系建立的 4 个阶段:

1.邻居发现阶段

2.双向通信(Bidirectional Communication)阶段：Hello 报文都列出了对方的 RID，则 BC 完成

3.数据库同步阶段：

4.完全邻接阶段: full adjacency

邻居关系的建立和维持都是靠Hello包完成的,在一般的网络类型中,Hello包是每经过1个HelloInterval发送一次,有 1 个例外:在 NBMA 网络中,路由器每经过一个 PollInterval 周期发送 Hello 包给状态为 down 的邻居(其他类型的网络是不会把 Hello 包发送给状态为 down 的路由器的).Cisco 路由器上 PollInterval 默认 60s

Hello Packet 以组播的方式发送给 224.0.0.5，在 NBMA 类型，点到多点和虚链路类型网络，以单播发送给邻居路由器。邻居可以通过手工配置或者 Inverse-ARP 发现

使用 show ip ospf neighbor 命令查看邻居的信息,如下:

```
Seurat#show ip ospf neighbor 192.168.30.105
Neighbor 192.168.30.105, interface address 192.168.16.41
   In the area 0 via interface Serial0
   Neighbor priority is 1, State is FULL
   Poll interval 60
   Options 2
   Dead timer due in 00:01:40
```

Poll interval:

这个参数只用于 NBMA 网络,因为在 NBMA 网络中邻居无法自动发现.如果邻居状态是 down,那么路由器将经 PollInterval 长的时间发送 Hello 包给这个状态为 down 的邻居

Dead timer due in 00:01:40:Dead Timer 将在 100 秒后超时,超时之前没收到新的 Hello 包的话将被宣告无效还有些未显示的信息:DR/BDR

Master/Slave: 在 ExStart 状态,邻居之间协商的主/从(Master/Slave)关系将控制数据库的同步

DD(Database Description) Sequence Number: 正向邻居发送的数据库描述序列号

Last Received Database Description Packet: 这个 packet 记录了 Initialize,More 和 Master 位和可选项,以及最后收到的数据库描述包的序列号.这个信息可以确定下一个数据库描述包是否重复

Link State Retransmission List: 没有得到确认的 LSA 列表

Database Summary List: 数据库同步期间,数据库描述包中向邻居发送的 LSA 列表,当路由器进入 exchange 状态后,这些 LSA 将组成 LSDB

Link State Require List:这个列表记录了来自邻居的数据库描述包的 LSA,这些 LSA 比 LSDB 中的 LSA 更新.链路状态请求(Link State Require,LSR)发送给邻居来请求 LSA 的拷贝,当这些请求的 LSA 通过链路状态更新(Link State Update,LSU)收到后,请求列表将会减少直至清空

OSPF 路由器在完全邻接之前,所经过的几个状态:

1.Down: 初始化状态

2.Attempt: 只适于 NBMA 网络,在 NBMA 网络中邻居是手动指定的,在该状态下,路由器将使用 HelloInterval 取代 PollInterval 来发送 Hello 包

3.Init: 表明在 RouterDeadInterval 里收到了 Hello 包,但是 2-Way 通信仍然没有建立起来

4.two-way: 双向会话建立

5.ExStart: 信息交换初始状态,在这个状态下,本地路由器和邻居将建立 Master/Slave 关系,并确定 DD Sequence Number,接口等级高的的成为 Master

6. Exchange: 信息交换状态,本地路由器向邻居发送数据库描述包,并且会发送 LSR 用于请求新的 LSA

7. Loading: 信息加载状态,本地路由器向邻居发送 LSR 用于请求新的 LSA

8. Full: 完全邻接状态,这种邻接出现在 Router LSA 和 Network LSA 中

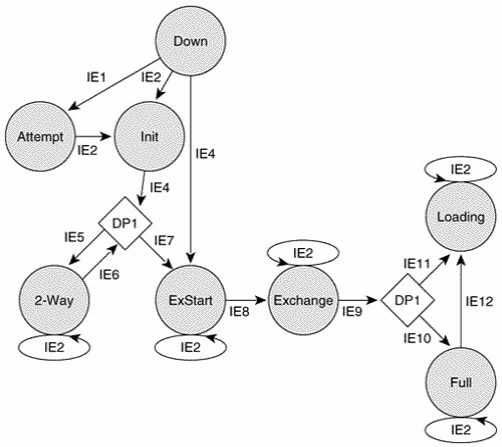


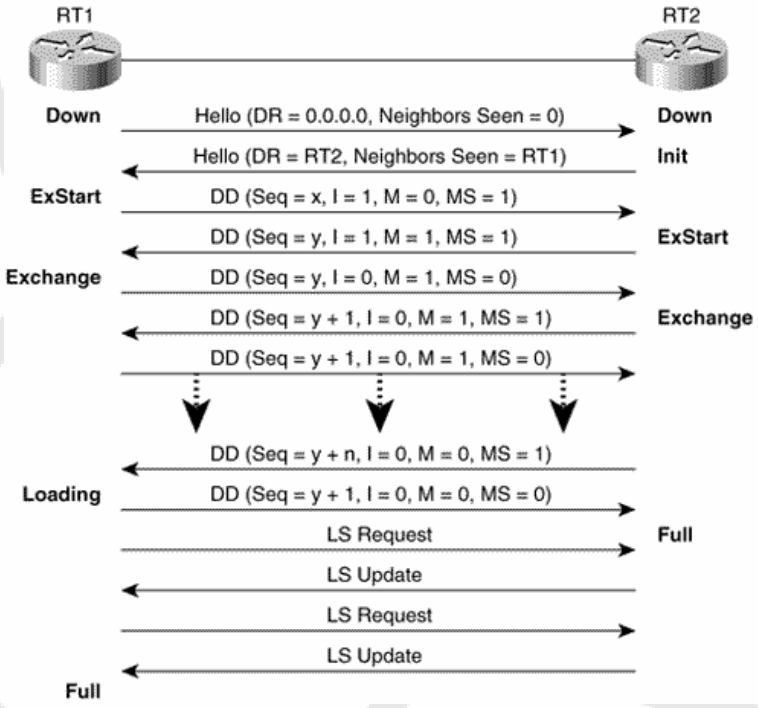
Figure 9.11. The neighbor state machine, from Down to Init.

在一个邻接关系的创建过程中，OSPF 协议使用如下 3 种报文

Database Description packets—DD (type 2)

Link State Request packets—LSR (type 3)

Link State Update packets—LSU (type 4)



1. RT1 首先变为有效状态,发送 Hello 包给 RT2,由于刚开始没有学习到任何邻居,虽然 Hello 包里 Neighbor 字段的值是空的,DR/BDR 字段设置为 0.0.0.0

2. RT2 收到来自 RT1 的 Hello 包以后,将 RT1 状态设置为 Init,并发送 Hello 包给 RT1,Neighbor 字段设置为 RT1 的 RID,并把 DR 字段设置成自己接口地址

3. RT1 收到 RT2 的 Hello 包并看到里面的 RID 以后,RT1 为了能够进行 Master/Slave 协商,将把 RT2 设置成 ExStart 状态.接着,RT1 将产生一个空的数据库描述包(DDP),并设置序列号为 x,同时设置 Init 位为 1,表示该

DDP 是用于本次信息交换的最初的 DDP, 并设置 More 位为 0 表示这个 DDP 不是最后一个 DDP; 设置 Masrer/Slave 位为 1 声明自己为 Master

4. RT2 收到 RT1 发来的这个 DDP 以后, 设置 RT1 状态为 ExStart, 并响应一个 DDP. 序列号设置为 y, 假设 RT2 的 RID 高于 RT1 的 RID, 因此它设置 MS 位为 1

5. Master/Slave 协商完成, RT2 为 Master, RT1 把 RT2 的状态设置为 Exchange 状态, 并发送 DDP 给 RT2, 序列号使用 y, MS 位设置为 0 声明自己是 Slave, I 位设置为 0, M 位设置为 1

6. RT2 收到 RT1 的这个 DDP 以后, 把 RT1 的状态设置为 Exchange, 并发送 DDP 给 RT1, 把序列号增加 1, 这个 DDP 包含了 RT2 的 Link State Summary List 中的 LSA 头部信息

7. RT1 收到这个 DDP 以后, 将响应一个序列号相同的 DDP, 这个过程将一直持续, 然后 RT2 将发送一个单独的 DDP 直到 RT1 以相同的序列号响应, 直到 RT2 发出包含最后 1 个 LSA Summary 的 DDP, 并设置 M 位为 0

8. Exchange 状态完成, 但是 RT1 的 LSR 列表中仍然还有未请求的 LSA 条目, 因此 RT1 进入 Loading 状态

9. 当 RT2 收到最后 1 个 DDP 以后, RT2 设置 RT1 的状态为 Full, 因为 RT1 的 LSR 列表中没有要请求的 LSA 条目了

10. RT1 发送 LSR 给 RT2, RT2 以 LSU 来响应这个请求, 这个过程持续到 RT1 的 LSR 列表成为空列表, 此时 RT1 也把 RT2 的状态设置为 Full

可以使用命令 debug ip ospf adj 查看邻接关系的建立情况

```
Degas#debug ip ospf adj
OSPF adjacency events debugging is on
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0x20E0 opt 0x2 flag 0x7 len 32
  state INIT
OSPF: 2 Way Communication to 192.168.30.70 on Ethernet0,
state 2WAY
OSPF: Neighbor change Event on interface Ethernet0
OSPF: DR/BDR election on Ethernet0
OSPF: Elect BDR 192.168.30.70
OSPF: Elect DR 192.168.30.175
    DR: 192.168.30.175 (Id) BDR: 192.168.30.70 (Id)
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x7 len 32
OSPF: First DBD and we are not SLAVE
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB17 opt 0x2 flag 0x2 len 92
state EXSTART
OSPF: NBR Negotiation Done. We are the MASTER
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x3 len 72
OSPF: Database request to 192.168.30.70
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB18 opt 0x2 flag 0x0 len 32
state EXCHANGE
OSPF: Send DBD to 192.168.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x1 len 32
OSPF: Rcv DBD from 192.168.30.70 on Ethernet0 seq 0xB19 opt 0x2 flag 0x0 len 32
state EXCHANGE
OSPF: Exchange Done with 192.168.30.70 on Ethernet0
OSPF: Synchronized with 192.168.30.70 on Ethernet0,
state FULL
```

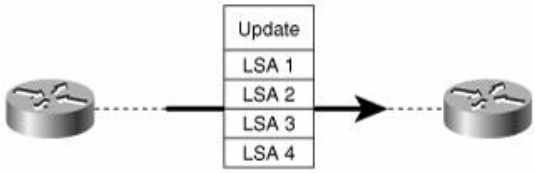
Flooding

Flooding 采用 2 种报文

LSU Type 4---链路状态更新报文

LSA Type 5---链路状态确认报文

如右图所示： 每一个链路状态更新报文和确认报文都可以携带多个 LSA。LSA 本身可以 flooding 到整个互联网络，但更新报文和确认报文只能在具有邻接关系的两个节点之间传送。



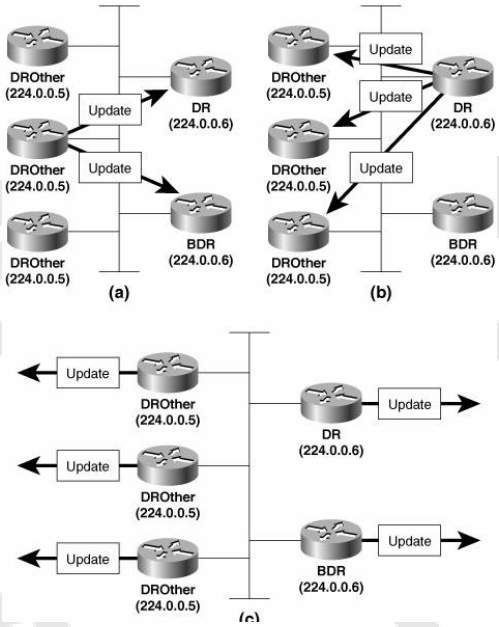
在 P-P 网络，路由器是以组播方式将更新报文发送到组播地址 224.0.0.5

在 P-MP 和虚链路网络，路由器以单播方式将更新报文发送至邻接邻居的接口地址

在广播型网络，DRother 路由器只能和 DR&BDR 形成邻接关系，所以更新报文将发送到 224.0.0.6，相应的 DR 以 224.0.0.5 泛洪 LSA

并且 BDR 只接收 LSA，不会确认和泛洪这些更新，除非 DR 失效

在 NBMA 型网络，LSA 以单播方式发送到 DR BDR，并且 DR 以单播方式发送这些更新



LSA的洪泛是可靠的,所以必须对每1个发送的LSA进行确认,确认分隐式确认(Implicit Acknowledge)和显式确认(Explicit Acknowledge)

当路由器要发送LSA的时候,会把LSA的拷贝放在链路状态重传列表中,这个LSA每隔RxmtInterval重传1次,直到该LSA得到确认,或邻接关系中断.无论哪种网络类型,重传的LSA总是以unicast的方式发送的

确认可以是delayed或direct,前者可以使用1个LSAck确认多个LSA,当然这个延迟的时间必须小于RxmtInterval;后者的确认是立即发送,采用单播的方式.当出现下面2种情况的时候将采用直接确认:

1.从邻居那里收到了重复的LSA

2.LSA的老化时间(Age)达到最大生存时间(MaxAge)

查看 LSDB 信息,使用 show ip ospf database 命令,如下:

LSA 通过序列号,校验和,和老化时间保证 LSDB 中的 LSA 是最新的

Seq: 序列号(Seq)的范围是 0x80000001 到 0x7fffffff

Checksum: 校验和(Checksum)计算除了 Age 字段以外的所有字段,每 5 分钟校验 1 次

Age: 范围是 0 到 3600 秒,16 位长.当路由器发出 1 个 LSA 后,就把 Age 设置为 0,当这个 LSA 经过 1 台路由器以后,Age 就会增加 1 个由 InfTransDelay 设定的时间(默认为 1 秒,这个时间可以通过命令 ip ospf transmit-delay <seconds>修改).LSA 保存在 LSDB 中的时候,老化时间也会增加

当收到相同的 LSA 的多个实例的时候,将通过下面的方法来确定哪个 LSA 是最新的:

1.比较 LSA 实例的序列号,越大的越新

2.如果序列号相同,就比较校验和,越大越新

3.如果校验和也相同,就比较老化时间,如果只有 1 个 LSA 拥有 MaxAge(3600 秒)的老化时间,它就是最新的

4.如果 LSA 老化时间相差 15 分钟以上,(叫做 MaxAgeDiff),老化时间越小的越新

5.如果上述都无法区分,则认为这 2 个 LSA 是相同的

Areas

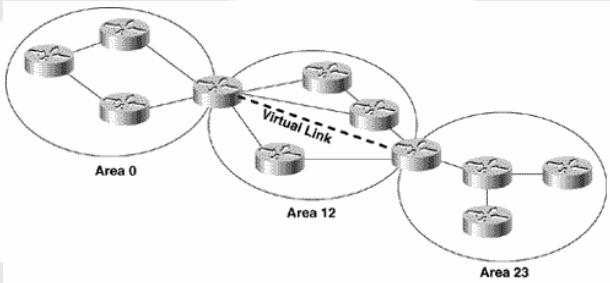
区域长度 32 位，可以用 10 进制，也可以类似于 IP 地址的点分十进制分 3 种通信量

1. Intra-Area Traffic:域内间通信量
2. Inter-Area Traffic:域间通信量
3. External Traffic:外部通信量

路由器类型

1. Internal Router:内部路由器
2. ABR(Area Border Router):区域边界路由器
3. Backbone Router(BR):骨干路由器
4. ASBR(Autonomous System Boundary Router):自治系统边界路由器

Virtual Link



1. 通过一个非骨干区域连接到一个骨干区域
2. 通过一个非骨干区域连接一个分段的骨干区域两边的部分区域

虚链接是一个逻辑的隧道（Tunnel）,配置虚链接的一些规则:

1. 虚链接必须配置在 2 个 ABR 之间
2. 虚链接所经过的区域叫 Transit Area,它必须拥有完整的路由信息
3. Transit Area 不能是 Stub Area
4. 尽口的避免使用虚链接,它增加了网络的复杂程度和加大了排错的难度

Link State Database

使用 show ip ospf database 可以查看 LSDB 的信息。显然，同时属于多个区域的 LSA 的路由器为 ABR

LSA 驻留在 LSDB 中，age 会加大，当超过 MaxAge 后，这些 LSA 将被清除掉，因此出现链路刷新(LSR)机制来防止这个问题,每 30min(LSRefreshTime)，发出这条 LSA 的源路由器将再次发送这个 LSA 的拷贝，并把序列号加 1，把 Age 设置为 0，来替换邻居 LSDB 中的较老的 LSA

```
Homer#show ip ospf database database-summary

            OSPF Router with ID (192.168.30.50) (Process ID 1)

Area ID         Router    Network   Sum-Net   Sum-ASBR  Subtotal  Delete    Maxage
0               8         4         185       27        224       0         0
4               7         0         216       26        249       0         0
5               7         0         107       13        127       0         0
56              2         1         236       26        265       0         0
AS External                                   580       0         0
Total           24        5         744       92        1445
Homer#
```

LSA Types

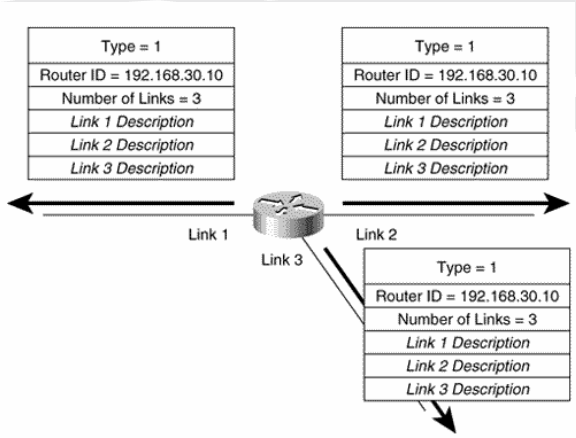
1. 类型 1:Router LSA
2. 类型 2:Network LSA
3. 类型 3:Network Summary LSA
4. 类型 4:ASBR Summary LSA
5. 类型 5:AS External LSA
6. 类型 6:Group Membership LSA

7. 类型 7:NSSA External LSA
8. 类型 8:External Attributes LSA
9. 类型 9:Opaque LSA(link-local scope,)
10. 类型 10:Opaque LSA(area-local scope)
11. 类型 11:Opaque LSA(AS scope)

使用 show ip ospf database database-summary 可以看到 LSDB 中的 LSA 类型

Router LSA:

每个路由器都将产生 Router LSA,这种 LSA 只在本区域内传播，描述了路由器所有的链路和接口，状态和开销

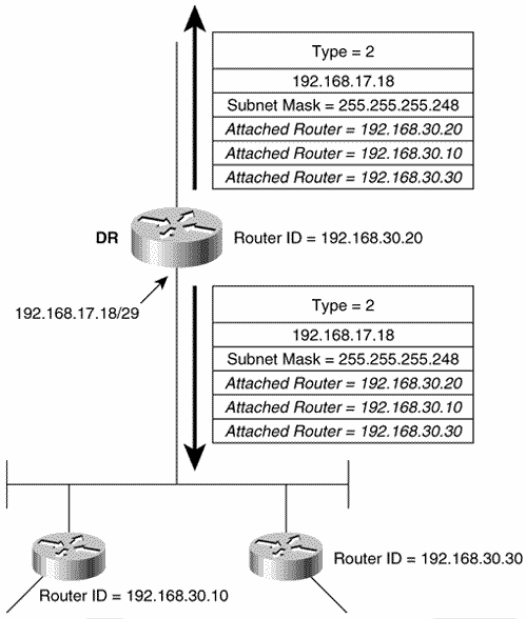


此种 LSA 可以通过 show ip ospf database router 查看

```
Homer#show ip ospf database router 192.168.30.10

       OSPF Router with ID (192.168.30.50) (Process ID 1)

                Router Link States (Area 0)

Routing Bit Set on this LSA
LS age: 680
Options: (No TOS-capability)
LS Type: Router Links
Link State ID: 192.168.30.10
Advertising Router: 192.168.30.10
LS Seq Number: 80001428
Checksum: 0x842A
Length: 60
Area Border Router
 Number of Links: 3

  Link connected to: another Router (point-to-point)
   (Link ID) Neighboring Router ID: 192.168.30.80
   (Link Data) Router Interface address: 192.168.17.9
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: a Stub Network
   (Link ID) Network/subnet number: 192.168.17.8
   (Link Data) Network Mask: 255.255.255.248
    Number of TOS metrics: 0
     TOS 0 Metrics: 64

  Link connected to: a Transit Network
   (Link ID) Designated Router address: 192.168.17.18
   (Link Data) Router Interface address: 192.168.17.17
    Number of TOS metrics: 0
     TOS 0 Metrics: 10

Homer#
```

Network LSA：

在每个多路访问网络中，DR 都会产生这种 Network LSA，它只在产生这条 Network LSA 的区域泛洪描述了所有和它相连的路由器（包括 DR 本身）

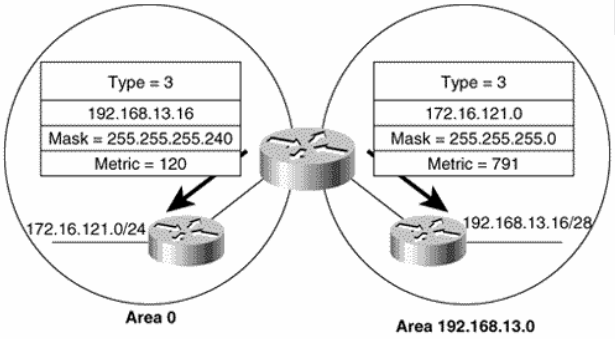


Show ip ospf database network 可以看到 Network LSA

```
Homer#show ip ospf database network 192.168.17.18

        OSPF Router with ID (192.168.30.50) (Process ID 1)

                Net Link States (Area 0)

Routing Bit Set on this LSA
LS age: 244
Options: (No TOS-capability)
LS Type: Network Links
Link State ID: 192.168.17.18 (address of Designated Router)
Advertising Router: 192.168.30.20
LS Seq Number: 800001BF
Checksum: 0x60AC
Length: 32
Network Mask: /29
      Attached Router: 192.168.30.20
      Attached Router: 192.168.30.10
      Attached Router: 192.168.30.30

Homer#
```

Network Summary LSA

由 ABR 路由器始发，用于通告该区域外部的目的地址，



可以使用 show ip ospf database summary 查看 LSA

```
Homer#show ip ospf database summary 172.16.121.0

   OSPF Router with ID (192.168.30.50) (Process ID 1)

      Summary Net Link States (Area 0)

Routing Bit Set on this LSA
LS age: 214
Options: (No TOS-capability)
LS Type: Summary Links(Network)
Link State ID: 172.16.121.0 (summary Network Number)
Advertising Router: 192.168.30.60
LS Seq Number: 800000B1
Checksum: 0xE864
Length: 28
Network Mask: /24
   TOS: 0 Metric: 791
```

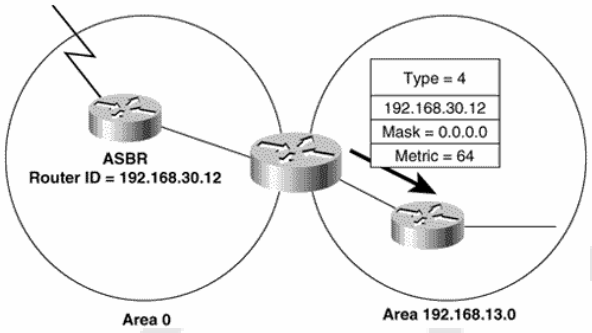
如果 ABR 知道有多条路径可以到达目标地址，但是它仍然只发送单个的 Network Summary LSA，并且是开销最低的那条；同样，如果 ABR 从其他的 ABR 那里收到多条 Network Summary LSA 的话，它会只选择开销最低的，并把这条 Network Summary LSA 宣告给其他区域

当其他的路由器收到来自 ABR 的 Network Summary LSA 以后，它不会运行 SPF 算法，它只简单的加上到达那个 ABR 的开销和 Network Summary LSA 中包含的开销，通过 ABR，到达目标地址的路由和开销一起被加进路由表里，这种依赖中间路由器来确定到达目标地址的完全路由(full route)实际上是距离矢量路由协议的行为

ASBR Summary LSA

由 ABR 发出，ASBR 汇总 LSA 除了所通告的目的地是一个 ASBR 而不是一个网络外，其他同 NetworkSummary LSA

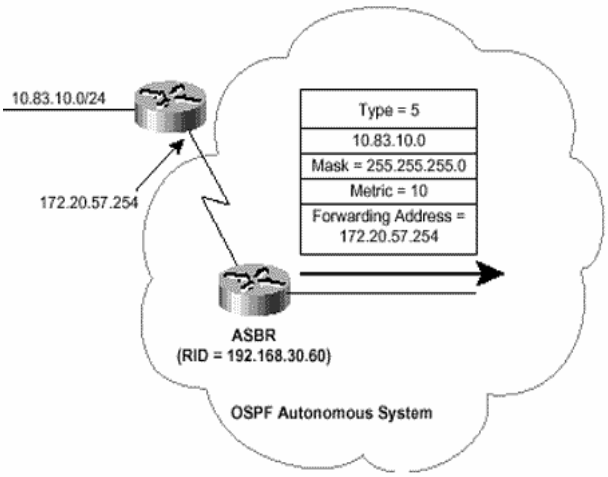
使用 show ip ospf database asbr-summary 可以看到




```
Homer#show ip ospf database asbr-summary

   OSPF Router with ID (192.168.30.50) (Process ID 1)

      Summary ASB Link States (Area 0)

Routing Bit Set on this LSA
LS age: 1640
Options: (No TOS-capability)
LS Type: Summary Links (AS Boundary Router)
Link State ID: 192.168.30.12 (AS Boundary Router address)
Advertising Router: 192.168.30.20
LS Seq Number: 80000009
Checksum: 0xF450
Length: 28
Network Mask: /0
   TOS: 0 Metric: 64

--More-
```

Autonmous System External LSA

发自 ASBR 路由器，用来通告到达 OSPF 自主系统外部的目的地或者 OSPF 自主系统那个外部的缺省路由的 LSA。这种 LSA 将在全 AS 内泛洪



可以使用 show ip ospf database external

```
Homer#show ip ospf database external 10.83.10.0

       OSPF Router with ID (192.168.30.50) (Process ID 1)

                   AS External Link States

Routing Bit Set on this LSA
LS age: 1680
Options: (No TOS-capability)
LS Type: AS External Link
Link State ID: 10.83.10.0 (External Network Number )
Advertising Router: 192.168.30.60
LS Seq Number: 80000D5A
Checksum: 0x7A1C
Length: 36
Network Mask: /24
      Metric Type: 1 (Comparable directly to link state metric)
      TOS: 0
      Metric: 10
      Forward Address: 172.20.57.254
      External Route Tag: 0

Homer#
```

NSSA External LSA

来自非完全 Stub 区域（not-so-stubby area）内 ASBR 路由器始发的 LSA 通告它只在 NSSA 区域内泛洪，这是与 LSA-Type5 的区别

Show ip ospf database nssa-external

```
Morisot#show ip ospf database nssa-external

         OSPF Router with ID (10.3.0.1) (Process ID 1)

                   Type-7 AS External Link States (Area 15)

LS age: 532
Options: (No TOS-capability, No Type 7/5 translation, DC)
LS Type: AS External Link
Link State ID: 10.0.0.0 (External Network Number)
Advertising Router: 10.3.0.1
LS Seq Number: 80000001
Checksum: 0x9493
Length: 36
Network Mask: /16
      Metric Type: 2 (Larger than any link state path)
      TOS: 0
      Metric: 100
      Forward Address: 10.3.0.1
      External Route Tag: 0

--More-
```

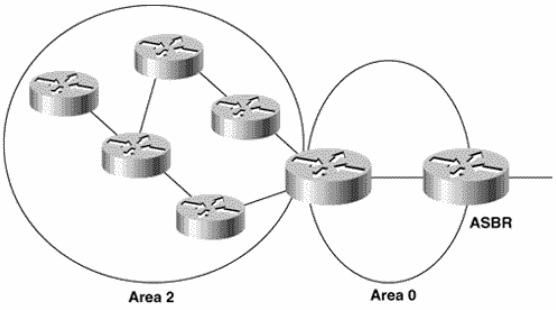
External Attributes LSA

被提议为作为 iBGP 的另一种选择，用来传输 BGP 协议信息穿越一个 OSPF 域。不过没有实现

Opaque LSA

特殊应用，透明 LSA，基于 MPLS。暂时没有实现

Stub Area



由于并不是每个路由器都需要外部网络的信息，为了减少 LSA 泛洪量和路由表条目，右图中的 Area 2 可以配置成为 Stub 区域

位于 Stub 边界的 ABR 将宣告一条默认路由到所有的 Stub 区域内的内部路由器，

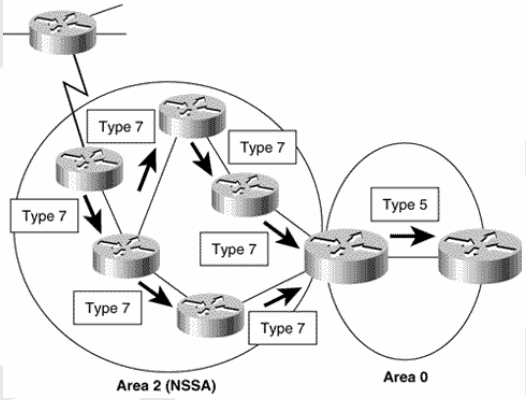
Stub 区域限制：

a) 所有位于 stub area 的路由器必须保持 LSDB 信息同步，并且它们会在它的 Hello 包中设置一个值为 0 的 E 位(E-bit),因此这些路由器是不会接收 E 位为 1 的 Hello 包,也就是说在 stub area 里没有配置成 stub router 的路由器将不能和其他配置成 stub router 的路由器建立邻接关系

b) 不能在 stub area 中配置虚链接(virtual link),并且虚链接不能穿越 stub area

c) stub area 里的路由器不可以是 ASBR

d) stub area 可以有多个 ABR,但是由于默认路由的缘故,内部路由器无法判定哪个 ABR 才是到达 ASBR 的最佳选择



Cisco 提出了 Totally Stubby Area,和 stub area 相比,它连 LSA 类型 3(Network Summary LSA)也阻塞掉了

NSSA 允许外部路由被宣告 OSPF 域中来，同时保留 Stub Area 的特征，因此 NSSA 里可以有 ASBR，ASBR 将使用 type7-LSA 来宣告外部路由，但经过 ABR，Type7 被转换为 Type5。 7 类 LSA 通过 OSPF 报头的一个 P-bit 作 Tag，如果 NSSA 里的 ABR 收到 P 位设置为 1 的 NSSA External LSA,它将把 LSA 类型 7 转换为 LSA 类型 5.并把它洪泛到其他区域中;如果收到的是 P 位设置为 0 的 NSSA External LSA,它将不会转换成类型 5 的 LSA,并且这个类型 7 的 LSA 里的目标地址也不会被宣告到 NSSA 的外部 NSSA 在 IOS11.2 后支持。

几种特殊的区域

Area Type	1&2	3&4	5	7
Backbone (area 0)	Yes	Yes	Yes	No
Non-backbone, non-stub	Yes	Yes	Yes	No
Stub	Yes	Yes	No	No
Totally stubby	Yes	No*	No	No
Not-so-stubby(NSSA)	Yes	Yes	No	Yes

*仅有1种例外，每一个ABR通过Type-3通告缺省路由

Stub

过滤 Type5LSA 并自动产生默认路由 O*IA 但不过滤掉域间路由

Stub-No Summary

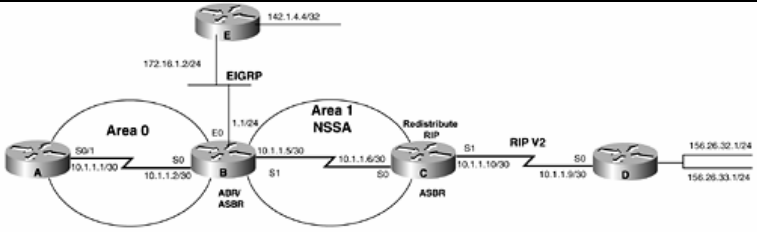
只放入 Type 2&6 类的路由，即域间路由被过滤掉，同时产生 0 *IA 的默认路由

NSSA

ASBR 使用 Type 7LSA 通告外部路由，路由条目由 O E2 改为 O N2,NSSA 不会自动产生默认路由，除非加入 area 2 nssa default-information-originate.且默认路由仅用在 ASBR，ABR 上，产生一条 0 *N2 的路由。同时过滤掉骨干区域来的 Type5 路由.

NSSA-No redistribution

如下图所示，B 上使用 area 1 nssa no-redistribution, B 上将不会把 EIGRP 做为 Type7LSA 通告，所以 C 上原本会收到的 2 条 O N2 路由将会消失



NSSA-No summary

即在路由器 B 上 area 1 nssa no-summary 过滤域间路由，并在自动产生一条 O *IA 的默认路由

The Route Tables

路由选择表是根据 LSDB 中 LSA 条目，使用 SPF 算法生成的。OSPF 是基于接口带宽进行度量的一种路由协议使用 10^8/BW 计算。

Interface Type	Cost (10^8 /BW)
FDDI, Fast Ethernet	1
HSSI (45M)	2
16M Token Ring	6
Ethernet	10
4M Token Ring	25
T1 (1.544M)	64
DS0 (64K)*	1562
56K*	1785
Tunnel (9K)	11111
Assumes the default bandwidth of the serial interface has been changed.	

通过命令 ip ospf cost 1~65535 可以修改端口的成本

不推荐 ip ospf cost 的方式修改

使用 auto-cost reference-bandwidth bandwidth(1~4296967Mb)

来修改默认的参考值，以便识别 1000M 和 10G 的 interface 两个不同的区域可以使用不同的参考带宽

Destination Type

每一个路由条目都可以被归类到目的类型中去网络条目(Network Entry)是数据包药转发的目的网络:

路由器条目(Routers Entry)记录的是如何到达 ABR/ASBR,这些路由器条目被单独放在一个内部的路由表里,

使用命令 show ip ospf border-routers 来查看这个信息,如下:i 代表区域内路由, I 代表区域间路由

```
Homer#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is 192.168.32.2 to network 0.0.0.0

O E1  192.168.118.0/24 [110/94] via 192.168.17.74, 02:15:01, Ethernet0
O E1  10.0.0.0/8 [110/84] via 192.168.17.41, 02:15:01, Serial0.19
存     O E1  192.168.119.0/24 [110/94] via 192.168.17.74, 02:15:01, Ethernet0
O E2  172.19.0.0/16 [110/21] via 192.168.32.2, 02:15:01, Ethernet1
      172.21.0.0/16 is variably subnetted, 2 subnets, 2 masks
O E2     172.21.0.0/16 [110/801] via 192.168.21.6, 02:15:01, Serial1.724
O        172.21.121.0/24 [110/791] via 192.168.21.6, 04:18:30, Serial1.724
      172.16.0.0/16 is variably subnetted, 104 subnets, 7 masks
O        172.16.21.48/30 [110/844] via 192.168.21.10, 04:18:48, Serial1.725
O IA     172.16.30.61/32 [110/856] via 192.168.17.74, 02:15:19, Ethernet0
O IA     172.16.35.0/24 [110/865] via 192.168.17.74, 02:15:19, Ethernet0
C        172.16.32.0/24 is directly connected, Ethernet1
O        172.16.17.48/29 [110/74] via 192.168.17.74, 06:19:46, Ethernet0
O E1     172.16.46.0/24 [110/30] via 192.168.32.2, 02:15:19, Ethernet1
O        172.16.45.0/24 [110/20] via 192.168.32.2, 3d10h, Ethernet1
O IA     172.16.30.54/32 [110/1061] via 192.168.17.74, 02:15:21, Ethernet0
O        172.16.17.56/29 [110/84] via 192.168.17.74, 06:19:48, Ethernet0
O        172.16.54.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
O        172.16.55.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
O        172.16.52.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
O        172.16.53.0/24 [110/11] via 192.168.32.2, 3d10h, Ethernet1
C        172.16.25.28/30 is directly connected, Tunnel29
--More--
```

```
Homer#show ip ospf border-routers

OSPF Process 1 internal Routing Table

Codes:  i - Intra-area route, I - Inter-area route
i 192.168.30.10 [74] via 192.168.17.74, Ethernet0, ABR, Area 0, SPF 391
I 192.168.30.12 [148] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
I 192.168.30.18 [205] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
i 192.168.30.20 [84] via 192.168.17.74, Ethernet0, ABR, Area 0, SPF 391
i 192.168.30.27 [781] via 192.168.21.6, Serial1.724, ASBR, Area 7, SPF 631
i 192.168.30.30 [74] via 192.168.17.74, Ethernet0, ABR/ASBR, Area 0, SPF 391
I 192.168.30.38 [269] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
i 192.168.30.37 [390] via 192.168.21.10, Serial1.725, ASBR, Area 7, SPF 631
i 192.168.30.40 [84] via 192.168.17.74, Ethernet0, ABR/ASBR, Area 0, SPF 391
i 192.168.30.47 [400] via 192.168.21.10, Serial1.725, ASBR, Area 7, SPF 631
i 192.168.30.50 [74] via 192.168.17.41, Serial0.19, ABR/ASBR, Area 0, SPF 391
I 192.168.30.62 [94] via 192.168.17.74, Ethernet0, ASBR, Area 0, SPF 391
i 192.168.30.60 [64] via 192.168.17.41, Serial0.19, ABR/ASBR, Area 0, SPF 391
i 192.168.30.60 [790] via 192.168.21.10, Serial1.725, ABR/ASBR, Area 7, SPF 631
i 192.168.30.80 [10] via 192.168.32.5, Ethernet1, ABR/ASBR, Area 78, SPF 158
i 192.168.30.80 [10] via 192.168.17.74, Ethernet0, ABR/ASBR, Area 0, SPF 391
i 172.20.57.254 [10] via 192.168.32.2, Ethernet1, ASBR, Area 78, SPF 158
Homer#
```

Path Type

到达目标网络的路径可以分为 **4** 类:**intra-area path(**区域内路由**),inter-area path(**区域间路由**),**
Type 1 external path(类型 **1** 的外部路径,**E1)**和 **Type 2 external path(**类型 **2** 的外部路径,**E2)**

路由标记如下:

1. O 区域内路由
2. O IA 区域间路由
3. O N1 O N2 Nssa 区域通告的 Type7LSA 生成的路由
4. O E1 O E2 外部路由 区别在于后者计算时会不必加上到达 ASBR 的 Metric

路由选择表的查找顺序

1. 路由器读取目的地址，并查找路由表，选择和目的地址最精确匹配(最长匹配)的路由进行转发
2. 如果存在 2 条相同的路由,查看管理距离
3. 根据 O > O IA > O E1 > O E2 的优先级顺序转发
4. 如果此时还存在等 Cost 路径，则作负载均衡转发

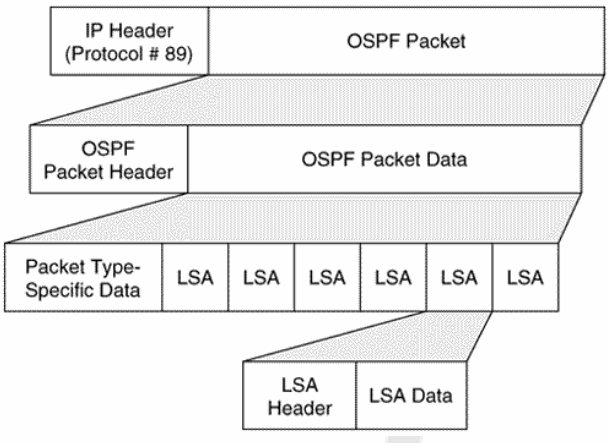
OSPF 默认 4 条路径负载均衡， 可以通过 maximum-paths 6 修改，最多 6 条路径负载均衡

OSPF over Demand Circuits

在 X.25 SVC, ISDN, PSTN 等即用即连的电路上, OSPF 在按需电路设计上具有抑制 Hello Packet 和 LSA-reflush 的功能, 最初会执行 LSDB 的同步, 随后指挥激活此条链路去 flooding 某些变化的 LSA

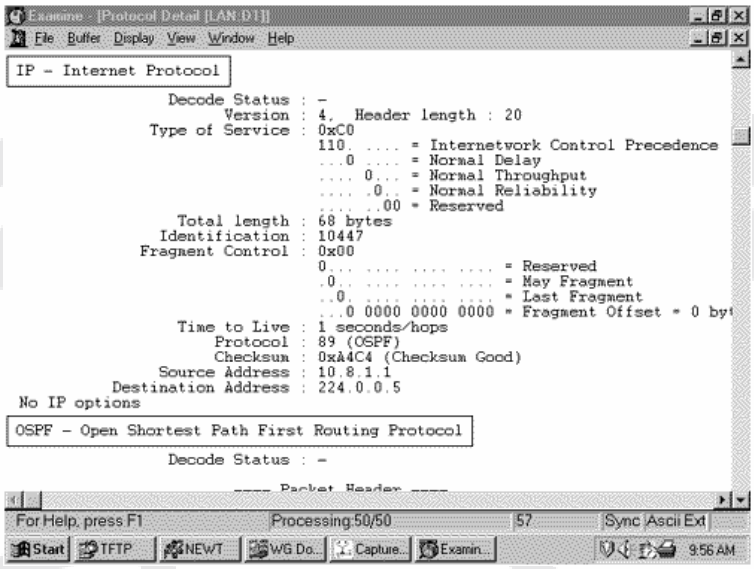
OSPF Packet Formats

OSPF 报文有多重封装构成, 如下图, 分为 IP 头, OSPF 报头, LSA 头等



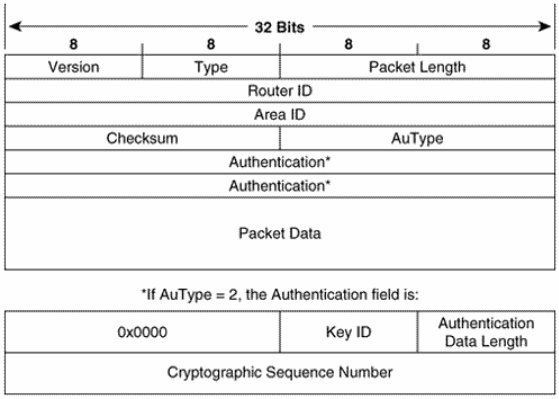
IP Packet Head

IP 报头可以看出协议号为 89, TTL=1 即 OSPF packet 转发不会超过 1 跳, 一些路由器可以设置 Precedence bit 来运行一些具有优先次序的报文处理, 包括 WFQ—加权公平队列 或者 WRED--加权随机预先检测序列



OSPF Packet

OSPF packet 的头部长度为 24 字节



Version: 当前最新为版本 2

Type: 类型 1 为 Hello 包;
类型 2 为数据库描述包(DDP);
类型 3 为链路状态请求包(LSR);
类型 4 为链路状态更新包(LSU);
类型 5 为链路状态确认包(LSAck)

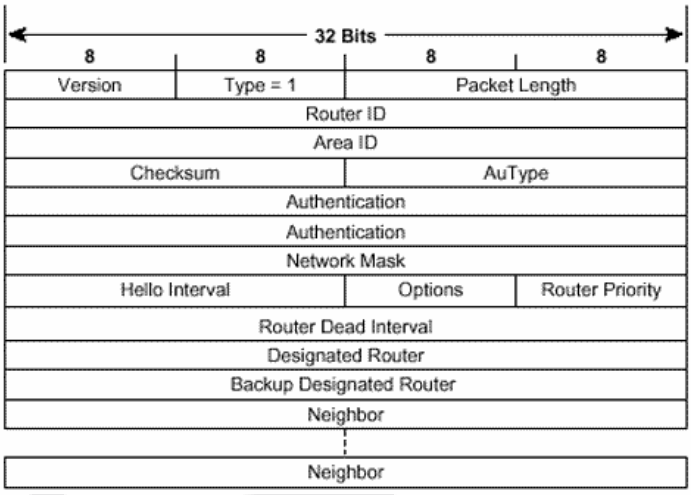
Area ID:
区域 ID, 如果是虚链接, 区域 ID 为 0.0.0.0(即 Backbone Area 的区域 ID), 因为虚链接被认识是骨干区域的 1 部分

AuType: 认证类型, 0 为 null, 即没有认证;1 为简单口令验证(即明文);2 为 MD5 加密验证

Hello Packet

Network Mask: 发送这个Hello包的接口的掩码, 如果收到这个Hello包的接口的掩码和它不能精确匹配, 这个 Hello 包将被丢弃

Router Dead Interval: 始发路由器在宣告邻居路由器无效前，将要等待的从邻居路由器发出的 Hello 包的时长

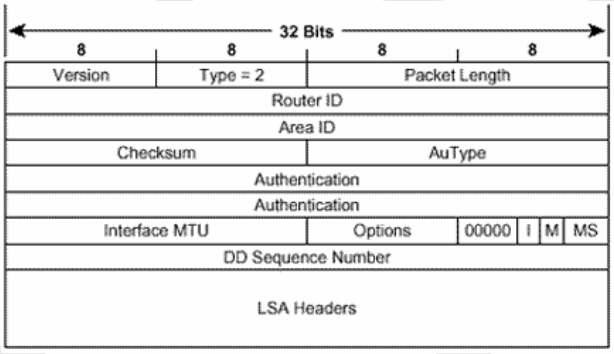


Router Priority: 优先级，参与 DR/BDR 选举之用，默认为 1，
如果为 0 的话将不能被选举为 DR/BDR

DR/BDR: DR/BDR 的接口的 IP 地址(而非 DR/BDR 的 RID). 在选举 DR/BDR 的过程中，这个 DR/BDR 字段的值并不一定就是最终选举出来的 DR/BDR 的值. 如果没有 DR/BDR(未选举或者是不需要 DR/BDR 的网络类型)，该字段的值就为 0.0.0.0

Neighbor: 是一个循环重复字段，列出了始发路由器在过去的一个 RDI 时间内受到的有效 Hello 包的网络上的所有 neighbor

DDP(Database Description Packets)

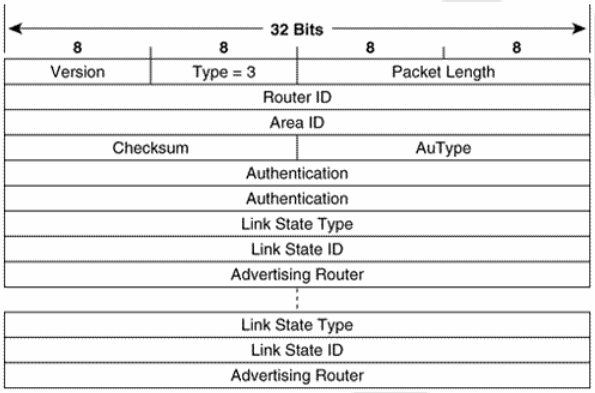


Interface MTU: 在报文不分段的情况喜爱，始发路由器接口可以发送的最大 IP 报文大小。在虚链接
上这个字段的值设置为 0x0000

I: Initial bit, 初始化位. 最初的 DDP 把该位设置为 1, 后续的设置为 0

M: More bit, 后续位, 当 DDP 不是这一系列的 DDP 中最后的一个 DDP 的时候, 该位设置为 1

MS: Master/Slave bit. 主/从位. Master 为 1, Slave 设置为 0



Link State Request Packet

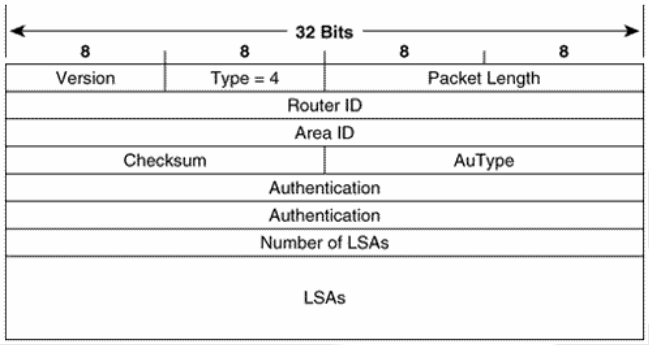
Link State Type:用来表明 LSA 的类型

Link State ID:是 LSA 头部中和类型无关的字段

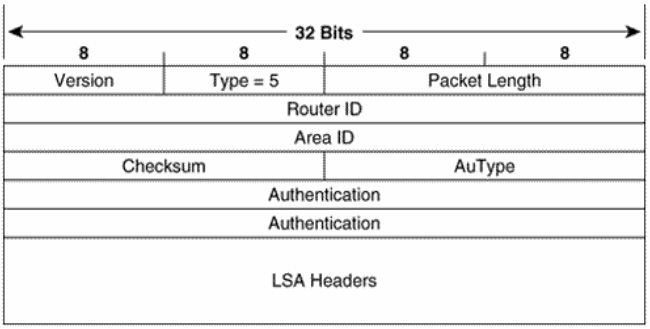
Advertising Router:宣告这条 LSA 的路由器的 RID

Link State Update Packet

Number of LSA -报文中 LSA 数量

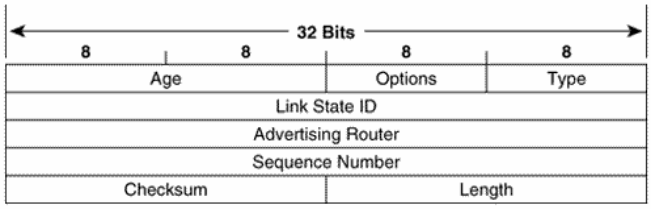


Link State Ack Packet

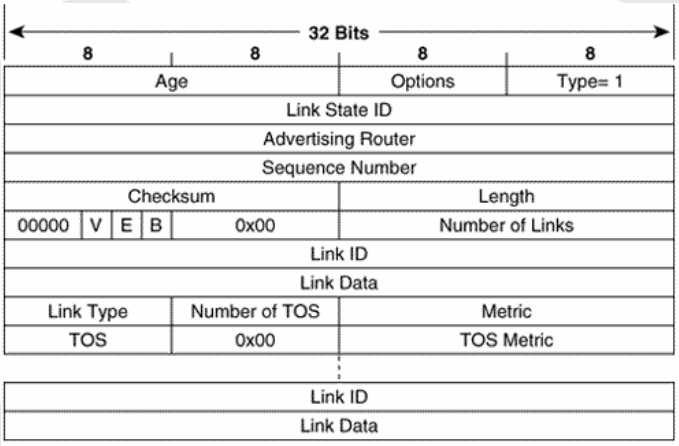


OSPF LSA Formats

LSA packet head:



Router LSA



Link State ID: 发出这个 Router LSA 的路由器的 RID

V: Virtual Link Endpoint bit,设置为 1 表示源路由器是一条或多条具有完全邻接的虚链接的节点

E: External bit,设置为 1 表示源路由器为 ASBR

B: Border bit,设置为 1 表示源路由器为 ABR

Link Type: 类型 1 为点到点的连接到另一台路由器;

类型 2 表示连接到 1 个 Transit Network;

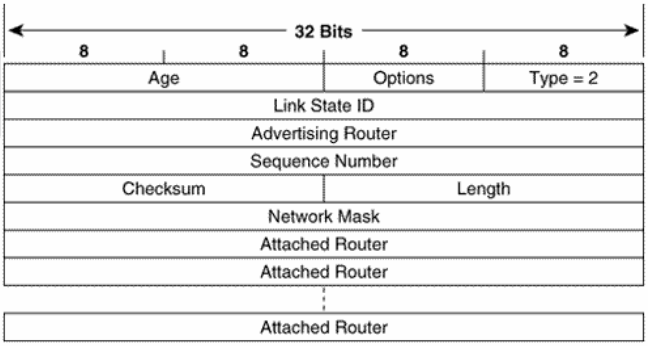
类型 3 表示连接到 1 个 Stub Network;
类型 4 表示虚链路

Link ID: 1, 邻居路由器的 RID
2, DR 路由器的接口的 IP 地址
3, IP 网络或者子网地址
4, 邻居路由器的 RID

Link Date: 依赖于链路类型字段的值

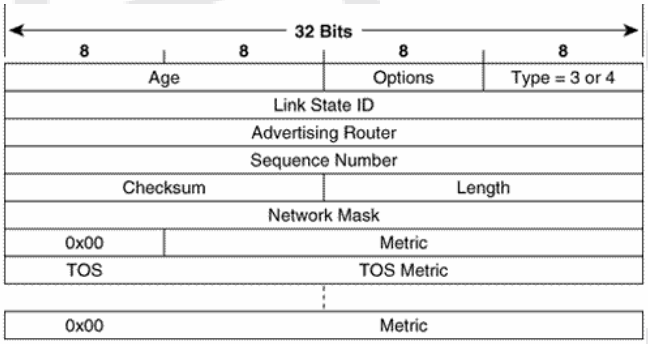
Notice: Cisco 的路由器只支持 TOS=0

Network LSA



Show ip ospf database network

Network Summary and ASBR Summary LSA

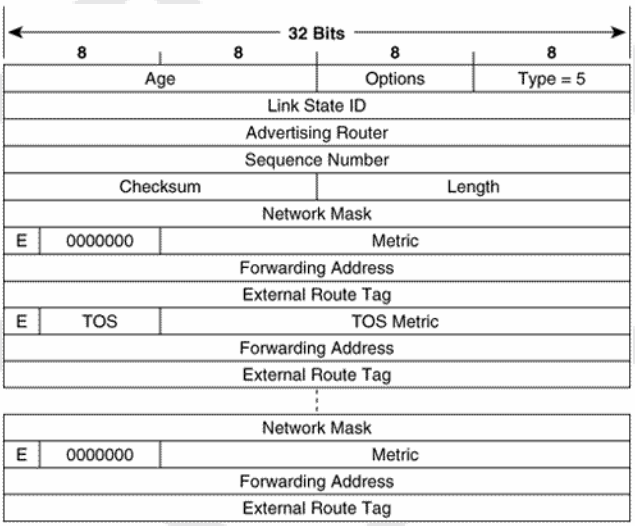


Link State ID: Type3 通告网络或子网的 IP
Type4 通告 ASBR 的 RID

Networkmask 0.0.0.0

Show ip ospf database summary
Show ip ospf database asbr-summary

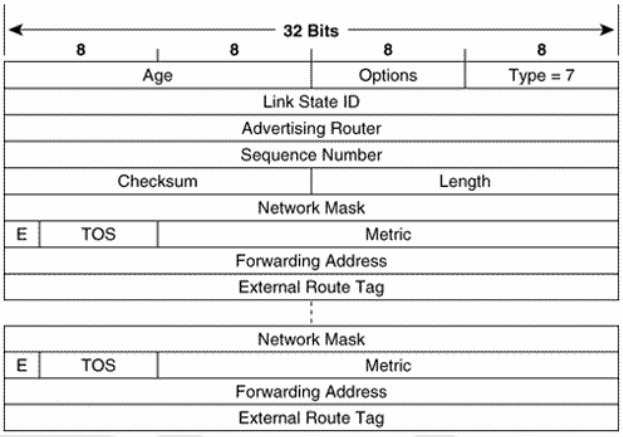
AS External LSA



Link State ID: 目标地址的 IP
Network Mask: 目标地址的子网掩码
如果类型 5 的 LSA 宣告的是默认路由, 那么上面 2 个字段都为 0.0.0.0
E: 1 度量类型 E2 0 度量类型 E1
Forwarding Address:
数据包应该被转发到的地址. 设置为 0.0.0.0 的话表示数据包将转发给源 ASBR
External Route Tag:
用于 route-map 的 tag

Show ip ospf database external

NSSA External LSA



Forwarding Address: 如果网络是 NSSA ASBR 和它邻接的 AS之间并作为内部路由进行宣告的话,这个转发地址就是网络的下一跳;如果不是,这个转发地址就的 NSSA ASBR 的 RID

Show ip ospf database nssa-external

The Options Field

* 不使用通常设置为 0

DC 当始发路由器具有支持按需电路上的 OSPF 能力时，此位将被设置

EA 当始发路由器具有接受和转发外部属性 LSA 的能力时，此位将被设置

N 只用在 Hello 报文， 1 将支持 NSSA 外部 LSA 0 将不接受和发送外部 LSA

P 只用在 NSSA 外部 LSA 头部， 这一位将告诉一个 NSSA 区域中的 ABR 将 Type 7 转换为 Type 5

MC 当始发路由器支持 IP 组播，此位将被设置

E 当始发路由器具有接受外部 LSA 能力时，此位将被设置 外部 LSA BB-LSA 非 stub 区域

T 当路由器支持 TOS，此位将被设置

OSPF Command and Configuration

OSPF Process Configuration Commands

Router ospf process-id :

*process-id 1~65535

***no router ospf** process-id 删除进程

启动 OSPF 进程，并分配 RID。RID 除非进程重新启动或者 RID 使用的接口状态是不活动的，否则不会改变 RID,配置 OSPF 前，检查 loopback 口是否配置 ip addr。Loopback 不是强制的，但可以稳定 OSPF 进程

Route ospf process-id **vrf** name:

*process-id 1~65535

*name: 一个 VPN 路由选择转发表<VRF>实例的名称。通过 OSPF 进程学到路由将替代全球的 ip 路由选择表被注入 vrf 的路由选择表

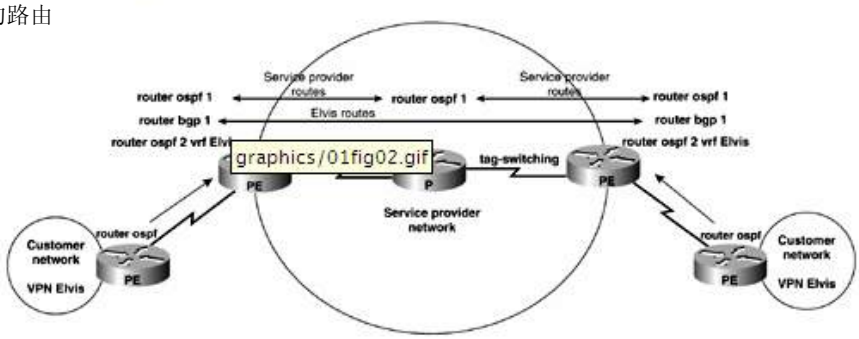
在 MPLS-VPN 环境下，此命令用于在 ISP 和 VPN 用户间传送 VPN 用户的路由

P ：ISP 提供商

CE：用户边缘路由器

PE：提供商边缘路由器

Figure 1-2. General MPLS/VPN Architecture



OSPF Area Commands

Area area-id **authentication**

*area-id 0~4294967295 或者 0.0.0.0~255.255.255.255 形式输入

*同时要在物理接口上启用 **ip ospf authentication-key** password

*在区域 0 使用认证的虚电路： **area** transit-area **virtual-link** router-id **authentication-key** password

在 OSPF 区域内启动一个简单的明文密码认证。OSPF 简单认证需要将密码配置在接口或者虚电路上。在 IOS12.X 中同区域内的不同接口使用认证的方法无需相同。 **ip ospf authentication null** 可以删除所选认证。配置密码时，空格符是有效字符，如 cisco<space>和 cisco 完全不同

Area area-id **authentication message-digest**

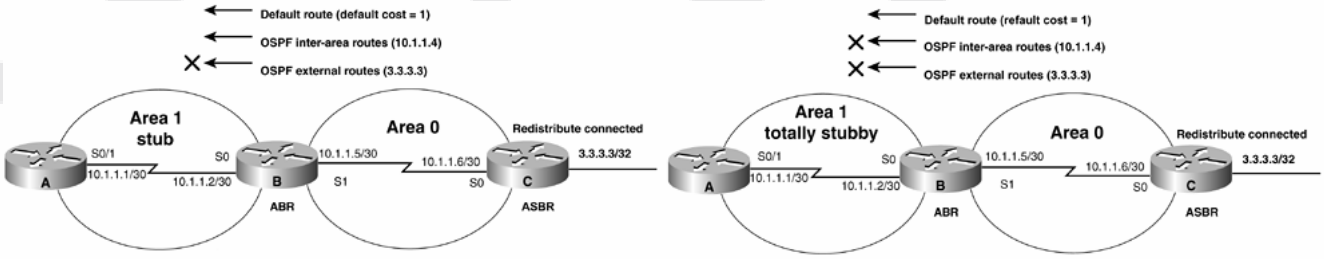
*采用 md5 的 authentication。在物理接口上启用 **ip ospf message-digest-key** key-id **md5** password

*虚电路上 **area** transit-area **virtul-link** router-id **message-digest-key** key-id **md5** password

配置验证时必须要求 key-id 和 password 都匹配，同时配置密码可以在全局配置下 **service password-encryption** 进行加密。启用新的密钥，先加入新的 key-id，然后在 no 旧密钥。

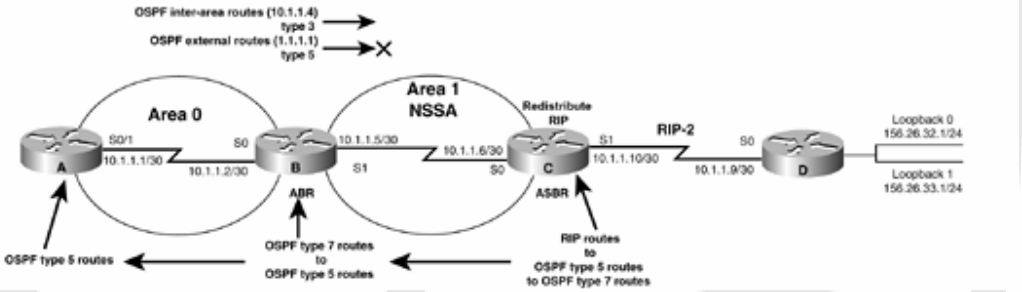
Area area-id **default-cost** cost

外部网络路由不能在一个 stub 区域或者 totally-stub 区域内广播，当一个 OSPF 区域配置为 stub 区域后，在 stub 区域的 ABR 会产生一个缺省路由代替外部路由，当 OSPF 配置为 totally-stub 时，缺省路由会代替外部和区域间路由。这个命令是设置广播到 stub 区域，totally-stub 区域或者 NSSA 区域中的缺省路由的成本的，默认缺省成本为 1。



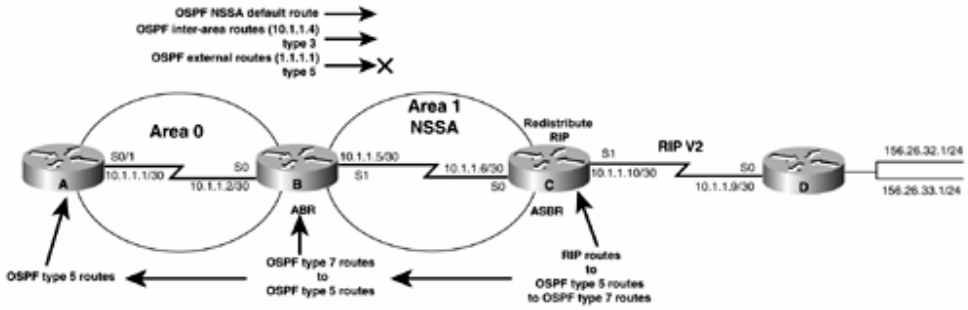
Area area-id **nssa**

在一个 stub 区域或者完全 stub 区域，stub 区域的 ABR 会阻止 OSPF 外部路由(LSA5)在 stub 区域内广播。故 ASBR 将不会成为 stub 区域的一部分。NSSA 区域中，允许路由器作为 ASBR redistribution，被配置为 NSSA 区域的 ASBR 将产生 7 类路由



Area area-id **nssa default-information-originate**

在 OSPF ABR 或者 ASBR 中产生一个能够进入 NSSA 区域的 OSPF NSSA 外部类型 2 的缺省路由，在 ABR 上无须定义。在 ASBR 上配置该命令，则需要配置缺省路由。在 B 上配置后，C 会产生一条 O*N2 的路由指向 B

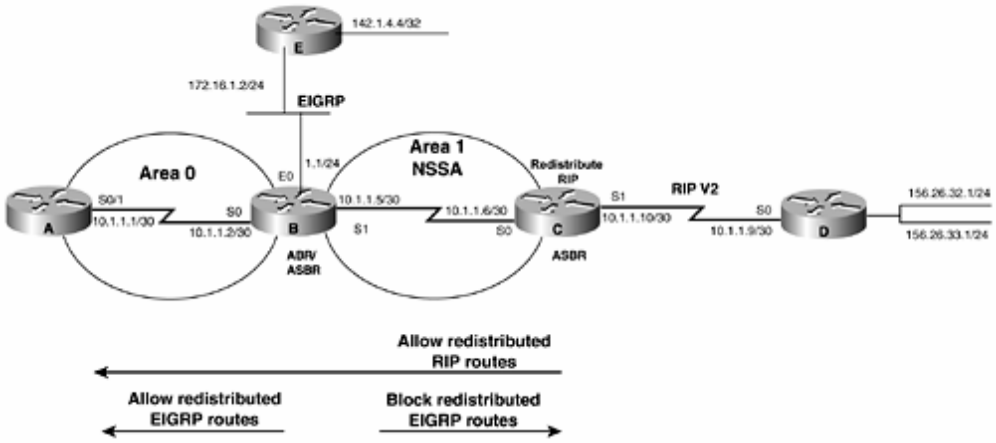


注意:此条命令仅用在 NSSA ABR 或者 ASBR 上

Area area-id **nssa no-redistribution**

在下图中，EIGRP 的信息将从 ABR 上 redistribution 进入到 NSSA 区域，故可以采用此条命令，防止 NSSA 产生由 EIGRP 产生的

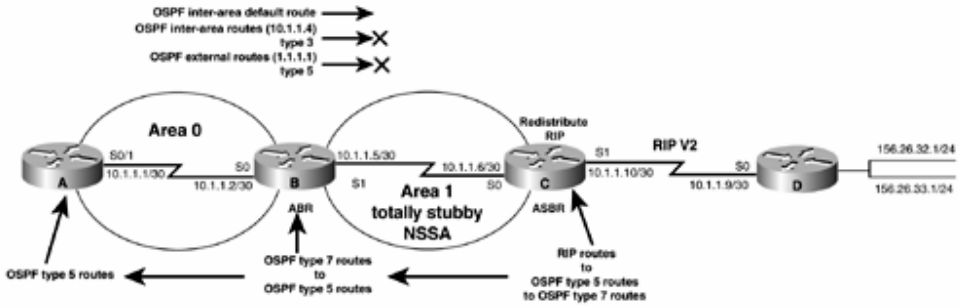
外部路由



注意:此条命令仅用在 ASBR 上，其它 NSSA 每台路由器只需 area **x** nssa

Area area-id **nssa no-summary**

该命令用于在 OSPF ABR,阻止 OSPF 区域间路由进入 NSSA 区域,该命令同时产生一条道路一个 OSPF 区域间的缺省路由



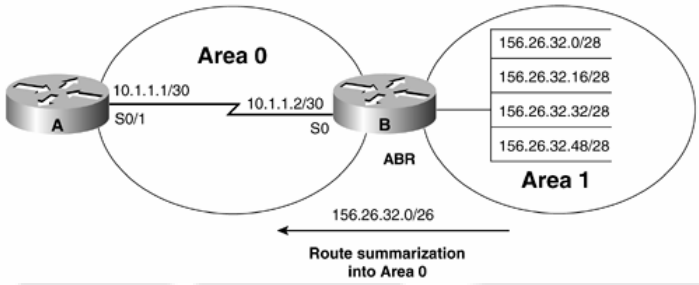
注意：no-summary 仅用在 NSSA ABR 上

Area area-id **range** ip-addr mask **[advertise | not-advertise]**

OSPF 可以会聚从骨干网或者区域 0 到非 0 OSPF 区域的 OSPF 路由，或者从非 0 区域到骨干网的 OSPF 路由

OSPF 路由汇聚只能在 ABR 上发生，ABR 应是一个在区域 0，并且在非 0 区域有接口的路由器。

默认情况下 advertise 是打开的， not-advertise 用于抑制 ABR 广播默认路由



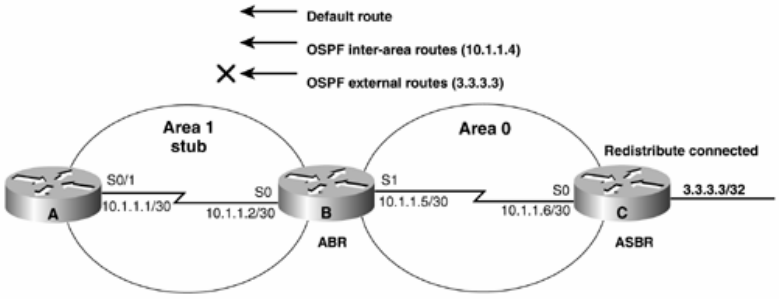
注意： 1。如果其中 156.26.32.0/28 down 了以后， IOS 将自动为这条中断链路创建一条默认路由

ip route 156.26.32.0 255.255.255.192 Null0

2。Area x range 仅用在 OSPF ABR 上

Area area-id stub

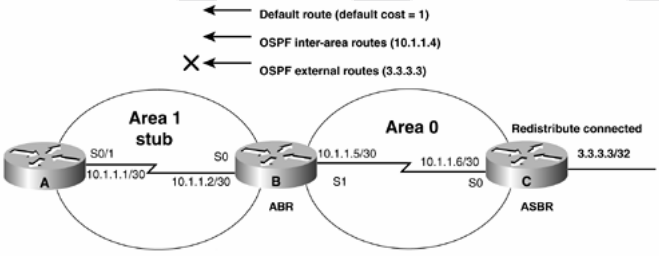
将区域配置成为 stub 区域后,ABR 将产生一条默认路由，通告到整个 stub 区域，域间路由能通过但 5 类的外部路由不能进入

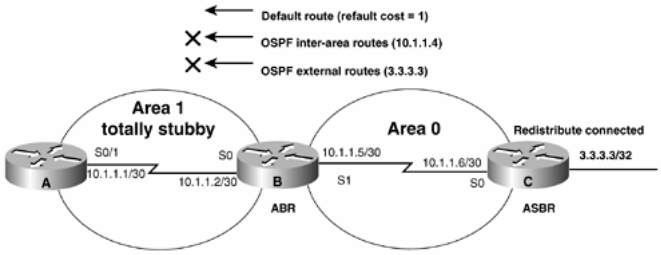


注意： Stub 区域不能作为 virtual-link 的穿透区域，Stub 区域内的每台路由器都应配置 stub area

Area area-id stub no-summary

将区域配置为完全 stub 区域后， 将会阻止区域间路由和外部路由通过

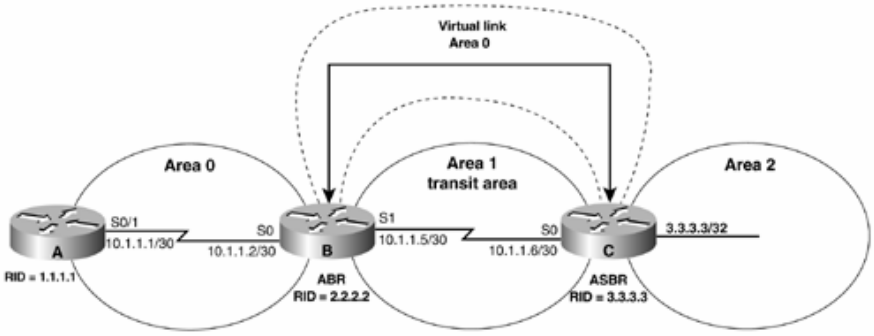




No-summary 仅用在 ABR 上，ASBR 不能成为完全末节的一部分。不能作为 virtual-link 的穿透区域

Area transit-area-id virtual-link router-id

所有非 0 OSPF 区域必须具有到骨干或者区域 0 的连接，虚拟链路用于修复被分割的区域，但传送区域不能为 stub 区域。



如上图所示，在 B,C 间建立 Virtual-link 只需要

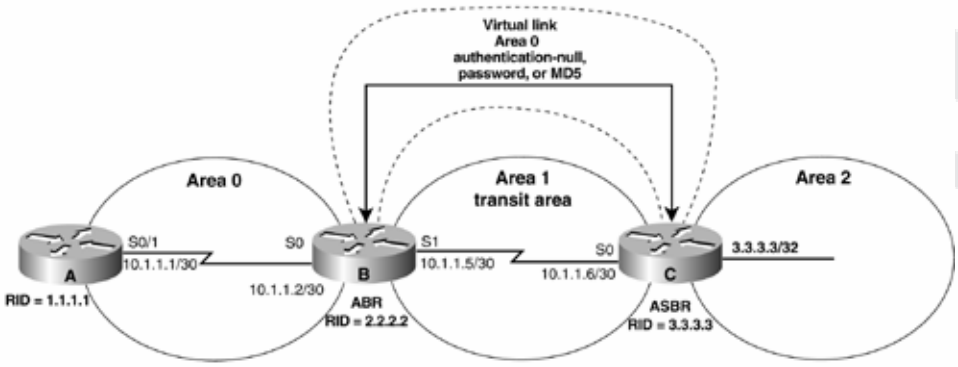
B<config-router>#area transit-area-id virtual-link router-C-ID

C<config-router>#area transit-area-id virtual-link router-B-ID

注意：虚拟电路不能频繁使用，设计网络时应该注意

Area transit-area-id **virtual-link** router-id **authentication**[authentication-key pass | message-disgest | null]

此命令用于在 Virtual-Link 上启用认证，可以采用明文或者 MD5 或者 null



1. 明文认证：

B: area 1 virtual-link 3.3.3.3 authentication authentication-key cisco

C: area 1 virtual-link 2.2.2.2 authentication authentication-key cisco

2. MD5

B: area 1 virtual-link 3.3.3.3 authentication message-digest

area 1 virtual-link 3.3.3.3 message-digest-key 1 md5 cisco

C: area 1 virtual-link 2.2.2.2 authentication message-digest

area 1 virtual-link 2.2.2.2 message-digest-key 1 md5 cisco

改变密码：

B，C 先声明 **key 2**，然后 **no key 1**

3. Null

如果区域 0 配置了认证，虚拟链路上会自动使用相同的认证，可以通过在虚拟链路上将认证设置为 Null 来取消

Eg：

A: ip ospf authentication-key Cisco

Area 0 authentication

B: ip ospf authentication-key Cisco

Area 0 authentication

Area 1 virtual-link 3.3.3.3

C: area 1 virtual-link 2.2.2.2

此时，在 B 上 debug 发现，由于 Mismatch authentication type，BC 不能建立邻居。 如果需要，则在 C 上加 key 但是为了安全和 key 范围限制，可以在 B，C 上设置 null

B: area 1 virtual-link 3.3.3.3 authentication null

C: area 1 virtual-link 2.2.2.2 authentication null

Area transit-area-id **virtual-link** router-id **authentication-key** password

如上例，在 Vlink 上不使用认证是不安全的，故可以在 v-link 链路上使用认证。

B: area 1 virtual-link 3.3.3.3 authentication-key kaka

C: area 1 virtual-link 2.2.2.2 authentication-key kaka

Area transit-area-id **virtual-link** router-id [**dead-interval** | **hello-interval** | **retransmit-interval** | **transmit-delay**]

修改虚拟链路上的 hello 和失效的时间间隔，如果二者其一不匹配都会导致链路中断，无法更新路由。默认时间为 10s/40s

Retransmit-interval: 当一台路由器在虚拟链路上广播 LSA 时，LSA 被夹在岛用于虚拟链路的重发列表，LSA 将被重发知道 LSA 被答复，广播间隔的时间即为 retransimit-interval,更改此参数不会导致链路中断

Transmit-delay: LSA 传送的时延.默认情况路由器将 LSA 的 age 域设置为 0，而此参数会直接加在 age 域上

Area transit-area-id **virtual-link** router-id **message-digiest-key** key-id **md5** password

在虚拟链路上使用 md5 的认证方式。

A: **ip ospf message-digest-key 1 md5 Cisco**
Area 0 authentication-message digest

B: **ip ospf message-digest-key 1 md5 Cisco**
Area 0 authentication-message digest
Area 1 virtual-link 3.3.3.3 message-digest-key 2 md5 kaka

C: **area 1 virtual-link 2.2.2.2 message-digest-key 2 md5 kaka**

更新密钥时，为了保证连接不中断，需要先在 B，C 上使用新的 key 3 ，然后再 no key 2

OSPF reference-bandwidth

Auto-cost reference-bandwidth bandwidth

默认情况下 cost 值对于大于 100M 的端口，都将表示为 1

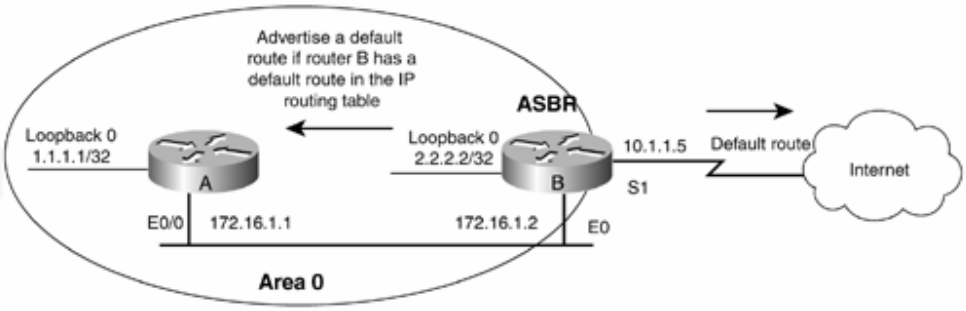
使用 auto-cost reference-bandwidth bandwidth(1~4296967Mb)

来修改默认的参考值，以便识别 1000M 和 10G 的 interface.更新后每个接口的成本都将改变，但可以使用 ip ospf cost 更改，但不建议使用这种方法

Default Route Generation

Default-information originate [always | metric | Metric-type]

在区域内产生默认路由，并广播到区域内，此命令用于 ASBR。



B: **ip route 0.0.0.0 0.0.0.0 S1**
Router ospf 1
Default-information originate

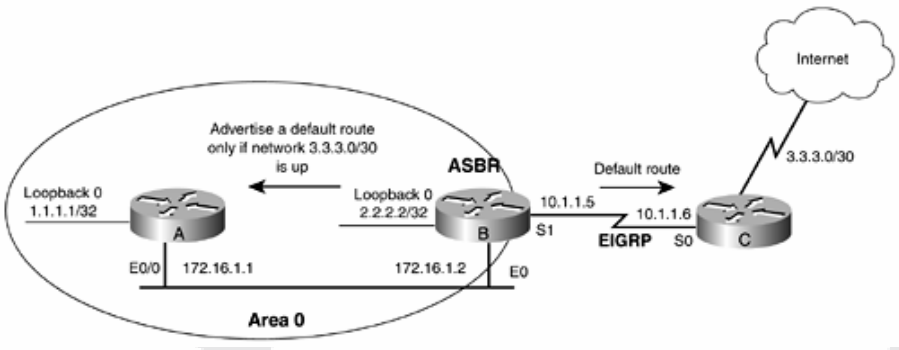
Aways: 此参数作用是防止由于缺省链路抖动(flapping)产生的缺省路由抖动，从而减少对 OSPF LSDB 的操作

Metric: Metric 缺省值为 1， 如果区域内有多条缺省路由，则可以更改 Metric，选择较佳缺省路由

Metric-type: 通告路由的类型。缺省成本是 1.缺省为 2 即 E2 类型的路由， 1 可以设置为 E1 类型

Default-information originate route-map route-map-name

使用 route-map 有条件的产生默认路由，如下图，当且仅当外部网络存在时，才发布缺省路由



Router B:

```
default-information originate route-map exist
!
access-list 1 permit 3.3.3.0 0.0.0.3
!
route-map exist permit 10
match ip address 1
```

Setting the Default Metric for Redistributed Protocols

Default-metric cost

对于重分布到 OSPF 的路由,如果没有使用 redistribute 命令分配一个度量,那么可以使用该命令分配一个成本,但使用此命令不会影响已由 redistribute 分配一个度量的路由,默认的 redistribute-cost BGP 的缺省度量是 1，其他协议是 20 在路由进程中，使用 Default-metric cost 可以更改

Administrative Distance

Distance administrative-distance

用于修改本地路由选择表中所有 OSPF 路由的管理距离，默认管理距离如下

connected— 0 static— 1 EBGP— 20 EIGRP— 90 IGRP— 100

OSPF— 110 IS-IS— 115 RIP— 120 IBGP— 200

Distance administrative-distance source-ip-address source-ip-mask access-list-number

用于修改特定来源的特定路由条目的管理距离，其中 source-ip-address 对于 OSPF 来说是源路由器的 router-id

Mask 是原地址的 ip 掩码，用反掩码表示

例如 R1 的 R-ID 为 1.1.1.1 则 distance 99 1.1.1.1 0.0.0.0 是修改来自 1.1.1.1 的管理距离，

若子网掩码为 0.0.0.255 则指来自 1.1.1.0/24 的所有路由器的管理距离。

同时可以可以使用 access-list 对特定的路由条目进行处理

Eg distance 80 1.1.1.1 0.0.0.0 1

Access-list 1 permit 3.3.3.0 0.0.0.255

Distance ospf [external | inter-area | intra-area] administrative-distance

修改本地路由器的 外部 域间 域内 路由条目的管理距离

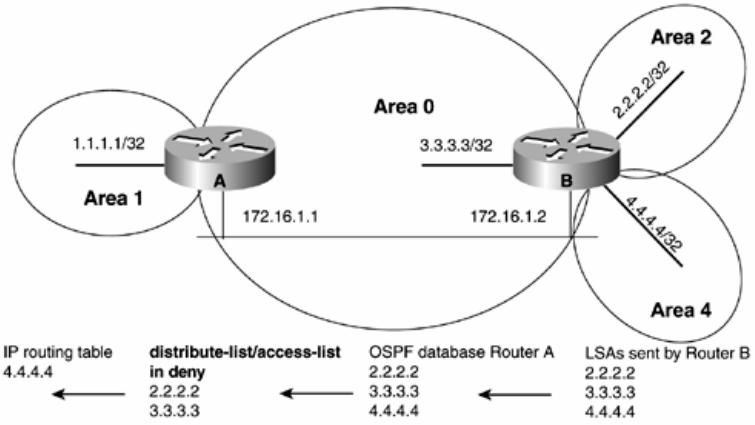
Distance opsf intra-area 70 inter-area 50 external 30

Filtering Routes with Distribute Lists

Distribute-list access-list-number in

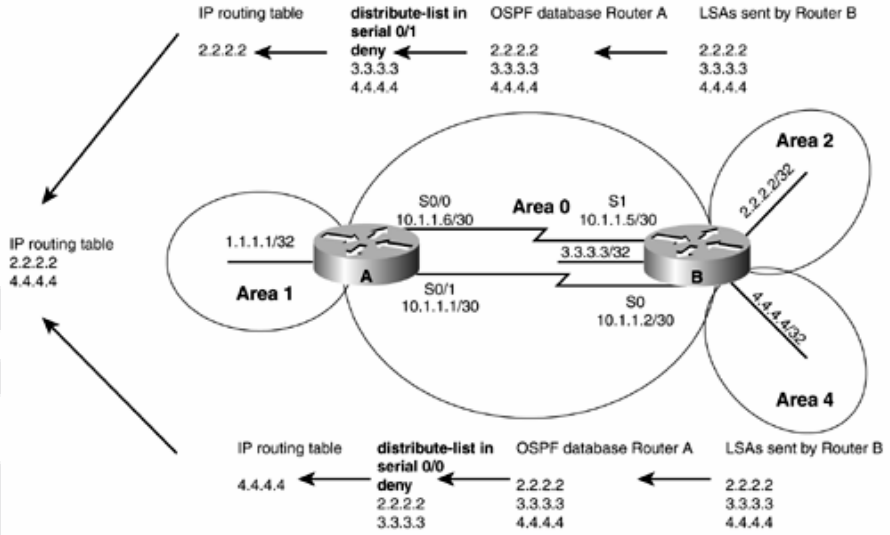
用于阻止从 OSPF 学到的路由条目放置到 IP 路由选择表如图所示，将 A 上过滤掉 2.2.2.0 和 3.3.3.0

```
distribute-list 1 in
!
access-list 1 deny    2.2.2.0 0.0.0.255
access-list 1 deny    3.3.3.0 0.0.0.255
access-list 1 permit any
```



Distribute-list access-list-number **in** interfacetype interface-number

用于阻止从特定接口学到的 OSPF 路由被放置到 ip 路由选择表中。

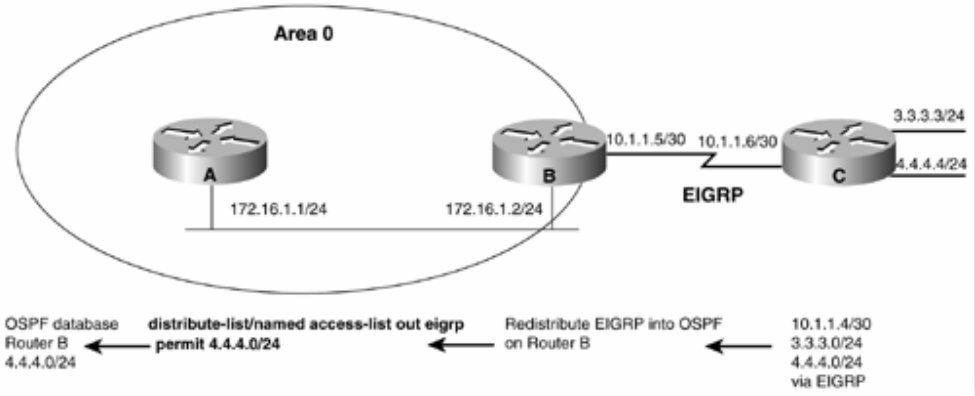



```
distribute-list 1 in Serial0/0
distribute-list 2 in Serial0/1
!
access-list 1 deny     2.2.2.0 0.0.0.255
access-list 1 deny     3.3.3.0 0.0.0.255
access-list 1 permit any
access-list 2 deny     3.3.3.0 0.0.0.255
access-list 2 deny     4.4.4.0 0.0.0.255
access-list 2 permit any
```

Distribute-list access-list-number **out** [interfacetype interface-number | routing-process]

对于距离矢量协议(RIP IGRP EIGRP)，该命令阻止被访问列表选定的路由广播到邻居处，OSPF 不会把路由广播到邻居，但会广播 LSDB，因此该命令在 OSPF 一起使用是无效的。

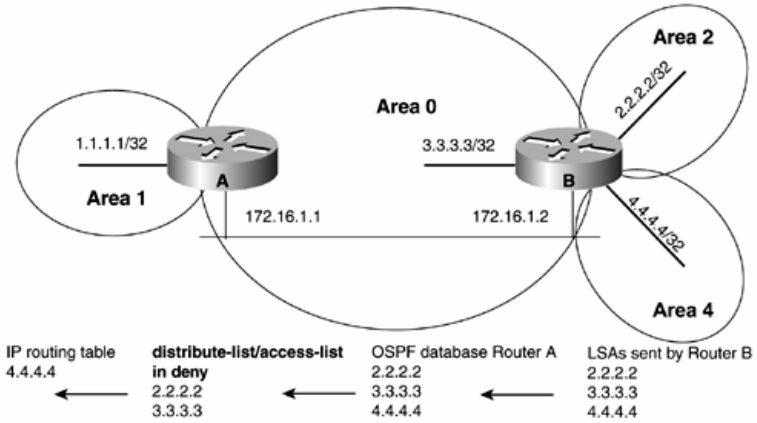
如果加入参数 routing-process 可以过滤相应的距离矢量路由协议重分布进 OSPF 的路由表项




```
distribute-list 1 out eigrp 1
!
access-list 1 permit 4.4.4.0 0.0.0.255
```

Distribute-list access-list-name **in**

用于阻止从 OSPF 学到的路由条目放置到 IP 路由选择表

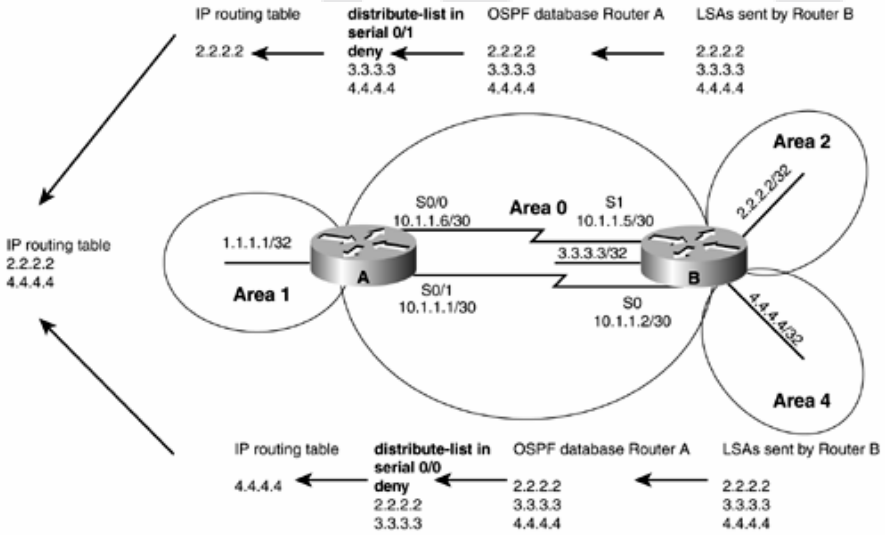


如图所示，将 A 上过滤掉 2.2.2.0 和 3.3.3.0

```
distribute-list filter-ospf in
!
ip access-list standard filter-ospf
 deny    2.2.2.0 0.0.0.255
 deny    3.3.3.0 0.0.0.255
 permit any
```

Distribute-list access-list-name **in** interfacetype interface-number

用于阻止从特定接口学到的 OSPF 路由被放置到 ip 路由选择表中。

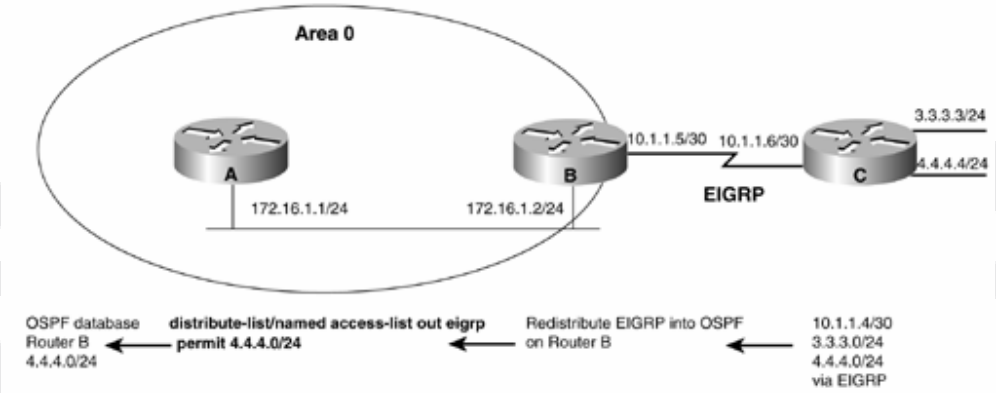



```
distribute-list filter-ospf1 in Serial0/0
distribute-list filter-ospf2 in Serial0/1
!
ip access-list standard filter-ospf1
 deny    2.2.2.0 0.0.0.255
 deny    3.3.3.0 0.0.0.255
 permit any
ip access-list standard filter-ospf2
 deny    3.3.3.0 0.0.0.255
 deny    4.4.4.0 0.0.0.255
 permit any
```

Distribute-list access-list-number out [interfacetype interface-number | routing-process]

对于距离矢量协议(RIP IGRP EIGRP)，该命令阻止被访问列表选定的路由广播到邻居处，OSPF 不会把路由广播到邻居，但会广播 LSDB，因此该命令在 OSPF 一起使用是无效的。

如果加入参数 routing-process 可以过滤相应的距离矢量路由协议重分布进 OSPF 的路由表项




```
distribute-list filter-eigrp out eigrp 1
!
ip access-list standard filter-eigrp
 permit 4.4.4.0 0.0.0.255
```

Distribute-list prefix prefix-name [in | out] {interface-type interface-number | routing-process}

如上 3 例使用 prefix 的配置如下

```
distribute-list prefix filter-ospf in
!
ip prefix-list filter-ospf seq 5 deny 2.2.2.2/32
ip prefix-list filter-ospf seq 10 deny 3.3.3.3/32
ip prefix-list filter-ospf seq 15 permit 0.0.0.0/0

distribute-list prefix filter-ospf1 in Serial0/0
distribute-list prefix filter-ospf2 in Serial0/1
!
ip prefix-list filter-ospf1 seq 5 deny 2.2.2.2/32
ip prefix-list filter-ospf1 seq 10 deny 3.3.3.3/32
ip prefix-list filter-ospf1 seq 15 permit 0.0.0.0/0
!
ip prefix-list filter-ospf2 seq 5 deny 3.3.3.3/32
ip prefix-list filter-ospf2 seq 10 deny 4.4.4.4/32
ip prefix-list filter-ospf2 seq 15 permit 0.0.0.0/0


distribute-list prefix filter-eigrp out eigrp 1
!
ip prefix-list filter-eigrp seq 5 permit 4.4.4.0/24
```

Handling of MOSPF LSAs

Ignore lsa mospf

Cisco 不支持 MOSPF 路由，缺省情况下会接受 Type6-LSA，路由器不支持，但会产生 syslog，此命令可以阻止 syslog 产生

Logging OSPF Neighbor Changes

Log-adjacency-changes {detail}

产生 OSPF 邻接关系变化的 log。

```
Logging buffered 4096 debugging
Log-adjacency-changes detail

#show logging
    00:56:38: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from FULL to DOW
    N, Neighbor Down: Interface down or detached
    00:56:43: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from DOWN to INI
    T, Received Hello
    00:56:43: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from INIT to 2WA
    Y, 2-Way Received
    00:56:43: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from 2WAY to EXS
    TART, AdjOK?
    00:56:43: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from EXSTART to
    EXCHANGE, Negotiation Done
    00:56:43: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from EXCHANGE to
     LOADING, Exchange Done
    00:56:43: %OSPF-5-ADJCHG: Process 1, Nbr 2.2.2.2 on Ethernet0/0 from LOADING to
    FULL, Loading Done
```

Multiple Path Configuration

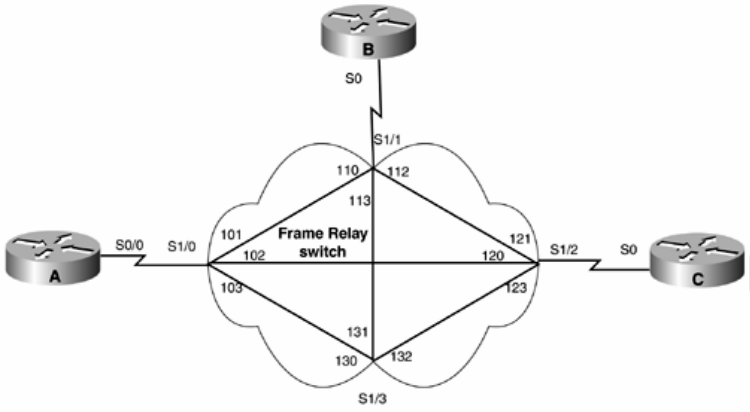
Maximum-paths number-of-paths

默认是 4 条路径的负载均衡，可以自行设置负载均衡链路条路　参数范围 1 ~ 6

OSPF neighbor Commands

Neighbor ip-address

OSPF 对待 NBMA 网络和其他任何广播网络一样，OSPF 认为这类网络具有广播特性，但必须使用 neighbor 命令建立一个 OSPF 邻居。在引入 ip ospf network 接口命令前，使用 neighbor 配置 OSPF 邻居

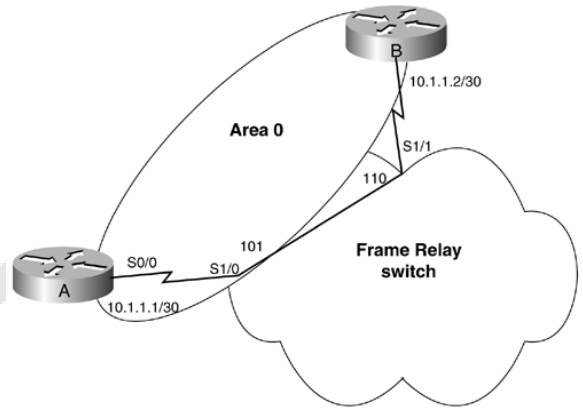


利用 Cisco 路由器模拟帧中继交换机配置如下

```
Frame Switch
hostname frame-relay
!
frame-relay switching
!
interface Serial1/0
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 no ip mroute-cache
 no fair-queue
 clockrate 2015232
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 101 interface Serial1/1 110
 frame-relay route 102 interface Serial1/2 120
 frame-relay route 103 interface Serial1/3 130
!
interface Serial1/1
 no ip address
no ip directed-broadcast
 encapsulation frame-relay
```

```
 clockrate 2015232
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 110 interface Serial1/0 101
 frame-relay route 112 interface Serial1/2 121
 frame-relay route 113 interface Serial1/3 131
!
interface Serial1/2
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 clockrate 2015232
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 120 interface Serial1/0 102
 frame-relay route 121 interface Serial1/1 112
 frame-relay route 123 interface Serial1/3 132
!
interface Serial1/3
 no ip address
 no ip directed-broadcast
 encapsulation frame-relay
 clockrate 2015232
 frame-relay lmi-type ansi
 frame-relay intf-type dce
 frame-relay route 130 interface Serial1/0 103
 frame-relay route 131 interface Serial1/1 113
 frame-relay route 132 interface Serial1/2 123
```

由于帧中继交换机是全互连的，路由器通过反向 ARP 获得所有特定接口的 DLCI，关闭路由器 A，B 的反向 ARP，并把适当 IP 映射到相应的 DLCI。

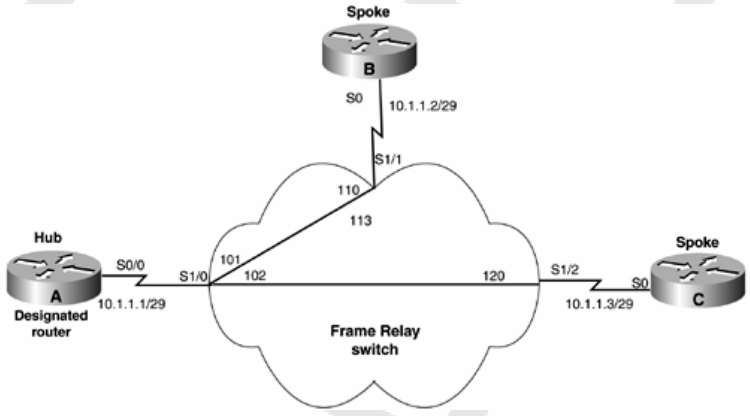


路由器 AB 配置如下

```
Router A
interface Serial0/0
 ip address 10.1.1.1 255.255.255.252
 encapsulation frame-relay
 frame-relay map ip 10.1.1.2 101 broadcast
 no frame-relay inverse-arp
 frame-relay lmi-type ansi
```

```
Router B
interface Serial0
 ip address 10.1.1.2 255.255.255.252
 encapsulation frame-relay
 frame-relay map ip 10.1.1.1 110 broadcast
 no frame-relay inverse-arp
 frame-relay lmi-type ansi
```

但一方加入 neighbor，邻居关系即将建立，并选举出 DR。 //OSPF neighbor 只需配置一端



如左图所示：
如果多个邻居，在同一子网内，则可以在 A 上指定 Neighbor

```
neighbor 10.1.1.3
neighbor 10.1.1.2
```

如果不在相同子网内，则在 B，C 上指定 Neighbor

```
B:  neighbor 10.1.1.1
C:  neighbor 10.1.1.1
```

Neighbor ip-address cost cost

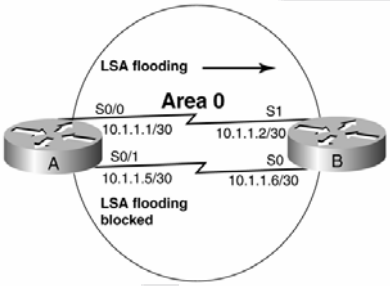
帧中继接口在 OSPF 下 Cost 为 48，Neighbor cost 命令仅用于点到多点的网络

可以在接口上使用 ip ospf network point-to-multipoint 更改接口网络类型

Neighbor 10.1.1.1 cost 10

Neighbor ip-address database-filter all out

此命令为了防止 LSA 泛洪到指定的邻居。 许多 ISP 在 OSPF 邻居间使用冗余电路，当一个 OSPF 路由器收到一个 LSA，则 LSA 在所有的 OSPF 接口上泛洪，除了收到 LSA 的接口。 可以使用该命令阻止其中一个邻接 LSA 的扩散



neighbor 10.1.1.6 database-filter all out

也可以在接口上启用

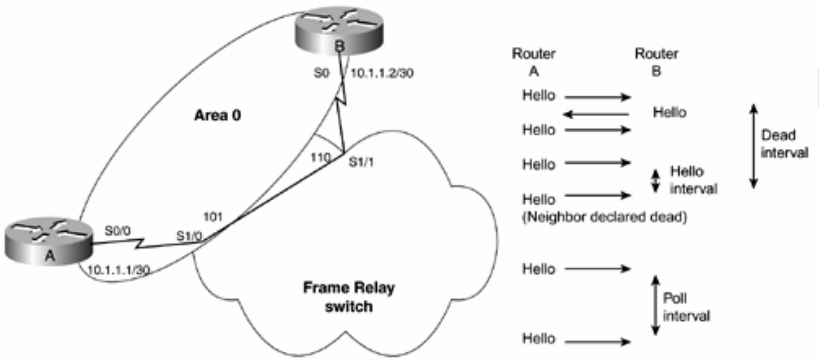
Ip ospf database-filter all out

注意： 该命令仅用于 p-to-mp 或者 NBMA 类型接口

如果 OSPF 邻居具有全连接，那么 database-filter all out 应该有效

Neighbor ip-address poll-interval interval

如果在终止间隔内没有从邻居处收到 hello packet，那么这个邻居是被宣告 down 的，当 down 后，hello 包会以论询间隔指定的速率发送到这个邻居，但这个选项不能用于 p-to-mp 接口上. 仅在 NBMA 网络使用



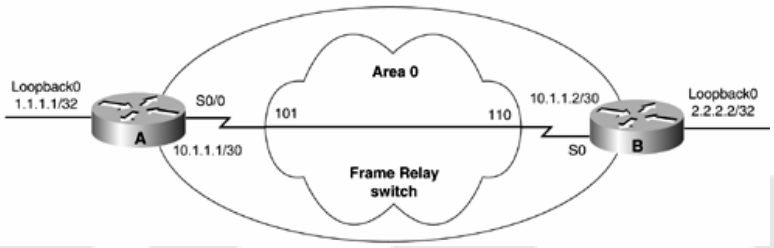

```
router ospf 1
 network 1.1.1.1 0.0.0.0 area 1
 network 10.1.1.0 0.0.0.3 area 0
 neighbor 10.1.1.2 poll-interval 90
```

可以通过 show ip ospf neighbor detail 查询

```
rtrA#show ip ospf neighbor detail
Neighbor 2.2.2.2, interface address 10.1.1.2
    In the area 0 via interface Serial0/0
    Neighbor priority is 0, State is FULL, 17 state changes
    DR is 10.1.1.2 BDR is 10.1.1.1
    Poll interval 90
    Options 2
    Dead timer due in 00:01:31
```

Neighbor ip-address priority priority

设置邻居的 priority 用于影响 DR 选举，默认 priority 为 10 不参与选择




```
Router B
router ospf 1
 network 2.2.2.2 0.0.0.0 area 2
 network 10.1.1.0 0.0.0.3 area 0
 neighbor 10.1.1.1 priority 2
```

则此时 neighbor 的优先级低于 routerB， routerB 成为 DR

OSPF network Command

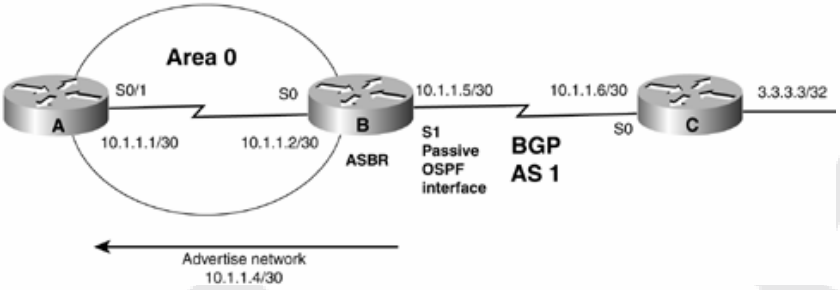
network ip-address wild-card-mask area area-id

用于通告和定义 OSPF 的区域,其中要使用 wild-card-mask Area-id 可以使用 点分 10 进制的形式

Passive OSPF Interfaces

passive-interface interface-name interface-number

使用被动接口减少协议流量，由于 OSPF 有 hello 机制，相对于 RIP，OSPF 端口 passive 以后，邻居关系立即断开，而 Rip 还可以接受路由更新信息



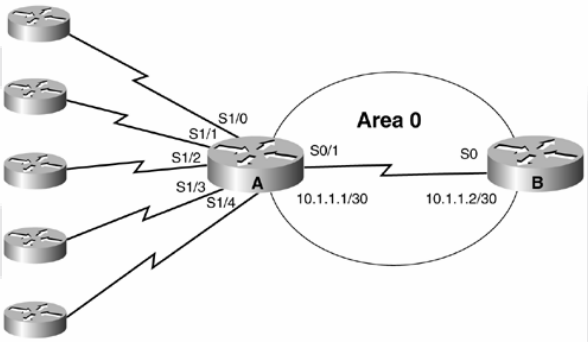

```
Router B
router ospf 1
redistribute bgp 1 subnets
passive-interface Serial1
network 2.2.2.2 0.0.0.0 area 2
network 10.1.1.0 0.0.0.3 area 0
network 10.1.1.4 0.0.0.3 area 0
```

passive-interface default

如果某些路由器需要 passive 的端口太多，可以设置 passive-interface default。

然后，no passive-interface 打开一些关键接口

```
router ospf 1
 passive-interface default
 no passive-interface Serial0/1
 network 10.1.0.0 0.0.255.255 area 0
```



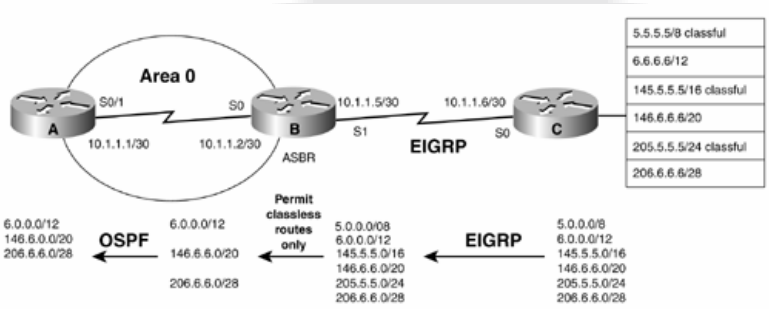
Route Redistribution

redistribute routing-process process-id [tag|metric|metic-type|subnets|route-map]

*routing-process: BGP EGP Connected EIGRP IGRP ISIS ISO-IGRP Mobile ODR OSPF RIP and Static

*ospf-metric: BGP 缺省重分布度量 1 其他协议为 20

*tag-value: 附加到重分布路由的一个 32 位的值，OSPF 没有使用路由标记，但可以在用于指定策略的路由映射中引用，缺省标记为 0



缺省情况下,OSPF 仅重分布有类路由

Redistribution eigrp 1

可以修改外部路由类型 E1 or E2 以及 metric

Redistribution eigrp 1 metric 66 metric-type 1

同时支持无类路由分布到 OSPF 中

Redistribution eigrp 1 metric 66 metric-type 1 subnets

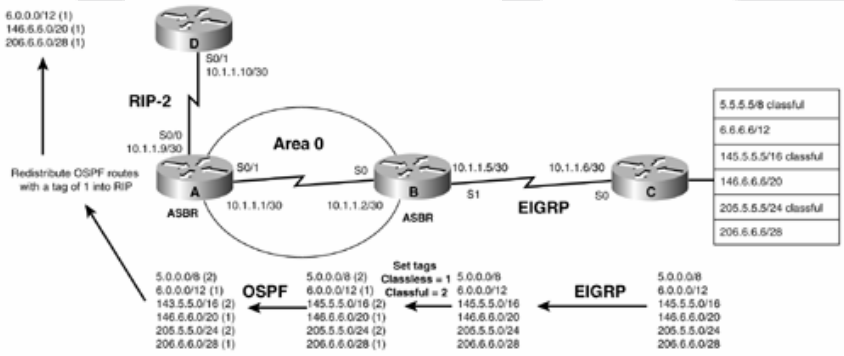
可以通过 tag 标记路由条目

Redistribution eigrp 1 metric 66 metric-type 1 subnets tag 555

利用 route-map 控制重分布

Router B

```
router ospf 1
redistribute eigrp 1 subnets route-map control-eigrp
!
access-list 1 permit 6.0.0.0 0.15.255.255
access-list 1 permit 146.6.0.0 0.0.15.255
access-list 1 permit 206.6.6.0 0.0.0.15
access-list 1 permit 10.1.1.4 0.0.0.3
route-map control-eigrp permit 10
 match ip address 1
```



利用 route-map 控制重分布，并修改 metric 值，并做标记

Router B

```
router ospf 1
 redistribute eigrp 1 subnets route-map control-eigrp
!
```

```
access-list 1 permit 6.0.0.0 0.15.255.255
access-list 1 permit 146.6.0.0 0.0.15.255
access-list 1 permit 206.6.6.0 0.0.0.15
access-list 1 permit 10.1.1.4 0.0.0.3
route-map control-eigrp permit 10
match ip address 1
 set metric 200
 set metric-type   type-1
 set tag 2
!
route-map control-eigrp permit 20
 set metric 100
 set metric-type type-1
 set tag 2
```

如上图，基于标签来控制路由的重分布

```
Router A
router rip
 version 2
 redistribute ospf 1 metric 1 route-map check-tags
 passive-interface Serial0/1
 network 10.0.0.0
 no auto-summary
!
route-map check-tags permit 10
 match tag 1
```

Controlling the OSPF Router ID

Router-id ip-address

手工指定路由器的 router-id，建议配置时手工设置，以增加 RID 的稳定性。CCIE 试验时，手工设置一次，避免扣分

另 ipv6 环境的 RID，仍然是现有的 ipv4 表示方法，但必须手工设置一次

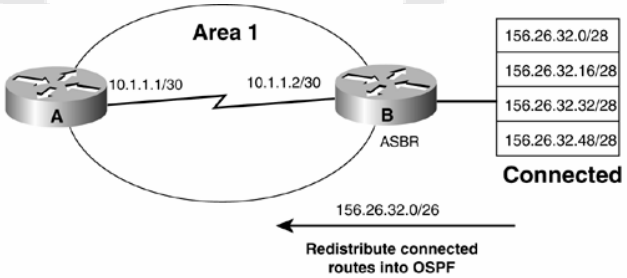
Summarizing External Routes

Summary-address ip-address [advertise | not-advertise]

汇聚路由可以应用到从动态路由选择协议，静态路由和连接路由再次分布的路由上。

只可用在 ASBR 和 ABR 上默认参数为 advertise

not-advertise 关键词阻止汇聚路由被 ABR，ASBR 广播



汇总前

```
O E2     2.2.2.2 [110/20] via 10.1.1.2, 00:03:17, Serial0/1
      156.26.0.0/28 is subnetted, 4 subnets
O E2     156.26.32.32 [110/20] via 10.1.1.2, 00:03:17, Serial0/1
O E2     156.26.32.48 [110/20] via 10.1.1.2, 00:03:17, Serial0/1
O E2     156.26.32.0 [110/20] via 10.1.1.2, 00:02:52, Serial0/1
O E2     156.26.32.16 [110/20] via 10.1.1.2, 00:03:17, Serial0/1
```

汇总

summary-address 156.26.32.0 255.255.255.192

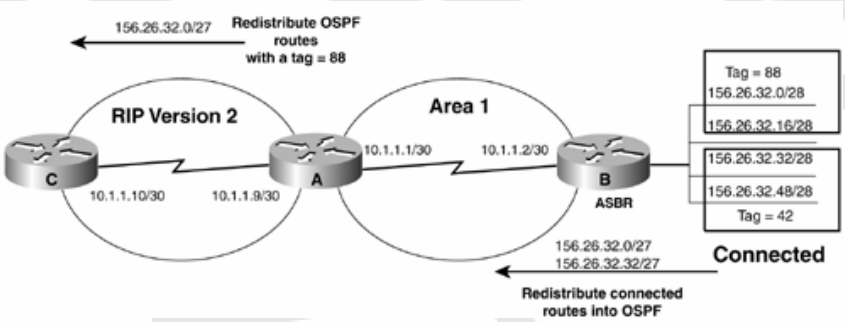
路由表：

```
O E2     2.2.2.2 [110/20] via 10.1.1.2, 00:05:36, Serial0/1
      156.26.0.0/26 is subnetted, 1 subnets
O E2     156.26.32.0 [110/20] via 10.1.1.2, 00:00:21, Serial0/1
```

注意：此命令仅用在 ASBR 上 用于汇总外部路由

Summary-address ip-address tag

用于对汇总路由标记，如下例在 OSPF 重分布到 RIP 时，对特定标记的路由进行重分布




```
Router B
router ospf 1
 summary-address 156.26.32.0 255.255.255.224 tag 88
 summary-address 156.26.32.32 255.255.255.224 tag 42
 network 10.1.1.0 0.0.0.3 area 1

Router A
router rip
 version 2
 redistribute ospf 1 metric 1 route-map checktags
 passive-interface Serial0/1
 network 10.0.0.0
 no auto-summary
!
route-map checktags permit 10
 match tag 88
```

OSPF Timers

timers lsa-group-pacing seconds

用于 LSA 更新，以前仅一个计时器，每 30min，路由器会检查整个 LSDB 并更新每个 LSA。但这样将会导致路由器 CPU 周期性的高负载及周期性的高网络使用率。LSA-group-pacing 就是用于解决这个问题的，路由器将 LSA 分组并指定更新 校验 老化功能的步调。对于一个快速路由器和一个慢速路由器点对点连接以及几个邻居同时向一个路由器发送分组的情况，为了增加效率和减少重传几率可以适当的增加 time， 默认时间为 33ms

```
timers lsa-group-pacing 180
```

```
rtrA#show ip ospf timer lsa-group

            OSPF Router with ID (1.1.1.1) (Process ID 1)

Group size 6, Head 0, Search Index 4, Interval 180 sec
Next update due in 00:00:22
Current time 134571
```

```
                    Index 0 Timestamp 134593
                    Index 1 Timestamp 134777
                    Index 2 Timestamp 134971
                    Index 3 Timestamp 135153
                    Index 4 Timestamp 135351
                    Index 5 Timestamp 135544
```

Timers spf delay interval

用于修改 SPF 计算的延迟和间隔，默认时间为 5s/10s

traffic-share min across-interfaces

用于和 IGRP EIGP 一起是用来支持非等价负载均衡，该命令作为 OSPF 下一个选项出现，但不应用到 OSPF 中

Interface Configuration Commands

ip ospf authentication

在 IOS12.0 以前，如果对 OSPF 一个区域进行认证，该区域的所有接口将配置相同的认证。

该命令允许在接口认证上与区域使用的认证类型不同

```
ip ospf authentication
ip ospf authentication-key laura

ip ospf authentication message-digest
ip ospf message-digest-key 1 md5 laura
ip ospf message-digest-key 2 md5 kaka   //相对于明文认证，MD5 可以更换密钥，同时不中断链路

ip ospf authentication null
```

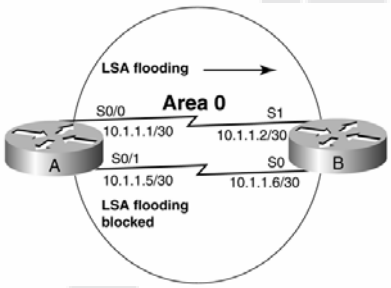
ip ospf cost

修改接口的 OSPF 成本值，但建议使用 autocost reference-bandwidth 来修改参考值

ip ospf database-filter all out

用来阻止接口上的 LSA 扩散。许多 ISP 在 OSPF 邻居将使用冗余的链路。可以使 ISP 在频繁扩散和可靠扩散间进行选择。

```
Router A
interface Serial0/1
 ip ospf database-filter all out
```



ip ospf hello/dead-interval

hello/失效计时器，默认为 hello-interval 的 4 倍。即邻居 down 了以后，发送 4 次 hello packet 无应答，则宣告邻居失效,注意：设置时间总应大于 hello-interval

1. 如果链路上两端 hello/dead interval 不匹配，邻居关系无法建立
2. 如果 dead < hello ，邻居关系会反复震荡

ip ospf demand-circuit

对于某些按流量或者时间收费的线路，此命令允许在拓扑相对稳定的情况下，定期禁用 hello 消息，LSA 的定期更新也不会传至该接口，关闭底层的数据链路层。从而减少流量和使用时间

ip ospf mtu-ignore

如果相邻的 OSPF 路由器的 MTU 单元不匹配，将无法形成邻接关系，使用此命令忽略 mtu 的影响

例如 Cisco Catalyst 3550 的 3 层口和路由器直连时，如果该口不是 SVI，则连接路由器运行 OSPF 将会发生邻接关系震荡， 此 bug 则是 MTU 不匹配造成的.

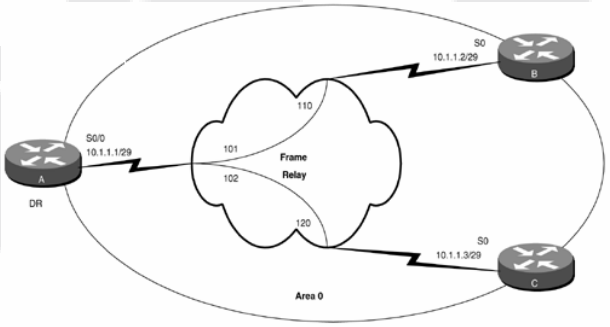
ip ospf network

OSPF 网络分以下 3 种形式

1. A broadcast multi-access network 所有该网络相关的路由器都可以与其他所有路由器直连通信，例如以太网 FDDI tokenring
2. A point-to-point network 这种网络只有 2 台路由器，如 HDLC PPP
3. NBMA 例如 FR x.25 等，所有的路由器都可能与其他有连接，其实质是逻辑上的点到点连接

Broadcast

如右图，对于 FR 线路将网络类型改为 broadcast 后，将建立邻居关系，并选举 DR/BDR。




```
interface Serial0/0
 bandwidth 64
 ip address 10.1.1.1 255.255.255.248
 encapsulation frame-relay
 ip ospf network broadcast
 frame-relay map ip 10.1.1.2 101 broadcast
 frame-relay map ip 10.1.1.3 102 broadcast
 no frame-relay inverse-arp
 frame-relay lmi-type ansi
```

non-Broadcast

OSPF 通过多点传送发送 Hello 包及其他协议包，如果网络不支持多点传送，或者只希望 OSPF 使用单点传送与邻居通信，这可以使用该命令，将网络类型转换为非广播型，必须使用 neighbor 指定邻居

Point-to-multipoint non-Broadcast

用来把一个 NBMA 型网络配置成多点网络。OSPF 使用多点传送在点到多点网络上发送 Hello 包和其他协议包。在一些网络上，如 ATM，连接是动态的，hello 包仅在已经建立的连接上传送

如上图，A 可以设置为 P-MP B,C 设置为 P-P

P-MP hello-interval 30s

P-P hello-interval 10s 故要修改默认的 hello/dead-interval

另一种方案是，将 BC 均改为 P-MP 类型。相对于广播型，不能选举 DR/BDR

Point-to-Point

如上图，如果要配置成 P2P 的网络结构，路由器 A 上将启用子接口，并将网络类型都设置为 P2P,此时 FR 的 Inverse-arp 将自动失效

ip ospf priority

用于在 DR，BDR 选举时，设置端口优先级，1 位最高优先级 0 为不参与 DR/BDR 选举。

注意：非 DR/BDR 最好将其优先级设置为 0，保证 DR/BDR 选择不会出现异常

ip ospf [retransmit-interval | transmit-delay]

LSA 重传间隔及传送延迟，默认时间 5s/1s

Show Commands

Show ip ospf process-id

查看 OSPF 进程的信息，可以使用如下正则表达式

show ip ospf | *begin regular-expression*

show ip ospf | *exclude regular-expression*

show ip ospf | *include regular-expression*

```
r2#show ip ospf
Routing Process "ospf 1" with ID 2.2.2.2
//显示 RID 及路由进程
Supports only single TOS(TOS0) routes
Supports opaque LSA
It is an area border router //路由器类型,可能为
ABR ASBR
SPF schedule delay 5 secs,Hold time between two
SPFs 10 secs
//SPF 保持时间及计算延迟
Minimum LSA interval 5 secs. Minimum LSA arrival
1 secs
Number of external LSA 1. Checksum Sum 0xAB1F
Number of opaque AS LSA 0. Checksum Sum 0x0
Number of DCbitless external and opaque AS LSA 0
Number of DoNotAge external and opaque AS LSA 0
Number of areas in this router is 4. 4 normal 0 stub
0 nssa
//OSPF 区域类型
External flood list length 0
   Area BACKBONE(0) //特定的区域信息分类输出
       Number of interfaces in this area is 3
        //区域内接口个数
       Area has no authentication
        //区域认证类型
       SPF algorithm executed 19 times
        //SPF 计算次数
       Area ranges are
       Number of LSA 15. Checksum Sum 0x7A630
       Number of opaque link LSA 0. Checksum Sum
0x0
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 5
       Flood list length 0
   Area 1
       Number of interfaces in this area is 1

Area has no authentication
       SPF algorithm executed 10 times
       Area ranges are
       Number of LSA 22. Checksum Sum 0xD15F3
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
   Area 2
       Number of interfaces in this area is 1
       Area has no authentication
       SPF algorithm executed 2 times
       Area ranges are
       Number of LSA 12. Checksum Sum 0x7AB94
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
   Area 9
       Number of interfaces in this area is 2
       Area has no authentication
       SPF algorithm executed 3 times
       Area ranges are
          8.8.8.0/27 Active(1) Advertise
     //指出 OSPF 路由是否使用 Area range 汇总
       Number of LSA 12. Checksum Sum 0x6FB7C
       Number of opaque link LSA 0. Checksum Sum 0x0
       Number of DCbitless LSA 0
       Number of indication LSA 0
       Number of DoNotAge LSA 0
       Flood list length 0
```

show ip ospf border-routers

show ip ospf process-id **border-routers**

查询边界路由器的信息

```
r6#show ip ospf border-routers
OSPF Process 1 internal Routing Table
Codes: i - Intra-area route, I - Inter-area route
I 4.4.4.4 [122] via 10.1.1.18, Serial0/1, ASBR, Area 7, SPF 5
i 1.1.1.1 [48] via 10.1.1.18, Serial0/1, ABR, Area 7, SPF 5
```

show ip ospf database

此命令用于查看 LSDB 的信息，针对不同的类型，查询参数较多

其中包含如下参数

show ip ospf *process-id* database
show ip ospf database adv-router *router-id*
show ip ospf *process-id* database adv-router *router-id*
show ip ospf database asbr-summary
show ip ospf *process-id* database asbr-summary
show ip ospf database asbr-summary *asbr-id*
show ip ospf *process-id* database asbr-summary *asbr-id*
show ip ospf database database-summary
show ip ospf *process-id* database database-summary
show ip ospf database external
show ip ospf *process-id* database external
show ip ospf database network
show ip ospf *process-id* database network
show ip ospf database nssa-external
show ip ospf *process-id* database nssa-external
show ip ospf database router
show ip ospf *process-id* database router
show ip ospf database self-originate
show ip ospf *process-id* database self-originate
show ip ospf database summary
show ip ospf *process-id* database summary

例如: show ip ospf database adv-router 1.1.1.1

```
            OSPF Router with ID (6.6.6.6) (Process ID 1)

                    Router Link States (Area 7)

Link ID         ADV Router      Age         Seq#       Checksum Link count
1.1.1.1         1.1.1.1         431         0x80000012 0x2714   2

                    Summary Net Link States (Area 7)

Link ID         ADV Router      Age         Seq#       Checksum
1.1.1.1         1.1.1.1         431         0x80000010 0x29FB
2.2.2.2         1.1.1.1         431         0x80000010 0x7D63
4.4.4.4         1.1.1.1         431         0x80000010 0x8549
8.8.8.0         1.1.1.1         431         0x80000003 0x18E4
10.1.1.0        1.1.1.1         431         0x80000010 0x24BC
10.1.1.4        1.1.1.1         431         0x80000010 0x3785
10.1.1.12       1.1.1.1         431         0x80000010 0xF1A8
//正在广播的路由  RID       这条 LSA 历时     LSA 序列号  检验和
172.16.1.0      1.1.1.1         1198        0x80000013 0x213
172.16.2.0      1.1.1.1         431         0x80000010 0x9888

                    Summary ASB Link States (Area 7)

Link ID         ADV Router      Age         Seq#       Checksum
4.4.4.4         1.1.1.1         431         0x80000010 0x6D61
查询 ASBR LSDB 汇总如下
r6#show ip ospf database asbr-summary

            OSPF Router with ID (6.6.6.6) (Process ID 1)

                Summary ASB Link States (Area 7)

  Routing Bit Set on this LSA
  LS age: 1124
```

```
Options: (No TOS-capability, DC, Upward)
LS Type: Summary Links(AS Boundary Router)
Link State ID: 4.4.4.4 (AS Boundary Router address) //ASBR 的 RID
Advertising Router: 1.1.1.1 //正在广播 LSA 的路由器的 RID
LS Seq Number: 80000010
Checksum: 0x6D61
Length: 28
Network Mask: /0
      TOS: 0  Metric: 74
```

show ip ospf flood-list

扩展参数: **show ip ospf process-id flood-list interface-name interface-number**

```
r4#show ip ospf flood-list

          OSPF Router with ID (4.4.4.4) (Process ID 1)

 Interface Serial0/0, Queue length 0
 Interface Ethernet1/0, Queue length 0
 Interface Ethernet0/0, Queue length 0
 Interface Loopback0, Queue length 0
```

show ip ospf {process-id} **interface** {int-name int-num}

```
r4#show ip ospf 1 interface ethernet 1/0
Ethernet1/0 is up, line protocol is up
  Internet Address 172.16.2.2/24, Area 0
  Process ID 1, Router ID 4.4.4.4, Network Type BROADCAST, Cost: 10
  Transmit Delay is 1 sec, State DR, Priority 1
  Designated Router (ID) 4.4.4.4, Interface address 172.16.2.2
  Backup Designated router (ID) 2.2.2.2, Interface address 172.16.2.1
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:08
  Index 2/2, flood queue length 0
  Next 0x0(0)/0x0(0)
  Last flood scan length is 3, maximum is 3
  Last flood scan time is 0 msec, maximum is 0 msec
  Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 2.2.2.2  (Backup Designated Router)
  Suppress hello for 0 neighbor(s)
```

show ip ospf {process-id} **neighbor detail** {neighbor-id} {int-name int-num}

查看邻居状态，详细信息等，根据路由进程 接口和邻居的 RID 可以分类查询

```
r2#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time   Address         Interface
4.4.4.4           1   FULL/DR         00:00:35    172.16.2.2      Ethernet0/0
3.3.3.3           1   FULL/  -        00:00:38    10.1.1.6        Serial0/1
1.1.1.1           1   FULL/  -        00:00:38    10.1.1.1        Serial0/0

r2#show ip ospf neighbor detail 1.1.1.1
 Neighbor 1.1.1.1, interface address 10.1.1.1
```

```
    In the area 1 via interface Serial0/0
    Neighbor priority is 1, State is FULL, 12 state changes
    DR is 0.0.0.0 BDR is 0.0.0.0
    Options is 0x2
    Dead timer due in 00:00:38
    Neighbor is up for 4d02h
    Index 1/1, retransmission queue length 0, number of retransmission
    First 0x0(0)/0x0(0) Next 0x0(0)/0x0(0)
    Last retransmission scan length is 1, maximum is 1
    Last retransmission scan time is 0 msec, maximum is 0 msec
```

show ip ospf {process-id} **request-list** {neighbor-id} {int-name int-num}

用于显示 OSPF 邻居的请求列表信息

```
r2#show ip ospf request-list 3.3.3.3

            OSPF Router with ID (2.2.2.2) (Process ID 1)

 Neighbor 3.3.3.3, interface Serial0/1 address 10.1.1.6

 Type  LS ID          ADV RTR         Seq NO      Age    Checksum
   1   3.3.3.3        3.3.3.3         0x8000020C  9       0x657
```

show ip ospf {process-id} **retransmission** } {int-name int-num}

用于显示等待再次发送 LSA 的列表

```
r2#show ip ospf retransmission-list serial 0/0

            OSPF Router with ID (2.2.2.2) (Process ID 1)

 Neighbor 1.1.1.1, interface Serial0/0 address 10.1.1.1
Link state retransmission due in 2969 msec, Queue length 2

  Type  LS ID          ADV RTR         Seq NO      Age    Checksum
    1   2.2.2.2        2.2.2.2         0x80000219   0      0xB123
```

show ip ospf {process-id} **summary-address**

用于显示在路由器上已经配置的汇聚路由

```
r4#show ip ospf summary-address

OSPF Process 1, Summary-address

169.254.0.0/255.254.0.0 Metric 10, Type 2, Tag 0
```

show ip ospf {process-id} **virtual-links**

```
r1#show ip ospf virtual-links
Virtual Link OSPF_VL0 to router 2.2.2.2 is up
  Run as demand circuit
  DoNotAge LSA allowed.
```

```
  Transit area 1, via interface Serial0/0, Cost of using 64
  Transmit Delay is 1 sec, State POINT_TO_POINT,
  Timer intervals configured, Hello 10, Dead 40, Wait 40, Retransmit 5
    Hello due in 00:00:01
    Adjacency State FULL (Hello suppressed)
```

debug Commands

debug ip ospf adj

用来显示有关邻居关系的信息

```
r6#debug ip ospf adj

OSPF adjacency events debugging is on

r6#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
r6#
6w3d: OSPF: Interface Loopback0 going Down
6w3d: OSPF: 6.6.6.6 address 6.6.6.6 on Loopback0 is dead, state DOWN
6w3d: OSPF: Interface Serial0/1 going Down
6w3d: OSPF: 6.6.6.6 address 10.1.1.17 on Serial0/1 is dead, state DOWN
6w3d: OSPF: 1.1.1.1 address 10.1.1.18 on Serial0/1 is dead, state DOWN
6w3d: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial0/1 from FULL to DOWN,
Neighbor Down: Interface down or detached
6w3d: OSPF: Interface Loopback0 going Up
6w3d: OSPF: Interface Serial0/1 going Up
6w3d: OSPF: Build router LSA for area 7, router ID 6.6.6.6, seq 0x80000001
6w3d: OSPF: 2 Way Communication to 1.1.1.1 on Serial0/1, state 2WAY
6w3d: OSPF: Send DBD to 1.1.1.1 on Serial0/1 seq 0x12F4 opt 0x42 flag 0x7 len 32
6w3d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x2071 opt 0x2 flag 0x7 len 32
  mtu 1500 state EXSTART
6w3d: OSPF: First DBD and we are not SLAVE
6w3d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x12F4 opt 0x2 flag 0x2 len 27
2  mtu 1500 state EXSTART
6w3d: OSPF: NBR Negotiation Done. We are the MASTER
6w3d: OSPF: Send DBD to 1.1.1.1 on Serial0/1 seq 0x12F5 opt 0x42 flag 0x3 len 52
6w3d: OSPF: Database request to 1.1.1.1
6w3d: OSPF: sent LS REQ packet to 10.1.1.18, length 144
6w3d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x12F5 opt 0x2 flag 0x0 len 32
  mtu 1500 state EXCHANGE
6w3d: OSPF: Send DBD to 1.1.1.1 on Serial0/1 seq 0x12F6 opt 0x42 flag 0x1 len 32
6w3d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x12F6 opt 0x2 flag 0x0 len 32
  mtu 1500 state EXCHANGE
6w3d: OSPF: Exchange Done with 1.1.1.1 on Serial0/1
6w3d: OSPF: Synchronized with 1.1.1.1 on Serial0/1, state FULL
6w3d: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial0/1 from LOADING to FULL,
Loading Done
```

部分错误信息

```
6w3d: OSPF: Rcv pkt from 10.1.1.18, Serial0/1, area 0.0.0.17
```

mismatch area 0.0.0.7 in the header

debug ip ospf events

查询 OSPF 事件，如邻接形成 Hello 包 LSA 扩散 DR 选择及 SPF 计算

```
r6#debug ip ospf events
OSPF events debugging is on
r6#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
r6#
6w4d: OSPF: Rcv hello from 1.1.1.1 area 7 from Serial0/1 10.1.1.18
6w4d: OSPF: 2 Way Communication to 1.1.1.1 on Serial0/1, state 2WAY
6w4d: OSPF: Send DBD to 1.1.1.1 on Serial0/1 seq 0x22BC opt 0x42 flag 0x7 len 32
6w4d: OSPF: End of hello processing

6w4d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x1D6E opt 0x2 flag 0x7 len 32
  mtu 1500 state EXSTART
6w4d: OSPF: First DBD and we are not SLAVE
6w4d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x22BC opt 0x2 flag 0x2 len 27
2  mtu 1500 state EXSTART

6w4d: OSPF: NBR Negotiation Done. We are the MASTER
6w4d: OSPF: Send DBD to 1.1.1.1 on Serial0/1 seq 0x22BD opt 0x42 flag 0x3 len 52
6w4d: OSPF: Database request to 1.1.1.1
6w4d: OSPF: sent LS REQ packet to 10.1.1.18, length 144

6w4d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x22BD opt 0x2 flag 0x0 len 32
  mtu 1500 state EXCHANGE
6w4d: OSPF: Send DBD to 1.1.1.1 on Serial0/1 seq 0x22BE opt 0x42 flag 0x1 len 32
6w4d: OSPF: Rcv DBD from 1.1.1.1 on Serial0/1 seq 0x22BE opt 0x2 flag 0x0 len 32
  mtu 1500 state EXCHANGE

6w4d: OSPF: Exchange Done with 1.1.1.1 on Serial0/1
6w4d: OSPF: Synchronized with 1.1.1.1 on Serial0/1, state FULL
6w4d: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial0/1 from LOADING to FULL,
Loading Done
```

调试一个 OSPF 邻接问题

```
r6#debug ip ospf events
OSPF events debugging is onr6#
6w4d: OSPF: Rcv hello from 1.1.1.1 area 7 from Serial0/1 10.1.1.18
6w4d: OSPF: Mismatched hello parameters from 10.1.1.18
6w4d: OSPF: Dead R 40 C 123, Hello R 10 C 10  //R 代表收到的参数  C 代表本地参数
```

debug ip ospf {process-id} flood

查看 OSPF 扩散协议所产生的事件

```
r6#debug ip ospf flood
OSPF flooding debugging is on
r6#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
```

```
r6#
6w4d: Inc retrans unit nbr count index 1 (0/1) to 1/1
6w4d: Set Nbr 1.1.1.1 1 first flood info from 0 (0) to 623AA2D8 (300)
6w4d: Init Nbr 1.1.1.1 1 next flood info to 623AA2D8
6w4d: OSPF: Add Type 1 LSA ID 6.6.6.6 Adv rtr 6.6.6.6 Seq 80000125 to Serial0/1
1.1.1.1 retransmission list
6w4d: OSPF: Start Serial0/1 1.1.1.1 retrans timer
6w4d: Set idb next flood info from 0 (0) to 623AA2D8 (300)
6w4d: OSPF: Add Type 1 LSA ID 6.6.6.6 Adv rtr 6.6.6.6 Seq 80000125 to Serial0/1
flood list
6w4d: OSPF: Start Serial0/1 pacing timer
6w4d: OSPF: Flooding update on Serial0/1 to 224.0.0.5 Area 7
... (many more pages of output!!!)
```

由于此命令将会产生大量 log，所以可以有选择性的输出

```
access-list 1 permit 172.16.1.0 0.0.0.255
r6#debug ip ospf flood 1
OSPF flooding debugging is on for access list 1
r6#clear ip ospf process
Reset ALL OSPF processes? [no]: yes
r6#
6w4d: Inc retrans unit nbr count index 1 (0/1) to 1/1
6w4d: Set Nbr 1.1.1.1 1 first flood info from 0 (0) to 623AA408 (303)
6w4d: Init Nbr 1.1.1.1 1 next flood info to 623AA408
6w4d: OSPF: Start Serial0/1 1.1.1.1 retrans timer
6w4d: Set idb next flood info from 0 (0) to 623AA408 (303)
6w4d: OSPF: Start Serial0/1 pacing timer
6w4d: Create retrans unit 0x623B71A8/0x623C5DE8 1 (0/1) 1
6w4d: OSPF: Set nbr 1 (0/1) retrans to 4504 count to 1
6w4d: Set idb next flood info from 623AA408 (303) to 0 (0)
6w4d: OSPF: Stop Serial0/1 flood timer
6w4d: %OSPF-5-ADJCHG: Process 1, Nbr 1.1.1.1 on Serial0/1 from FULL to DOWN,
Neighbor Down: Interface down or detached
6w4d: Dec retrans unit nbr count index 1 (0/1) to 0/0
6w4d: Free nbr retrans unit 0x623B71A8/0x623C5DE8 0 total 0. Also Free nbr
retrans block
6w4d: Set Nbr 1.1.1.1 1 first flood info from 623AA408 (303) to 0 (0)
... (many more pages of output!!!)
```

debug ip ospf lsa-generation ip-access-list-number

用来显示和摘要 LSA 的发生和扩散有关的信息

```
r4#debug ip ospf lsa-generation
OSPF summary lsa generation debugging is on

6w4d: OSPF: Start redist-scanning
6w4d: OSPF: Scan the RIB for both redistribution and translation
6w4d: OSPF: net 169.254.0.0 up, new metric decreases: old 16777215, new 10
6w4d: OSPF: Generate external LSA 169.254.0.0, mask 255.254.0.0, type 5, age 0,
metric 10, tag 0, metric-type 2, seq 0x80000001
6w4d: OSPF: generate external LSA for summary 169.254.0.0 255.254.0.0, metric 10
```

debug ip ospf packet

```
r6#debug ip ospf packet
OSPF packet debugging is on
r6#
6w4d: OSPF: rcv. v:2 t:1 l:48 rid:1.1.1.1
      aid:0.0.0.7 chk:0 aut:2 keyid:8 seq:0xC from Serial0/1
```

v: 版本号 t:包类型 1-hello 2- Database Description 3-LSR 4-LSU 5-LSAck

l: 用字节表示包长度 rid: router-id aid: area-id aut 0-null 1-text 2-md5 keyid: md5-key

debug ip ospf retransmission

用于显示 OSPF-LSA 的重发事件

debug ip ospf spf [inter | intra | external] acl-num

用于显示 SPF 算法的计算

```
r4#debug ip ospf spf external
OSPF spf external events debugging is on
r6#
6w4d: OSPF: Started Building Type 5 External Routes
6w4d: OSPF: Start processing Type 5 External LSA 169.254.0.0, mask 255.254.0.0,
adv 4.4.4.4, age 39, seq 0x80000095, metric 10, metric-type 2
6w4d: OSPF: Did not find route to ASBR 4.4.4.4
6w4d: OSPF: ex_delete_old_routes
6w4d: OSPF: Started Building Type 7 External Routes
6w4d: OSPF: ex_delete_old_routes
6w4d: OSPF: Started Building Type 5 External Routes
6w4d: OSPF: Start processing Type 5 External LSA 169.254.0.0, mask 255.254.0.0,
adv 4.4.4.4, age 49, seq 0x80000095, metric 10, metric-type 2
6w4d:    Add better path to LSA ID 169.254.0.0, gateway 10.1.1.18, dist 10
6w4d:    Add path: next-hop 10.1.1.18, interface Serial0/1
6w4d:    Add External Route to 169.254.0.0. Metric: 10, Next Hop: 10.1.1.18
6w4d: OSPF: insert route list LS ID 169.254.0.0, type 5, adv rtr 4.4.4.4
6w4d: OSPF: ex_delete_old_routes
6w4d: OSPF: Started Building Type 7 External Routes
6w4d: OSPF: ex_delete_old_routes
```

clear Commands

clear ip ospf {process-id} **couters** {neighbor neighbor-id} {int-name int-number}

用于重新设置邻居状态变换计数器

clear ip ospf process

用于重新启动 OSPF 进程

clear ip ospf redistribution

用于清除重分布到 OSPF 的路由

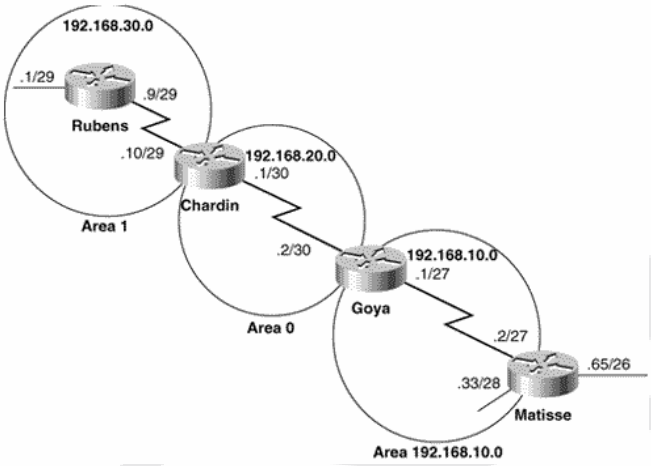
Configuring OSPF

Case A: basic OSPF Configuration

配置 OSPF 的 3 个基本步骤:

1. 确保每隔路由器接口相连的区域
2. 使用 router ospf process-id 启动 OSPF
3. 使用 network addr area area-id 来定义参与 OSPF 的网络

process-id 为本地标识符,仅用于标示本地 OSPF 进程



Rubens(config)#router ospf 10

Rubens(config-router)#network 0.0.0.0 255.255.255.255 area 1

最简略的指定参与OSPF进程的方法

Chardin(config)#router ospf 20

Chardin(config-router)#network 192.168.30.0 0.0.0.255 area 1

Chardin(config-router)#network 192.168.20.0 0.0.0.255 area 0

采用Inverse Mask(反掩码)来指定网络

Goya(config)#router ospf 30

Goya(config-router)#network 192.168.20.0 0.0.0.3 area 0.0.0.0

Goya(config-router)#network 192.168.10.0 0.0.0.31 area 192.168.10.0

使用点分十进制的方式指定区域

Matisse(config)#router ospf 40

Matisse(config-router)#network 192.168.10.2 0.0.0.0 area 192.168.10.0

Matisse(config-router)#network 192.168.10.33 0.0.0.0 area 192.168.10.0

Case B: Setting Router IDs with Loopback Interfaces

```
Cisco Internetwork Operating System Software
IOS (tm) 2500 Software (C2500-J-L), Version 11.2(7a), RELEASE SOFTWARE (fc1)
Copyright (c) 1986-1997 by cisco Systems, Inc.
Compiled Tue 01-Jul-97 15:31 by kuong
Image text-base: 0x0303E1EC, data-base: 0x00001000

cisco 2509 (68030) processor (revision C) with 16384K/2048K bytes of memory.
Processor board ID 01210416, with hardware revision 00000000
Bridging software.
SuperLAT software copyright 1990 by Meridian Technology Corp).
X.25 software, Version 2.0, NET2, BFE and GOSIP compliant.
TN3270 Emulation software.
1 Ethernet/IEEE 802.3 interface(s)
2 Serial network interface(s)
32K bytes of non-volatile configuration memory.
8192K bytes of processor board System flash (Read ONLY)

OSPF: Could not allocate router id
 network 192.168.10.2 0.0.0.0 area 192.168.10.0
    ^
% Invalid input detected at '^' marker.

 network 192.168.10.334 0.0.0.0 area 192.168.10.0
    ^
% Invalid input detected at '^' marker.


Press RETURN to get started!
```

如果一个 OSPF 进程找不到有效的 IP 地址作为 RID，则可以启用 Loopback 接口

Router<config>#int lo 0

Router<config-if>#ip addr 192.168.1.1 255.255.255.0

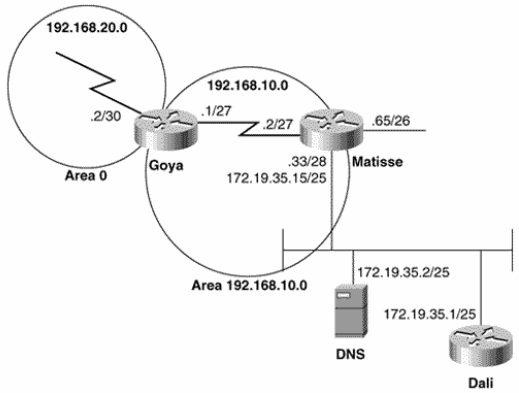
Case C: Domain name Service Lookups

ip ospf name-lookup

通过开启 DNS 服务来识别路由器

```
Goya#show ip ospf neighbor

Neighbor ID     Pri  State     Dead Time    Address         Interface
chardin           1  FULL/ -   00:00:38     192.168.20.1    Serial0
matisse           1  FULL/ -   00:00:36     192.168.10.2    Serial1
Goya#show ip ospf database

      OSPF Router with ID (192.168.50.3) (Process ID 30)

                  Router Link States (Area 0.0.0.0)

Link ID           ADV Router     Age      Seq#       Checksum    Link count
192.168.50.2      chardin        151      0x80000097  0x1B3F     2
192.168.50.3      goya           1568     0x8000000C  0x2A1C     3

                  Summary Net Link States (Area 0.0.0.0)

Link ID           ADV Router     Age      Seq#       Checksum
192.168.10.0      goya           1568     0x80000009  0xA35E
192.168.10.33     goya           1568     0x80000009  0x1DA3
192.168.30.1      chardin        1058     0x80000009  0x6984
192.168.30.8      chardin        1059      0x80000009 0xEEFF
```

Case D: OSPF and Secondary Addresses



为路由器 Matisse 的 E0 口配置一个辅助地址,并运行 OSPF

```
interface Ethernet0
ip address 172.19.35.15 255.255.255.128 secondary
ip address 192.168.10.33 255.255.255.240
!
router ospf 40
network 192.168.10.2 0.0.0.0 area 192.168.10.0
network 192.168.10.33 0.0.0.0 area 192.168.10.0
network 172.19.35.15 0.0.0.0 area 192.168.10.0
```

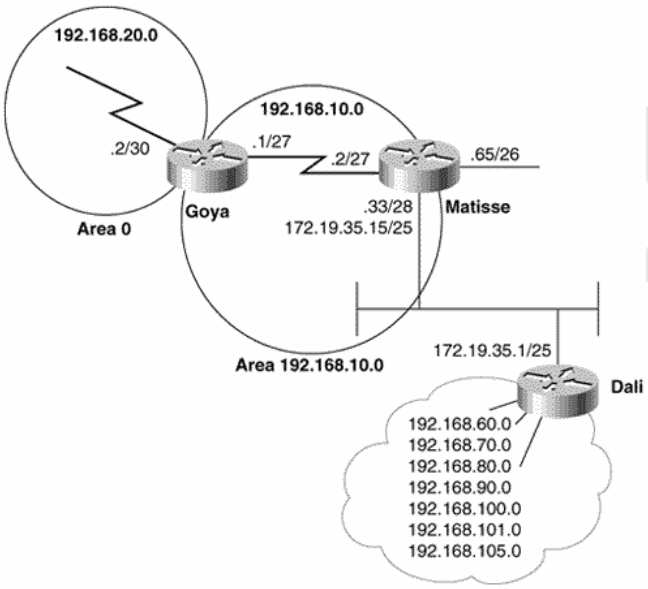
辅助地址是和 DNS 服务器以及路由器 Dali 位于同一个子网,根据这个配置,路由器 Matisse 会像它的邻居宣告子网 172.19.35.0/25,但是如果主地址 192.168.10.33 的 network 语句被删除后,这个辅助网络 172.19.35.0/25 将不会被宣告出去

由于路由器 Matisse 是通过辅助地址和网络 172.19.35.0/25 相连的,因此它不会和 172.19.35.0/25 的路由器建立邻接关系.并且如果有要到达 DNS 服务器的 packet,它将经过路由器 Matisse 的 E0 口转发,由于 DNS 服务器必须发送一些应答给不属于它所在的那些网络,它将把这些应答的 packet 发送给作为 DNS 服务器默认网关的路由器 Dali.但是路由器 Dali 没有和路由器 Matisse 交换了路由信息,所以路由器 Dali 不知道如何到达 OSPF 路由域.解决方法是在路由器 Dali 上配置一条指向 OSPF 路由域的静态路由:

Dali(config)#ip route 192.168.0.0 255.255.0.0 172.19.35.15

由于静态路由的基于无类的,它可以使用超网来匹配到达 OSPF 路由域的所有目标地址.并且在这个例子中,路由器 Matisse 并不是 ASBR,它只发送到外部的路由信息,并不接受外部路由信息,因此它也没有产生任何类型 5 的 LSA(AS External LSA)

Case E: Stub Areas



可以把区域1设置为stub area,可以在OSPF进程里加入area <area-id> stub命令把区域设置为stub area.配置如下:

```
Rubens(config)#router ospf 10
Rubens(config-router)#network 0.0.0.0 255.255.255.255 area 1
Rubens(config-router)#area 1 stub

Chardin(config)#router ospf 10
Chardin(config-router)#network 192.168.30.0 0.0.0.255 area 1
Chardin(config-router)#network 192.168.20.0 0.0.0.255 area 0
Chardin(config-router)#area 1 stub
```

区域1被配置成stub area之前路由器Rubens的LSDB中LSA条目为14条

```
Rubens#show ip ospf database database-summary

            OSPF Router with ID (192.168.50.1) (Process ID 10)

Area ID         Router    Network  Sum-Net  Sum-ASBR   Subtotal   Delete  Maxage
1               2         0        4        1          7          0       0
AS External                                           7          0       0
Total           2         0        4        1          14
Rubens#
```

区域1被配置成stub area之后路由器Rubens的LSDB中LSA条目为7条

```
   Rubens#show ip ospf database database-summary

               OSPF Router with ID (192.168.50.1) (Process ID 10)

   Area ID         Router    Network  Sum-Net  Sum-ASBR   Subtotal   Delete  Maxage
   1               2          0       5        0          7          0       0
   AS External                                           0          0       0
   Total           2         0        5        0          7
   Rubens#
```

当 stub area 和 ABR 相连,ABR 将通过 Type3 LSA(Network Summary LSA)向 stub area 宣告一条默认路由

```
   O IA    172.19.35.0 [110/202] via 192.168.30.10, 00:05:43, Serial1
   O*IA 0.0.0.0/0 [110/65] via 192.168.30.10, 00:05:44, Serial1
   Rubens#
```

Case F:Totally-stub Areas

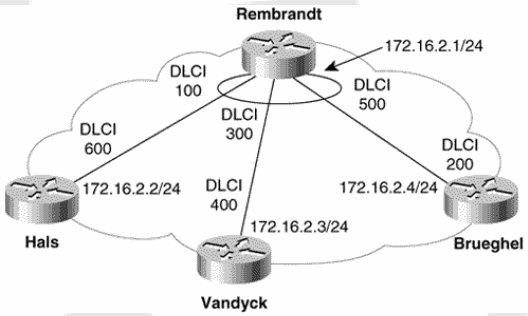
只需在 area stub 后挂接 no-summary,仅在 ABR 上执行，此时将过滤掉域间路由，只剩下域内路由和默认路由

```
Rubens#show ip ospf database database-summary

            OSPF Router with ID (192.168.50.1) (Process ID 10)

Area ID  Router    Network  Sum-Net  Sum-ASBR  Subtotal  Delete   Maxage
1        2         0        1        0         3         0        0
AS External                                   0         0        0
Total    2         0        1        0         3
Rubens#
```

```
     192.168.50.0/24 is subnetted, 1 subnets
C       192.168.50.1 is directly connected, Loopback1
     192.168.30.0/24 is subnetted, 2 subnets
C       192.168.30.0 is directly connected, Ethernet0
C       192.168.30.8 is directly connected, Serial1
O*IA 0.0.0.0/0 [110/65] via 192.168.30.10, 00:03:33, Serial1
Rubens#
```

Case G: OSPF on NBMA networks

在NBMA网络比如ATM,X.25和帧中继等,需要人工指定邻居.在IOS版本10.0之前,可以使用命令neighbor <interface ip address>来指定邻居.拓扑图如下:




```
Rembrandt
interface Serial0
   encapsulation frame-relay
   ip address 172.16.2.1 255.255.255.0
   frame-relay map ip 172.16.2.2 100
   frame-realy map ip 172.16.2.3 300
   frame-relay map ip 172.16.2.4 500
    !
router ospf 1
   network 172.16.0.0 0.0.255.255 area 0
   neighbor 172.16.2.2
   neighbor 172.16.2.3
   neighbor 172.16.2.4

Hals
interface Serial0
   encapsulation frame-relay
   ip address 172.16.2.2 255.255.255.0
   frame-relay map ip 172.16.2.1 600
   frame-relay map ip 172.16.2.3 600
   frame-relay map ip 172.16.2.4 600
!
```

```
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.1 priority 10
//这里加了一个参数,priority 10,意思是同时指定邻居的优先级为 10,默认为 0(即不参与 DR/BDR 选举)因此刚才路由器 Rembrandt 在指定邻居的同时分别指定它们的优先级都为 0(意图明显,让路由器 Rembrandt 作为 DR)
Vandyck
interface Serial0
 encapsulation frame-relay
 ip address 172.16.2.3 255.255.255.0
 frame-relay map ip 172.16.2.1 400
 frame-relay map ip 172.16.2.2 400
 frame-relay map ip 172.16.2.4 400
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.1 priority 10
//在这里别忘记指定邻居的优先级
Brueghel
interface Serial0
 encapsulation frame-relay
 ip address 172.16.2.4 255.255.255.0
 frame-relay map ip 172.16.2.1 200
 frame-relay map ip 172.16.2.2 200
 frame-relay map ip 172.16.2.3 200
!
router ospf 1
 network 172.16.0.0 0.0.255.255 area 0
 neighbor 172.16.2.1 priority 10
```

1. neighbor 命令只是用在老版本的 IOS(10.0 之前)上,在之后的 IOS 版本里,可以在接口配置模式下使用 ip ospf network 命令改变默认的 OSPF 网络类型
2. 采用全互连的 NBMA 上 DR/BDR 的选举是最有利的,但是这样的成本太高;另 1 种解决方法是把网络类型改为点到多点,来避免 DR/BDR 的选举问题.点到多点链路把 PVC 当作点到点链路的集合.
3. 但是这种点到多点的配置,在动态连接的网络比如帧中继 SVC,ATM SVC 等出现问题.在 IOS 版本 11.3AA 之后,可以使用在 ip ospf network point-to-multipoint 之后加个参数 non-broadcast 来解决.但是仍然要人工指定邻居
4. 在 IOS 版本 11.3AA 之后,多了个特性,就是在指定邻居的同时还可以基于每条 VC 来指定开销
5. 最后一种解决方案是,把每条 PVC 当作 1 条点到点链路,通过划分子接口的方式完成,这样也消除了 DR/BDR 的选举带来的问题

关于 **NBMA** 做如下补充:

网络类型	DR/BDR选举	hello interval	dead interval
broadcast	Y	10s	40s
P-P	N	10s	40s
P-MP	N	30s	120s
NBMA	Y	30s	120s
P-MP(非广播)	N	30s	120s

在配置 **NBMA Broadcast** 型时注意:

1. 更改 OSPF 接口优先级方式确保中心路由器为 DR
2. 更改接口类型为广播型
3. 为了保证端到端 ping 通，边缘路由器没有直连的 PVC，故在边缘路由器上除了要做对中心路由器 IP 地址到本

地 DLCI 的映射，还要做邻居边缘路由器 IP 到本地 DLCI 的映射，让中心路由器引导流量

在配置 **NBMA P-P** 型时注意：

1. 中心路由器要启用子接口，子接口之间出于不同子网，映射子接口到本地 DLCI
2. 在边缘路由器上把接口类型更改为 P-P，并映射同一子网下中心路由器子接口的 IP 地址到本地 DLCI

在配置 **NBMA non-broadcast** 型时注意：

1. 更改 OSPF 接口优先级方式确保中心路由器为 DR
2. 为了保证端到端 ping 通，边缘路由器没有直连的 PVC，故在边缘路由器上除了要做对中心路由器 IP 地址到本地 DLCI 的映射，还要做邻居边缘路由器 IP 到本地 DLCI 的映射，让中心路由器引导流量

在配置 **NBMA P-MP** 型时注意：

1. 中心路由器类型改为 P-MP
2. 边缘路由器如果做为 P-P，只需设置一条中心路由器 IP 到本地 DLCI 的映射，并且 hello_interval 改为 30s
3. 边缘路由器作为 P-MP，需要将邻居路由器和中心路由器的 IP 映射到本地 DLCI
4. 凡是 P-MP 的接口，都会宣告一条该接口的 32 位主机路由给邻居

Case H：OSPF over Demand Circuits

通过在按需电路相连的接口上增加 ip ospf demand-circuits。在按需电路上实现 OSPF 需要注意：

1. 只有在区域的 LSDB 中的所有 LSA 设置了 DC-bit 后，设置 DoNotAge 位的 LSA 才被允许进入该区域
2. 实现按需电路上的 OSPF 区域内的说有路由器都必须具有支持这种配置方法的能力
3. 如果运行在按需电路上的 OSPF 是在一个非末节区域实现的，那么所有的非末节区域必须支持这种方式 因为 DC-bit 是在 type-5LSA 中实现的，Type-5 LSA 可以泛洪到所有的非末节区域
4. 应该尽量限制在末节，完全末节，NSSA 区域上实现 DC，因为这样可以限制 LSA 的数量，亦可取消 OSPF 域内的所有路由器支持 DC
5. 如果配置的 DC 上有虚电路穿过，此虚电路亦被认为是按需电路
6. 每 30min 刷新 LSA。但由于设置了 donotAge 的 LSA 在穿越按需电路时不需要刷新，因此丧失了 OSPF 的稳定特性
7. 重新刷新过程可以在一条按需电路两端外的其他所有接口上发生。结果导致链路 2 端的 LSA 序列号可能会是不同的

Design Solutions

推荐的 OSPF 网络规模如下表

参数	Min	Aver	Max
每个域内路由器数目	1	500	1000
每个区域的路由器数目	1	100	350
每个域内区域数目	1	25	75
每个路由器邻居数目	1	50	100
每个路由器的区域数目	1	3	5

ABR 的区域数目：每个 ABR 有 2 个区域是最优的

DR/BDR 选择：可以使用 show process cpu/memory 检查

尽量选择负载低，且链路稳定的路由器为了网络稳定，最好设置端口优先级，避免路由进程重启后导致的 DR/BDR 改变

NBMA 网络： 部分互联网络要比全互联好很多，在一些情况下仔细设计的 P-P 或者 P-MP 网络要比多点网络工作起来好很多

LSDB 大小： LSDB 的大小对于路由器是否稳定工作十分重要

*路由选择表的每一项需要消耗 200~280 字节，加上每条链路消耗 44 字节

*每个 LSA 消耗 100 字节，加上 LSA 实际长度，大概增加 60~100 字节

因此通常要给少于 500KB 的路由选择表 2~16MB 的 RAM。大于 500KB 大型网络需要 16~64MB 的储存器

检测注意： show memory 最好不要在工作日时间做，在 show memory 之前最好备份路由器配置

保持较小 OSPF 区域并使用路由汇总将会急剧减少 CPU 使用

OSPF 区域设计概述：

1. 优先考虑物理的临近
2. 如果链路不稳定则减小区域规模
3. 确保连续的区域，尽量避免不必要的虚链路
4. 调整 hello-interval/dead-interval/retransmission-interval/transmit-delay 来降低链路使用率
5. 区域命名，点分 10 进制和普通 10 进制 2 种。大型多区域最好采用点分 10 进制
6. BB 区域必须连续，否则会对网络稳定性产生影响，当然非连续也可以通过 VL 来修补
7. E1/E2 路由：通常网络有多个从 OSPF AS 到相同外部网络的接入点，使用 E1 路由可以让路由器更好的选择到外部的最短路径，同时大规模网络亦使用 E1。当网络规模较小时使用 E2
8. 尽量把 ABR 的 loopback 口放入 area 0

OSPF Redistribution：

1. 尽可能将功能不够强大的路由协议重分布到相对功能强大的路由协议中， 例如 RIP 重分布进 OSPF OSPF 重分布进 BGP。但 BGP 重分布进 OSPF 将是一个非常糟糕的做法
2. 尽可能在一个点上重分布。多点重分布容易产生环路，例如 CCIE_lab 160 的路由环路
3. 使用路由过滤器，尽量过滤无用路由信息
4. 必要时进行重分布，而不是尽可能重分布
5. 调整管理距离和路由选择度量标准
6. 一直都要设置度量值。
7. 对部分路由信息打标，通过 route-map 过滤分类是一个不错的做法

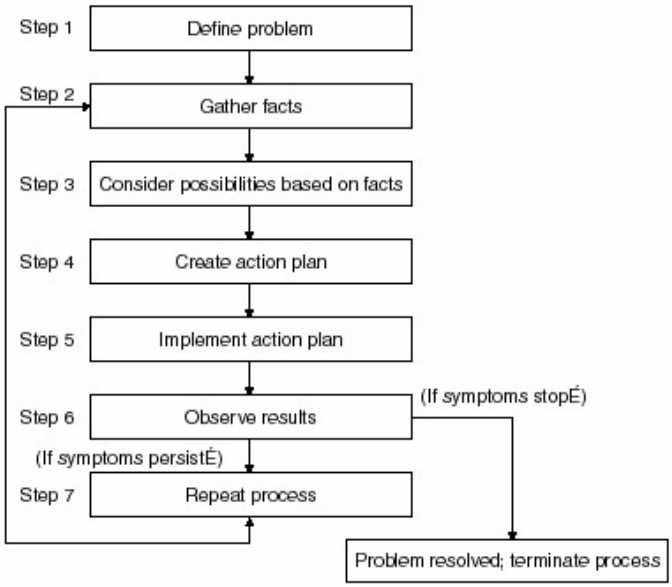
OSPF Summary-address：

1. IP 编址时尽量考虑到子网的规则分配
2. IP 编址尽量考虑到以后用户的增长
3. 汇总仅在 ABR，ASBR 进行
4. 如果网络中有 Classful 的路由协议，则必须在边界上进行汇总。

Troubleshooting OSPF

Troubleshooting 常用步骤：

1. 清楚地定义问题
2. 收集信息，例如受影响的用户，网络管理员，改变的配置，NMS 报告内容，路由器 debug 命令，Cisco 设计缺陷等
3. 思考可能的问题
4. 制定一个行动计划
5. 执行计划
6. 收集结果
7. 重新执行此过程



Debug 命令使用：

参见前文 command and configurations

OSPF 协议故障排除方法如下

1. 一般故障是由路由信息丢失，错误的或者不精确的路由信息引起的。
2. show ip ospf database 查看 LSA，如果一条链路是不稳定的，则通告将十分频繁，故其序列号明显高于其他 LSA
3. 检查 network area 语句是否把所有接口都指定到正确的区域，wild-netmask 是否正确

4. 检查邻接关系时，考虑 hello 报文是否在发送？两端计时器是否相同，可选性能字段是否相同，接口是否在同一子网邻居是否同一种网络类型
5. 认证类型是否有问题？
6. 虚链路是否正确设置？
7. 如果怀疑 LSDB 有问题，可以 show ip ospf database 查看同区域内每台路由器的 LSA 数量及类型
8. 分区域检查，如果 stub NSSA 区域，每台路由器都将申明 area stub or area nssa
9. 地址汇总配置是否正确
10. 防止重分布产生的路由环路

OSPF协议常见故障：

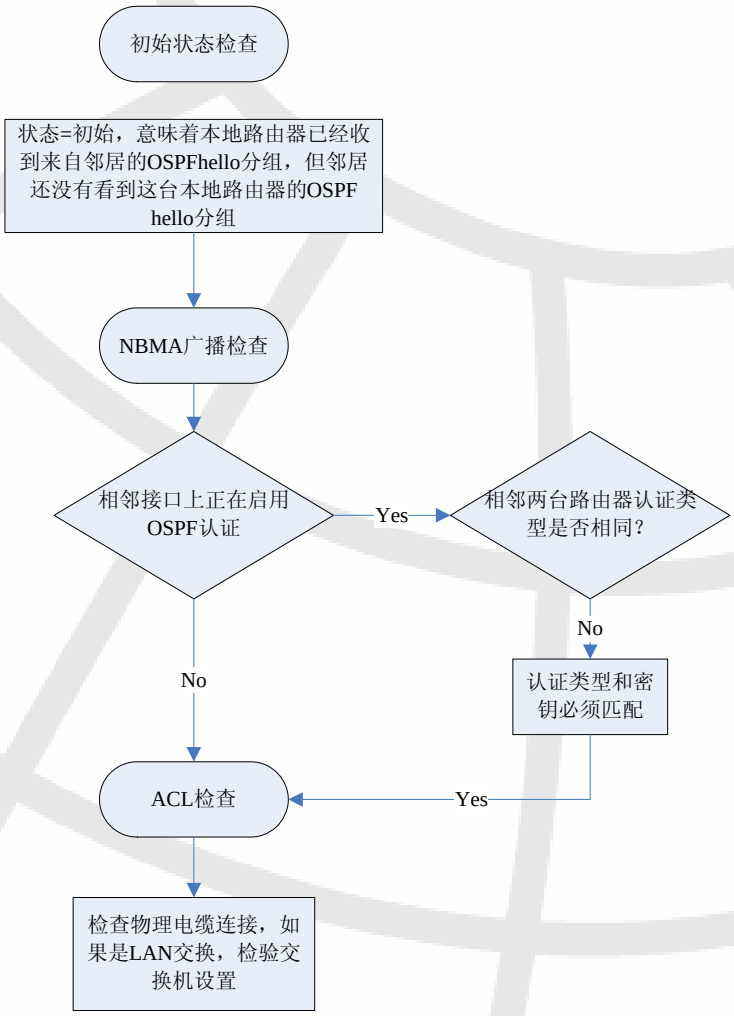
邻居及邻接关系出现问题大致情况如下

1. OSPF 没有在接口上启用，或者一条网络路由器配置命令错误或丢失
2. 不匹配的 Hello 或者计时器，E 位，区域 ID，认证类型或者网络掩码
3. 访问列表错误配置，可能正在阻塞 OSPF hello 包
4. 虚拟链路和末节设置不匹配
5. 接口被定义为 passive-mode
6. 不匹配的认证类型
7. 不匹配的认证密钥
8. 第 2 层停机
9. 没有在 NBMA 上定义网络类型
10. FR 或 dialer 缺少 broadcast 关键字

OSPF 停滞在 INIT 状态

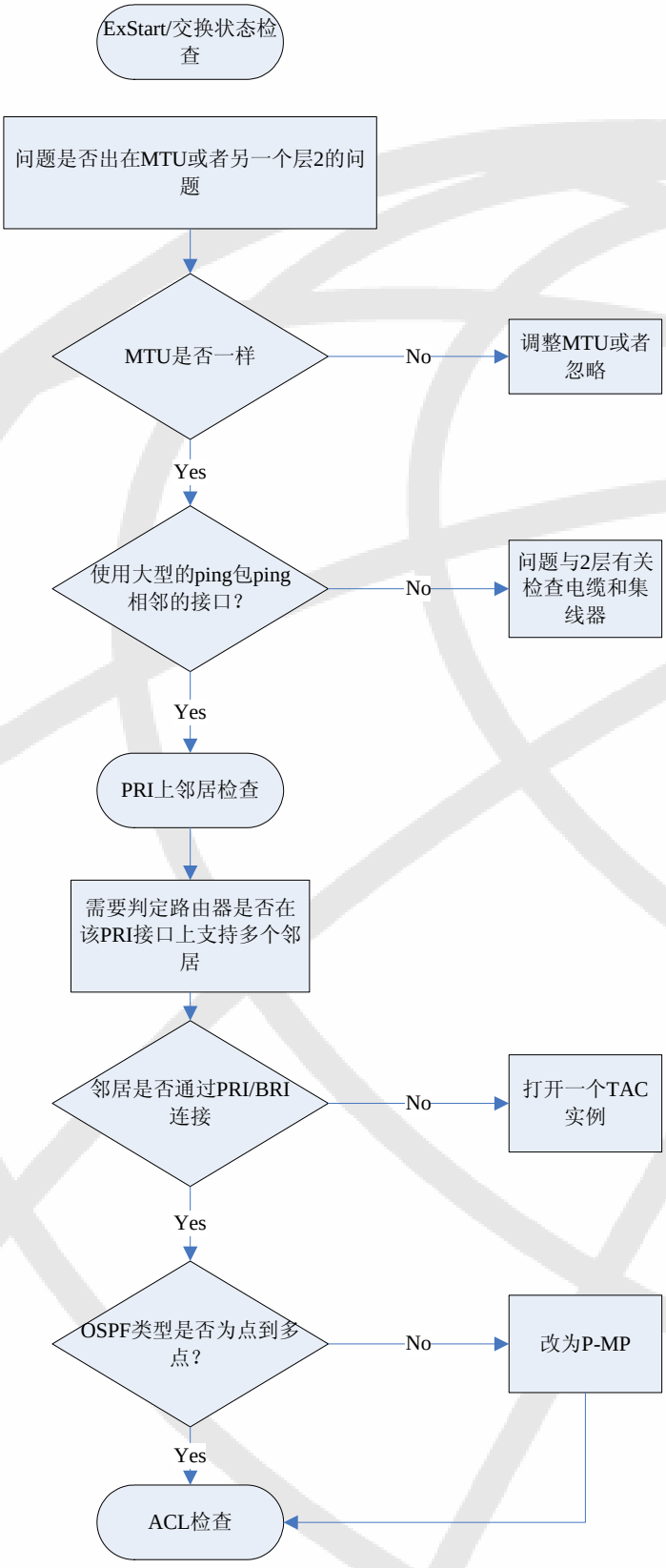
检查方法如右图：

1. 一端正在阻塞 hello 分组
2. 一端 NAT 转换了 OSPF hello 包
3. 一端的组播能力被破坏
4. 肯定是一个 2 层的问题
5. Dialer 或 FR 缺少 broadcast
6. 路由器发送 hello，但没有收到响应
7. 邻居 hello 在 NBMA 云团中消失
8. ACL 或者某些 2 层原因拒绝了 hello 包



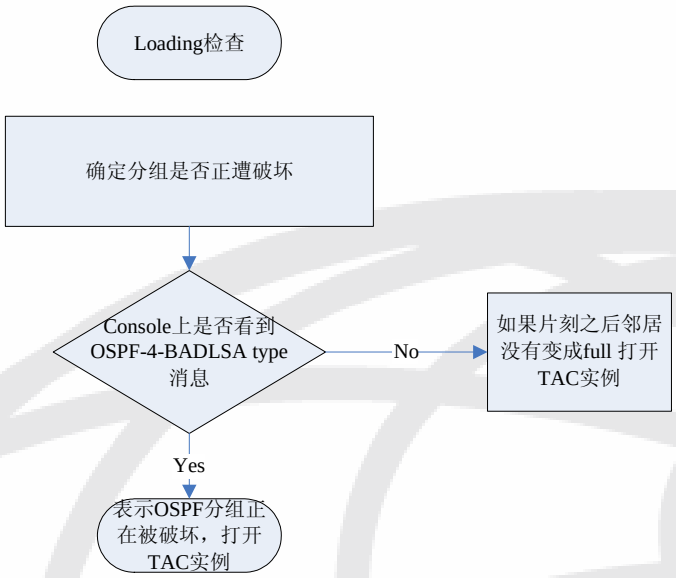
OSPF 停滞在 Exstart/Switching 状态

检查方法如下图：



1. MTU 不匹配
2. 邻居 RID 是一致的
3. 单播被破坏
   a. 在一个高度冗余的网络中，Frame/ATM 环境中错误的 VC/DLCI 映射
   b. MTU 的问题，不能使用超过某长度的分组
   c. ACL 阻塞单播
   d. NAT 正在转换单播分组
4. 在 PRI 和 BRI/dialer 和网络类型之间是点到点的

OSPF 停滞在 Loading 状态



a. LSR 产生，但是邻居正在发送一个坏的分组或者被破坏的内存

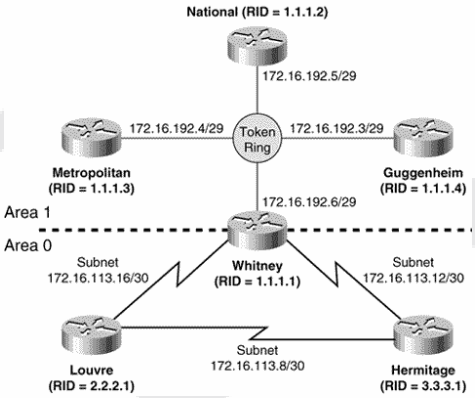
-show ip ospf request-list 查看坏的 LSA

-show loggin 查看 OSPF-4-BASLSATYPE 消息

b. LSR 产生，但邻居正在忽略这个请求

c. MTU 不匹配，但旧的 IOS 没有发现它

Case Study: An Isolated Area



现在这个环境里，区域 1 里的数据包的路由的正常的，但是区域间的通信却失败了

首先想到的是，区域间通信失败，问题可能出现在 ABR 上

检查 ABR：

```
National#show ip ospf neighbor

Neighbor ID     Pri   State           Dead Time       Address         Interface
1.1.1.1           1   FULL/DR         00:00:33        172.16.192.6    TokenRing0
1.1.1.3           1   FULL/BDR        00:00:34        172.16.192.4    TokenRing0
1.1.1.4           1   FULL/  -        00:00:30        172.16.192.3    TokenRing0
National#
```

发现区域 0 的邻接关系没有建立，查询 LSDB

```
National#show ip ospf database

       OSPF Router with ID (1.1.1.2) (Process ID 8)

                Router Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum Link count
172.16.192.6    1.1.1.1         132         0x80000034 0xAC4D   3
172.16.219.120  1.1.1.2 1       458         0x80000028 0x6B46   2

                Net Link States (Area 1)

Link ID         ADV Router      Age         Seq#       Checksum
172.16.192.6    1.1.1.1         132         0x8000002E 0x2078
National#
```

从图可以看出有邻接，但是没有通告区域间的目的地址

检查 OSPF 配置，发现 network area 没有精细配置

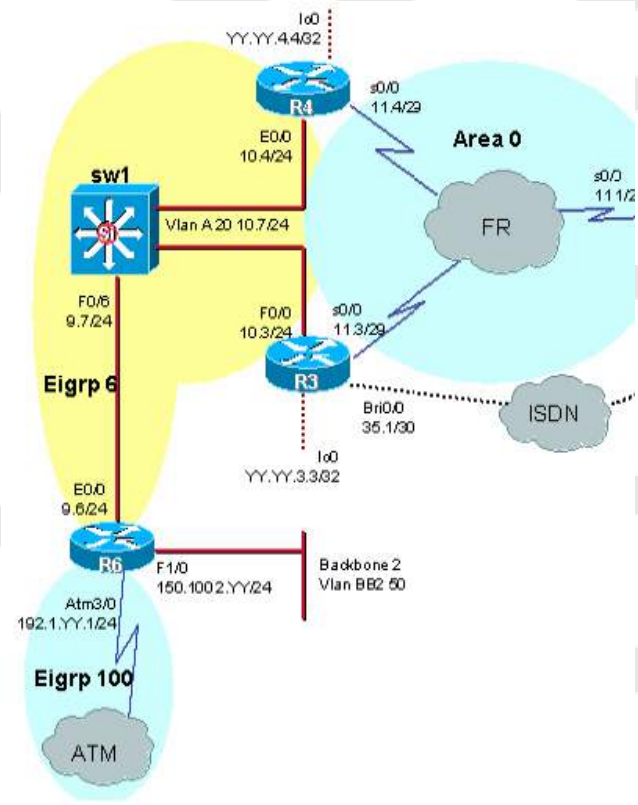
导致 S1 口被通告到 Area1。

```
Whitney(config)#router ospf 8
Whitney(config-router)#network 172.16.192.0 0.0.0.255 area 1
Whitney(config-router)#network 172.16.113.0 0.0.0.255 area 0
```

此故障到此解决。

Case Study: Routing-loops

此 troubleshooting 取自 CCIE_lab 160



Eigrp 100 redistribute to Eigrp 6

EIGRP 6 redistribute to OSPF Area 0

由于 EIGRP 重分布到 EIGRP6

EIGRP6 中将收到 EIGRP 外部路由，管理距离 170

将 EIGRP 重分布到 OSPF 中，此时 R3 通告 OSPF 的路由到 R4，此时 R4 将会收到 2 条同目的网络的路由:一条管理距离为 170 来自 EIGRP，另一条管理距离为 110 来自 OSPF,故 R4 将选择 R3 作为下一跳

对于这种环路，可以有多种方法解决

A ： EIGRP 100 重分布进 EIGRP 6 时打标，通过 route-map 过滤

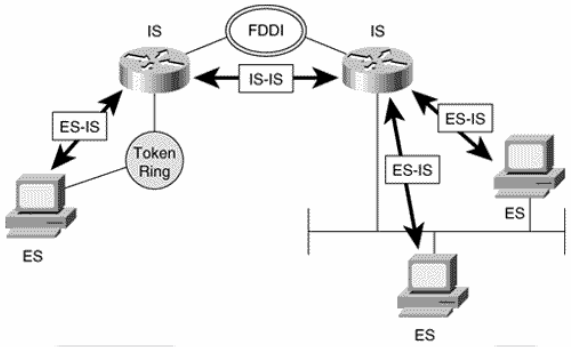
B ： 修改管理距离，将 EIGRP 6 中外部路由改为 90~110z 之间

Integrated IS-IS

Operation of Integrated IS-IS

IS-IS 是一种链路状态路由协议,它是由 ISO 定义和发展,
IS-IS 支持的是 ISO 的 CLNS(Connectionless Network Protocol,无连接的网络协议),
为了能够支持 IP,后来将 IS-IS 扩展为可以同时支持 OSI 和 TCP/IP 的集成 IS-IS(Integrated IS-IS)

在 IS-IS 中,路由器被描述为是一个中间系统(Intermediate System,IS),主机被描述为端系统(End System,ES).因此提供主机和路由器之间的通信的协议即为 ES-IS;而路由器之间的通信即为 IS-IS,如下图:

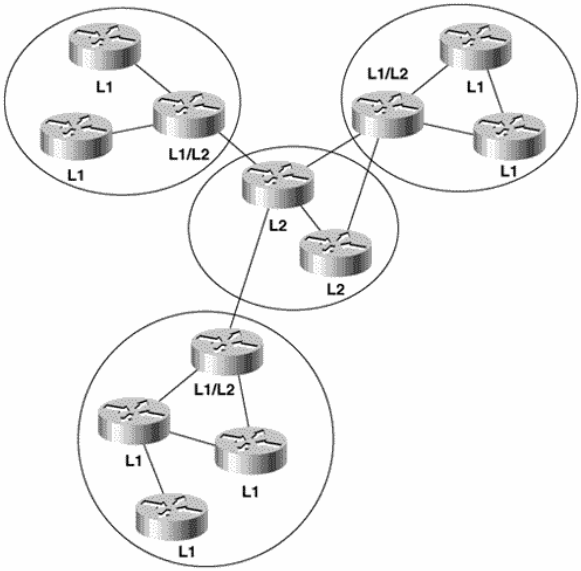


和 IP 中使用 Proxy ARP 或 IRDP,或在主机上配置默认网关不同的是,CLNP 会 ES 和 IS 之间形成 ES-IS 的邻接关系

与 1 个子网相连的接口就叫 SNPA(Subnetwork Point of Attachment).SNPA 并不是物理接口
它的基本概念和子网的基本概念是相同的

IS-IS Areas

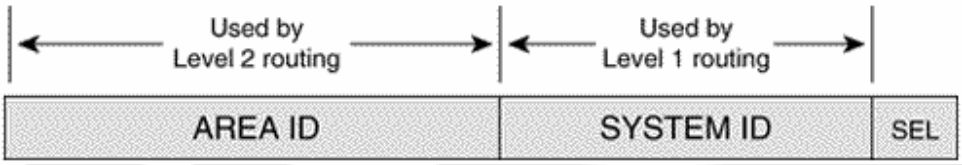
IS-IS 也定义了 2 层区域的概念,和 OSPF 不同的是,OSPF 的区域是以路由器为边界;而 IS-IS 中是以链路为边界,如下图:



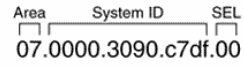
连接不同区域的IS为Level 2(L2)路由器,
或者是同时具有Level 1和Level 2的L1/L2路由器,
L1/L2路由器要同时分别维持L1和L2的LSDB
在IS-IS中,也有类似于RID的System ID.
由于IS-IS中区域是以路由器为边界,因此,1个路由器的每个接口上的区域ID都是一样的.
在IS-IS中,1个路由器最多可以具有3个区域ID,
这样对区域中的过渡是很有用的

Network Entity Titles

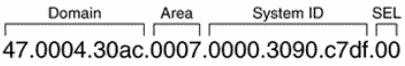
在 IS-IS 中,可以通过 Network Entity Tiyle(NET)来同时定义区域 ID 和系统 ID.即使集成 IS-IS 只使用在一个 TCP/IP 的环境中,由于它是 CLNP 协议,使用的是 CLNS PDU(Protocol Data Unit),所以仍然要有 1 个 ISO 地址.ISO 地址就是 NET,长度为 8 到 20 字节.NET 的结构如下图:



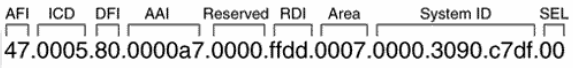
如下为 3 种 NET 格式



(a)



(b)



(c)

第 1 个是简单的 NET,区域 ID,系统 ID(为路由器某个接口的 MAC 地址)和选择符(SEL),SEL 被设置为 0x00;如果当 SEL 不为 0,它就是一个 NSAP(Network Service Access Point);第 3 个为 GOSIP NSAP.无论是那种格式的 NET,都要满足下面的几个规则:

1. NET 必须是以单个的 8 位位组开始(比如,47.xxxx…)
2. NET 必须是以担搁的 8 位位组结束(比如,…xxxx.00).如果 SEL 不为 0,IS-IS 也能正常工作,但是在 CLNP/IP 这种双环境中可能会出问题
3. 在 Cisco 路由器上,NET 中的系统 ID 必须为 6 字节

Subnetwork Dependent Functions

IS-IS 中只定义 2 种网络类型: 广播型和点到点型.

IS-IS hello packet

IS-IS 路由器通过 IS-IS Hello PDU 形成邻接关系,
IS-IS Hello PDU 每 10 秒发送 1 次
在接口配置模式下使用 **isis hello-interval <seconds>**来修改.

当然 L1 路由器可以和 L1 路由器和 L1/L2 路由器形成邻接关系,而 L1 路由器是不可以和 L2 路由器形成邻接关系的,
IS-IS Hello PDU 中也有 Hold Timer,默认为 IS-IS Hello PDU 发送间隔的 3 倍长,
可以通过接口配置命令 **isis hello-multiplier <seconds>**来修改

使用命令**show clns is-neighbors**查看IS-IS的邻居表

```
Brussels#show clns is-neighbors

System Id      Interface   State  Type Priority  Circuit Id         Format
0000.0C04.DCC0 Se0         Up     L1   0         06                 Phase V
0000.0C04.DCC0 Et1         Up     L1   64        0000.0C76.5B7C.03  Phase V
0000.0C0A.2C51 Et0         Up     L2   64        0000.0C76.5B7C.02  Phase V
0000.0C0A.2AA9 Et0         Up     L1L2 64/64     0000.0C76.5B7C.02  Phase V
Brussels#
```

State 为邻接的状态,一般为 Init(学习到但是还未邻接)和 Up(邻接成功建立)

Circuit ID 为电路 ID,8 字节长,用来标识 IS-IS 接口.

如果这个接口是和广播型多路访问的网络相连,那么电路 ID 和该网络上的指定路由器(Designated Router,DR)的系统 ID 相连,这个完整的字符串就叫 LAN ID.在这种用法里,把电路 ID 称为仿真节点 ID(Pseudonode ID).如上的例子里,0000.0C76.5B7C 就是 DR 的系统 ID,00 就是仿真节点 ID

Format 为邻接关系的格式,

对于集成的 IS-IS,这里的值永远为 Phase V,代表 OSI/DECnet Phase V;另外的一种值是 DECnet Phase IV

IS-IS 会广播型网络上选举 1 个 DR,与 OSPF 不同的是,这种广播型多路访问的网络中所有的 IS-IS 路由器都要和它所有的邻居建立邻接关系,而不仅仅是只和 DR 邻接.路由器通过组播的方式发送 LSP(Link State Packet)给它的邻居,DR 采用 SNP(Sequence Number PDU)来保证 LSP 的洪泛是可靠的

每个 IS-IS 路由器的接口都有 1 个基于 L1 和 L2 的优先级,这个优先级的范围是 0 到 127,Cisco 默认的基于 L1 和 L2 的优先级都是 64,这个值可以通过命令 **isis priority <value>**来修改

IS-IS 中 DR 的选举也是根据优先级来判定的,IS-IS 路由器发送的 Hello 包分都包含的有优先级字段(L1 的 Hello 包宣告 L1 的优先级;L2 的 Hello 包宣告 L2 的优先级).如果优先级为 0,它就不能成为 DR,如上图:

第一行显示的优先级为0,因为它是串性点到点链路,不需要选举DR,所以设置为0(非广播型网络不需要选举DR).优先级最高的成为DR;如果优先级一样,就比较系统ID,系统ID最大的成为DR.与L1和L2的优先级分别对应的是,要分别选举1个L1的DR和L2的DR

与OSPF不同的是,IS-IS中,不会选举BDR, 另外DR会抢占, 只要网络上出现1个优先级高过现有DR的优先级的路由器,它将立即取代现有DR的位置而成为新的DR.这样,每次DR的更改都将产生1组LSP的洪泛

Subnetwork Independent Functions

子网独立子层定义了CLNS如何把书数据包通过CLNP的网络把这些服务提供给传输层路由功能被分为4个过程:

1. 更新(Update)过程
2. 决策(Decision)过程
3. 转发(Forwarding)过程
4. 接收(Receive)过程

更新过程负责洪泛滥L1和L2的LSP来构建LSDB(Link State Database),L1的LSP会在整个区域洪泛,而L2的LSP只会在所有建立了L2邻接的路由器上洪泛

每个LSP都包含提个剩余生存时间(Remaining Lifetime),一个序列号(Sequence Number)和一个校验和(Checksum).

和OSPF中LSA的Age不同的是,IS-IS中的生存时间是递减的.IS-IS中LSP的最大生存周期(MaxAge)是1200秒(20分钟),和OSPF的LSA一样,当LSP驻留在LSDB中,这个剩余生存时间也会随之推移的.为防止LSP的剩余生存时间为0,源路由器会周期刷新它的LSP,这个刷新的时间周期为900秒(15分钟), Jitter 25%.

当LSP的剩余生存时间递减到0以后,这个LSP还会在LSDB中保留60秒,这个时间叫做ZeroAgeLifetime

如果1个IS-IS路由器收到了校验和错误的LSP,它将把这个LSP的剩余生存时间设置为0来移除这条LSP,源路由器将重新发送这个LSP的1个新的实例(这点和OSPF不同,OSPF的源路由器仅仅清除这条LSA)

在一个错误机率比较高的网络上,允许路由器清除错误的LSF可能会导致LSP流量过高,因为这将使得源路由器不断重发LSP的新的实例.可以使用命令**ignore-lsp-errors**来忽略这种错误的LSP而不去清除它(当然发出这条错误的LSP的源路由器可以通过SNP得知这条LSP没有被收到)

序列号是一个无符号的32位长的线性数字,当IS-IS路由器初次发送LSP的时候,序列号设置为1,以后这个LSP的每个实例的序列号都会递增1,如果序列号达到了最大的0xFFFFFFFF,IS-IS进程将关闭至少21分钟(MaxAge+ZeroAgeLifetime),以便允许旧的LSP从LSDB中清除

在点到点的网络上,路由器直接发送L1和L2的LSP给邻居.
在广播型的网络上,LSP通过组播的方式发给邻居,运载L1的LSP的帧的目标MAC地址为0180.c200.0014,又叫AllL1ISs;运载L2的LSP的帧的目标MAC地址为0180.c200.0015,又叫AllL2ISs

IS-IS通过SNP来确认LSP的接收和维持LSDB信息的同步,有2种类型的SNP:

1. Parial SNP(PSNP)
2. Complete SNP(CSNP)

在点到点网络上,IS-IS路由器通过PSNP来确认每个LSP是否收到.PSNP通过下面的信息来描述它所确认的LSP:

1. LSP ID
2. LSP的序列号
3. LSP的校验和
4. LSP的剩余生存时间

在点到点的环境中,当有路由器要发送LSP的时候,它会设置个叫做minimumLSPTransmissionInterval的时间,在这个时间超出之前没有收到关于这个LSP的PSNP时,新的LSP将被重传.Cisco路由器上这个时间默认为5秒,可以通过接口配置命令**isis retransmit-interval <seconds>**来修改

在广播型网络上,LSP不需要被每个路由器都确认.DR将周期性的组播CSNP,用来描述LSDB中的每个LSP,默认发送时间为10秒,这个时间可以通过命令**isis csnp-interval <seconds>**来修改,

L1的CSNP是组播到AllL1ISs
L2的CSNP是组播到AllL2ISs

当路由器收到CSNP的时候,它会把这个CSNP中的LSP的信息和它LSDB中的LSP做比较.如果路由器发现LSDB中存在CSNP中没有的LSP,或者有比CSNP中更新的LSP,那么这个路由器就将组播这个LSP,但是如果其他的路由器先发送这个更新的LSP,这个路由器就不会发送这个LSP的另1个拷贝.如果路由器的LSDB中没有包含CSNP中所列出来的LSP,或者LSDB中的LSP比CSNP中列出来的LSP更旧,路由器将组播PSNP,这个PSNP列举了路由器所需要的LSP的清单.虽然PSNP是组播出去的,但是只有DR会以合适的LSP响应它

IS-IS还有个特性,当路由器因内存不足的原因而不能记录完整的LSDB信息的时候,它可以通知其他的路由器,它将在LSP中设置1个超载(Overload,OL)位,它用来表示路由器可能不能够再做路由决策了,因为它的LSDB不完整.再这个发生超载现象的路由器发送1个清除OL位的LSP之前,其他的路由器就会避免经过它来把数据包转发到其他网络上去,因此OL位也被叫做hippity(或hippity-hop)位

show isis database命令查看IS-IS路由器的LSDB信息,如下:

```
Brussels#show isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C04.DCC0.00-00  0x00000036   0x78AE        1152          0/0/0
0000.0C0A.2AA9.00-00  0x0000011B   0x057B        416           0/0/0
0000.0C76.5B7C.00-00* 0x00000150   0xD5D4        961           1/0/0
0000.0C76.5B7C.02-00* 0x00000119   0xD9C3        407           0/0/0
0000.0C76.5B7C.03-00* 0x000000FA   0x896E        847           0/0/0

IS-IS Level-2 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2AA9.00-00  0x0000013E   0x319A        666           0/0/0
0000.0C0A.2C51.00-00  0x00000133   0x762D        654           0/0/0
0000.0C76.5B7C.00-00* 0x0000014C   0x4E91        886           0/0/0
0000.0C76.5B7C.02-00* 0x0000011F   0x3CC3        1174          0/0/0
0000.3090.C7DF.00-00  0x0000011A   0xDDF0        858           0/0/0
Brussels#
```

可以看出路由器Brussels是一个L1/L2路由器,其中LSP ID后面跟的星号表示这个LSP是正在查看的路由器所发出去的

LSP Holdtime也就是剩余生存时间

ATT表示Attached位,设置为1表示包含有达到其他区域的路由

P表示Partition位,该位一般设置为0

如下是加上detail参数查看具体某条LSP的信息:

```
London#show isis database detail level-2 0000.0C0A.2C51.00-00

IS-IS Level-2 LSP 0000.0C0A.2C51.00-00
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2C51.00-00* 0x0000013B   0x6635        815           0/0/0
  Area Address: 47.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.3.2
  Metric: 10 IS 0000.0C76.5B7C.02
  Metric: 10 IP 10.1.3.0 255.255.255.0
  Metric: 20 IP 10.1.2.0 255.255.255.0
  Metric: 10 IP 10.1.255.4 255.255.255.252
  Metric: 20 IP 10.1.255.0 255.255.255.0
  Metric: 30 IP 10.1.255.8 255.255.255.252
London#
```

一旦更新过程建立了完整的LSDB后,决策过程就将使用LSDB中的LSP信息来构建一条最短路径树,然后形成路由表.对于L1和L2的路由,路由器会分别执行不同的SPF计算

IS-IS的metric标准如下(1个是必须,另3个是可选):

1. Default:必须的.每台IS-IS路由器都必须支持和理解的metric
2. Delay:可选的,反映了子网的传输
3. Expense:可选的,反映子网的成本开销
4. Error:可选的,反映了子出错的机率(类似IGRP/EIGRP中的RELY)

每1种metric的范围都是0到63之间的整数,并且每条路由要根据每种metric进行单独的计算,并且有可能产生不同的路由表.Cisco的路由器只支持Default类型的metric.

在Cisco路由器上,接口默认的metric都为10,可以使用命令**isis metric <value>**进行修改,

因此一条路由的总的开销就是接口开销之和.对于任何一条路由来说,最大的metric为1023

IS-IS路由不光分为L1和L2,还分为内部路由和外部路由,

对于L2,它有可能是内部路由也有可能是外部路由;

对于L1,它始终都是内部路由.

如果存在到达某个目标地址的多条路由,L1的路由将优先与L2的路由.在同一Level中的路由,支持可选metric的路由将优先与只支持Default metric的路由.在每个Level里,开销最低的路由将优先被采用,如果在同一Level中多条路由的等价的,那么它们将都被放入路由表里.Cisco支持最大6条路径的等价的负载均衡

LSP ID中最后的1个字节是LSP Number,用来跟踪分片(因LSP过大而分片)的LSP.决策过程将注意到LSP Number的原因是:如果路由器的LSDB中没有LSP Number为0并且剩余生存时间不为0的LSP,决策过程将不会处理具有相同的系统ID,但是LSP Number不为0的LSP.这将保证不会因不完整的LSP信息而导致不正确的路由决策.另外,决策过程将仅从LSP Number为0的LSP接受下面几种信息:

1. OL位
2. IS的类型字段

3. 区域地址可选字段的信息

在LSP Number不为0的LSP中,决策过程将忽略上面3个信息,在分片的LSP中,上述3个字段的信息由LSP Number为0的LSP来宣告

IS-IS支持VLSM,并且管理距离(administrative distance,AD)是115

如果ATT位设置为1,集成IS-IS就会增加一条到达最近的L1/L2路由器的默认路由,如下:

```
Paris#show isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C0A.2C51.00-00  0x0000016D   0xA093        730           1/0/0
0000.3090.6756.00-00* 0x00000167   0xC103        813           0/0/0
0000.3090.6756.04-00* 0x0000014E   0x227F        801           0/0/0
0000.3090.C7DF.00-00  0x00000158   0x78A6        442           0/0/0
Paris#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 10.1.255.6 to network 0.0.0.0

     10.0.0.0 is variably subnetted, 5 subnets, 2 masks
i L1    10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C       10.1.2.0 255.255.255.0 is directly connected, TokenRing0
i L1    10.1.255.4 255.255.255.252 [115/20] via 10.1.255.6, Serial0
C       10.1.255.0 255.255.255.0 is directly connected, Serial0
i L1    10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
i*L1 0.0.0.0 0.0.0.0 [115/10] via 10.1.255.6, Serial0
Paris#
```

如上反映了一个问题,就是CLNP信息和IP信息的关联不太方便.解决办法是使用命令**which-route**,如下:

```
Paris#which-route 0000.0C0A.2C51.00
Route look-up for destination 00.000c.0a2c.5100
  Using route to closest IS-IS level-2 router

Adjacency entry used:
System Id      SNPA            Interface   State  Holdtime  Type Protocol
0000.0C0A.2C51 *HDLC*          Se0         Up     26        L1   IS-IS
  Area Address(es): 47.0001
  IP Address(es):  10.1.255.6
  Uptime: 22:08:52
Paris#
```

这个命令的作用是确定某个CLNS地址在路由表中的位置,把CLNS信息和IP信息相互关联

IS-IS PDU Formats

IS-IS使用9种PDU,可以分为3个大类,如下:

Hello PDUs:

1. L1 LAN IS-IS Hello PDU:类型号为15
2. L2 LAN IS-IS Hello PDU:类型号为16
3. Point-to-point IS-IS Hello PDU:类型号为17

Link State PDUs:

1. L1 LSP:类型号为18
2. L2 LSP:类型号为20

Sequence Numbers PDUs(SNPs):

1. L1 CSNP:类型号为24
2. L2 CSNP:类型号为25
3. L1 PSNP:类型号为26

	Length, in Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R | R | R | PDU Type	1
Version	1
Reserved	1
Maximum Area Addresses	1
PDU- Specific Fields	
Variable-Length Fields	

4. L2 PSNP:类型号为27

这些IS-IS PDU的前8字节的头部信息格式如右图:

Interdomain Routeing Protocol Discriminator:

ISO9577 分配的一个恒定不变的数,值为 0x83,用来标识网络 层的 PDU(Network PDU,NPDU)

Length Indicator:

标记固定头部信息的长度,1 字节长

Version/Protocol ID Extension:始终设置为 1

ID Length:

表示路由域里 NSAP 和 NET 使用的系统 ID 的长度.该字段的值可以是以下几种数值之一:

1. 值为 1 到 8 的整数:表示系统 ID 具有和这个值相同的长度
2. 值为 0:表示系统 ID 为 6 字节长
3. 值为 255:表示系统 ID 字段为 null(0 字节)

R:保留,设置为 0

PDU Type:PDU 类型,5 位长

Version:版本号,始终为 1,这和之前的 Version/Protocol ID Extension 是一样的

Maximum Area Addresses:

最大区域数,表示区域内允许的最大区域地址数量.值可以是下面几种数值之一:

1. 值为 0:表示最大只支持 3 个区域地址
2. 值为 1 到 254 的整数,表示支持的区域地址数和这个值相同

Cisco 路由器中,只支持值为 0,即最大支持 3 个区域地址

CLV Fields

Variable-Length Fields:可变长字段,为CLV字段,即代码/长度/值(Code/Length/Value)

	Length, in Octets
Code	1
Length	1
Value	Length

其中Code和Length字段长度分别为1字节,Value字段就是它本身的信息.Value字段的大小最大可为255字节.

The IS-IS Hello PDU Format

有2种类型的IS-IS Hello PDU:

IS-IS LAN Hello PDU

IS-IS Point-to-point Hello PDU

IS-IS LAN Hello PDU还可以分为L1和L2,但是这2种类型是一样的.

Circuit Type:

2位长,之前的6个R位保留,设置为0.在这个字段里,如果值为01表示是L1路由器,如果为02则表示是L2路由器,为11的话表示是L1/L2路由器,如果值为00,这个PDU将被忽略

Source ID:发出这个IS-IS Hello PDU的源路由器的系统ID

LAN ID:系统ID加上一个1字节长的仿真节点ID

	Length, in Octets
Intradomain Routeing Protocol Discriminator	1
Length Indicator	1
Version/Protocol ID Extension	1
ID Length	1
R R R PDU Type	1
Version	1
Reserved	1
Maximum Area Addresses	1
R R R R R R Circuit Type	1
Source ID	ID Length
Holding Time	2
PDU Length	2
R Priority	2
LAN ID	ID Length + 1
Variable-Length Fields	

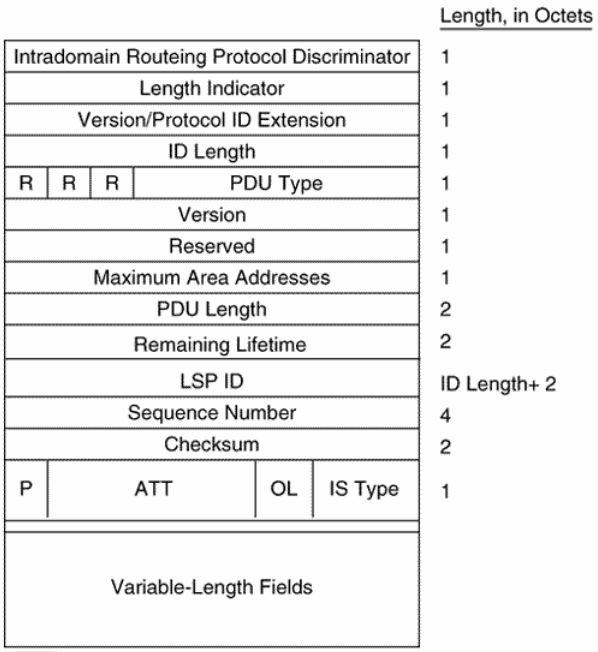
The IS-IS Link State PDU Format

P:Cisco 路由器中设置为 0

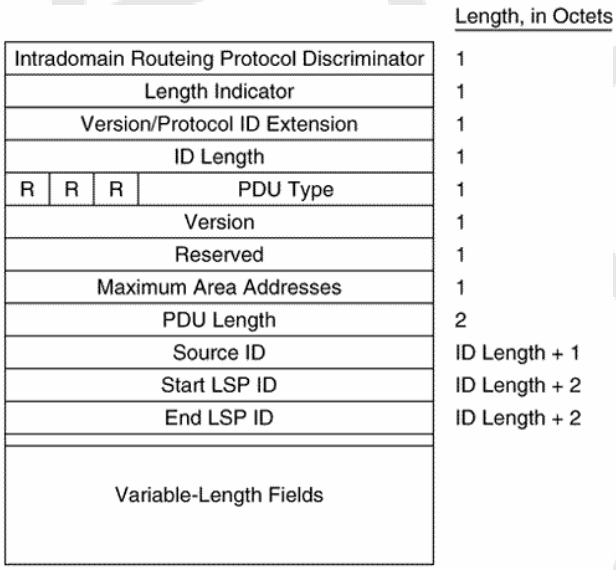
ATT:4 位长,表示相连的区域使用了哪种的 metric 类型,
从右(第 4 位)到左(第 7 位)依次是:
Default,Delay,Expense 和 Error,

因为 Cisco 的路由器只支持 Default,因此其他 3 位设置为 0

IS Type:2 位长,用来表示源路由器的类型,如下:
00:未使用
01:L1 路由器
10:未使用
11:L2 路由器



The IS-IS Sequence Numbers PDU Format

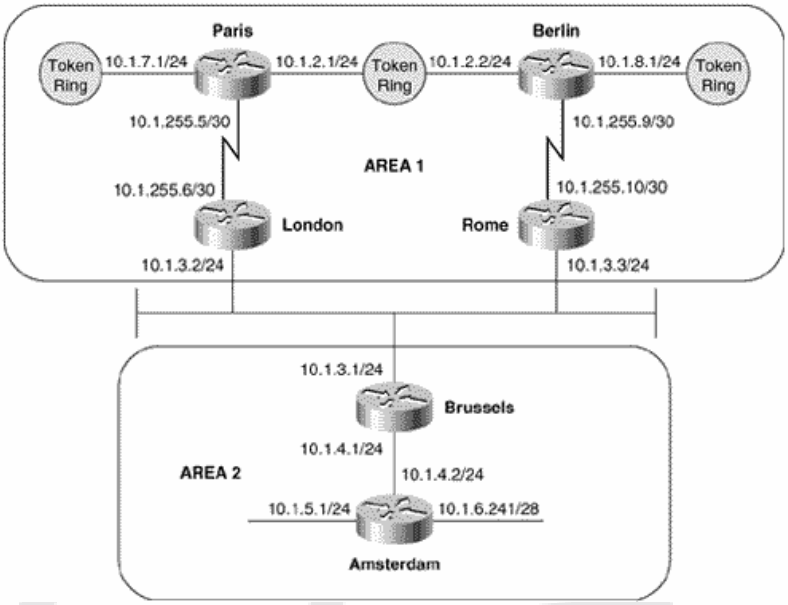


Configuring Integrated IS-IS

Case Study: A Basic Integrated IS-IS Configuration

配置IS-IS的基本步骤如下:

1. 确定路由器所在的区域和要起用IS-IS的接口
2. 使用router isis来启动IS-IS(router isis也可以跟一个名字,比如router isis dancer)
3. 使用net命令配置NET地址
4. ip router isis在相应的接口上启动集成IS-IS



NET 地址表如下

Router	Area	MAC	Net
Paris	00.0001	0000.3090.6756	00.0001.0000.3090.6756.00
Berlin	00.0001	0000.3090.c7df	00.0001.0000.3090.c7df.00
London	00.0001	0000.0c0a.2c51	00.0001.0000.0c0a.2c51.00
Rome	00.0001	0000.0c0a.2aa9	00.0001.0000.0c0a.2aa9.00
Brussels	00.0002	0000.0c76.5b7c	00.0002.0000.0c76.5b7c.00
Amsterdam	00.0002	0000.0c04.dcc0	00.0002.0000.0c04.dcc0.00

路由器Paris配置如下:

```
clns routing
!
interface Serial0
ip address 10.1.255.5 255.255.255.252
ip router isis
!
interface TokenRing0
ip address 10.1.2.1 255.255.255.0
ip router isis
ring-speed 16
!
interface TokenRing1
ip address 10.1.7.1 255.255.255.0
ip router isis
ring-speed 16
!
router isis
net 00.0001.0000.3090.6756.00
```

路由器London配置如下:

```
clns routing
!
interface Ethernet0
ip address 10.1.3.2 255.255.255.0
ip router isis
!
interface Serial0
ip address 10.1.255.6 255.255.255.252
ip router isis
!
router isis
net 00.0001.0000.0c0a.2c51.00
```

路由器Brussels配置如下:

```
clns routing
!
interface Ethernet0
ip address 10.1.3.1 255.255.255.0
ip router isis
!
interface Ethernet1
ip address 10.1.4.1 255.255.255.0
ip router isis
```

```
!
router isis
net 00.0002.0000.0c76.5b7c.00
```

路由器Amsterdam配置如下:

```
clns routing
!
interface Ethernet0
ip address 10.1.4.2 255.255.255.0
ip router isis
!
interface Ethernet1
ip address 10.1.5.1 255.255.255.0
ip router isis
!
interface Ethernet2
ip address 10.1.6.241 255.255.255.240
ip router isis
!
router isis
net 00.0002.0000.0c04.dcc0.00
```

路由器Rome和Berlin的配置方法类似,要注意的是命令**clns routing**在配置IS-IS的时候不是必需的
因为在启用IS-IS的时候,路由器自动加入这个命令

Case Study: Changing the Router Types

我们可以发现路由器Paris和路由器Amsterdam都分别维持的有L1和L2的LSDB,

```
Paris#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is not set

     10.0.0.0 is variably subnetted, 9 subnets, 3 masks
i L1    10.1.8.0 255.255.255.0 [115/20] via 10.1.2.2, TokenRing0
i L1    10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C       10.1.2.0 255.255.255.0 is directly connected, TokenRing0
C       10.1.7.0 255.255.255.0 is directly connected, TokenRing1
i L2    10.1.5.0 255.255.255.0 [115/40] via 10.1.255.6, Serial0
i L2    10.1.4.0 255.255.255.0 [115/30] via 10.1.255.6, Serial0
C       10.1.255.4 255.255.255.252 is directly connected, Serial0
i L1    10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
i L2    10.1.6.240 255.255.255.240 [115/40] via 10.1.255.6, Serial0
Paris#
```

但是实际上路由器Berlin,Amsterdam还有Paris配置成L1的路由器就可以了,这样可以减轻路由器的负担,使用命令is-type来更改路由类型,因此,要把路由器Berlin更改为L1路由器,只需做如下配置:

```
Berlin(config)#router isis
Berlin(config-router)#net 00.0001.0000.3090.c7df.00
Berlin(config-router)#is-type level-1
```

```
Paris#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is not set

     10.0.0.0 is variably subnetted, 6 subnets, 2 masks
i L1    10.1.8.0 255.255.255.0 [115/20] via 10.1.2.2, TokenRing0
i L1    10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C       10.1.2.0 255.255.255.0 is directly connected, TokenRing0
C       10.1.7.0 255.255.255.0 is directly connected, TokenRing1
C       10.1.255.4 255.255.255.252 is directly connected, Serial0
i L1    10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
Paris#

Amsterdam#show isis database
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.0C04.DCC0.00-00* 0x0000002C   0x2E77        726           0/0/0
0000.0C76.5B7C.00-00  0x0000002A   0xC515        733           1/0/0
0000.0C76.5B7C.03-00  0x00000026   0x3399        733           0/0/0
Amsterdam#
```

可以看到LSDB和路由表的体积都减小了.但是还有个问题,就是虽然L1/L2路由器会在LSP中设置ATT位来告诉L1路由器它具有区域间的连接,因此路由器Paris会有1条到达路由器London或Rome的默认路由,但是之前Paris的路由表中并没有这条默认路由.原因是ATT位是CLNS的特性,在IP环境中它不能被直接的理解.解决办法有2种,

1种是不光在路由器London和Paris的接口上启用IP的IS-IS,还要启用CLNS的IS-IS,如下:

```
London(config)#interface Serial0
London(config-if)#ip address 10.1.255.6 255.255.255.252
London(config-if)#ip router isis
London(config-if)#clns router isis
Paris(config)#interface Serial0
Paris(config-if)#ip address 10.1.255.5 255.255.255.252
Paris(config-if)#ip router isis
Paris(config-if)#clns router isis
```

```
Paris#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 10.1.255.6 to network 0.0.0.0

     10.0.0.0 is variably subnetted, 6 subnets, 2 masks
i L1    10.1.8.0 255.255.255.0 [115/20] via 10.1.2.2, TokenRing0
i L1    10.1.3.0 255.255.255.0 [115/20] via 10.1.255.6, Serial0
C       10.1.2.0 255.255.255.0 is directly connected, TokenRing0
C       10.1.7.0 255.255.255.0 is directly connected, TokenRing1
C       10.1.255.4 255.255.255.252 is directly connected, Serial0
i L1    10.1.255.8 255.255.255.252 [115/20] via 10.1.2.2, TokenRing0
i*L1 0.0.0.0 0.0.0.0 [115/10] via 10.1.255.6, Serial0
Paris#ping 10.1.6.241
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.6.241, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/36/40 ms
Paris#
```

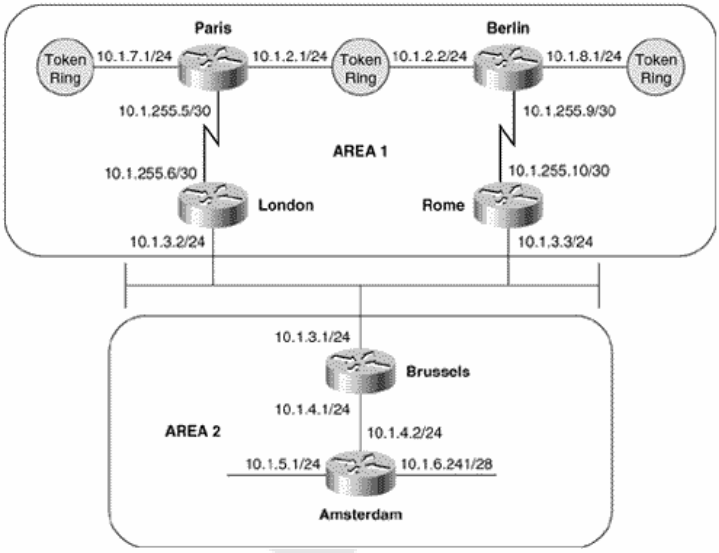
第2种解决方法是在L1/L2路由器上配置一条默认路由,并使用default-information originate命令使得IS-IS来宣告这条默认路由.以路由器Brussels为例,配置如下:

```
Brussels(config)#router isis
Brussels(config-router)#net 00.0002.0000.0c76.5b7c.00
Brussels(config-roiuter)#default-information originate
Brussels(config)#ip route 0.0.0.0 0.0.0.0 Null0
```

```
Amsterdam#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is 10.1.4.1 to network 0.0.0.0

     10.0.0.0 is variably subnetted, 4 subnets, 2 masks
i L1    10.1.3.0 255.255.255.0 [115/20] via 10.1.4.1, Ethernet0
C       10.1.5.0 255.255.255.0 is directly connected, Ethernet1
C       10.1.4.0 255.255.255.0 is directly connected, Ethernet0
C       10.1.6.240 255.255.255.240 is directly connected, Ethernet2
i*L1 0.0.0.0 0.0.0.0 [115/10] via 10.1.4.1, Ethernet0
Amsterdam#ping 10.1.8.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.1.8.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 32/36/40 ms
```

Case Study: An Area Migration

在OSPF中,区域的迁移必须在中断网络的情况下尽情;而IS-IS可以在不中断网络的情况下进行区域的迁移,因为Cisco的路由器最多可以配置3个区域地址.



现假设要用GOSIP格式的NET地址代替原有NET地址格式,

Router	NET
Paris	47.0005.80.00ab7c.0000.ffe9.0001.0000.3090.6756.00
Berlin	47.0005.80.00ab7c.0000.ffe9.0001.0000.3090.c7df.00
London	47.0005.80.00ab7c.0000.ffe9.0001.0000.0c0a.2c51.00
Rome	47.0005.80.00ab7c.0000.ffe9.0001.0000.0c0a.2aa9.00
Brussels	47.0005.80.00ab7c.0000.ffe9.0002.0000.0c76.5b7c.00
Amsterdam	47.0005.80.00ab7c.0000.ffe9.0002.0000.0c04.dcc0.00

配置步骤是在不删除原有NET地址的基础上,增加新的GOSIP NET地址,以路由器Rome为例,如下:

```
Rome(config)#router isis
Rome(config-router)#net 00.0001.0000.0c0a.2aa9.00
Rome(config-router)#net 47.0005.8000.ab7c.0000.ffe9.0001.0000.0c0a.2aa9.00
```

使用**show isis database detail**或**show clns is-neighbors**验证,如下:

```
Rome#show isis database detail
IS-IS Level-1 Link State Database
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/
0000.0C0A.2AA9.00-00* 0x00000059   0x705E        592           1/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.3.3
  Metric: 10 IP 10.1.3.0 255.255.255.0
  Metric: 10 IP 10.1.255.8 255.255.255.252
  Metric: 10 IS 0000.0C0A.2C51.01
  Metric: 10 IS 0000.3090.C7DF.00
  Metric: 0  ES 0000.0C0A.2AA9
0000.0C0A.2C51.00-00  0x00000059   0xD495        652           1/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.3.2
  Metric: 10 IP 10.1.3.0 255.255.255.0
  Metric: 10 IP 10.1.255.4 255.255.255.252
  Metric: 10 IS 0000.0C0A.2C51.01
  Metric: 10 IS 0000.3090.6756.00
  Metric: 0  ES 0000.0C0A.2C51
0000.0C0A.2C51.01-00  0x00000052   0xD81B        507           0/0/0
  Metric: 0  IS 0000.0C0A.2C51.00
  Metric: 0  IS 0000.0C0A.2AA9.00
0000.3090.6756.00-00  0x0000005C   0xDB0D        678           0/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
  NLPID:        0x81 0xCC
  IP Address:   10.1.7.1
  Metric: 10 IP 10.1.7.0 255.255.255.0
  Metric: 10 IP 10.1.255.4 255.255.255.252
  Metric: 10 IP 10.1.2.0 255.255.255.0
  Metric: 10 IS 0000.3090.6756.04
  Metric: 10 IS 0000.0C0A.2C51.00
  Metric: 0  ES 0000.3090.6756
0000.3090.6756.04-00  0x00000054   0x1983        835           0/0/0
  Metric: 0  IS 0000.3090.6756.00
  Metric: 0  IS 0000.3090.C7DF.00
0000.3090.C7DF.00-00  0x0000005B   0x18A5        545           0/0/0
  Area Address: 00.0001
  Area Address: 47.0005.8000.ab7c.0000.ffe9.0001
--More--
```

增加了新的GOSIP NET地址以后,使用no net命令删除原有NET地址,迁移完成

Case Study: Route Summarization

```
Zurich(config)#router isis
Zurich(config-router)#net 01.0000.0c76.5b7c.00
Zurich(config-router)#summary-address 172.16.0.0 255.255.248.0
```

Case Study: Authentication

IS-IS只支持明文口令的验证,Cisco IOS支持3个级别的验证:邻居间,整个区域和整个IS-IS路由域.这3个级别的验证可以分开来使用也可以结合在一起使用,规则如下:

1. 邻居间的验证,相互连接的路由器的接口上配置的口令必须相同
2. 配置邻居间的验证,要分别为L1和L2的邻接各自配置单独的口令
3. 整个区域发验证,区域内所有的IS-IS路由器必须启用验证,并配置相同的口令
4. 整个IS-IS路由域的验证,每个L1/L2路由器和L2路由器都必须启用验证,并配置相同的口令

邻居间验证的配置,使用接口配置命令**isis password <password> {level-1|level-2}**,L1和L2的口令可以相同,如果不跟

{level-1|level-2}参数,默认为L1.

在整个区域的验证,使用**area-password <password>**命令启用验证,如下是配置区域3的验证:

```
Bonn(config-router)#area-password Rhine
```

要注意的是isis password命令指定的口令是包含在IS-IS Hello PDU中的;而area-password所指定的口令是包含在L1的LSP,CSNP和PSNP中的.因此,前者指定的口令只能用来验证邻接关系的建立;后者是验证L1的LSP的交换,如果口令不一样,仍然可以建立邻接关系,但是不会交换L1的LSP信息

IS-IS路由域的验证,使用命令**domain-password <password>**来指定,IS-IS路由域的验证只需要在L2或L1/L2路由器上配置

IS-IS路由域的验证口令是出现在L2的LSP,CSNP和PSNP中的,是验证L2的LSP的交换,如果口令不一样,仍然可以建立邻接关系,但是不会交换L2的LSP信息

Troubleshooting Integrated IS-IS

Case Study: Troubleshooting IS-IS Adjacencies

查看IS-IS邻居表使用**show clns is-neighbors {detail}**命令,在查看邻居表的时候,要靠拉到诸如以下问题:

1. 路由器Level的配置是否正确
2. 邻居路由器是否都正常发送IS-IS Hello PDU,可以通过命令**debug isis adj-packets**来查看,如下:

```
Bonn#debug isis adj-packets
IS-IS Adjacency related packets debugging is on
Bonn#
ISIS-Adj: Sending serial IIH on Serial0
ISIS-Adj: Rec L1 IIH from 0000.0c04.dcc0 (Ethernet0), cir type 1, cir id 0000.0C
0A.2AA9.02
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Rec serial IIH from *HDLC* on Serial0, cir type 2, cir id 02
ISIS-Adj: rcvd state 0, old state 0, new state 0
ISIS-Adj: Action = 2, new_type = 0
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Sending L2 IIH on Ethernet0
ISIS-Adj: Sending serial IIH on Serial0
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Rec serial IIH from *HDLC* on Serial0, cir type 2, cir id 02
ISIS-Adj: rcvd state 0, old state 0, new state 0
ISIS-Adj: Action = 2, new_type = 0
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Rec L1 IIH from 0000.0c04.dcc0 (Etheir type 1, cir id 0000.0C0A.2AA9.0
2
ISIS-Adj: Sending L1 IIH on Ethernet0
ISIS-Adj: Sending L2 IIH on Ethernet0
ISIS-Adj: Sending serial IIH on Serial0
```

3. 邻居之间的**isis hello-interval**和**isis hello-multiplier**的设置是否相同
4. 如果启用了验证,注意3个验证级别的口令是否符合之前所说的几个规则
5. 是否存在堵塞IS-IS或CLNS的ACL

Case Study: Troubleshooting the IS-IS Link State Database

使用**show isis database {level-1|level-2} {detail} {LSP ID}**来查看IS-IS路由器的LSDB信息

```
Zurich#show isis database detail 0000.3090.6756.00-00

IS-IS Level-2 LSP 0000.3090.6756.00-00
LSPID                 LSP Seq Num  LSP Checksum  LSP Holdtime  ATT/P/OL
0000.3090.6756.00-00  0x00000080   0x9EA1        480           0/0/0
  Auth:         Length: 12
  Area Address: 02
  NLPID:        0xCC
  IP Address:   172.16.8.1
  Metric: 10 IS 0000.0C76.5B7C.00
  Metric: 10 IS 0000.0C0A.2AA9.00
  Metric: 10 IP 172.16.8.0 255.255.255.0
  Metric: 10 IP 172.16.9.0 255.255.255.0
Zurich#
```

如果1个LSP的序列号明显高于其他LSP的序列号,很可能是由于区域不稳定或L2骨干路由器不稳定造成的;另外要注意的是LSP的Holdtime从来都不会很小,否则就有可能是网络不稳定造成的.如果怀疑网络不稳定,可以使用命令**show isis spf-log**显示本地路由器最近执行的所有SPF计算的日志信息,如下:

```
Geneva#sh isis spf-log

   Level 1 SPF log
  When   Duration  Nodes  Count     Last trigger LSP   Triggers
02:43:09      12      3      1                         PERIODIC
02:28:08      12      3      1                         PERIODIC
02:13:06      12      3      1                         PERIODIC
01:58:05      12      3      1                         PERIODIC
01:43:03      12      3      1                         PERIODIC
01:28:02      12      3      1                         PERIODIC
01:13:00      12      3      1                         PERIODIC
00:57:59      12      3      1                         PERIODIC
00:42:58      12      3      1                         PERIODIC
00:27:56      12      3      1                         PERIODIC
00:12:55      12      3      1                         PERIODIC
00:03:08       8      3      1  0000.0C76.5B7C.00-00   LSPHEADER
00:02:35       8      3      1  0000.0C76.5B7C.00-00   LSPHEADER
00:02:23       8      3      1  0000.0C76.5B7C.00-00   LSPHEADER
00:01:50       8      3      1  0000.0C76.5B7C.00-00   LSPHEADER
00:01:14       4      1      1  0000.0C0A.2C51.00-00   TLVCONTENT
00:00:46       4      2      2  0000.0C0A.2C51.04-00   NEWLSP TLVCONTENT
00:00:20       4      1      3  0000.0C0A.2C51.00-00   NEWADJ TLVCONTENT
00:00:08       8      3      1  0000.0C76.5B7C.02-00   TLVCONTENT
Geneva#
```

前几行显示的是周期性的LSP的刷新(每15分钟),而后几行显示了频繁的SPF计算,说明网络发生了变化.为了进一步跟踪问题所在,可以使用3个调试命令:

1. **debug isis spf-triggers**:显示的是与触发SPF计算相关的事件信息

```
Geneva#debug isis spf-triggers
IS-IS SPF triggering events debugging is on
Geneva#
ISIS-SPF-TRIG: L1, LSP fields changed 0000.0C76.5B7C.00-00
ISIS-SPF-TRIG: L1, LSP fields changed 0000.0C76.5B7C.00-00
Geneva#
```

2. **debug isis spf-events**:显示引起触发SPF计算的1个详细报告,如下:

```
Geneva#debug isis spf-events
IS-IS SPF events debugging is on
Geneva#
ISIS-SPF: L1 LSP 3 (0000.0C76.5B7C.00-00) flagged for recalculation
from 34F561A
ISIS-SPF: Calculating routes for L1 LSP 3 (0000.0C76.5B7C.00-00)
ISIS-SPF: Add 172.16.4.0/255.255.255.0 to IP route table, metric 20
ISIS-SPF: Next hop 0000.0C76.5B7C/172.16.4.2 (Ethernet0) (rejected)
ISIS-SPF: Add 0000.0C76.5B7C to L1 route table, metric 10
ISIS-SPF:   Next hop 0000.0C76.5B7C (Ethernet0)
ISIS-SPF: Aging L1 LSP 3 (0000.0C76.5B7C.00-00), version 132
ISIS-SPF: Aging IP 172.16.8.0/255.255.255.0, next hop 172.16.4.2
ISIS-SPF: Deleted NDB
ISIS-SPF: Compute L1 SPT
ISIS-SPF: Move 0000.0C0A.2C51.00-00 to PATHS, metric 0
ISIS-SPF: thru 2147483647/2147483647/2147483647, delay 0/0/0, mtu 2147483647/214
7483647/2147483647, hops 0/0/0, ticks 0/0/0
ISIS-SPF: Add 0000.0C76.5B7C.02-00 to TENT, metric 10
ISIS-SPF:  Next hop local
ISIS-SPF: Add 0000.0C0A.2C51 to L1 route table, metric 0
ISIS-SPF: Move 0000.0C76.5B7C.02-00 to PATHS, metric 10
ISIS-SPF: thru 2147483647/2147483647/2147483647, delay 0/0/0, mtu 2147483647/214
7483647/2147483647, hops 0/0/0, ticks 0/0/0
ISIS-SPF: considering adj to 0000.0C76.5B7C (Ethernet0) metric 10
ISIS-SPF:   (accepted)
ISIS-SPF: Add 0000.0C76.5B7C.00-00 to TENT, metric 10
ISIS-SPF:   Next hop 0000.0C76.5B7C (Ethernet0)
ISIS-SPF: Move 0000.0C76.5B7C.00-00 to PATHS, metric 10
ISIS-SPF: Add 172.16.4.0/255.255.255.0 to IP route table, metric 20
ISIS-SPF: N0C76.5B7C/172.16.4.2 (Ethernet0) (rejected)
ISIS-SPF: Add 0000.0C76.5B7C to L1 route table, metric 10
ISIS-SPF:   Next hop 0000.0C76.5B7C (Ethernet0)
ISIS-SPF: Aging L1 LSP 1 (0000.0C0A.2C51.00-00), version 126
ISIS-SPF: Aging L1 LSP 2 (0000.0C76.5B7C.02-00), version 127
ISIS-SPF: Aging L1 LSP 3 (0000.0C76.5B7C.00-00), version 133
```

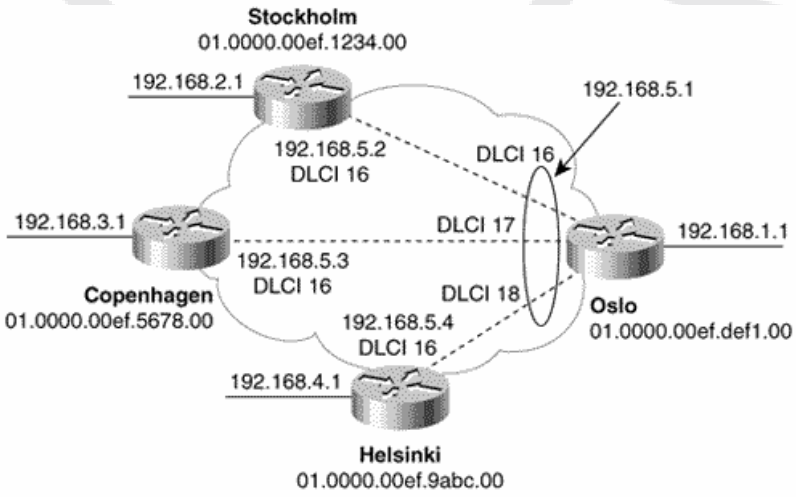
3. **debug isis spf-statistics**:显示关于SPF计算它本身的统计信息,如下:

```
Geneva#debug isis spf-statistics
IS-IS SPF Timing and Statistics Data debugging is on
Geneva#
ISIS-Stats: Compute L1 SPT
ISIS-Stats: Complete L1 SPT, Compute time 0.008, 3 nodes, 2 links on SPT, 0 suspends
ISIS-Stats: Compute L1 SPT
ISIS-Stats: Complete L1 SPT, Compute time 0.008, 3 nodes, 2 links on SPT, 0 suspends
```

如果你怀疑IS-IS路由器的LSDB信息的同步发生了问题,可以检查LSP ID和校验和.使用下面2个命令进行调试:

1. **debug isis update-packets**:显示路由器发送和接收SNP和LSP的相关信息
2. **debug isis snp-packets**:显示路由器发送和接收的PSNP和CSNP的相关信息

Case Study: Integrated IS-IS on NBMA Networks



路由器Oslo:

```
interface Serial0
ip address 192.168.5.1 255.255.255.0
ip router isis
encapsulation frame-relay
frame-relay interface-dlci 16
```

```
frame-relay interface-dlci 17
frame-relay interface-dlci 18
```

路由器Stockholm:

```
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.16 point-to-point
ip address 192.168.5.2 255.255.255.0
ip router isis
frame-relay interface-dlci 16
```

路由器Copenhagen:

```
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.16 point-to-point
ip address 192.168.5.3 255.255.255.0
ip router isis
frame-relay interface-dlci 16
```

路由器Helsinki:

```
interface Serial0
no ip address
encapsulation frame-relay
!
interface Serial0.16 point-to-point
ip address 192.168.5.4 255.255.255.0
ip router isis
frame-relay interface-dlci 16
```

除路由器Oslo以外的路由器都配置了子接口,而默认情况下,Cisco路由器的串行接口在封装帧中继的时候都被看作是多点接口,因此其他的3个路由器发送的是Point-to-point IS-IS Hello PDU,而路由器Oslo发送的是L1和L2的IS-IS LAN Hello PDU

解决办法是把路由器Oslo的串行接口配置为点到点的子接口,并且要注意的是每条PVC要位于不同的子网内

Route Redistribution

Principles of Redistribution

路由再发布的作用:可以使得多种路由协议之间,多重厂商环境中进行路由信息交换

Metrics

在做路由再发布的时候要考虑到的一个问题是:

metric.比如把OSPF路由再发布到EIGRP里,EIGRP和OSPF之间没有办法理解对方的metric,

因此在做路由再发布之前,要分配一个对方可以理解的metric,

Administrative Distances

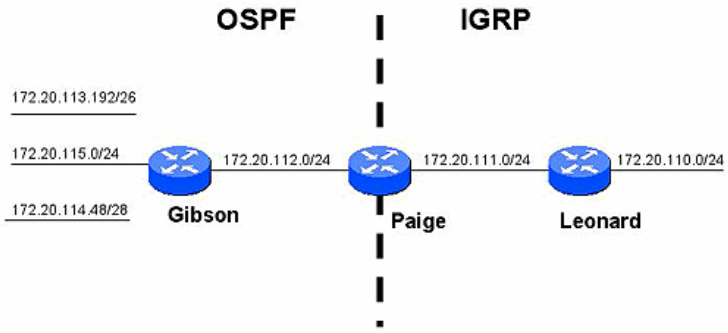
根据管理距离(AD)来判定多种协议学来的路由.Cisco默认的几种路由协议的AD如下:

1. 直连接口:0
2. 静态路由:1(有个例外,使用接口来代替下1跳地址的时候它会被认为是直连接口)
3. EIGRP汇总路由:5
4. External(外部) BGP:20
5. EIGRP:90
6. IGRP:100
7. OSPF:110
8. IS-IS:115
9. RIP:120
10. EGP:140
11. External(外部) EIGRP:170
12. Internal(内部) BGP:200
13. 未知:255

Redistributing from Classless to Classful Protocols

OSPF是基于无类的路由协议,将IGRP再发布到OSPF以后,路由器Paige它可以知道OSPF路由域和IGRP路由域的所有子网信息;而路由器Leonard只能学习到OSPF中掩码为/24的子网,因为IGRP是基于类的路由协议

Configuring Redistribution



路由器Paige的IGRP配置如下:

```
Paige(config)#router igrp 1
Paige(config-router)#redistribute ospf 1 metric 10000 100 255 1 1500
Paige(config-router)#passive-interface Ethernet1
Paige(config-router)#network 172.20.0.0
```

如上把OSPF(源路由协议)向IGRP(接受再发布的路由协议)再发布,同时分配了该路由的metric,

10000:带宽

100:延迟

255:可靠性

1:负载

1500:MTU

路由器Paige的OSPF配置如下:

```
Paige(config)#router ospf 1
Paige(config-router)#redistribute igrp 1 metric 30 metric-type 1 subnets
Paige(config-router)#network 172.20.112.2 0.0.0.0 area 0
```

如上是把IGRP再发布到OSPF中去,指定metric为30(OSPF的metric标准为cost),经过再发布以后,路由器Paige就成为了ASBR,经过再发布的IGRP路由是作为外部路由宣告进OSPF路由域的,同时使用metric-type命令指定外部路由类型为E1.subnets参数只在把路由再发布到OSPF中使用,它指明经过再发布后的子网的细节信息

另一种分配metric的方法是使用default-metric命令,比如刚才才把IGRP再发布到OSPF里的配置也可以写成下面的形式:

```
Paige(config)#router ospf 1
Paige(config-router)#redistribute igrp 1 metric-type 1 subnets
Paige(config-router)#default-metric 30
Paige(config-router)#network 172.20.112.2 0.0.0.0 area 0
```

2种不同的配置其实是相同的效果,

default-metric命令的优点是,当要再发布多种路由协议的时候,可以同时指定这些经过再发布的路由的metric

```
Paige(config)#router ospf 1
Paige(config-router)#redistribute igrp 1 metric-type 1 subnets
Paige(config-router)#redistribute rip metric-type 1 subnets
Paige(config-router)#redistribute eigrp 2 metric-type 1 subnets
Paige(config-router)#default-metric 30
Paige(config-router)#network 172.20.112.2 0.0.0.0 area 0
```

这里使用default-metric 30同时指定了再发布到OSPF里的RIP,IGRP和EIGRP路由的metric都为30

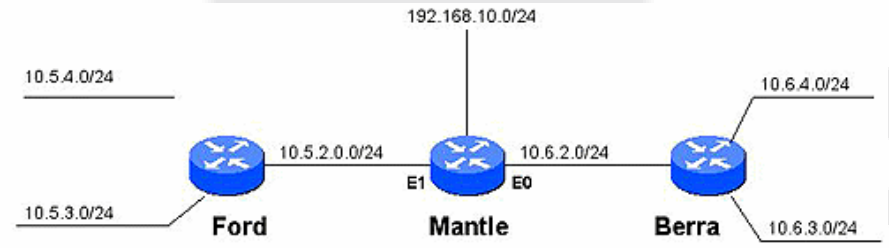
不过这2种分配metric的命令可以结合在一起使用,如下:

```
Paige(config)#router igrp 1
Paige(config-router)#redistribute ospf 1
Paige(config-router)#redistribute rip metric 50000 500 255 1 1500
```

```
Paige(config-router)#redistribute eigrp 2
Paige(config-router)#default-metric 10000 100 255 1 1500
Paige(config-router)#passive-interface Ethernet1
Paige(config-router)#network 172.20.0.0
```

如果metric和default-metric命令没有指定具体的参数,再发布到OSPF里的路由默认的metric为20,
而其他的路由协议为0.metric为0不能被RIP正确理解,并且与IGRP和EIGRP不兼容,IS-IS可以正确理解

Case Study: Redistributing IGRP and RIP

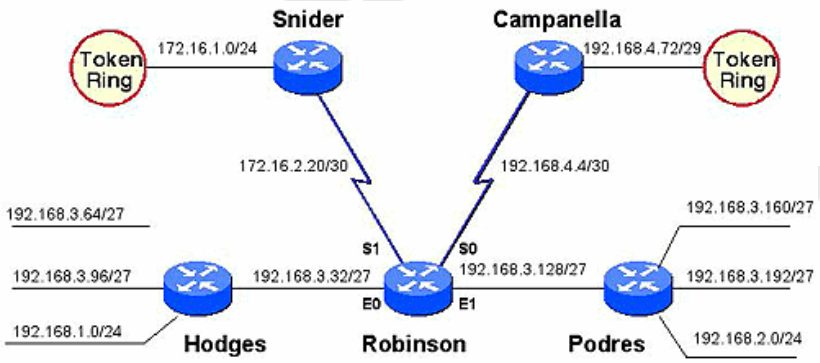



```
router rip
 redistribute igrp 1 metric 5
 passive-interface Ethernet1
 network 10.0.0.0
!
router igrp 1
 redistribute connected
 redistribute rip
 default-metric 1000 100 255 1 1500
 passive-interface Ethernet0
 network 10.0.0.0
```

注意路由器Mantle还连接了个stub网络192.168.10.0/24,如果要求把这个网络宣告进IGRP路由域,
可以使用redistribute connected命令

```
router rip
 redistribute connected metric 5
 redistribute igrp 1 metric 5
 passive-interface Ethernet1
 network 10.0.0.0
!
router igrp 1
 redistribute connected
 redistribute rip
 default-metric 1000 100 255 1 1500
 passive-interface Ethernet0
 network 10.0.0.0
```

Case Study: Redistributing EIGRP and OSPF



在这个拓扑图中,路由器Hodges运行OSPF进程1,路由器Podres运行EIGRP进程1,,路由器Snider和Campanella运行EIGRP进程2,路由器Robinsion配置如下:

```
router eigrp 1
redistribute ospf 1 metric 1000 100 1 255 1500
redistribute eigrp 2
passive-interface Ethernet0
network 192.168.3.0
!
router eigrp 2
redistribute ospf 1 metric 1000 100 1 255 1500
redistribute eigrp 1
network 192.168.4.0
network 172.16.0.0
!
router ospf 1
redistribute eigrp 1 metric 50
redistribute eigrp 2 metric 100
network 192.168.3.33 0.0.0.0 area 0
```

注意不同进程的EIGRP的再发布不需要分配metric,因为它们是同一种路由协议,可以相互理解对方的metric

查看路由器Podres的路由表,如下:

```
Podres#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is not set

D EX 192.168.1.0/24 [170/2611200] via 192.168.3.129, 00:39:14, Ethernet0
C    192.168.2.0/24 is directly connected, Ethernet3
     192.168.3.0/24 is variably subnetted, 7 subnets, 3 masks
D EX    192.168.3.96/27 [170/2611200] via 192.168.3.129, 00:41:18, Ethernet0
D EX    192.168.3.64/27 [170/2611200] via 192.168.3.129, 00:41:18, Ethernet0
D       192.168.3.32/27 [90/307200] via 192.168.3.129, 00:44:06, Ethernet0
D       192.168.3.0/24 is a summary, 00:52:21, Null0
C       192.168.3.192/27 is directly connected, Ethernet2
C       192.168.3.160/27 is directly connected, Ethernet1
C       192.168.3.128/27 is directly connected, Ethernet0
     192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
D EX    192.168.4.72/29 [170/2211584] via 192.168.3.129, 00:07:25, Ethernet0
D EX    192.168.4.4/30 [170/281600] via 192.168.3.129, 00:07:25, Ethernet0
D EX    192.168.4.0/24 [170/2195456] via 192.168.3.129, 00:07:25, Ethernet0
     172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
D EX    172.16.2.20/30 [170/281600] via 192.168.3.129, 00:07:27, Ethernet0
D EX    172.16.0.0/16 [170/2195456] via 192.168.3.129, 00:07:27, Ethernet0
D EX    172.16.1.0/24 [170/2211584] via 192.168.3.129, 00:07:27, Ethernet0
Podres#
```

再查看路由器Hodges的路由表,如下:

```
Hodges#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, Ethernet2
O E2 192.168.2.0/24 [110/50] via 192.168.3.33, 00:11:59, Ethernet0
     192.168.3.0/27 is subnetted, 3 subnets
C       192.168.3.96 is directly connected, Ethernet1
C       192.168.3.64 is directly connected, Ethernet3
C       192.168.3.32 is directly connected, Ethernet0
Hodges#
```

为什么只有1条为指向192.168.2.0/24的E2外部OSPF路由?答案是把其他类型的路由再发布到OSPF里的时候没有使用参数subnets,因此将把没有连接到做路由再发布的路由器(Robinson)的主网络地址(192.168.2.0/24)做再发布.使用subnets参数

```
router ospf 1
redistribute eigrp 1 metric 50 subnets
redistribute eigrp 2 metric 100 subnets
network 192.168.3.33 0.0.0.0 area 0
```

```
Hodges#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

C    192.168.1.0/24 is directly connected, Ethernet2
O E2 192.168.2.0/24 [110/50] via 192.168.3.33, 00:17:31, Ethernet0
     192.168.3.0/27 is subnetted, 6 subnets
C       192.168.3.96 is directly connected, Ethernet1
C       192.168.3.64 is directly connected, Ethernet3
C       192.168.3.32 is directly connected, Ethernet0
O E2    192.168.3.192 [110/50] via 192.168.3.33, 00:02:51, Ethernet0
O E2    192.168.3.160 [110/50] via 192.168.3.33, 00:02:51, Ethernet0
O E2    192.168.3.128 [110/50] via 192.168.3.33, 00:00:36, Ethernet0
     192.168.4.0/24 is variably subnetted, 3 subnets, 3 masks
O E2    192.168.4.72/29 [110/100] via 192.168.3.33, 00:00:19, Ethernet0
O E2    192.168.4.4/30 [110/100] via 192.168.3.33, 00:00:19, Ethernet0
O E2    192.168.4.0/24 [110/100] via 192.168.3.33, 00:00:19, Ethernet0
     172.16.0.0/16 is variably subnetted, 3 subnets, 3 masks
O E2    172.16.2.20/30 [110/100] via 192.168.3.33, 00:00:20, Ethernet0
O E2    172.16.0.0/16 [110/100] via 192.168.3.33, 00:00:20, Ethernet0
O E2    172.16.1.0/24 [110/100] via 192.168.3.33, 00:00:20, Ethernet0
Hodges#
```

如上,所有子网都能被路由器Hodges学习到

另外,OSPF外部路由类型默认为E2,如果要把上面的OSPF外部路由类型更改为E1,可以在再发布外部路由到OSPF中的时候使用命令metric-type 1,如下:

```
Robinson(config)#router ospf 1
Robinson(config-router)#redistribute eigrp 1 metric 50 subnets
Robinson(config-router)#redistribute eigrp 2 metric 100 metric-type 1 subnets
Robinson(config-router)#network 192.168.3.33 0.0.0.0 area 0
```

Case Study: Redistribution and Route Summarization

Cisco的EIGRP,OSPF和IS-IS都可以对再发布的路由进行路由汇总.拓扑图如下:

可以在OSPF进程下使用summary-address指定汇总的地址和掩码,不过这个命令是用在ASBR上的,

ABR上的路由汇总是使用area <area-id> range <address>命令,如下:

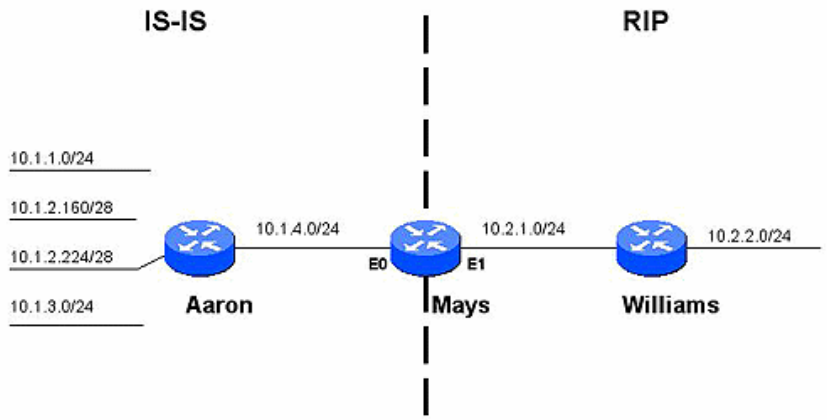
```
router eigrp 1
redistribute ospf 1 metric 1000 100 1 255 1500
redistribute eigrp 2
passive-interface Ethernet0
network 192.168.3.0
!
router eigrp 2
redistribute eigrp 1
network 192.168.4.0
network 172.16.0.0
!
router ospf 1
summary-address 192.168.3.128 255.255.255.128
summary-address 172.16.0.0 255.255.0.0
```

```
redistribute eigrp 1 metric 50 subnets
redistribute eigrp 2 metric 100 metric-type 1 subnets
network 192.168.3.33 0.0.0.0 area 0
```

对于EIGRP的汇总,是基于接口的,使用命令ip summary-address eigrp {process-id},如下:

```
interface Ethernet0
ip address 192.168.3.33 255.255.255.224
!
interface Ethernet1
ip address 192.168.3.129 255.255.255.224
ip summary-address eigrp 1 192.168.3.0 255.255.255.128
ip summary-address eigrp 1 172.16.0.0 255.255.0.0
ip summary-address eigrp 1 192.168.4.0 255.255.255.0
!
interface Serial0
ip address 192.168.4.5 255.255.255.252
ip summary-address eigrp 2 192.168.3.0 255.255.255.0
!
interface Serial1
ip address 172.16.2.21 255.255.255.252
ip summary-address eigrp 2 192.168.0.0 255.255.0.0
!
router eigrp 1
redistribute ospf 1 metric 1000 100 1 255 1500
redistribute eigrp 2
passive-interface Ethernet0
network 192.168.3.0
!
router eigrp 2
redistribute eigrp 1
network 192.168.4.0
network 172.16.0.0
!
router ospf 1
summary-address 192.168.3.128 255.255.255.128
summary-address 172.16.0.0 255.255.0.0
redistribute eigrp 1 metric 50 subnets
redistribute eigrp 2 metric 100 metric-type 1 subnets
network 192.168.3.33 0.0.0.0 area 0
```

Case Study: Redistributing IS-IS and RIP



路由器Mays的配置如下:

```
router isis
redistribute rip metric 0 metric-type internal level-2
net 01.0001.0000.0c76.5432.00
!
router rip
redistribute isis level-1-2 metric 1
passive-interface Ethernet0
network 10.0.0.0
```

参数internal为内部路由的含义,默认为内部,并且为L1.路由器Aaron的路由表如下:

```
Aaron#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     10.0.0.0/8 is variably subnetted, 7 subnets, 2 masks
C       10.1.3.0/24 is directly connected, Ethernet4
i L2    10.2.1.0/24 [115/10] via 10.1.4.2, Ethernet0
i L2    10.2.2.0/24 [115/10] via 10.1.4.2, Ethernet0
C       10.1.1.0/24 is directly connected, Ethernet1
C       10.1.4.0/24 is directly connected, Ethernet0
C       10.1.2.160/28 is directly connected, Ethernet2
C       10.1.2.224/28 is directly connected, Ethernet3
Aaron#
```

路由器Aaron所连的子网可以汇总为10.2.0.0/16,再发布到IS-IS里的路由和OSPF一样,也是使用summary-address命令,但是还要额外的指定IS-IS的Level,如下:

```
router isis
summary-address 10.2.0.0 255.255.0.0 level-1
redistribute rip metric 0 metric-type external level-1
net 01.0001.0000.0c76.5432.00
!
router rip
redistribute isis level-1-2 metric 1
passive-interface Ethernet0
network 10.0.0.0
```

Case Study: Redistributing Static Routes

```
router rip
redistribute static metric 1
redistribute isis level-1-2 metric 1
passive-interface Ethernet0
network 10.0.0.0
```

Default Routes and ODR

Fundamentals of Default Routes and ODR

默认路由(Default Route)最大的好处就是减少路由表的条目,从而减小了路由表体积,降低了对路由器CPU资源的占用

ODR(On-Demand Routing,按需路由)是从Cisco IOS版本11.2出现的,它为Cisco所私有,并且不是真正意义上路由协议.它依赖于Cisco发现协议(CDP,Cisco Discovery Protocol)

```
Router#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is not set

     192.168.1.0/24 is variably subnetted, 3 subnets, 2 masks
o       192.168.1.40/30 [160/1] via 192.168.1.37, 00:00:27, Serial0
C       192.168.1.36/30 is directly connected, Serial0
C       192.168.1.192/27 is directly connected, Ethernet1
o    192.168.3.0/24 [160/1] via 192.168.1.37, 00:00:27, Serial0
     192.168.4.0/24 is variably subnetted, 2 subnets, 2 masks
o       192.168.4.48/29 [160/1] via 192.168.1.37, 00:00:27, Serial0
o       192.168.4.128/27 [160/1] via 192.168.1.37, 00:00:27, Serial0
Router#
```

标记为o的代表ODR,它的管理距离为160,并且metric永远不会超过1

Configuring Default Routes and ODR

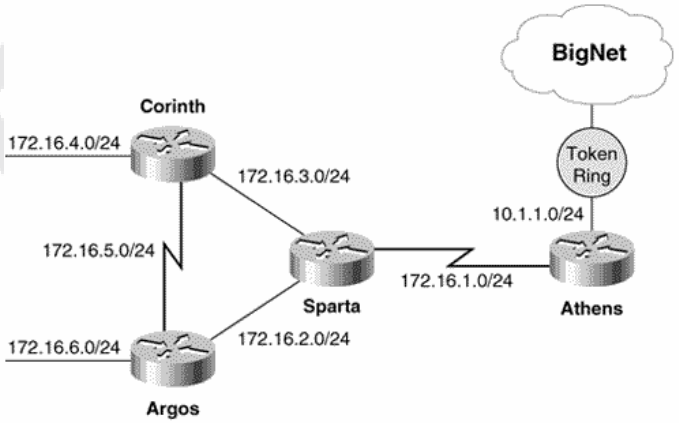
Case Study: Static Default Routes

```
Memphis(config)#ip classless
Memphis(config)#ip route 0.0.0.0 0.0.0.0 192.168.1.82
```

配置的静态默认路由，RIP,IGRP和EIGRP将自动宣告默认路由(OSPF和IS-IS要做额外的配置)

Case Study: The default-network Command

另一种配置默认路由的方法是使用ip default-network命令.



路由器Athens配置如下:

```
router rip
network 172.16.0.0
!
ip classless
ip default-network 10.0.0.0
```

```
Athens#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route

Gateway of last resort is not set

*    10.0.0.0/8 is subnetted, 1 subnets
C       10.1.1.0 is directly connected, TokenRing0
R    192.168.1.0/24 [120/2] via 172.16.1.2, 00:00:12, Serial0
     172.16.0.0/16 is subnetted, 6 subnets
R       172.16.4.0 [120/2] via 172.16.1.2, 00:00:12, Serial0
R       172.16.5.0 [120/2] via 172.16.1.2, 00:00:12, Serial0
R       172.16.6.0 [120/2] via 172.16.1.2, 00:00:12, Serial0
C       172.16.1.0 is directly connected, Serial0
R       172.16.2.0 [120/1] via 172.16.1.2, 00:00:12, Serial0
R       172.16.3.0 [120/1] via 172.16.1.2, 00:00:12, Serial0
Athens#
```

可以看到10.0.0.0被标记为侯选的默认路由,但是没有指定默认网关,原因是路由器Athens就是到这个默认网络的网关,即使在配置RIP的时候不声明network 10.0.0.0,ip default-network命令会使得路由器Athens宣告一个默认网络

```
Sparta#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR

Gateway of last resort is 172.16.1.1 to network 0.0.0.0

R    192.168.1.0/24  [120/1] via 172.16.2.2, 00:00:10, Ethernet0
                     [120/1] via 172.16.3.2, 00:00:14, Ethernet1
     172.16.0.0/24 is subnetted, 6 subnets
R       172.16.4.0 [120/1] via 172.16.3.2, 00:00:14, Ethernet1
R       172.16.5.0 [120/1] via 172.16.3.2, 00:00:14, Ethernet1
R       172.16.6.0 [120/1] via 172.16.2.2, 00:00:10, Ethernet0
C       172.16.1.0 is directly connected, Serial0
C       172.16.2.0 is directly connected, Ethernet0
C       172.16.3.0 is directly connected, Ethernet1
R*   0.0.0.0/0 [120/1] via 172.16.1.1, 00:00:17, Serial0
Sparta#
```

对于IGRP和EIGRP的默认路由稍微有些不同,它们不能理解0.0.0.0,所以通常会宣告一个真实的地址作为外部路由,然后这个外部路由会被IGRP和EIGRP理解成默认路由

如果路由器Athens运行的是IGRP,如下:

```
router igrp 1
network 10.0.0.0
network 172.16.0.0
!
ip classless
ip default-network 10.0.0.0
```

注意和配置RIP不同的是,在配置IGRP的时候增加了network 10.0.0.0语句

Case Study: The default-information originate Command

可以看到路由器Athens上设置的有默认路由,可路由器Sparta上却没有,这时就要用到default-information originate命令,告诉该OSPF路由器成为1个ASBR(默认路由以类型5的LSA被宣告进OSPF路由域中),并指定metric和OSPF外部路由类型,如下:

```
Athens(config)#router ospf 1
```

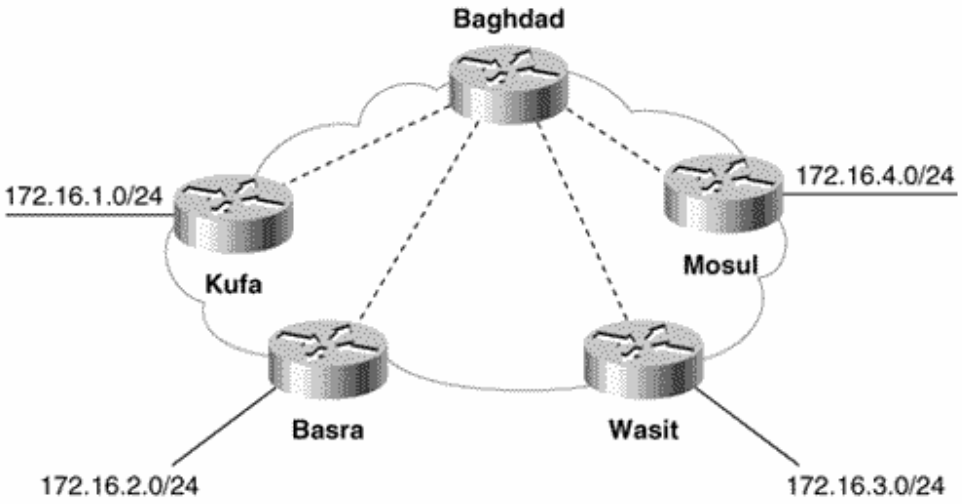
```
Athens(config-router)#network 172.16.0.0 0.0.255.255 area 0
Athens(config-router)#default-information originate metric 10 metric-type 1
Athens(config)#ip classless
Athens(config)#ip route 0.0.0.0 0.0.0.0 10.1.1.2
```

Case Study: Configuring On-Demand Routing

ODR的启用只需输入命令**router odr**无需指明网络和其他参数.
ODR传送地址前缀，而不是整个地址，因此路由器必须支持VLSM
ODR可以被重分布到其他路由协议
ODR的管理距离为160，度量永远都不会超过1

ODR路由得传输机制是Cisco发现协议（CDP），路由器上用**cdp run**启用，用**cdp enable**在特定的接口上启用
CDP运行在任何支持子网访问协议（SNAP）的介质上，即ODR也依赖SNAP的支持

在中心路由器上用**router odr**启用
分支路由器上配置一条指向中心路由器的静态路由就可以了



Route Filtering

Configuring Route Filters

路由过滤可以通过下面2种方法实现:
1. 使用distribute-list过滤特定路由
2. 使用distance命令来控制路由的AD
路由过滤器的用途：把一个路由选择域分割成多个子域，在连接不同子域的路由器上过滤
建立路由防火墙

过滤器是基于访问列表的基础上用**distribute-list acl-no in/out** 接口/路由协议
NO. 是被应用的访问列表编号
在“路由协议”关键字中，仅有out是有意义的
由于链路状态协议不从自身路由表中通告路由，所以在“接口”关键字中用out是没有意义的
要过滤什么进程，就把过滤器放在什么进程下
例：在OSPF 1下过滤RIP，则在OSPF进程下用**distribute-list 10 in rip**
在OSPF 1下过滤OSPF 1，这在OSPF进程中用**distribute-list 10 in**

要在一个本来运行一种路由协议的网络中运用另一种路由协议时，为了防止出错和路由黑洞，如果新协议的管理距离小于旧协议，在新路由进程中用distance增大新协议的管理距离，等到网络中的每个路由器上新协议都配置好后再改回去，

再删除旧协议，最后用clear ip route * 清空路由表，让其重新学习

在路由进程中用**distance AD IP-addr wildcat-mask acl-NO.**

Route Maps

Basic Uses of Route Maps

route map和ACL很类似,它可以用于路由的再发布和策略路由,还经常使用在BGP中.策略路由(policy route)实际上是复杂的静态路由,静态路由是基于数据包的目标地址并转发到指定的下一跳路由器,策略路由还利用和扩展IP ACL链接,这样就可以提供更多功能的过滤和分类

route map的一些命令:

match命令可以和路由的再发布结合使用:

1. **match interface {type number} [...type number]**:匹配指定的下一跳路由器的接口的路由
2. **match ip address {ACL number|name} [...ACL number|name]**:匹配ACL所指定的目标IP地址的路由
3. **match ip next-hop {ACL number|name} [...ACL number|name]**:匹配ACL所指定的下一跳路由器地址的路由
4. **match ip route-source {ACL number|name} [...ACL number|name]**:匹配ACL所指定的路由器所宣告的路由
5. **match metric {metric-value}**:匹配指定metric大小的路由
6. **match route-type {internal|external[type-1|type-2]|level-1|level-2}**:匹配指定的OSPF,EIGRP或IS-IS的路由类型的路由
7. **match tag {tag-value} [...tag-value]**:匹配带有标签(tag)的路由

set命令也可以和路由的再发布一起使用:

1. set level {level-1|level-2|level-1-2|stub-area|backbone}:设置IS-IS的Level,或OSPF的区域,匹配成功的路由将被再发布到该区域
2. set metric {metric-value|bandwidth delay RELY load MTU}:为匹配成功的路由设置metric大小
3. set metric-type {internal|external|type-1|type-2}:为匹配成功的路由设置metric的类型,该路由将被再发布到OSPF或IS-IS
4. set next-hop {next-hop}:为匹配成功的路由指定下一跳地址
5. set tag {tag-value}:为匹配成功的路由设置标签

match命令还可以和策略路由一起使用:

1. match ip address {ACL number|name} […ACL number|name]:匹配ACL所指定的数据包的特征的路由
2. match length {min} {max}:匹配层3的数据包的长度

set命令也可以和策略路由一起使用:

1. set default interface {type number} […type number]:当不存在指向目标网络的显式路由(explicit route)的时候,为匹配成功的数据包设置出口接口
2. set interface {type number} […type number]:当存在指向目标网络的显式路由的时候,为匹配成功的数据包设置出口接口
3. set ip default next-hop {ip-address} […ip-address]:当不存在指向目标网络的显式路由的时候,为匹配成功的数据包设置下一跳路由器地址
4. set ip precedence {precedence}:为匹配成功的IP数据包设置服务类型(Type of Service,ToS)的优先级
5. set ip tos {tos}:为匹配成功的数据包设置服务类型的字段的TOS位

Configuring Route Maps

route map是通过名字来标识的,每个route map都包含许可或拒绝操作以及一个序列号,序列号在没有给出的情况下默认是10,并且route map允许有多个陈述,如下:

```
Linus(config)#route-map Hagar 20
```

```
Linus(config-route-map)#match ip address 111
Linus(config-route-map)#set metric 50
Linus(config-route-map)#route-map Hagar 15
Linus(config-route-map)#match ip address 112
Linus(config-route-map)#set metric 80
```

尽管先输入的是20,后输入的是15,IOS将把15放在20之前.还可以允许删除个别陈述,如下:

```
Linus(config)#no route-map Hagar 15
```

在删除的时候要特别小心,假如你输入了**no route-map Hegar**而没有指定序列号,那么整个route map将被删除.并且如果在添加match和set语句的时候没有指定序列号的话,那么它们仅仅会修改陈述10.在匹配的时候,从上到下,如果匹配成功,将不再和后面的陈述进行匹配,指定操作将被执行

关于拒绝操作,是依赖于route map是使用再路由的再发布中还是策略路由中,如果是在策略路由中匹配失败(拒绝),那么数据包将按正常方式转发;如果是用于路由再发布,并且匹配失败(拒绝),那么路由将不会被再发布

如果数据包没有找到任何匹配,和ACL一样,route map末尾也有个默认的隐含拒绝所有的操作,如果是在策略路由中匹配失败(拒绝),那么数据包将按正常方式转发;如果是用于路由再发布,并且匹配失败(拒绝),那么路由将不会被再发布

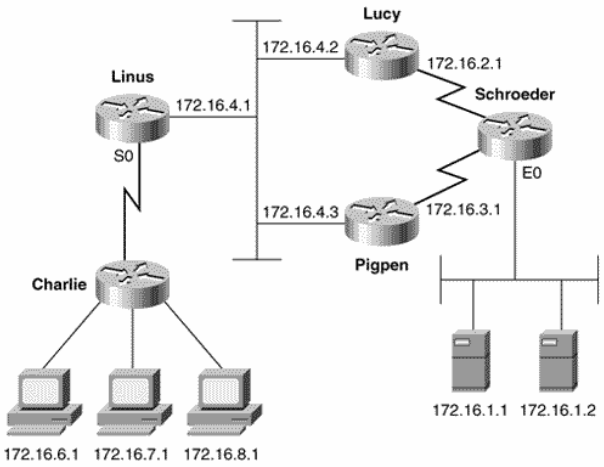
如果route map的陈述中没有match语句,那么默认的操作是匹配所有的数据包和路由;每个route map的陈述可能有多个match和set语句,如下:

```
! route-map Garfield permit 10
match ip route-source 15
match interface Serial0
set metric-type type-1
set next-hop 10.1.2.3
!
```

在这里,为了执行**set**语句,每个**match**语句中都必须进行匹配

Case Study: Policy Routing

使用**ip policy route-map**命令定义策略路由,这个命令是基于接口的,并且只对进站(incoming)的数据包有影响



现在要求在路由器Linus上使用策略路由,来自网络172.16.6.0的数据包只被转发到路由器Lucy上;而来自172.16.7.0的数据包只转发到路由器Pigpen上.相应配置如下:

```
interface Serial0
ip address 172.16.5.1 255.255.255.0
ip policy route-map Sally
!
access-list 1 permit 172.16.6.0 0.0.0.255
access-list 2 permit 172.16.7.0 0.0.0.255
```

```
!
route-map Sally permit 10
match ip address 1
set ip next-hop 172.16.4.2
!
route-map Sally permit 15
match ip address 2
set ip next-hop 172.16.4.3
```

至于172.16.8.0的数据包将按正常方式被转发,并做负载均衡

再假设策略规定从172.16.1.0的服务器上发出的FTP和Telnet流量分别转发到Lucy和Pigpen上,

在路由器Schroeder上做如下配置:

```
interface Ethernet0
ip address 172.16.1.4 255.255.255.0
ip policy route-map Rerun
!
access-list 105 permit tcp 172.16.1.0 0.0.0.255 eq ftp any
access-list 105 permit tcp 172.16.1.0 0.0.0.255 eq ftp-data any
access-list 106 permit tcp 172.16.1.0 0.0.0.255 eq telnet any
!
route-map Rerun permit 10
match ip address 105
set ip next-hop 172.16.2.1
!
route-map Rerun permit 20
match ip address 106
set ip next-hop 172.16.3.1
```

还可以根据数据包的大小来做策略路由,在路由器Schroeder上做如下配置:

```
interface Ethernet0
ip address 172.16.1.4 255.255.255.0
ip policy route-map Woodstock
!
route-map Woodstock permit 20
match length 1000 1600
set ip next-hop 172.16.2.1
!
route-map Woodstock permit 30
match length 0 400
set ip next-hop 172.16.3.1
```

对于路由器自己所产生的流量,要对它们做策略路由的话,使用ip local policy route-map,该命令是基于全局配置模式下的,如下,在路由器Schroeder上的配置:

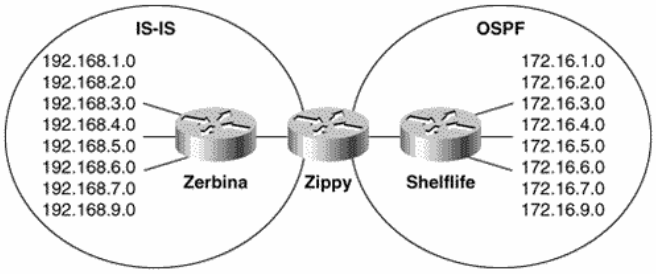
```
interface Ethernet0
ip address 172.16.1.4 255.255.255.0
ip policy route-map Woodstock
!
ip local policy route-map Woodstock
!
access-list 120 permit ip any 172.16.1.0 0.0.0.255
access-list 120 permit ospf any any
!
route-map Woodstock permit 10
match ip address 120
!
route-map Woodstock permit 20
```

```
match length 1000 1600
set ip next-hop 172.16.2.1
!
route-map Woodstock permit 30
match length 0 400
set ip next-hop 172.16.3.1
```

只有match语句没有set语句,如果没有permit 10,那么OSPF Hello包会和permit 30匹配并被转发到路由器Pigpen上,这将切断路由器Lucy和Schroeder之前的邻接关系;如果匹配permit 10,OSPF Hello包将被正常转发,不影响邻接关系

Case Study: Route Maps and Redistribution

在路由的再发布中使用route map,只需要在使用redistribute的时候调用相应的route map即可.拓扑图如下:



现在要求只相互再发布第三个8位位组为奇数的路由,在路由器Zippy上做如下配置:

```
router ospf 1
redistribute isis level-1 metric 20 subnets route-map Griffy
network 172.16.10.2 0.0.0.0 area 5
!
router isis
redistribute ospf 1 metric 25 route-map Toad metric-type internal level-2
net 47.0001.1234.5678.9056.00
!
access-list 1 permit 192.168.2.0
access-list 1 permit 192.168.4.0
access-list 1 permit 192.168.6.0
access-list 2 permit 172.16.1.0
access-list 2 permit 172.16.3.0
access-list 2 permit 172.16.5.0
access-list 2 permit 172.16.7.0
access-list 2 permit 172.16.9.0
!
route-map Griffy deny 10
match ip address 1
!
route-map Griffy permit 20
!
route-map Toad permit 10
match ip address 2
```

其中permit 20没有match命令,因此默认匹配所有地址(这个例子中有多种配置,不妨自己写写其他的配置)

上个例子使用distribution-list做简单的路由过滤也可达到相同效果,但是route map还可以有更高级的功能,如下配置:

```
router ospf 1
redistribute isis level-1 metric 20 subnets route-map Griffy
network 172.16.10.2 0.0.0.0 area 5
!
router isis
redistribute ospf 1 metric 25 route-map Toad metric-type internal level-2
```

```
 net 47.0001.1234.5678.9056.00
 !
 ip classless
 access-list 1 permit 192.168.2.0
 access-list 1 permit 192.168.4.0
 access-list 1 permit 192.168.6.0
 access-list 2 permit 172.16.9.0
 access-list 2 permit 172.16.5.0
 access-list 2 permit 172.16.7.0
 access-list 2 permit 172.16.1.0
 access-list 2 permit 172.16.3.0
 !
 route-map Griffy permit 10
 match ip address 1
 set metric-type type-1
 !
 route-map Griffy permit 20
 !
 route-map Toad permit 10
 match ip address 2
 set metric 15
 set level level-1
 !
 route-map Toad permit 20
```

如上就是使用route map来控制再分布的OSPF外部路由类型和IS-IS路由Level

关于Route-map的更多应用参见后文BGP中的策略控制

Border Gateway Protocol

Who needs BGP

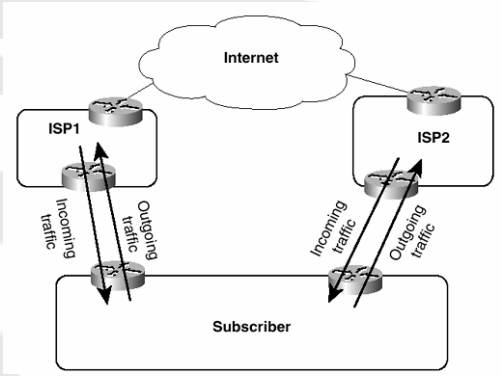
BGP 是一种路径矢量路由协议，用于传输自治系统间的路由信息，

BGP 在启动的时候传播整张路由表，以后只传播网络变化的部分触发更新

它采用 TCP 连接传送信息，端口号为 179

在 Internet 上，BGP 需要通告的路由数目极大，由于 TCP 提供了可靠的传送机制，同时 TCP 使用滑动窗口机制，使得 BGP 可以不断地

发送分组，而无需像 OSPF 或 EIGRP 那样停止发送并等待确认。



使用 BGP 一般有如下情况：

1. 一个 AS 允许包穿越它，到达其他的 AS
2. 一个 AS 连接多个 AS
3. 必须对数据流进入和离开 AS 进行控制

不使用 BGP 一般有如下情况：

1. AS 只有一个出口
2. AS 的所有出口均为 1 个 ASP
3. 路由性能不高，内存较小，CPU 较慢，带宽不大

对于如右图的多 ISP 连接，需要注意：

1. 必须劝说最初供应商通告将他的 CIDR 块打一个洞，通告一条更精细的路由
2. 必须劝说第二个供应商公布属于不同供应商的地址空间
3. 两个供应商必须在用户公布这块地址上愿意紧密合作
4. 如果用户地址空间小于/19 一些骨干 ISP 不会接受此路由

   注：一般前缀小于/19 的地址称为全球路由可达地址，因为骨干 ISP 通常为了控制骨干级路由表大小，仅接受/19 或更小的前缀。但由于用户的抱怨，此限制逐渐放松

BGP Basic

BGP 是路径矢量协议，它使用一个 AS 号列表，数据包必须通过这些 AS 才能到达目的，同时对产生的 AS-path 做一定的策略。AS-Path 对于路由环路非常容易检测到，如果路由器接受到一条含有本地 AS 号的 AS-path，说明出现环路。

BGP 没有给出每个 AS 域内的拓扑结构，因此 BGP 只能看到 AS 树，而 IGP 只能看到 AS 域内拓扑结构，

下图是一个典型的 BGP 路由表

```
route-server>show ip bgp
BGP table version is 4639209, local router ID is 12.0.1.28
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*  3.0.0.0          192.205.31.225                         0 7018 701 80 i
*                   192.205.31.161                         0 7018 701 80 i
*>                  192.205.31.33                          0 7018 701 80 i
*                   192.205.31.97                          0 7018 701 80 i
*  4.0.0.0          192.205.31.225                         0 7018 1 i
*                   192.205.31.161                         0 7018 1 i
*>                  192.205.31.33                          0 7018 1 i
*                   192.205.31.97                          0 7018 1 i
*  6.0.0.0          192.205.31.226                         0 7018 568 721 1455 i
*                   192.205.31.225                         0 7018 568 721 1455 i
*                 192.205.31.161                 0 7018 701 6113 568 721 1455 i
*>                  192.205.31.34                          0 7018 568 721 1455 i
*                   192.205.31.33                          0 7018 568 721 1455 i
```

```
*                  192.205.31.97                0 7018 1239 568 721 1455 i
*  9.2.0.0/16      192.205.31.225               0 7018 1 1673 1675 i
*                  192.205.31.161               0 7018 701 1673 1675 i
 --More--
```

当某个特殊目的网络有并列的，等开销的路径时，Cisco 缺省执行 EBGP 只选择一条路径，但可以使用 **maximum-paths** 改变并行路径缺省的最大数目，但仅对 EBGP 有效。

BGP Message Types

BGP 具有 4 种消息类型

1. Open
2. Keepalive
3. Update
4. Notification

Open Message

TCP 对话建立以后，两个邻居都要发送一个 Open 消息，每个邻居都用该消息来标示自己，并规定自己的 BGP 运行参数

BGP version

它明确了发起者正在运行的 BGP 版本号(2,3,4)，可以通过 **neighbor version** 修改，缺省版本号为 4。

如果版本号不相同，路由器将自动降低版本号重发 Open 消息，直到版本一致

AS number

发起会话路由器的 AS 号，用于确认 EBGP 或者 IBGP 会话

Hold time

路由器必须收到一个 keepalive 或者更新消息之前允许经过的最大秒数。

Holdtime 必须是 0(在这种情况下,必须是没有发送 Keepalive)或者至少 3s

Cisco 默认的 holdtime 为 180s,如果两个邻居间 holdtime 不一致,选较短的那个做为两者可接受的时间

BGP router-ID

选取方式和 OSPF 相同,使用数值最大的 loopback 口地址,没有 loopback 则使用物理接口上数值最大的地址

Optional parameters

用于一些可选功能的支持.例如鉴别，多协议支持及路由刷新等

Keepalive Message

如果路由器接受了他在邻居的 Open 消息中的参数，它就会发送一个应答的 Keepalive 消息。

默认情况 Keepalive 间隔 60s，或者是达成一致的保持时间的 1/3

Update Message

Update message 用来公布可用的路由，撤销的路由或者两者兼顾

Network Layer Reachability Information (NLRI)

用来公布 IP 地址前缀和前缀长度的字节组，例如<19,198.24.160.0>

Path Attributes

该属性为 BGP 提供了选择最短路径，检查到路由环路以及决定路由策略的信息

Withdrawn Routes

用来描述已经变成不可达并正从业务中撤销的目的地址字节组(长度和前缀)

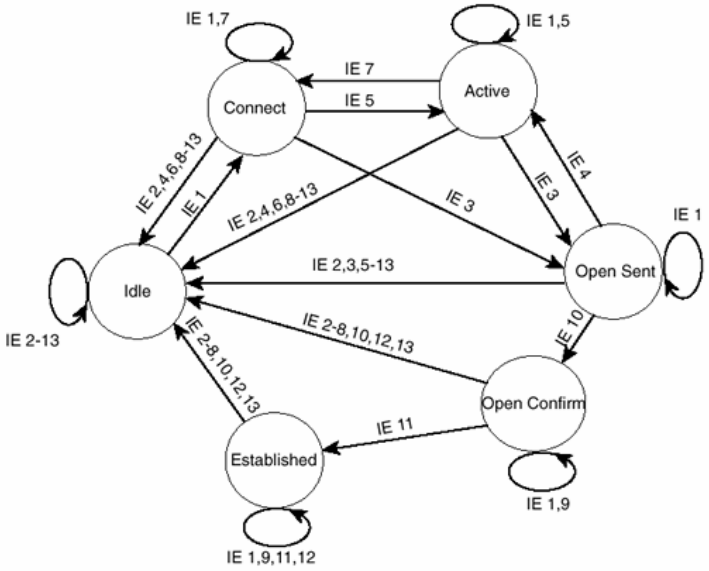
注：虽然 NLRI 字段可以包含多个前缀，但每一个更新消息只描述一条 BGP 路由(因为路径属性只描述一条路径，但该路径可能会到达多个目的地)

Notification Message

当检测到差错的时候就会发送 Notification 消息，通常会导致 BGP 连接终止

例如使用 Notification 消息进行 BGP 版本的协商

The BGP Finite State Machine



IE	描 述
1	BGP 开始
2	BGP 结束
3	BGP 传输连接打开
4	BGP 传输连接终止
5	BGP 传输连接打开失败
6	BGP 传输致命差错
7	重试连接计时器超时
8	持续时间终止
9	Keepalive 计时器终止
10	收到 Open 消息
11	收到 Keepalive 消息
12	收到 Update 消息
13	收到 Notification 消息

Idle State

a) BGP 通常以 Idle State 开始(此时拒绝接收所有入连接)。当一个开始事件出现,BGP 过程初始化所有 BGP 资源打开重试连接(ConnectRetry)计时器,初始化到邻居的 TCP 连接,接听来自邻居的 TCP 初始化消息并将它的状态转到 Connect 状态.

b) 开始事件是由一个操作者配置一个 BGP 过程,或者重置一个已经存在的过程或者路由器软件重置 BGP 过程引起

c) 一个差错的出现会将 BGP 过程的状态转为 Idle.路由器可能会试图发起另外一个开始事件.为了防止在持续差错条件下导致的摆动,在第一次转回到空闲状态后,路由器会自动开启重试连接计时器,当计时器终止后,路由器就会放弃重新开始 BGP.重试计时器第一次的时间为 60s,下一次为前一次的 2 倍 120s,成指数形式增加

Connect State

此状态下 BGP 过程会等到 TCP 连接完成以后再决定后续的动作.

1. 如果 TCP 连接建立成功,BGP 连接将 ConnectRetry 清零,完成初始化并给邻居发送一个 Open 消息,转移到 Open 状态
2. 如果 TCP 连接建立失败,BGP 继续监听由邻居发起的连接,重置 ConnectRetry 计时器并转移到 Active 状态
3. 如果在连接状态下,ConnectRetry 超时，计时器将重新开始，并再一次试图与邻居建立 TCP 连接，BGP 保持 Connect 状态,此时如果有任何其他输入事件,转入 Idle 状态

Active State

在此状态,BGP 过程试图与邻居建立一个 TCP 连接

1. 如果连接成功,BGP 过程将 ConnectRetry 计时器清零,完成初始化,给邻居发送一个 Open 消息并转移到发送 Open 消息状态,Hold 计时器设置为 4mins
2. 如果在激活状态,ConnectRelay 计时器超时，将回到 ConnectState 并且重置 ConnectRelay 计时器.也发起一个到对等的 TCP 连接并继续监听来自对等体的连接.
3. 如果邻居试图与一个未知 IP 建立 TCP 会话,同时 ConnectRelay 计时器重置,连接被拒绝并保持在 Active 状态
4. 任何一个事件(除开始事件)都回导致状态转向 idle

Open send State

在此状态下，已经发送了 Open 消息，BGP 等待邻居发来的 Open 消息，

1. 当收到一个 Open 消息，如果发现差错，将给邻居发一个 Notification 消息并转入 Idle 状态
2. 如果收到的 Open 消息没有差错，将给邻居发送一个 Keepalive 消息并将 Keepalive 计时器清零，此时协商一个较短的 holdtime，如果为 0，则没有启动 Hold 和 keepalive 计时器，根据 AS 号选择 IBGP 或者 EBGP，同时将状态转移到 OpenConfirm 状态
3. 如果收到一个 TCP 断开消息，本地断开 BGP 连接，重置 ConnectRetry 计时器，并转入 Active 状态

Open Confirm State

此状态下 BGP 会等待一个 Keepalive 消息或者 Notification 消息

1. 如果收到一个 Keepalive 消息，转移到 Establish 状态
2. 如果收到一个 Notification 消息，转入 Idle 状态，并断开 TCP 连接
3. 如果 Hold 计时器超时，检测到一个差错或出现 stop 事件，BGP 将给邻居发送一个 Notification 并断开连接转入 Idle 状态

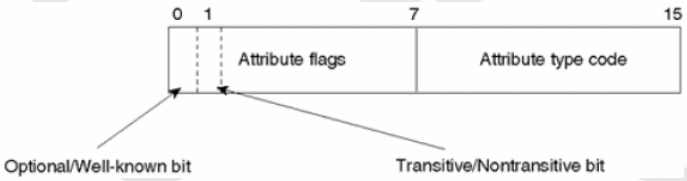
Establish State

此状态下，BGP 对等体间的连接已经完全建立，可以交换 Update Keepalive 和 Notification 消息，如果收到 Notification 自动转入 Idle，并中断连接

Path-Attributers

BGP 路径属性分为 4 类

1. Well-known mandatory 公认必遵 - 所有的 BGP 路由器必须识别
   (Update 消息必须包含该属性)
2. Well-known discretionary 公认可遵 - 所有的 BGP 路由器都能识别，但是不是一定需要
   (Update 消息可以不包含该属性或者该属性任选)
3. Optional transitive 可选传递 - 不是所有的 BGP 路由器都能识别，但所有 BGP 路由器都能传递它
   (此属性即使 BGP 路由器不接受也可以传递)
4. Optional nontransitive 可选非传递 - 不是所有的 BGP 路由器都能识别，不能识别 BGP 路由器丢弃它
   (此属性如果 BGP 路由器不接受则立即丢弃此消息，不再传递)



TYPE	Attribute	Class
1	ORIGIN	Well-known mandatory
2	AS_PATH	Well-known mandatory
3	NEXT_HOP	Well-known mandatory
4	LOCAL_PREF	Well-known discretionary
5	ATOMIC_AGGREGATE	Well-known discretionary
6	AGGREGATOR	Optional transitive
7	COMMUNITY	Optional transitive
8	MULTI_EXIT_DISC (MED)	Optional nontransitive
9	ORIGINATOR_ID	Optional nontransitive
10	CLUSTER_LIST	Optional nontransitive

ORIGIN

属于公认必选属性，明确了路由更新的来源，用于判断路由可信度，当 BGP 有多条路由来源时，路由器会将 ORIGIN 做为路由决策的参考

来源有如下几类：

IGP---从 AS 内部学到，ORIGIN 为 0

EGP---NLRI 从 EGP 学到，ORIGIN 为 1

Incomplete---NLRI 通过其他手段获得，ORIGIN 为 3

一般来说具有较低 ORIGIN 值得前缀被优先选取， IGP>BGP>重分布

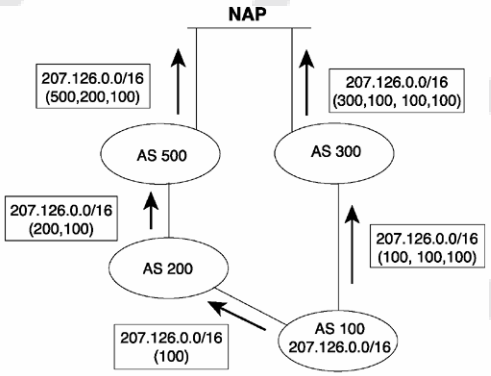
例如通过重分布进入 BGP，ORIGIN 属性为 3，通过 Network 命令注入其 ORIGIN 为 0

AS_PATH

描述一个路由传递过程中经过那一些 AS（不算自己，从离自己最近的 AS 开始，以目的网段的 AS 结束），为了避免

AS 环路，如果从外部收到一条包含自己 AS 的路由，就说明有环路，此时 BGP 将丢弃该路由

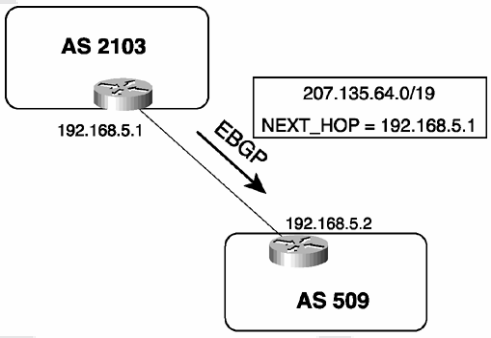
通常一条 AS_PATH 含有多个同一 AS 号，用于加长 AS_PATH，提供策略选路，如下图



AS500 为主要流量提供商，

通过修改 AS100 到 AS300 通告的 AS_PATh，让流量走 AS500，

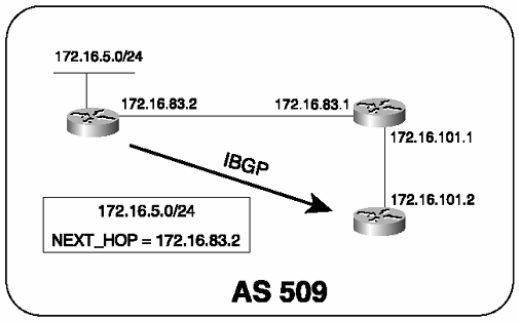
此方法亦保证 AS300 做为备份可用



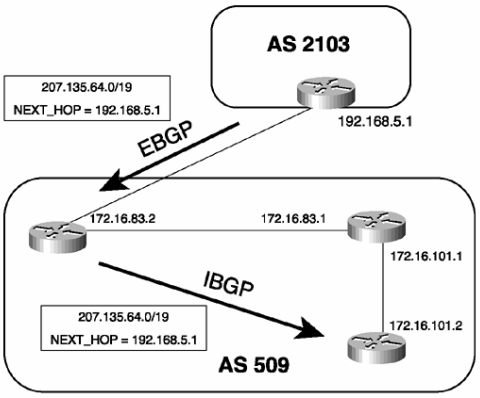
Next_Hop

该属性描述了到公布目的地址的路径的下一跳路由器的 IP 地址

1. 如果正在进行路由通告的路由器和接收的路由器在不同的 AS 中，Next_Hop 为正在宣告的路由器接口的 ip
2. 如果正在宣告的路由器和接收的路由器在同一个 AS 内，并且更新消息中 NLRI 目的地也在同一个 AS 中，则 next_hop 为一宣告的路由的邻居的 ip



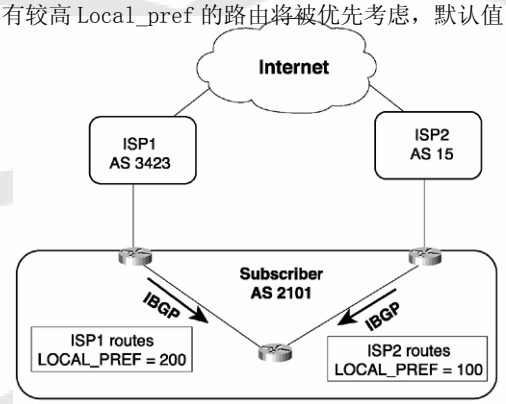
3. 如果正在宣告的路由器和接收的路由器是内部对等体，并且更新消息的 NLRI 指向不同 AS，则 Next_hop 为学习到路由的外部对等实体的 ip



Local_Pref

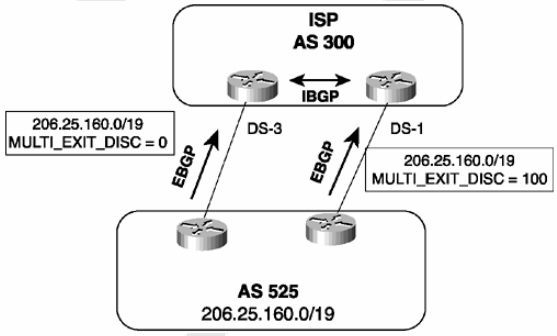
仅用在本地 AS，不会传到其他 AS，具有较高 Local_pref 的路由将被优先考虑，默认值 100

如右图，优先考虑 ISP1 的路由



Multi_Exit_Disc(MED)

Local_Pref 仅影响离开 AS 的业务量，而 MED 用于影响流入 AS 的业务量，它允许一个 AS 将其首选入口通知给另一个 AS，具有最低 MED 值的路由作为首选

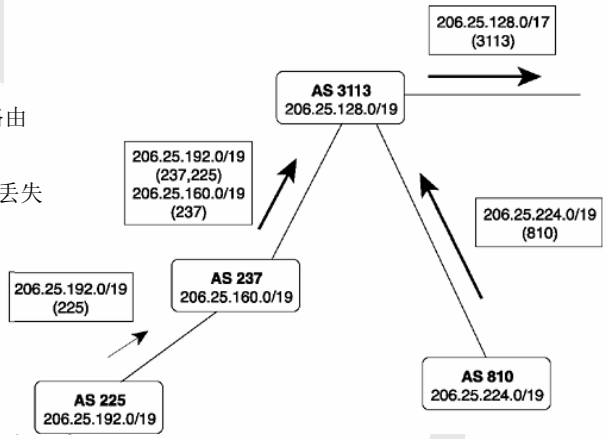


Atomic_Aggregate and Aggregator

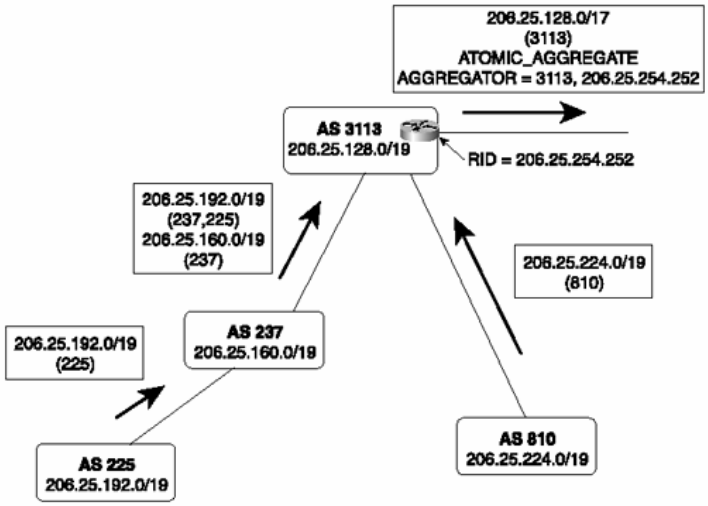
BGP 支持向另一个 BGP 路由器传递重叠的路由，BGP 有如下重叠处理选项

1. 同时公布精细和初略的路由
2. 只公布精细路由
3. 之公布路由中没有重叠的部分
4. 聚合两条路由并公布聚合后的路由
5. 两者都不公布

如右图即为在进行聚合时导致了路由信息丢失



Atomic_Aggregate 属性即是用来警告下游路由器聚合时产生路径信息丢失,可设置附加属性 Aggregate 来通告汇聚点,Cisco 采用 BGP-RID 来作为 Aggregator 地址,如下图



Weight

Cisco 专有,对于离开 AS 的报文，从多条路径中选择哪一条。它是不传递的。缺省下从对等体学到的所有路由器的权重值是 0，由本地路由器产生的所有路由的权重值是 32768

选路时优先选用权值最高的路径

Comunity

使可以向一组源路由使用相同的策略，即一个目的地作为一些目的地团体中的一个成员，这些目的地共享一个或多个共同特性。它有 4 个字节 - 前面两个字节的 AS 号,后面两个字节的管理上定义表示符,而 Cisco 正好反过来，用 **ip bgpcommuity new-format** 改过来

当对团体路由进行聚合时，聚合路由继承了所有路由的全部团体属性

NO_EXPORT 的团体属性，携带该属性的路由允许在邻居 AS 内公布但不允许邻居 AS 把路由公布其他 AS

NONE 属性删除现存的团体属性

NO_ADVERTISE 属性指不在 IBGP 邻居间传递带有该属性的路由

DELETE 属性用于只删除匹配特定团体列表的属性

ORIGINATOR_ID

由路由反射器(RR)使用，它是有路由发起者产生的一个 32 比特的值，该值是本地 AS 里路由发起者的 RID，如果路由器发起者从该属性值中看到了自己的 RID，就说明有环路，该路由忽略

Cluster_LIST

由路由反射器使用，它是路由经过反射器簇 ID 的一个序号。如果路由反射器在该属性值中发现自己的本地簇 ID，就说明有环路，忽略掉。如果一个簇里不止一个 RR，要在进程下用 **bgp cluster-id** 手工指定簇 ID,因为默认 RR 将自己的 RID 当成 **cluster-id**

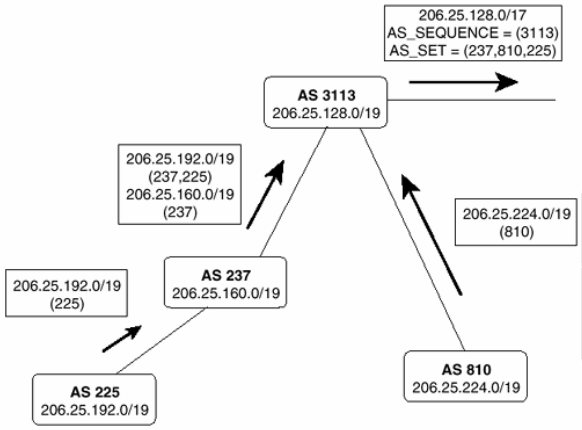
AS_SET

AS_PATH 有 4 种类型

1. AS_SEQUENCE ---路径上 AS 号是有序的
2. AS_SET ---路径上 AS 号是无规则的
3. AS_CONFED_SEQUENCE
4. AS_CONFED_SET ---用于联盟的有规则和无规则 AS 号序列(亦称为子有序和子无序 PATH)

由于聚合时 AS_PATH 中数据丢失,导致产生环路的潜在因素增加,因此加入 AS_SET 字段,通告汇聚时包含的 AS 号 AS_PATH 替代了 ATOMIC_AGGREGATE 的功能， 但也有其缺点：如聚合的网络出现故障，AS_SET 的改变将通

告到聚合点以外



BGP Dscision Process

BGP 路由信息库(RIB)包括

Adj-RIBs-In: 储存未经处理的路由信息,来自对等接收到的更新消息,所包含的路由是可用的

Loc-RIB: 包含的路由是运行 BGP 的路由器通过对 Adj-RIBs-In 中的路由使用他的本地路由策略而选择的路由

Adj-RIBs-Out:包含运行 BGP 的路由器向其对等公布的路由

3 个数据库间的处理方法如下

1. 为每一条可用路由计算优先级,并对 BGP update 消息中包含的路由信息变更作出处理
2. 将最优路由放入 Loc-RIB
3. 将适合的路由放入 Adj-RIBs-out(调用仅在上一步完成后)

BGP 路径选择顺序:

1， Weight---权重（越大越优先）
2， Local_Pref---本地优先（越大越优先）
3， 所传递路由的本地起源优先，即下一跳是 0.0.0.0(在 BGP 表中,当前路由器通告的路由的下一跳为 0.0.0.0)
4， AS-Path（越短越优先）
5， 起源（优先：IGP > EGP > Incomplete）
6， MED（越小越优先）
7， 首选 EBGP 的路由,在联盟 EBGP 和 IBGP 中首选联盟 EBGP 路由
8， BGP 优先选择到 BGP 下一跳的 IGP 度量最低的路径
9， 如果有多条来自相同相邻 AS 的路由并通过 Maximum-paths 使多条路径可用,则将所有开销相同的路由加入 Loc-RIB
10，BGP 优先选择最老的路径，降低滚翻的影响
11，BGP 邻居的 RID（越小越优先）
12，如果多条路径始发路由器 ID 或路由器 ID 相同，那么优选 Cluster-List 最短的路径
13，选择邻居 ip 地址最小的路由
14，

Route Dampening

路由抑制可以阻止公布不稳定的路由，它为每条路由分配一个动态的度量数字用来反映稳定程度，当一条路由出现摆动，就给他分配一个惩罚值，摆动得越多，惩罚值越大。当一段时间不摆动，惩罚值降低，在一个半衰期后，降到原来的一半。如果惩罚值超过抑制上限，该路由就被抑制，只有当一个半衰期后惩罚值降低到重新使用界限时，才重新使用。

缺省下， 惩罚值 - 每次摆动 1000
抑制界限 - 2000
重新使用界限 - 750

半衰期 - 15 分钟
最大抑制时间 - 60 分钟,或者半衰期的 4 倍

IBGP and IGP Synchronization

在某些情况下将 IGBP 当 IGP 使用,但是每个 IBGP 路由器必须与其它每一个 IBGP 路由器建立对等,即必须保证 AS 内 IBGP 全互连, 全互连可以防止 AS 内产生 BGP 环路,同时保证 BGP 路由上的所有路由器都知道如何转发数据包到目的地

IBGP 和 IGP 同步规则

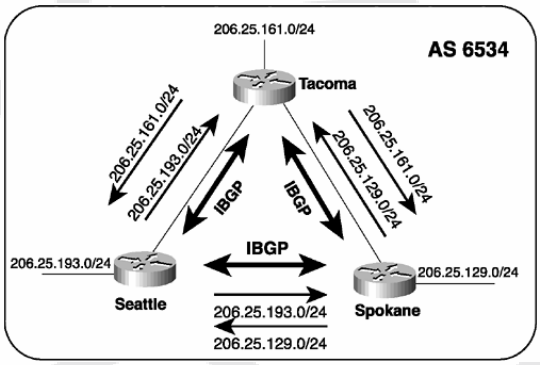
一条从 IBGP 邻居学习到的路由在进入 IGP 路由表或者公布给一个 BGP 对端之前,通过 IGP 必须知道该路由.

同步可以防止没有足够信息的 IGP 造成路由黑洞.但也会带来很多缺陷,为了使 IBGP 工作正常,可以采用 2 种方法:

1. 将外部路由重分发到 IGP 中从而保证 IGP 和 BGP 同步,但对于 IGP 路由器一般无法承受巨大的 internet 路由条目,在几个大型的中断事故中,多数是 BGP 无意中被分发到了 OSPF 或者 IS-IS 中
2. IBGP 为逻辑上全网状连接,且关闭同步功能,所有路由器通过 BGP 知道外部路由,且不用通知 IGP 就能将路由加入到路由表中,但如果一个 AS 中有多个 IBGP 路由器,一个路由器将于其他每个路由器建立对等,十分耗时,故常采用联盟或者路由反射器来控制 IBGP 逻辑上的全互连.

同步是一个较老的 BGP 特性, 在新版的 IOS 中, 同步默认被关闭

下图为关闭同步前后的区别,前者为同步打开,后者为同步关闭




```
Seattle#show ip bgp
BGP table version is 7, local router ID is 206.25.193.1
Status codes: s suppressed, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.1.0      0.0.0.0                  0         32768 i
* i                 192.168.1.1              0    100      0 i
*>i192.168.2.0      192.168.1.1              0    100      0 i
*>i206.25.161.0     192.168.1.1              0    100      0 i
*> 206.25.193.0     0.0.0.0                  0         32768 i
Seattle#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
Gateway of last resort is not set

C    206.25.193.0 is directly connected, Loopback0
C    192.168.1.0 is directly connected, Serial0
Seattle#
```

关闭同步以后

```
Seattle#show ip bgp
BGP table version is 11, local router ID is 206.25.193.1
Status codes: s suppressed, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 192.168.1.0      0.0.0.0                  0         32768 i
* i                 192.168.1.1              0    100      0 i
*>i192.168.2.0      192.168.1.1              0    100      0 i
* i                 192.168.2.1              0    100      0 i
*>i206.25.129.0     192.168.2.1              0    100      0 i
*>i206.25.161.0     192.168.1.1              0    100      0 i
*> 206.25.193.0     0.0.0.0                  0         32768 i

Seattle#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default

Gateway of last resort is not set

C    206.25.193.0 is directly connected, Loopback0
B    206.25.129.0 [200/0] via 192.168.2.1, 00:07:34
C    192.168.1.0 is directly connected, Serial0
B    192.168.2.0 [200/0] via 192.168.1.1, 00:07:42
B    206.25.161.0 [200/0] via 192.168.1.1, 00:07:43

Seattle#ping 206.25.129.1
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 206.25.129.1, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 4/5/8 ms
Seattle#
```

Managing Large-Scale BGP Peering

在一个较大规模的 AS 中试图生成全互连的 IBGP 对等关系是一个规模庞大的工作，同时对于 BGP 对等关系的管理也十分复杂。BGP 设计了如下 4 种方式对对等关系进行管理

Peer groups

在大型互联网络中，一个路由器的策略会应用到多个对等关系，在 Cisco 路由器中可以使用一个名字和一系列的路由策略来定义一个对等组(Peer Groups)，任何对等策略对整个 group 生效，对等组提高了路由器性能，只需访问一次策略数据库，生成一个 Update 消息，并将副本发给所有的对等体即可。

Communities

相对于 PeerGroups 是对一组路由器使用策略，而团体(Communities)是对一组路由是用策略，团体可设置成为一个公用的值或者管理员定义的特有团体，一条路由可以属于多个团体，路由聚合可以继承所有路由的全部团体属性

Route reflectors

在大型的 AS 中，全互连带来极大的工作量，通过路由反射器(RR)可以建立一种 C/S 结构，如一个含有 N 个路由器的 AS 内，全互连将使用对等会话数目为(N-2)N/2，如果选取一个 RR，则对等会话数目将降为 N-1。对一个 C/S 结

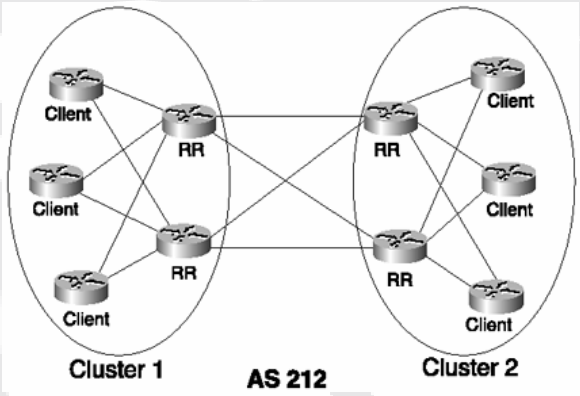
构称其为一个 RR-Cluster。RR 公布路由的规则：

1，如果路由是从非客户的 IBGP 对等学习到的，只将它反射给客户

2，如果路由是从客户处学习到的，将它反射给除了发起该路由的客户外所有的客户以及非客户

3，如果路由是从 EBGP 邻居学来的，将它反射给所有的客户和非客户

将一个路由器配置成路由反射器（RR）,用 **neigbhor route-reflect-client** 把自己配成反射器，由该命令所定义的 IBGP 邻居路由器当成客户机，这些客户机只与 RR 建立对等关系。RR 不能改动它从客户处收到的路由的属性。在一个 AS 内可以做 RR 冗余，因为客户并不知道自己是客户，所有一个 RR 可以是另一个路由反射器的客户。只需要 RR 支持路由反射，客户不需要支持



RR 使用了 2 个 BGP 路径属性：

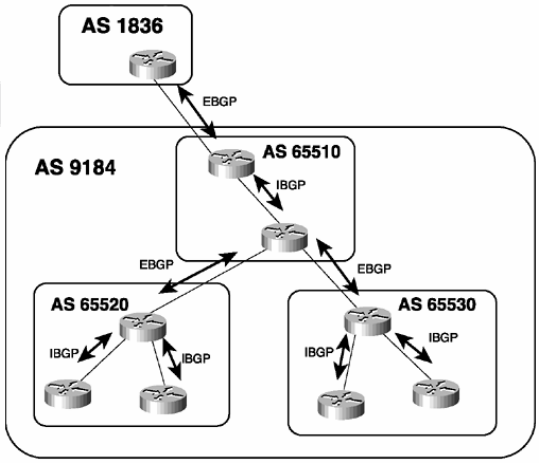
ORIGINATOR_ID

由路由反射器(RR)使用，它是有路由发起者产生的一个 32 比特的值，该值是本地 AS 里路由发起者的 RID，如果路由器发起者从该属性值中看到了自己的 RID，就说明有环路，该路由忽略

Cluster_LIST

由路由反射器使用，它是路由经过反射器簇 ID 的一个序号。如果路由反射器在该属性值中发现自己的本地簇 ID，就说明有环路，忽略掉。如果一个簇里不止一个 RR，要在进程下用 **bgp cluster-id** 手工指定簇 ID,因为默认 RR 将自己的 RID 当成 **cluster-id**

Confederations



联盟(confederations)是一组分成子自治系统组的 AS 如上图

1. 每一个联盟分配一个联盟 ID,对于外端而言,此联盟 ID 代表的是整个联盟的 AS 号.联盟其实质是对自治系统的再次细分.

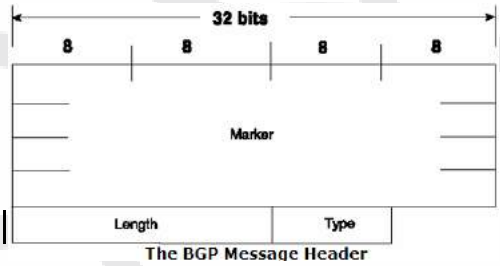
2. AS_PATH 中加入了 AS_CONFED_SEQUENCE 和 AS_CONFED_SET 用法和 AS_SEQUENCE 及 AS_SET 完全相同,
3. 在联盟环境下,所有路由器必须支持联盟
4. 用预留 AS(64512~65535)作为联盟中的 AS 编号是一中通用做法
5. 选路优先级：联盟的外部 EBGP〉AS 成员的 EBGP〉IBGP
6. 联盟相对于标准的 AS，Next_hops MED 可以不加修改的公布给联盟中的其他 AS 成员的 EBGP 对端，而且可以发送 Local_Pref 属性
7. 大型系统中，联盟和 RR 同时使用可以更好的控制 IBGP 对等关系

BGP Message Formats

BGP Message Header

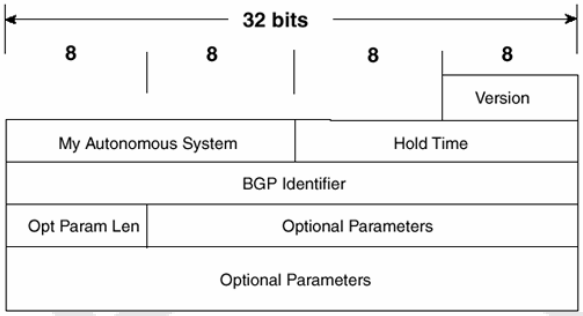
BGP 消息信头包含了标记，长度和类型 3 个字段

Code	Type
1	Open
2	Update
3	Notification
4	Keepalive

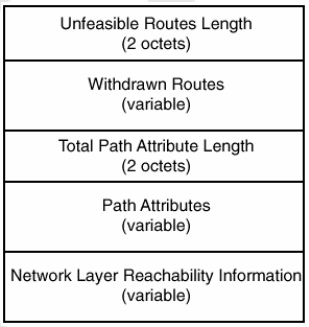


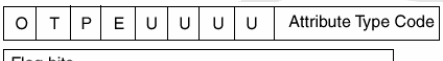
The BGP Message Header

Open Message



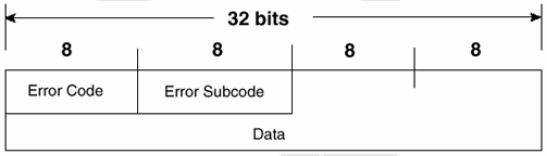
Update Message





Flag bits
O: Optional bit
0 = Optional
1 = Well-Known
T: Transitive bit
0 = Transitive
1 = Non-Transitive
P: Partial bit
0 = Optional Transitive attribute is partial
1 = Optional Transitive attribute is complete
E: Extended length bit
0 = Attribute Length is one octet
1 = Attribute Length is two octets
U: Unused

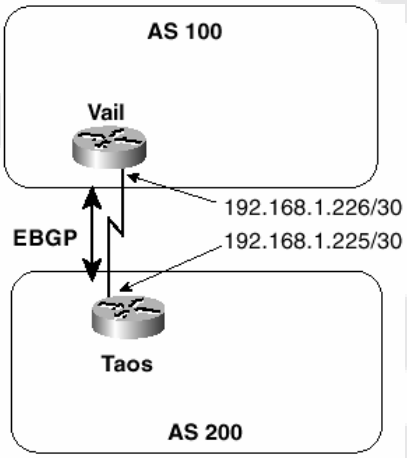
Notification Message



Error Code	Error	Error Subcode	Subcode Detail
1	Message Header Error	1	连接不同步
		2	错误的消息长度
		3	错误的消息格式
2	Open Message Error	1	不支持的版本号
		2	错误的对等 AS(peer AS)
		3	错误的 BGP 标识符(BGP-RID)
		4	错误的可选字段
		5	鉴别失败(Authentication failure)
		6	不可接受的 Holdtime
3	Update Message Error	1	畸形的属性列表
		2	不可识别的公认(well-known)属性
		3	丢失的公认属性
		4	属性标志错误
		5	属性长度错误
		6	非法的 ORIGIN 属性
		7	AS 路由环路
		8	非法的下一跳属性
		9	可选属性错误
		10	非法的网段(invalid network)
		11	畸形的 AS_Path
4	Hold 计时器超时	0	—
5	有限状态机错误	0	—
6	终止	0	—

Basic BGP Configuration

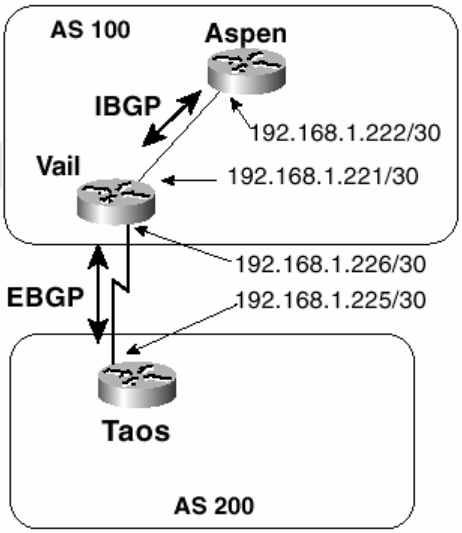
Case Study: Peering BGP Routers



Vail
router bgp 100
neighbor 192.168.1.225 remote-as 200

Taos
router bgp 200
neighbor 192.168.1.226 remote-as 100

```
Vail#show ip bgp neighbors
BGP neighbor is 192.168.1.225,  remote AS 200, external link
 Index 1, Offset 0, Mask 0x2
  BGP version 4, remote router ID 192.168.1.225
  BGP state = Established, table version = 1, up for 19:32:02
  Last read 00:00:03, hold time is 180, keepalive interval is 60 seconds
  Minimum time between advertisement runs is 30 seconds
  Received 1175 messages, 0 notifications, 0 in queue
  Sent 1175 messages, 0 notifications, 0 in queue
  Prefix advertised 0, suppressed 0, withdrawn 0
  Connections established 1; dropped 0
  Last reset never
  0 accepted prefixes consume 0 bytes
  0 history paths consume 0 bytes
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 192.168.1.226, Local port: 11025
Foreign host: 192.168.1.225, Foreign port: 179
Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)
Event Timers (current time is 0x45FDF2C):
Timer          Starts    Wakeups            Next
Retrans          1176          0             0x0
TimeWait            0          0             0x0
AckHold          1175        885             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0
iss: 4072889888  snduna: 4072912224  sndnxt: 4072912224     sndwnd:  16004
irs: 4121607729  rcvnxt: 4121630065  rcvwnd:      16004  delrcvwnd:    380
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 4 ms, maxRTT: 340 ms, ACK hold: 200 ms
Flags: higher precedence, nagle
Datagrams (max data segment is 1460 bytes):
Rcvd: 2220 (out of order: 0), with data: 1175, total data bytes: 22335
Sent: 2077 (retransmit: 0), with data: 1175, total data bytes: 22335
Vail#
```



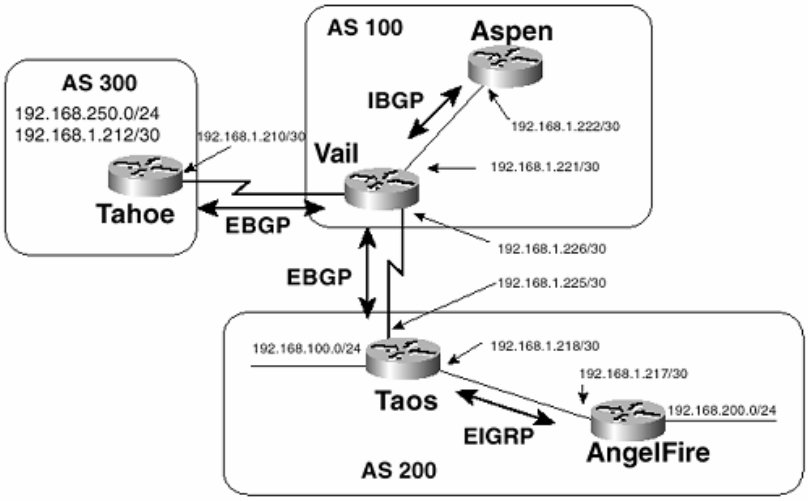
Vail 的配置

```
router bgp 100
 bgp router-id 1.1.3.2 //routerid 更改需要重启 BGP 进程 clear ip bgp
 neighbor 192.168.1.222 remote-as 100
```

```
 neighbor 192.168.1.225 remote-as 200
Aspen(config)#router bgp 100
Aspen(config-router)#neighbor 192.168.1.221 remote-as 100
Aspen(config-router)#^Z
Aspen#
18:24:13: %SYS-5-CONFIG_I: Configured from console by console
Aspen#   //debug bgp events
18:24:33: BGP: 192.168.1.221 went from Idle to Active
18:24:41: BGP: 192.168.1.221 went from Active to OpenSent
18:24:42: BGP: 192.168.1.221 went from OpenSent to OpenConfirm
18:24:42: BGP: 192.168.1.221 went from OpenConfirm to Established
18:24:43: BGP: 192.168.1.221 computing updates, neighbor version 0, table version
n 1, starting at 0.0.0.0
18:24:43: BGP: 192.168.1.221 update run completed, ran for 0ms, neighbor version
 0, start version 1, throttled to 1, check point net 0.0.0.0
```

Case Study: BGP and IGP redistributions

通过在 BGP 路由进程模式下 **redistribute eigrp 300** 可将 IGP 注入到 BGP 中，其度量值为以前路由协议的度量值，可以通过 **Default-metric** 修改，重分发到 BGP 后有部分管理员不想公开的 IGP 路由，需使用路由过滤器过滤掉。



另一方面,AS 内的非运行 BGP 的路由可能需要一定的外部路由信息,如左图中的 AngelFire 需要从 Taos 知道 AS300 AS100 的路由信息，则需要在 Taos 上做重分发

redistribute bgp 200 metric 10000 100 255 1 1500

不过一般禁止这样使用，如果 Taos 是一个 internet 上的 BGP 路由器，重分发后极大数目的路由将会进入 IGP，导致 IGP 崩溃，此种重分发一般只用在具有有限前缀的企业网络

此时 Taos 可以加入一条静态路由

ip route 192.168.250.0 255.255.255.0 Serial0

此时 192.168.1.212/30 将不会通告给 AngelFire，同时 192.168.250.0 的摆动仅会影响到 Taos，不会影响到 Angelfire

Case Study: BGP default-route

对于上图的 AS200，仅有一个出口，则可以在出口上做默认路由

ip route 0.0.0.0 0.0.0.0 Serial0

也可以配置成 BGP 通告的默认路由，此时只需 **ip route 0.0.0.0 0.0.0.0 Null0**

同时也可以使用 **neighbor 192.168.1.225 default-originate** 此时类似于 OSPF 的 **default-information-originate**

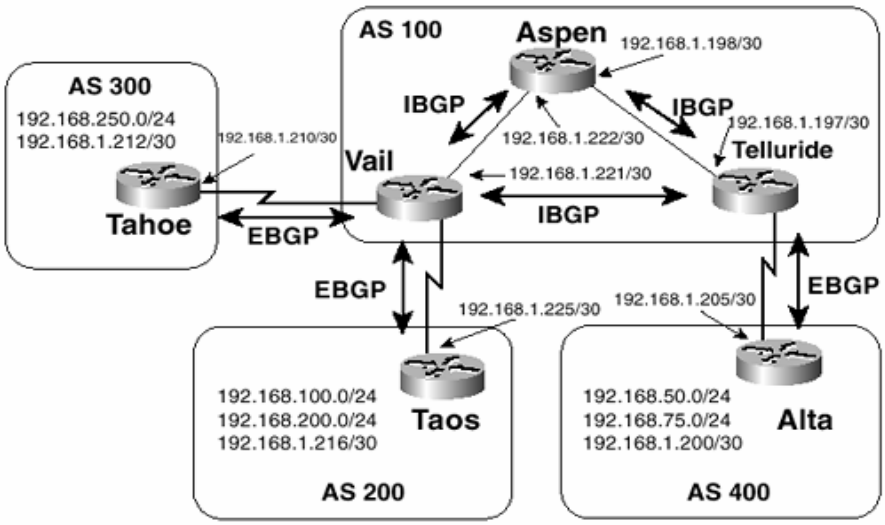
always 此时不管路由器实际上有没有缺省路由，都将公布一条默认路由。如果只想发布默认路由，通过 distribute-list 过滤

```
neighbor 192.168.1.225 distribute-list 1 out
!
access-list 1 permit 0.0.0.0
access-list 1 deny any
```

Case Study: IBGP without an IGP

如图，通过配置 AS100 的 3 台路由器的 neighbor 使其全互连此时必须关闭同步功能，如果同步功能打开，IBGP 路由无法加入路由表

注：关闭工作状态的 BGP 进程的同步后，需要重新启动 BGP 进程



在 Vail 和 Telluride 上可以使用 neighbor next-hop-self 此后，在公布学到的 EBGP 路由时将会以自己的地址做为下一跳，使用回环口做 BGP 进程接口

```
neighbor 1.1.1.1  update-source loopback 0
```

Case Study: EBGP-Multi hops

由于 EBGP 需要直接连接，且数据报文 TTL 值为 1，update-source 为 loopback 时，需要经过多跳，所以可以定义：

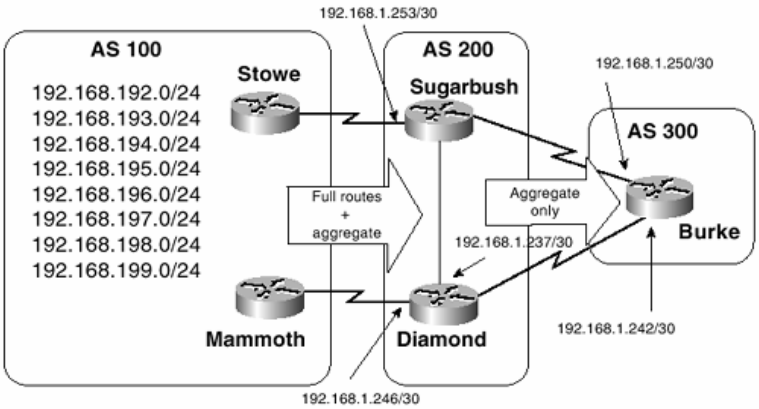
```
Neighbor 1.1.1.1 remote-as 100
Neighbor 1.1.1.1 update-source loopback 0
Neighbor 1.1.1.1 ebgp-multihop 2
```

Case Study: Aggregate Routes

一种方法是建立一条静态的到 Null0 的路由，让 BGP 通告出去，

另一种方法是采用 Aggregate-address 192.168.192.0 255.255.248.0

Summary-only：用于只公布聚会和的路由，不公布具体地址，并且在 BGP 路由表中用 S 表示



如上图， AS100->AS200 需要通告精细路由和汇聚路由，200-〉300 只通告汇聚地址。配置如下：

Stowe

```
router eigrp 100
 network 192.168.199.0
!
router bgp 100
 aggregate-address 192.168.192.0 255.255.248.0
 redistribute eigrp 100
 neighbor 192.168.1.253 remote-as 200
 neighbor 192.168.1.253 send-community
 neighbor 192.168.1.253 route-map COMMUNITY out
!
ip classless
!
access-list 101 permit ip host 192.168.192.0 host 255.255.248.0
!
route-map COMMUNITY permit 10
 match ip address 101
 set community none
!
route-map COMMUNITY permit 20
 set community no-export
```

mammoth //采用 prefix 来匹配

```
router bgp 100
 aggregate-address 192.168.192.0 255.255.248.0
 redistribute eigrp 100
 neighbor 192.168.1.246 remote-as 200
 neighbor 192.168.1.246 send-communit
 neighbor 192.168.1.246 route-map COMMUNITY out
!
ip classless
ip route 192.168.255.251 255.255.255.255 192.168.1.205
!
ip prefix-list AGGREGATE seq 5 permit 192.168.192.0/21
```

//seq 类似于 basic 的行号

```
!
route-map COMMUNITY permit 10
 match ip address prefix-list AGGREGATE
 set community none
!
route-map COMMUNITY permit 20
 set community no-export
```

suppress-map: 利用 suppress-map 也可以抑制选中的前缀

例如需要在 Stowe 链路上公布 192.168.192~194.0 在 mammoth 上公布 196~197.0 不公布 195.0 199.0

Stowe

```
router bgp 100
 aggregate-address 192.168.192.0 255.255.248.0 suppress-map VERMONT
 redistribute eigrp 100
 neighbor 192.168.1.253 remote-as 200
 neighbor 192.168.1.253 send-community
 neighbor 192.168.1.253 route-map COMMUNITY out
!
access-list 1 permit 192.168.195.0 0.0.0.255
```

```
access-list 1 permit 192.168.196.0 0.0.3.255
!
route-map VERMONT permit 10
 match ip address 1
```

Mammoth

```
router bgp 100
 aggregate-address 192.168.192.0 255.255.248.0 suppress-map CALIFORNIA
 redistribute eigrp 100
 neighbor 192.168.1.246 remote-as 200
 neighbor 192.168.1.246 send-community
 neighbor 192.168.1.246 route-map COMMUNITY out
!
ip prefix-list SUPPRESSEDROUTES seq 5 permit 192.168.192.0/22 le 24
ip prefix-list SUPPRESSEDROUTES seq 10 permit 192.168.199.0/24
!
route-map CALIFORNIA permit 10
 match ip address prefix-list SUPPRESSEDROUTES
```

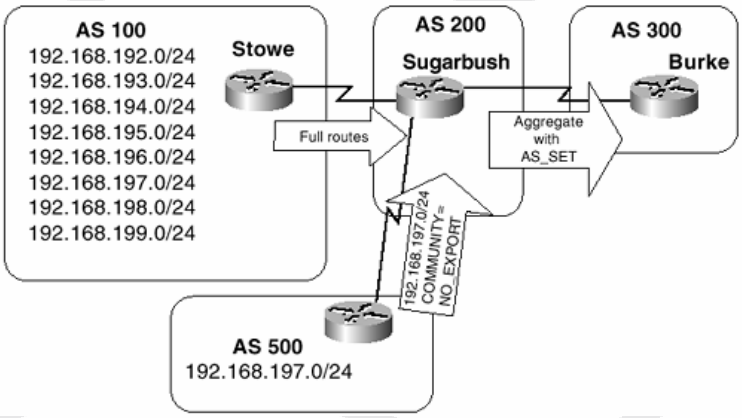
attribute-map:用于改变聚合路由属性

```
aggregate-address 192.168.192.0 255.255.248.0 attribute-map ORIGIN
!
route-map ORIGIN permit 10
set origin incomplete
```

AS-set: 使用 As-set 属性，防止汇聚时导致的 AS_path 丢失

```
aggregate-address 192.168.192.0 255.255.248.0 as-set
```

dvertise-map:在公布带有 as-set 的聚合路由时，不希望该聚合路由继承所有被聚合路由的所有属性



Sugarbush:

```
router bgp 200
 aggregate-address 192.168.192.0 255.255.248.0 as-set summary-only advertise-map ALLOW_ROUTE
 neighbor 192.168.1.1 remote-as 500
 neighbor 192.168.1.250 remote-as 300
 neighbor 192.168.1.254 remote-as 100
!
access-list 1 deny    192.168.197.0
access-list 1 permit any
!
route-map ALLOW_ROUTE permit 10
 match ip address 1
```

Managing BGP Connections

```
service password-encryption   //全局模式下使用此命令将配置文件中的口令加密
!
router bgp 100
 redistribute eigrp 100
 neighbor 192.168.1.253 remote-as 200
 neighbor 192.168.1.253 description ****56K to Sugarbush, Ckt. ID 54.DWDA.987654
 neighbor 192.168.1.253 password 7 14191D3F5831782574
```

密码仅启用在链路上，不同邻居可以设置不同的密码，同时 Ip 地址可以改为对等组的名字。加密方式默认为 MD5

Neighbor version: 修改通告出去的 BGP 版本号

Neighbor advertisement-interval:修改 update 间隔时间

Bgp bestpath as-path ignore :bgp 选路时忽略 as-path 长度

Neighbor maximum-prefix 300 90 warning-only 最大允许邻居通告的前缀数量，默认超过 75%警告，可以修改这个比例。如果超过 max-prefix ，中断邻居关系

Clear ip bgp ip 重置某个邻居间的 BGP 连接

Neighbor shutdown //配置中期，暂时中断到某个路由器的连接

Timers bgp 修改 keepalive 时间和 holdtime

Routing Policies

Resetting BGP connections

运行 BGP 的路由器，当如下配置出现变化后必须重置 BGP 连接

1. 补充或者改变与 BGP 有关的访问列表
2. 改变与 BGP 有关的权值
3. 改变与 BGP 有关的分配列表
4. 改变与 BGP 有关的计时器的相关规定
5. 改变 BGP 的管理距离
6. 改变与 BGP 有关的 route-map

可以使用软重置连接，但软重置连接将会带来较大的内存消耗，需要留意内存使用情况

首先需要对邻居配置 **neighbor 192.168.1.253 soft-reconfiguration inbound**

Clear ip bgp 192.168.1.253 soft out|in

distribute-list 过滤路由

```
neighbor 192.168.1.1 distribute-list 1 out
!
Access-list 1 deny 192.168.3.0
Access-list 1 permit any
```

As-path 过滤路由

Neighbor distribute-list 可以对 NLRI 处理，as_path 过滤则是对 AS 的处理。

```
Neighbor 1.1.1.1 filter-list 1 out
!
Ip as-path access-list 1 permit ^$
```

AS-PATH 过滤采用正则表达式，正则表达式解释如下：

1. 匹配行开始和结束：

Ip as-path access-list 20 permit 850

将匹配（850） （850, 23, 33）（33, 27, 7850, 33)这样的路经

如果仅匹配 850，则可以使用 **ip as-path access-list 20 permit ^850$**

仅匹配一个空的 AS_PATH **ip as-path access-list 20 permit ^$**

2. 匹配字符集和

Ip as-path access-list 22 permit ^85[0123459]$

将匹配含有 850~855 and 859 的 AS_PATH

3. 否定：匹配出字符集和外的其他任何字符

Ip as-path access-list 23 permit ^85[^0-5]$

将匹配含有 856~859 的 AS_PATH

4. 通配符

Ip as-path access-list 24 permit ^85.

将匹配含有 850~859 的 AS_PATH 由于.支持空格符，所以也匹配 85

5. 替代，即或操作

Ip as-path access-list 24 permit ^(851|852)$

将匹配含有 851 或者 852 的 AS_PATH

6. 选择字符，匹配一个可能存在或者不存在的字符

Ip as-path access-list 24 permit ^(850)?$

将匹配单一 AS 号 850 或者空列表，如果表达式为 850? 匹配 85 or 850

7. 重复，匹配多个字符

Ip as-path access-list 24 permit ^(850)*$

将匹配重复多次的 AS 号 850 或者空列表，匹配(850),(850,850,850),(850,850,888)

Ip as-path access-list 24 permit ^(850)*$

将匹配至少包含一个 AS 号为 850

8. 分界线,描述多个字

Ip as-path access-list 24 permit ^5610_148_3_2$

只有 as_path(5610_148_3_2)才能被匹配

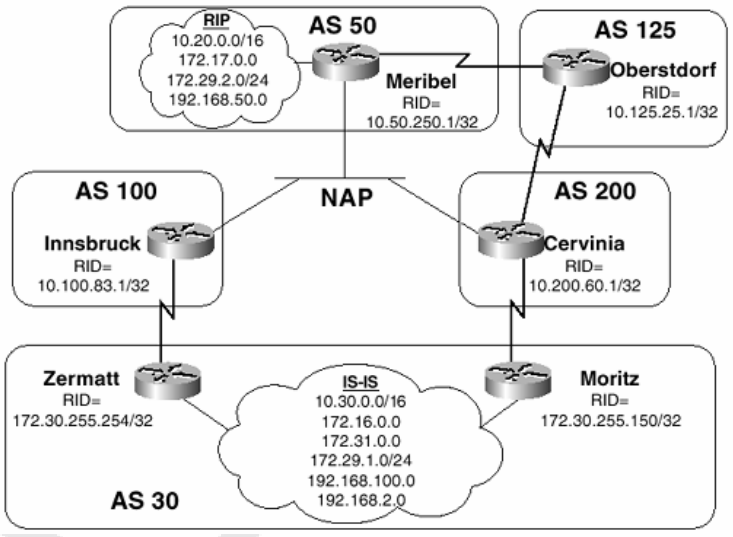
Ip as-path access-list 24 permit -5610_148_3_2$

包含 as_path(5610_148_3_2)都能被匹配

Ip as-path access-list 24 permit ^(550)+_[12|34]?_1805_.*

即匹配 as_path 在多个 550 后，选择 12 或者 34 接下来必须要有 1805

元字符特殊字符	匹配内容
.	任何单一字符，包括空格
[]	在方括号中罗列的任何字符
[^]	除了在方括号中所罗列字符外的任何字符（^必须放在字符列表之前）
-	（连字符）在由连字符所分配的两个字符之间的任意字符
?	字符或模式出现 0 次或 1 次
*	字符或模式出现 0 次或多次
+	字符或模式出现 1 次或多次
^	一行的开始
$	一行的结束
|	由元字符特殊字符分隔的字之一
_	（下划线）一个逗号，行的开始，行的结束或空格



```
router bgp 200
neighbor 10.50.250.1 remote-as 50
neighbor 10.50.250.1 ebgp-multihop 2
neighbor 10.50.250.1 update-source Loopback0
neighbor 10.50.250.1 filter-list 2 in
neighbor 10.100.83.1 remote-as 100
neighbor 10.100.83.1 ebgp-multihop 2
neighbor 10.100.83.1 update-source Loopback0
neighbor 10.100.83.1 filter-list 1 out
neighbor 10.125.25.1 remote-as 125
neighbor 10.125.25.1 ebgp-multihop 2
neighbor 10.125.25.1 update-source Loopback0
neighbor 172.30.255.150 remote-as 30
neighbor 172.30.255.150 ebgp-multihop 2
neighbor 172.30.255.150 update-source Loopback0
no auto-summary
  !
 ip as-path access-list 1 deny ^50$
 ip as-path access-list 1 permit .*
 ip as-path access-list 2 permit ^50$
```

此配置通过对 as-path 的过滤将 as125 做为到 as50 的转接 AS

Route-map 过滤路由

```
router bgp 30
 redistribute isis level-2
 neighbor 10.100.83.1 remote-as 100
 neighbor 10.100.83.1 ebgp-multihop 2
 neighbor 10.100.83.1 update-source Loopback0
 neighbor 10.100.83.1 route-map Innsbruck out
 no auto-summary
!

access-list 1 permit 192.168.100.0
access-list 1 permit 10.30.0.0
access-list 1 permit 192.168.2.0
access-list 1 permit 172.29.1.0
access-list 1 permit 172.31.0.0
access-list 1 permit 172.16.0.0
!
route-map Innsbruck permit 10
match ip address 1
```

Administrative Weights

CISCO 通过其专有的权重属性控制选路。配置如下

```
neighbor 10.100.83.1 remote-as 100
neighbor 10.100.83.1 weight 50000
neighbor 10.200.60.1 remote-as 200
neighbor 10.200.60.1 filter-list 3 weight 20000
```

也可以使用 route-map 设置权重

```
neighbor 10.200.60.1 route-map Cervinia in
!
ip as-path access-list 1 permit ^$
ip as-path access-list 2 permit _75$
ip as-path access-list 3 permit _50$
!
```

```
route-map Cervinia permit 10
 match as-path 2
 set weight 60000
!
route-map Cervinia permit 20
 match as-path 3
 set weight 40000
```

通过 route-map 设置 weight 时，只有 AS_PATH 与之匹配，

route-map 优先级高于带 filter 的 neighbor weight 高于 neighbor 的 weight

Administrative Distance and backdoor route

Distance bgp 20 95 200 //修改 EBGP IBGP local BGP 的 AD 从而对选路产生影响

在两个 AS 间可以存在一条单独的链路交换相互间的数据量，但并不通告给其他 AS

此时对于这条链路两端的路由器加入 network link-ip backdoor 即可

Local_Pref Attribute and MED

使用 ip default local-preference 修改默认值，并可以通过 route-map set local_preference

使用 default-metric 修改默认 MED 值。在 route-map 使用 set metic 修改 MED 值

在路由图中用 set metric-type internal 将一个 BGP 路由的 MED 设置成与到同一目的地的 IGP 路由的相同的度量。

在接收的一方用 bgp always-compare-med 来打破 MED 属性只能用于两个 AS 之间

Prepend as-path

通过附加 as-path，通过 AS_PATH 的长度控制选路.

```
ip as-path access-list 1 permit ^$
!
access-list 3 permit 192.168.100.0
access-list 3 permit 10.30.0.0
access-list 3 permit 172.29.1.0
!
route-map PATH permit 10
 match ip address 3
 set as-path prepend 30
```

Route Tagging

在 BGP 进程下用 table-map kaka 启动路由标记列表，

配上路由图的 set as-path tag（用于 IGP 重分布进 BGP 的路由恢复 AS_PATH 信息）

set automatic-tag（用于 BGP 重分布进 IGP 时恢复 AS_PATH 和 ORIGIN 信息）

Route Dampening

bgp dampening half-life reuse suppress max-suppress 被抑制的 BGP 路由表里标记 d 和 h

show ip bgp flap-statistics | dampened-paths 可以查看被抑制的路由

可以使用 clear ip bgp flap-statistics|dampened-paths 清楚被抑制的路由条目

例如 clear ip bgp flap-statistics regexp _30_来清除所有 As_path 含有 AS 30 的被抑制的路由条目

Large-Scale BGP

BGP Peer Groups

通过使用对等体组简化邻居配置,并对一组路由器使用共同策略。

```
neighbor CLIENTS peer-group
 neighbor CLIENTS ebgp-multihop 2
 neighbor CLIENTS update-source Loopback2
 neighbor CLIENTS filter-list 2 in
 neighbor CLIENTS filter-list 1 out
 neighbor 10.1.255.2 remote-as 200
 neighbor 10.1.255.2 peer-group CLIENTS
```

```
...
no auto-summary
!
ip as-path access-list 1 permit ^$
ip as-path access-list 2 permit ^[2-6]00$
```

BGP Communities

团体可以对一组路由使用相同的策略

1. 通过 route-map 识别已经设置了团体属性的路由
2. 通过 set community 设置团体属性
3. 使用 neighbor send-community 来明确该属性将要发送给哪个路由器

NO_EXPORT 接收到的携带该值的路由不能公布给 EBGP 或者 IBGP 的对等实体，不能再联盟范围以外公布

NO_ADVERTISE 接收到的携带该值的路由不能公布给 EBGP 或者 IBGP 的对等实体

LOCAL_AS 不能将携带该值的路由公布给 EBGP 对等实体，以及在联盟内的其他 AS

NONE 属性删除现存的团体属性

```
neighbor 10.1.255.8 send-community
 neighbor 10.1.255.8 route-map AUSTRIA out
 no auto-summary
!
access-list 1 permit 10.2.2.0
access-list 2 permit 10.3.3.0
!
route-map AUSTRIA permit 10
 match ip address 1
 set community no-export
!
route-map AUSTRIA permit 20
 match ip address 2
 set community none
```

自定义团体：

自定义团体号格式： AA：NN AA 为 AS 号 NN 为自定义编号，CISCO 刚好相反，

使用 ip bgp-community new-format 更改

```
Ip bgp-community newformat
access-list 1 permit 10.1.1.0
access-list 2 permit 10.1.2.0
!
route-map UTAH permit 10
 match ip address 1
 set community 200: 5
!
route-map UTAH permit 20
 match ip address 2
 set community 200：10
```

```
!
ip community-list 1 permit  .*:5
ip community-list 2 permit  .*:10
!
route-map COMM_PREF permit 10
 match community 1
 set local-preference 150
!
route-map COMM_PREF permit 20
 match community 2
 set local-preference 200
```

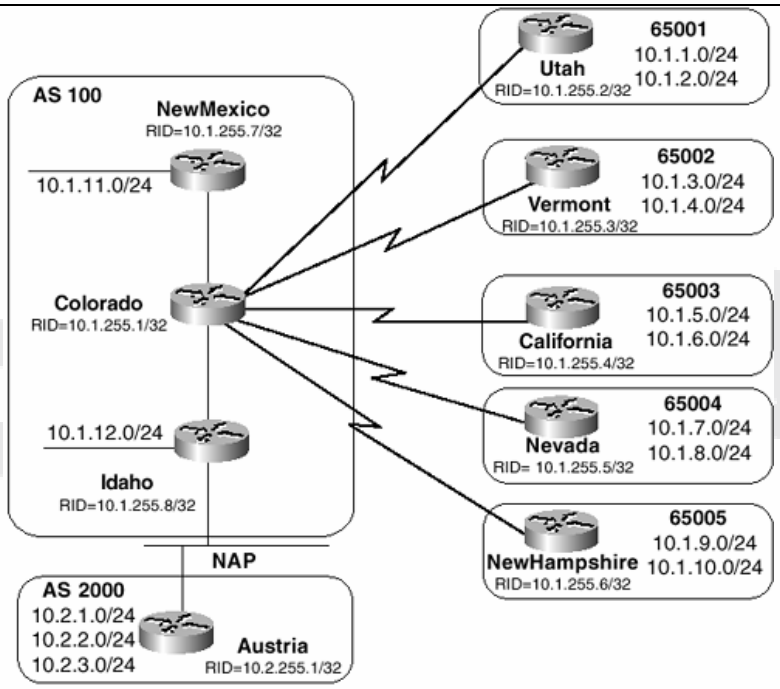
ip community-list 1 permit 5 10 表示与带有团体 5，或 10，或两个都有的标记的路由，

如果后面加上关键字 exact-match 表示只带有 5，10 或两个都有的表示的路由，还带有其他团体标记的路由不匹配

Private-AS

为了防止 AS 不够用，采用了私有 AS 号的设计，只需在出口路由器上将私有 AS 从 AS_PATH 删除即可

```
neighbor 10.2.255.1 remove-private-AS
```

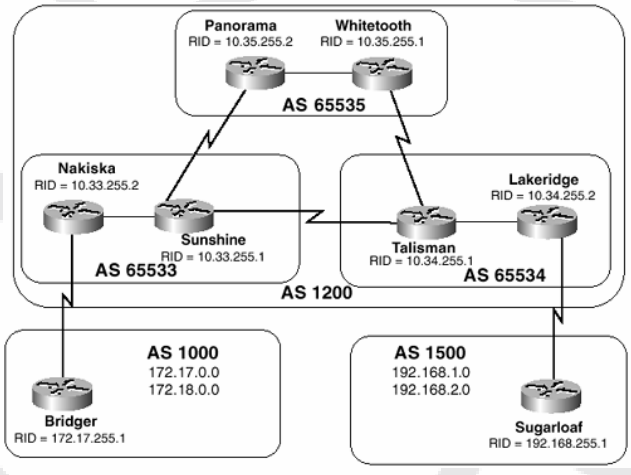


BGP Confederations

BGP 联盟可以使管理者将大型的 AS 分割成小型的 AS，而这些小型 AS 对外部是隐藏的

联盟属性如下：

1. 在联盟内部保留联盟外部的路由的 NEXT_HOP 属性
2. 公布给联盟的路由的 MED 属性在整个联盟范围内予以保留
3. 路由的 Local_pref 属性在整个联盟范围内予以保留
4. 在联盟范围内，将成员 AS 号加入 AS_PATH，但并不公布到联盟外，使用 type 3，4 的 AS_PATH
5. AS_PATH 中的联盟 AS 号用于避免环路，当联盟内选择一个最短的 AS_PATH 可以考虑不使用联盟 AS 号




```
bgp confederation identifier 1200
 bgp confederation peers 65533 65535
 neighbor Confed peer-group
 neighbor Confed ebgp-multihop 2
 neighbor Confed update-sourc Loopback 0
 neighbor Confed next-hop-self
 neighbor MyGroup peer-group
 neighbor MyGroup remote-as 65534
 neighbor MyGroup update-source Loopback 0
 neighbor 10.33.255.1 remote-as 65533
 neighbor 10.33.255.1 peer-group Confed
 neighbor 10.34.255.2 peer-group MyGroup
 neighbor 10.35.255.1 remote-as 65535
 neighbor 10.35.255.1 peer-group Confed
```

在整个联盟内使用 next-hop-self，因此 IGP 可以了解到所有下一跳。

Route Reflectors

路由反射器用于减小 IBGP 对等连接数量的一种方法

在成为 RR 的路由器上加入 **neighbor 10.33.255.2 route-reflector-client**

RR 公布路由的规则：

1，如果路由是从非客户的 IBGP 对等学习到的，只将它反射给客户

2，如果路由是从客户处学习到的，将它反射给除了发起该路由的客户外所有的客户以及非客户

3，如果路由是从 EBGP 邻居学来的，将它反射给所有的客户和非客户

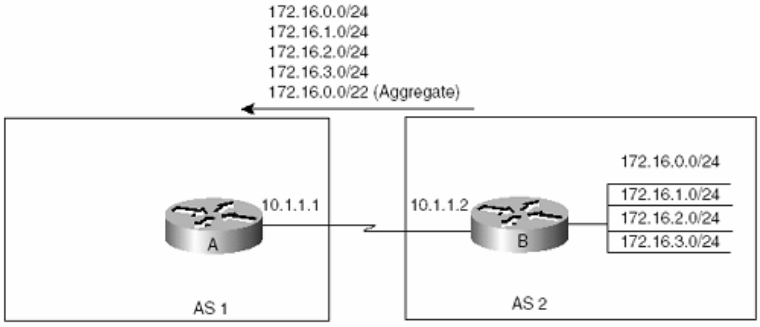
如果路由反射器在该属性值中发现自己的本地簇 ID，就说明有环路，忽略掉。

如果一个簇里不止一个 RR，要在进程下用 **bgp cluster-id** 手工指定簇 ID,因为默认 RR 将自己的 RID 当成 cluster-id

BGP 4 command and Configurations

Aggregate-address address mask

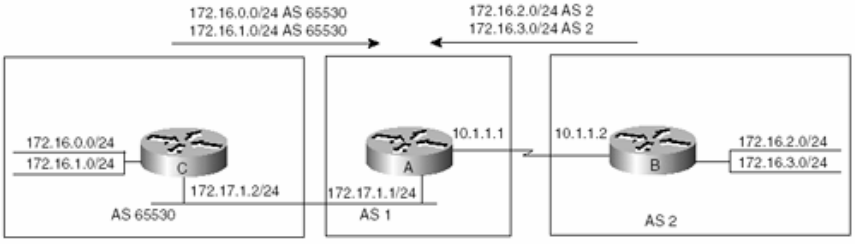
用于手工汇聚路由，不加参数表示创建一条汇聚路由，但并不抑制精细路由




```
Router B
router bgp 2
 network 172.16.0.0 mask 255.255.255.0
 network 172.16.1.0 mask 255.255.255.0
 network 172.16.2.0 mask 255.255.255.0
 network 172.16.3.0 mask 255.255.255.0
 aggregate-address 172.16.0.0 255.255.252.0
 neighbor 10.1.1.1 remote-as 1
RouterB#show ip bgp
 Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0/24    0.0.0.0                  0         32768 i
*> 172.16.0.0/22    0.0.0.0                            32768 i
*> 172.16.1.0/24    0.0.0.0                  0         32768 i
*> 172.16.2.0/24    0.0.0.0                  0         32768 i
*> 172.16.3.0/24    0.0.0.0                  0         32768 i


rtrA#show ip route
...
B        172.16.0.0/16 [200/0] via 0.0.0.0, 00:03:01, Null0
...
```

Aggregate-address address mask as-set

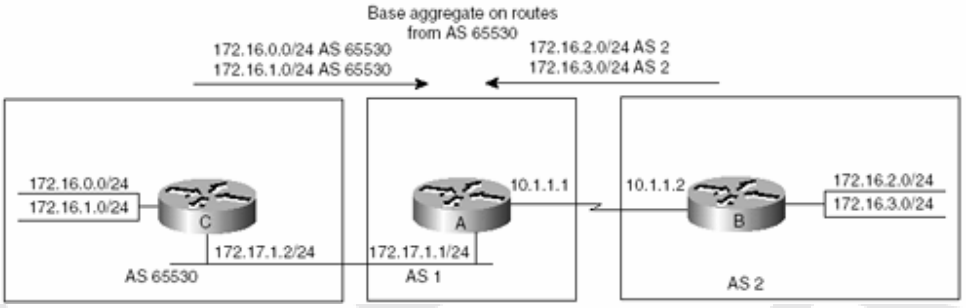
用于手工汇聚路由，AS-set 用于在汇聚时加入汇聚地址和 AS_SET 字段，防止产生路由环路



```
Router A
router bgp 1
aggregate-address 172.16.0.0 255.255.252.0 as-set
 neighbor 10.1.1.2 remote-as 2
 neighbor 172.17.1.2 remote-as 65530
rtrA#show ip bgp 172.16.0.0/22
BGP routing table entry for 172.16.0.0/22, version 10
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.1.1.2 172.17.1.2
  {65530,2}, (aggregated by 1 144.223.1.1)    //此处加入了 AS-SET 字段
    0.0.0.0 from 0.0.0.0 (144.223.1.1)
      Origin IGP, localpref 100, weight 32768, valid, aggregated, local, atomic-aggregate, best
```

Aggregate-address address mask as-set advertise-map route-map-name

用于手工汇聚路由，advertise-map 可以决定 AS_PATH 中的信息是否被保留到聚合路由中




```
aggregate-address 172.16.0.0 255.255.252.0 as-set

rtrA#show ip bgp 172.16.0.0/22
BGP routing table entry for 172.16.0.0/22, version 10
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.1.1.2 172.17.1.2
  {65530,2}, (aggregated by 1 172.17.1.1)
    0.0.0.0 from 0.0.0.0 (172.17.1.1)
      Origin IGP, localpref 100, weight 32768, valid, aggregated, local, atomic-aggregate, best

aggregate-address 172.16.0.0 255.255.252.0 as-set
 advertise-map select-as
 !
access-list 1 permit 172.16.0.0 0.0.0.255
access-list 1 permit 172.16.1.0 0.0.0.255
!
ip as-path access-list 1 permit ^65530$
!
route-map select-as permit 10
 match as-path 1

rtrA#show ip bgp 172.16.0.0/22
BGP routing table entry for 172.16.0.0/22, version 5
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  10.1.1.2 172.17.1.2
```

```
65530, (aggregated by 1 172.17.1.1)
  0.0.0.0 from 0.0.0.0 (172.17.1.1)
    Origin IGP, localpref 100, weight 32768, valid, aggregated, local, atomic-aggregate, best
```

Aggregate-address address mask **attribute-map** route-map-name

用于手工汇聚路由,attribute-map 用于改变聚合路由属性

```
aggregate-address 172.16.0.0 255.255.252.0 attribute-map ORIGIN
 !
 route-map ORIGIN permit 10
  set origin incomplete
  set metric 50
```

Aggregate-address address mask **route-map** route-map-name

用于手工汇聚路由时改变聚合路由属性，等同于 attribute-map

Aggregate-address address mask **summary-only**

用于手工汇聚路由时抑制精细路由的发送

Aggregate-address address mask **suspress-map** route-map-name

用于手工汇聚路由，并抑制相关的路由，一般需要按 RFC1918 抑制路由

```
aggregate-address 172.16.0.0 255.255.252.0 suppress-map suppress
!
access-list 1 permit 172.16.2.0 0.0.0.255
route-map suppress permit 10
 match ip address 1
```

Auto-summary

自动汇总到 ABC 三类地址，在 Cisco 较新的 IOS 中，no auto-summary 自动打开

bgp always-compare-med

打开这条命令后，路由器将始终比较 MED 的大小来决定路径，配置这条命令需在 AS 内的所有路由器上
原算法只比较来自己相同 external-AS 的通告的路由，此命令将比较所有路由，而不考虑外部 AS 是否相同

bgp bestpath as-path ignore

在选路时，忽略 as-path 长度的比较.但需要注意，根据选路算法，weight local_pref origin 具有更高优先级

bgp bestpath med confed

打开这条命令，在选路时，路由器将只比较所有带有 As_confe_seq 属性的路由条目，此命令用于联盟路由器，同时 weight 和 local_pref 比 MED 有更高的优先级

bgp bestpath med missing-as-worst

对于收到的一条路由前缀，如果缺失 MED 值，默认将 MED 设置为 0，此条命令用于将没有 MED 的路由前缀设置为一个最大值 4,294,967,295

bgp client-to-client reflection

在配置路由反射器时，客户到客户间的反射是默认打开的。但如果客户间是全互连的，此条命令加 no 前缀，关闭客户间的反射

bgp cluster-id 32-bit_id

BGP 可以收工指定 cluster-id，cluster-id 可以用普通 10 进制输入，也可以用点分 10 进制输入。

bgp confederation identifier as_number

bgp confederation peers 1_or_more_as_number

前面一条命令定义整个联盟的 AS 号，后面一条命令定义同属于这个 AS 的其它对等 AS 的号。

bgp dampening

开启 BGP 的路由惩罚。

bgp dampening half-life reuse suppress max_suppress_time **route-map** route-map-name

half-time: 半衰期，时间 1~45 分钟，即在半衰期时间惩罚值降低一半

Default 15mins

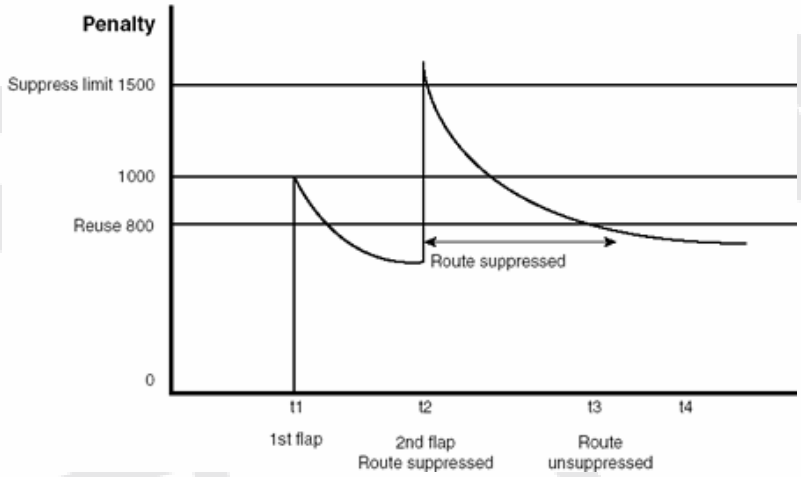
Reuse: 惩罚下限，即一条路由惩罚值降低到此阀值以下，路由将不会被抑制

Range 1~20000 default 750

Suppress: 惩罚上限，即惩罚值如果超过此值，路由将被抑制

Range 1~20000 default 2000

Max_suppress_time：最大抑制时间，默认其值为 half-time 的 4 倍 range 1~255minutes




```
router bgp 2
 bgp dampening route-map dampen
 neighbor 10.1.1.1 remote-as 1
!
ip as-path access-list 1 permit ip as-path access-list 1 permit ^1$
ip as-path access-list 2 permit ip as-path access-list 1 permit ^3$
!
route-map dampen permit 10
 match as-path 1
 set dampening 15 750 2000 60
route-map dampen permit 20
 match as-path 2
 set dampening 20 800 2200 80
```

bgp default local-preference local-preference

设置默认的 local_pref 值，range 0~4,294,967,295,默认 local_pref 100

bgp deterministic-med

打开此命令后，用于确保使用来自相同 AS 的不同对等通告的路由进行 MED 比较

此命令和 bgp always-compare-med 的关系如下

例如有相同前缀的路由如下（默认按路由接受顺序的逆序排序）

entry1: AS(PATH) 500, med 150, external, rid 172.16.13.1
entry2: AS(PATH) 100, med 200, external, rid 1.1.1.1
entry3: AS(PATH) 500, med 100, internal, rid 172.16.8.1

1. 都关闭

1,2 将先比较，2 将优选，因为 2 有更小的 RID，然后 2 与 3 比较，2 为 external，所以最优路径为 2

2. 打开 always-compare-med

先 1，2 比较，选择 med 更低的 1 和 3 比较，由于 1，3 在一个相同 AS 中，所以再次比较 med，选择 3

3. Deterministic-med 打开，路由前缀的信息将按 AS 重新分组

entry1: AS(PATH) 100, med 200, external, rid 1.1.1.1
entry2: AS(PATH) 500, med 100, internal, rid 172.16.8.4

entry3: AS(PATH) 500, med 150, external, rid 172.16.13.1

此时先组内比较，由于 entry2 的 MED 高于 entry3,所以 entry2 将和 entry1 比较，由于在不同的 AS，所以 MED 不比较，且 1 为 external 的路由，所以，最有路径为 1

4. 都打开

此时先组内比较，由于 entry2 的 MED 高于 entry3,所以 entry2 将和 entry1 比较，由于 always-compare-med 打开，所以 2 的 MED 小于 1 的 MED，2 将作为最佳路径

bgp fast-external-fallover

bgp fast-external-fallover 默认是打开的，如果一个接口 down 了以后，BGP 会话会立即中断，如果这个接口处与振荡状态，那么大量的 BGP 的 update 和 withdrawn 消息将充斥着整个链路，同时从 idle 到 establish 状态不停改变

对一个 flapping 的端口，建议使用 no bgp fast-external-fallover，此时端口 shutdown 以后，BGP 会话和邻居关系还存在，直到超过 BGP 的 Holdtime 才会中断

bgp log-neighbor-changes

此命令用于记录邻居关系变更的一些信息，这些信息可以发到带有 syslog-daemon 的 UNIX 主机或者保存在路由器的内部缓存中，记录消息类型如下

BGP protocol initialization	Error during connection collision
No memory for path entry	Peer closing down the session
No memory for attribute entry	Peer exceeding maximum prefix limit
No memory for prefix entry	Interface flap
No memory for aggregate entry	Router ID changed
No memory for dampening info	Neighbor deleted
No memory for BGP updates	Member added to peer group
BGP notification received	Administrative shutdown
Erroneous BGP update received	Remote AS changed
User reset request	RR client configuration modification
Peer time-out	Soft reconfiguration modification
Password change	

配置如下：

```
logging buffered 4096 debugging
!
router bgp 1
 bgp log-neighbor-changes
 neighbor 172.17.1.1 remote-as 2

rtr#show logging
Syslog logging: enabled (0 messages dropped, 0 flushes, 0 overruns)
    Console logging: level debugging, 48 messages logged
    Monitor logging: level debugging, 0 messages logged
    Buffer logging: level debugging, 3 messages logged
    Trap logging: level informational, 51 message lines logged

Log Buffer (4096 bytes):
01:18:16: %SYS-5-CONFIG_I: Configured from console by console
01:18:31: %BGP-5-ADJCHANGE: neighbor 172.17.1.2 Down - Peer closed the session
01:19:00: %BGP-5-ADJCHANGE: neighbor 172.17.1.2 Up
```

bgp router-id ip-address

手工设置 BGP 的 router-id,自动设置方法和 OSPF 相同,如果有 Loopback 口,设置最大地址的 loopback 的 ip-addr 否则设置活动的物理接口的最大 ip 做为 rid

default-information originate

用于产生一条默认路由，产生默认路由的属性为 incomplete

```
router bgp 2
 default-information originate
 neighbor 172.17.1.2 remote-as 1
!
ip route 0.0.0.0 0.0.0.0 serial 2/0
```

default-metric metric

用于设置指派给重分布的路由的 MED 值，默认为 0。手工设置 ospf 的 metric 为 5，静态路由为默认值 10

```
router bgp 2
 redistribute static
 redistribute ospf 1 metric 5
 neighbor 172.17.1.2 remote-as 1
 default-metric 10
```

distance admin-distance ip-source-address ip-address-mask ip-access-list-number

用于设置 BGP 路由的管理距离。

Admin-distance 设置 BGP 管理距离值

Ip-source-address/mask 设置路由信息通告源的 RID 地址和子网掩码(子网掩码采用反掩码)

Ip-access-list-number 设置一个 acl，对 acl 所选择的路由条目进行管理距离修改

```
neighbor 10.1.1.2 remote-as 2
 distance 50 10.1.1.2 0.0.0.0
```

distance bgp external internal local

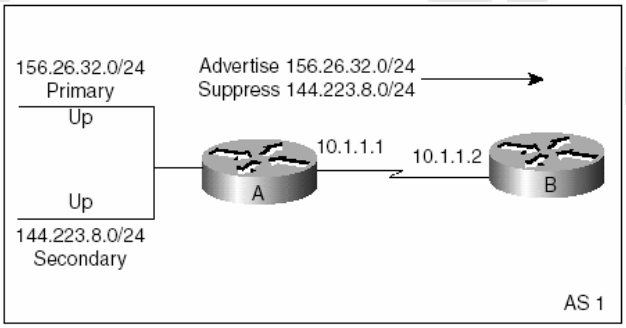
用于修改 BGP 3 种类型路由的管理距离，默认值为 20 200 200

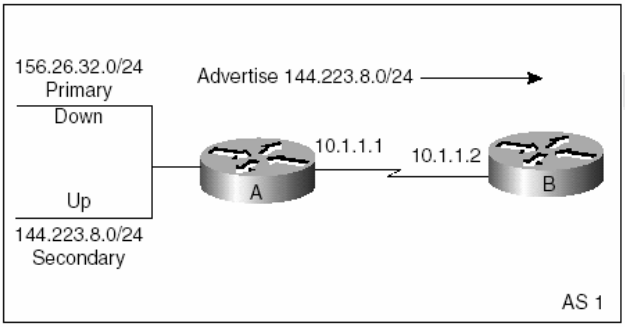
Distribute-list

利用 distribute-list 进行过滤，其命令方式为 **neighbor** {ipaddr|peer-group} **distribute-list** acl-number {in|out}

```
neighbor 192.168.1.225 distribute-list 1 out
!
access-list 1 permit 0.0.0.0
access-list 1 deny any
```

maximum-paths number-of-paths





在负载均衡情况下，设置最大负载均衡的路径数目，默认为 1，最大值为 6

neighbor {ip-address | peer-group-name} advertise-map route-map-name1 non-exist-map route-map-name2

对于左图配置，当两条链路都存在时，启动 primary，抑制 secondary

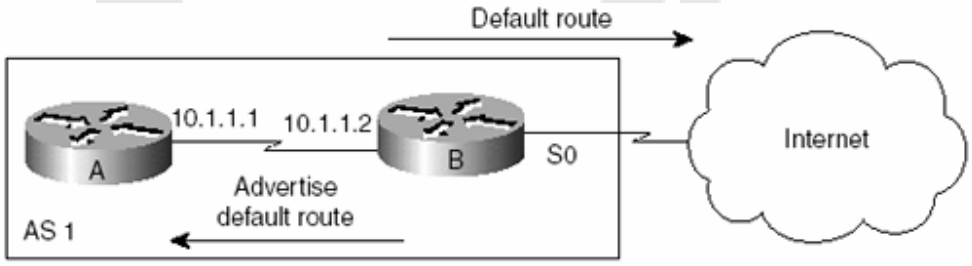
当 primary down 了以后，自动通告 secondary 的路由

```
Router A
interface loopback 0
 description primary prefix
 ip address 156.26.32.1 255.255.255.0
!
interface loopback 1
 description secondary prefix
 ip address 144.223.8.1 255.255.255.0
!
router bgp 1
 network 156.26.32.0 mask 255.255.255.0
 network 144.223.8.0 mask 255.255.255.0
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 advertise-map secondary non-exist-map primary
!
access-list 1 permit 156.26.1.0 0.0.0.255
access-list 2 permit 144.223.8.0 0.0.0.255
!
route-map primary permit 10
 match ip address 1
!
route-map secondary permit 10
 match ip address 2
```

neighbor {ip-address | peer-group-name} advertisement-interval seconds

调整 BGP 邻居通告的间隔时间，range 0~600 default IBGP 5s EBGP 30s

neighbor {ip-address | peer-group-name} default-originate

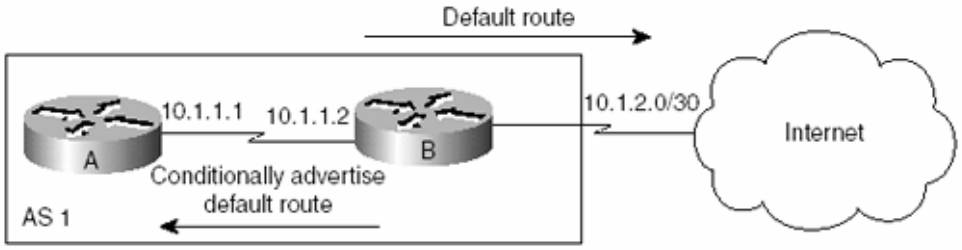


产生一条默认路由，但这种做法并不推荐，因为 BGP 所通告的默认路由，很有可能没有相应的路由或者默认路由 down 掉

```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 1
```

```
Router B
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 default-originate
!
ip route 0.0.0.0 0.0.0.0 serial 0
```

neighbor {ip-address | peer-group-name} default-originate route-map route-map-name



利用 route-map，有选择性的公布默认路由，

```
Router B
router bgp 1
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 default-originate route-map exists
!
access-list 1 permit 10.1.2.0 0.0.0.3
!
route-map exists permit 10
 match ip address 1
```

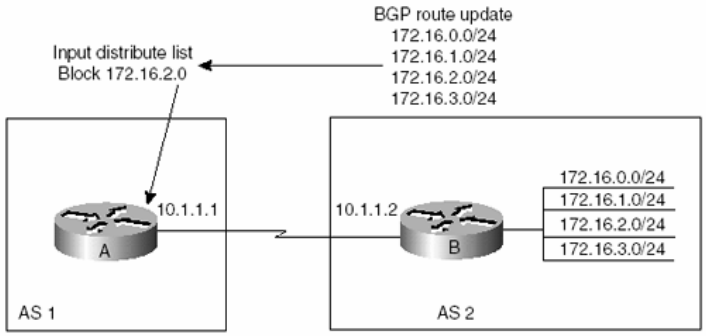
neighbor {ip-address | peer-group-name} description text

用于描述一个邻居的详细信息

```
Neighbor 1.1.1.1 description 56k-ISP1
```

neighbor {ip-address | peer-grp} distribute-list {acl-number-or-name prefix-list-name} in|out

用于对通告的路由条目进行出站或者入站的过滤

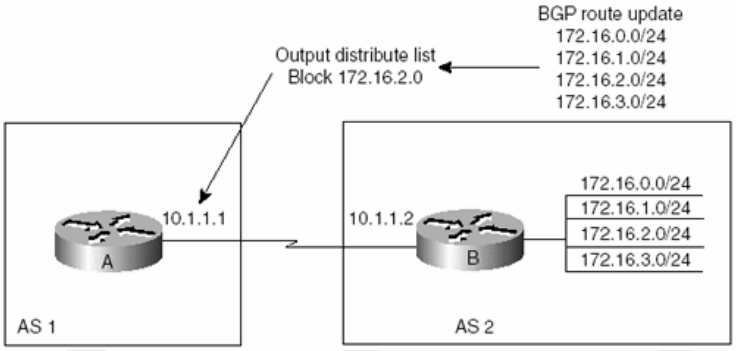



```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 distribute-list 1 in
!
access-list 1 deny    172.16.2.0 0.0.0.255
access-list 1 permit any
rtrA#show ip bgp
BGP table version is 4, local router ID is 172.17.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0/24    10.1.1.2                 0             0 2 i
*> 172.16.1.0/24    10.1.1.2                 0             0 2 i
*> 172.16.3.0/24    10.1.1.2                 0             0 2 i
```

如果 ACL 改为 access-list 1 permit 172.16.2.0 0.0.0.255，此时 BGP 表将改为

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.2.0/24    10.1.1.2                 0             0 2 i
```

出站过滤类型



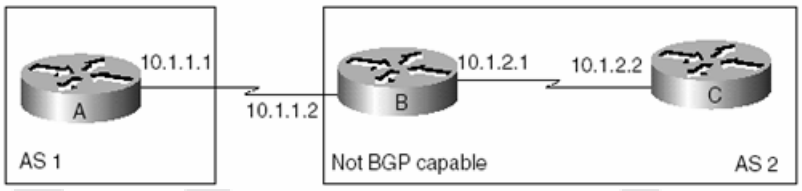

```
Router B
router bgp 2
neighbor 10.1.1.1 remote-as 2
neighbor 10.1.1.1 distribute-list 1 out
access-list 1 deny    172.16.2.0 0.0.0.255
access-list 1 permit any
```

例如 CCIE_lab 考试一道题目：要求 aggregate 不使用 no-summary 参数而抑制精细路由，其配置如下：

```
router bgp 2
 aggregate-address 172.16.0.0 255.255.252.0
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 distribute-list 100 out
access-list 100 permit 172.16.0.0 0.0.3.255 255.255.252.0 0.0.0.0
```

neighbor {ip-address | peer-group-name} ebgp-multihop maximum-hop-count

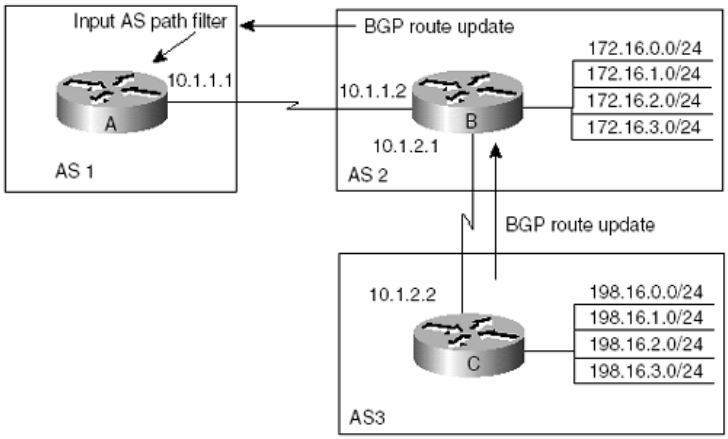
EBGP 多跳，用于如下的网络拓扑结构，EBGP 邻居间没有直接连接，需要 EBGP 多跳解决，默认 EBGP 多跳开启后，Default 255 不打开，EBGP 多跳为 1。




```
Router A
router bgp 1
 neighbor 10.1.2.2 remote-as 2
 neighbor 10.1.2.2 ebgp-multihop 255
!
ip route 10.1.2.0 255.255.255.252 serial0

Router C
router bgp 2
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.1.1 ebgp-multihop 2
!
ip route 10.1.1.0 255.255.255.252 serial 0
```

neighbor {ip-address | peer-group-name} filter-list as-path-list-number in|out

利用 AS_PATH 对路由条目进行过滤



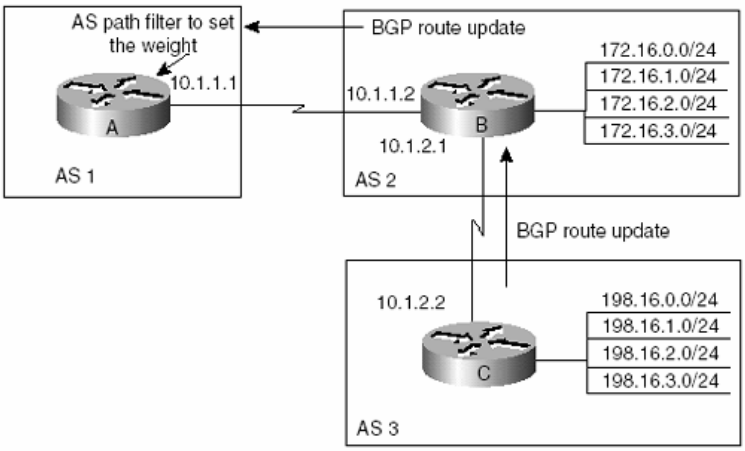

```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 filter-list 1 in
!
ip as-path access-list 1 deny _3$
ip as-path access-list 1 permit .*
```

也可以在 B 上做出站的路由过滤

```
Router B
router bgp 2
 neighbor 10.1.1.1 remote-as 1
 neighbor 10.1.2.2 remote-as 3
 neighbor 10.1.1.2 filter-list 1 out
!
ip as-path access-list 1 deny ^3$
ip as-path access-list 1 permit .*
```

neighbor {ip-address | peer-group-name} **filter-list** as-path-list-number **weight** weight

用于对来自特定的 AS_PATH 的路由，修改他们的权重




```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 filter-list 1 weight 850
!
ip as-path access-list 1 permit _3$
```

```
rtrA#show ip bgp
BGP table version is 9, local router ID is 172.17.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.0.0/24    10.1.1.2                 0             0 2 i
*> 172.16.1.0/24    10.1.1.2                 0             0 2 i
*> 172.16.2.0/24    10.1.1.2                 0             0 2 i
*> 172.16.3.0/24    10.1.1.2                 0             0 2 i
*> 198.16.0.0       10.1.1.2                               850 2 3 i
*> 198.16.1.0       10.1.1.2                               850 2 3 i
*> 198.16.2.0       10.1.1.2                               850 2 3 i
*> 198.16.3.0       10.1.1.2                               850 2 3 I
```

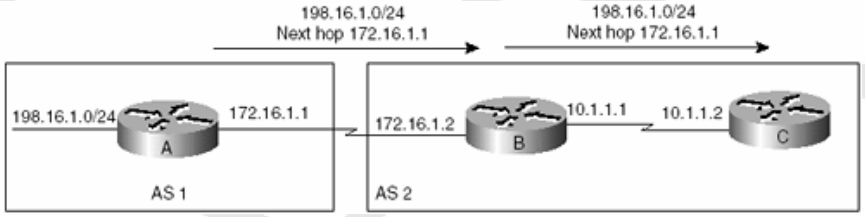
neighbor {ip-address | peer-group-name} **maximum-prefix** prefix-limit threshold-value **warning-only**

设置最多从邻居收到的路由前缀的条目，threshold-value 是一个比例值，当超过这个比例时，发送 warning,默认值为 75%

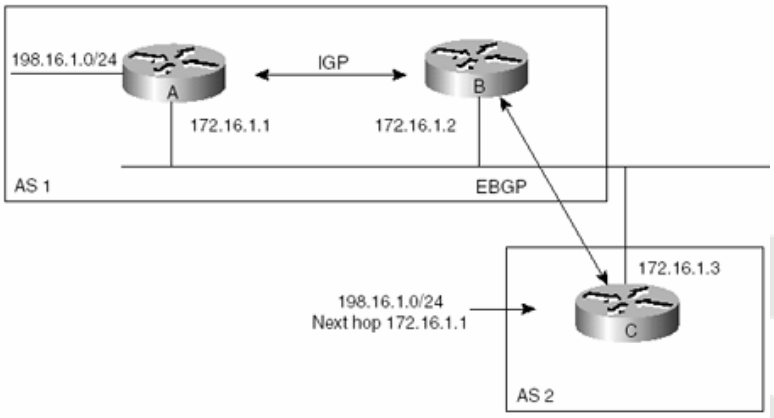
```
rtrA#show ip bgp neighbors
BGP neighbor is 10.1.1.2,  remote AS 2, external link
Index 1, Offset 0, Mask 0x2
  BGP version 4, remote router ID 10.1.1.2
  BGP state = Established, table version = 7, up for 00:53:07
  Last read 00:00:08, hold time is 180, keepalive interval is 60 seconds
  Minimum time between advertisement runs is 30 seconds
  Received 375 messages, 0 notifications, 0 in queue
  Sent 343 messages, 0 notifications, 0 in queue
  Prefix advertised 0, suppressed 0, withdrawn 0
  Connections established 20; dropped 19
  Last reset 00:53:28, due to User reset
  6 accepted prefixes consume 192 bytes, maximum limit 8 (warning-only)
  Threshold for warning message 75%
  0 history paths consume 0 bytes
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 10.1.1.1, Local port: 11015
Foreign host: 10.1.1.2, Foreign port: 179
```

neighbor {ip-address | peer-group-name} **next-hop-self**

当一个 BGP 路由器学到一条 EBGP 路由，他发送给 IBGP 路由器时 Next-hop 属性是默认不更改的

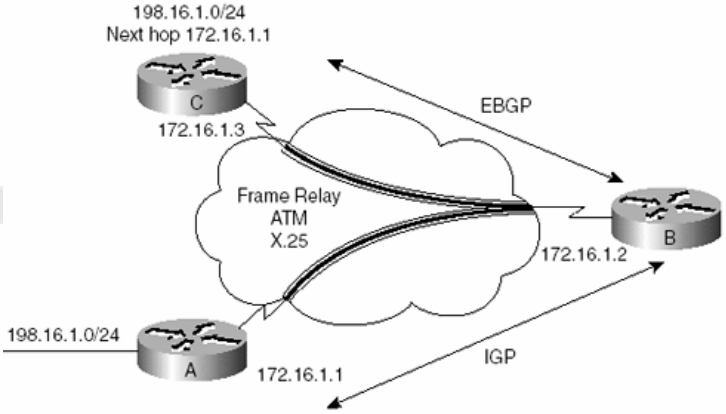


对于多路访问的拓扑结构，为了避免额外的跳数，Next-hop 属性如下



RouterB 从 A 学到一条 IBGP 路由，当他通告给 EBGP 邻居 C 时，将不会修改下一跳，依旧保留 A 的下一跳属性

对于一个 NBMA 型的网络



由于 A 和 C 之间没有直接的链路

需要人为在 RouterB 上修改 Next-hop 属性

```
Router B
router bgp 1
 neighbor 172.16.1.3 remote-as 2
 neighbor 172.16.1.3 next-hop-self
```

```
rtrC#show ip bgp
BGP table version is 22, local router ID is 172.16.1.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 198.16.1.0/24    172.16.1.2               0             0 2 i
```

neighbor {ip-address | peer-group-name} **password** password

在 BGP 对等链路上启用 TCP MD5 认证

neighbor ip-address **peer-group** peer-group-name

定义对等体组，简化配置

```
Router A (Using peer groups)
router bgp 1
 neighbor local peer-group
 neighbor local remote-as 1
 neighbor 10.1.1.2 peer-group local
 neighbor 10.1.2.2 peer-group local
 neighbor local distribute-list 1 out
 neighbor 10.1.2.2 peer-group local
```

```
 neighbor external peer-group
 neighbor 172.16.1.2 remote-as 2
 neighbor 172.16.2.2 remote-as 3
 neighbor 172.16.1.2 peer-group external
 neighbor 172.16.2.2 peer-group external
 neighbor external distribute-list 2 out
```

neighbor {ip-address | peer-group-name} prefix-list prefix-list-name in|out

利用前缀进行路由过滤

```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 prefix-list aggregate in
!
 ip prefix-list aggregate seq 5 permit 172.16.0.0/22
```

neighbor {ip-address | peer-group-name} remote-as number

定义一个邻居，并标明邻居所在的 AS 号，用于区分 EBGP 和 IBGP 邻居

neighbor {ip-address | peer-group-name} remove-private-as

通过此命令在宣告路由时，将 AS_PATH 中的私有 AS 号取消掉，一般用于联盟的出口路由器

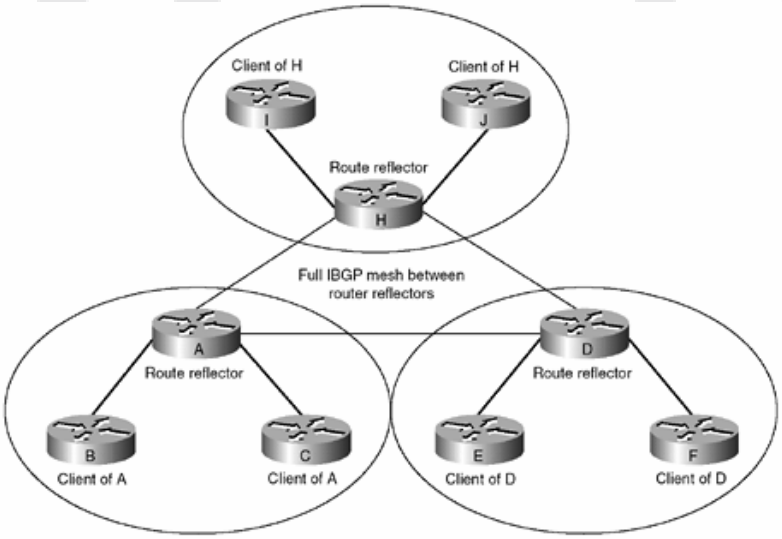
neighbor {ip-address | peer-group-name} route-map route-map-name in|out

通过 route-map 过滤路由，或者对某些路由加上 weight MED tag 等特征属性

```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 route-map filter in
!
Access-list 100 deny ip 1.1.0.0 0.0.255.255
Access-list 100 permit ip any any
route-map filter permit 10
match ip address 100
 set weight 90
```

neighbor {ip-address | peer-group-name} route-reflector-client

定义 BGP 路由反射器，此命令仅作用于 RR 上，client 不需要定义路由反射器



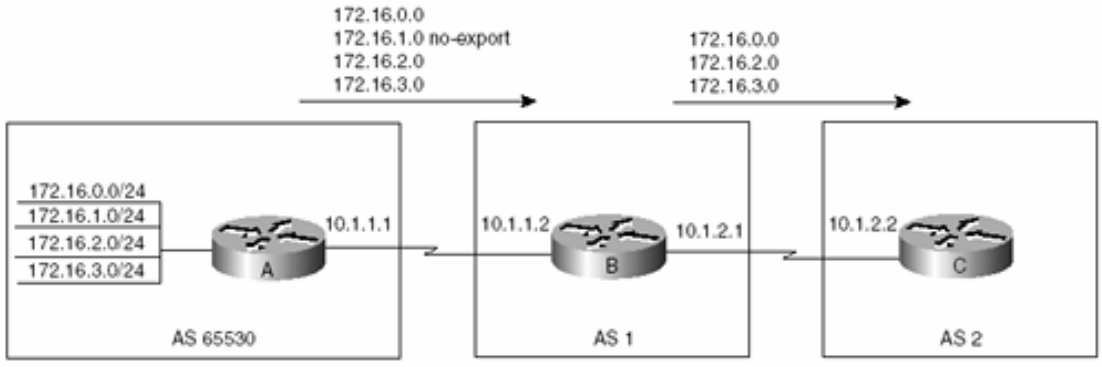
Router A
router bgp 1
neighbor (IP address for Router B) remote-as 1
neighbor (IP address for Router B) route-reflector-client
neighbor (IP address for Router C) remote-as 1
neighbor (IP address for Router C) route-reflector-client
neighbor (IP address for Router D) remote-as 1
neighbor (IP address for Router H) remote-as 1

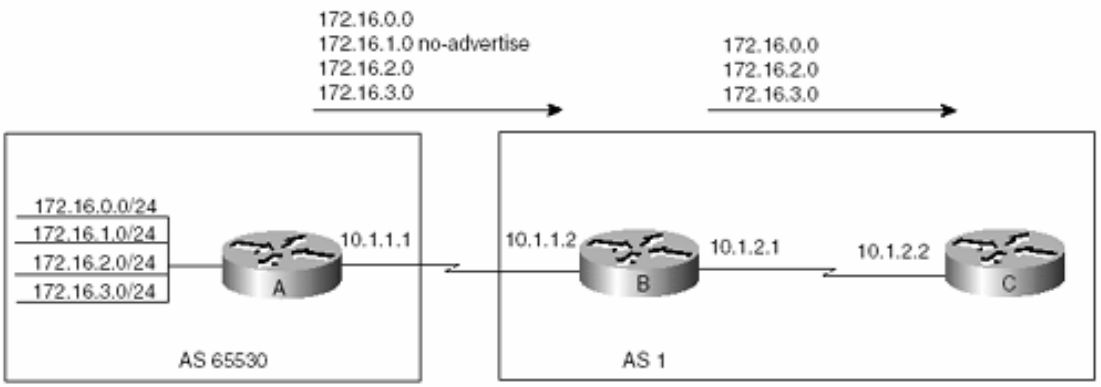
Router B
router bgp 1
neighbor (IP address for Router A) remote-as 1

Router C
router bgp 1
neighbor (IP address for Router A) remote-as 1

Router D
router bgp 1
neighbor (IP address for Router E) remote-as 1
neighbor (IP address for Router E) route-reflector-client
neighbor (IP address for Router F) remote-as 1
neighbor (IP address for Router F) route-reflector-client
neighbor (IP address for Router A) remote-as 1
neighbor (IP address for Router H) remote-as 1

neighbor {ip-address | peer-group-name} send-community
在配置 BGP 路由器团体属性时必须加入 send-community 参数





Router A
router bgp 65530

```
 network 172.16.0.0 mask 255.255.255.0
 network 172.16.1.0 mask 255.255.255.0
 network 172.16.2.0 mask 255.255.255.0
 network 172.16.3.0 mask 255.255.255.0
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 send-community
 neighbor 10.1.1.2 route-map setnoexport out
!
access-list 1 permit 172.16.1.0 0.0.0.255
!
route-map setnoexport permit 10
 match ip address 1
 set community no-export
route-map setnoexport permit 20
```

neighbor {ip-address | peer-group-name} shutdown

此命令用于关闭一个 BGP 会话，但在配置中却不关闭，仅为 administratively shut down

neighbor {ip-address | peer-group-name} soft-reconfiguration inbound

优雅重启动，用于在 BGP 策略更改后，例如 distribute-list route-map 等，默认情况下需要重新启动 BGP 会话

使用此条命令后将不需要重启 BGP 会话，soft-reconfiguration 将不管入站策略，将邻居的 update 存放到内存中，因此 inbound 将消耗很多内存，而 outboound 由于不消耗内存，所以是默认打开的

```
router bgp 2
 neighbor 10.1.1.2 remote-as 1
 neighbor 10.1.1.2 soft-reconfiguration inbound
router#
BGP neighbor is 10.1.1.2,  remote AS 1, external link
 Index 1, Offset 0, Mask 0x2
  Inbound soft reconfiguration allowed
  BGP version 4, remote router ID 172.16.1.1
  BGP state = Established, table version = 6, up for 00:00:33
  Last read 00:00:33, hold time is 180, keepalive interval is 60 seconds
 ....
```

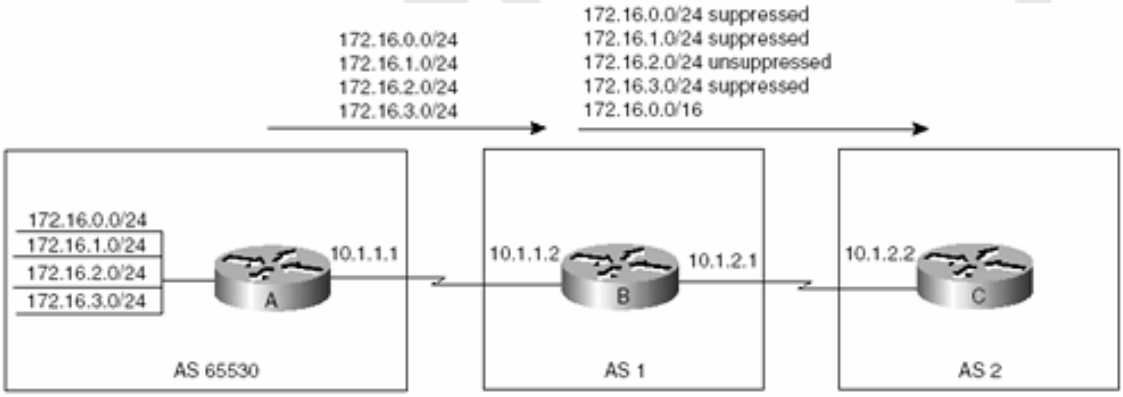
neighbor {ip-address | peer-group-name} timers keepalive holdtime

设置到邻居的 keepalive 和 hold 时间，默认时间 60 180

```
Router A
router bgp 1
 neighbor 10.1.1.2 remote-as 2
 neighbor 10.1.1.2 timers 50 150
 no auto-summary
```

neighbor {ip-address | peer-group-name} unsuppress-map route-map-name

将某条被抑制的路由改为非抑制路由



```
Router B
router bgp 1
 aggregate-address 172.16.0.0 255.255.0.0 summary-only
neighbor 10.1.2.2 unsuppress-map allow
!
access-list 1 permit 172.16.2.0 0.0.0.255
route-map allow permit 10
 match ip address 1
```

neighbor {ip-address | peer-group-name} update-source interface-name

设置 BGP update 消息来源的端口，一般把 BGP 进程建立在 loopback 口上，以获得更大的稳定性

```
router bgp 1
 neighbor 172.16.1.3 remote-as 1
 neighbor 172.16.1.3 update-source Loopback0
```

neighbor {ip-address | peer-group-name} version version-number

修改到邻居的 BGP 版本号，默认出现不匹配会自动协商，但协商过程较慢，可以手工设置版本号

neighbor {ip-address | peer-group-name} weight default-weight

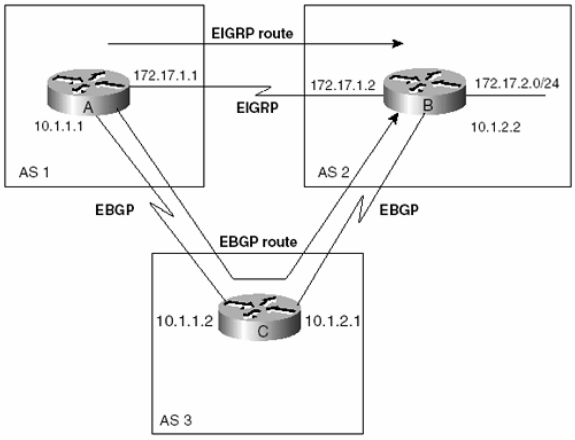
设置邻居通告来的路由的默认权重

network ip-address mask network-mask

用于通告网络，如果不加 mask 则自动按 ABC 类汇总，例如 10.0.1.0 自动汇总为 A 类

network ip-address mask network-mask backdoor

通告后门路由，即 AS1-AS2 间那条链路为后门链路，这条链路不会公布给 EBGP 对端




```
Router A
router bgp 1
 network 172.17.2.0 mask 255.255.255.0 backdoor
 neighbor 10.1.1.2 remote-as 3
rtrA#show ip route
Codes: C - connected, S - static, I - IGRP, R - RIP, M - mobile, B - BGP
       D - EIGRP, EX - EIGRP external, O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2, E - EGP
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, * - candidate default
       U - per-user static route, o - ODR, P - periodic downloaded static route
       T - traffic engineered route
Gateway of last resort is not set
     172.17.0.0/24 is subnetted, 2 subnets
          C       172.17.1.0 is directly connected, Ethernet0
          D       172.17.2.0 [90/409600] via 172.17.1.2, Ethernet0
```

```
        10.0.0.0/30 is subnetted, 2 subnets
B       10.1.2.0 [20/0] via 10.1.1.2
C       10.1.1.0 is directly connected, Serial0
```

network ip-address **mask** network-mask **route-map** route-map-name

此命令可以对通告的路由属性通过 route-map 修改

network ip-address **weight** weight

修改通告的网络的权重

redistribute protocol **route-map** route-map-name **metric** metric

将 IGP 直连和静态路由重分布到 BGP 中，同时挂接 route-map 可以对相应的路由进行权重等属性的设置。同时也可以设置分布进入后的 metric 值

```
redistribute static route-map setwt
 redistribute eigrp 1 route-map block
 neighbor 172.17.1.2 remote-as 2
 no auto-summary
!
access-list 1 deny 172.16.2.0 0.0.0.255
access-list 1 permit any
route-map block permit 10
 match ip add 1
route-map setwt permit 10
 set weight 88
```

redistribute protocol **weight** weight

对于重分布进入的路由设置相应的权重

synchronization

一条从 IBGP 邻居学习到的路由在进入 IGP 路由表或者公布给一个 BGP 对端之前,通过 IGP 必须知道该路由.

同步可以防止没有足够信息的 IGP 造成路由黑洞.但也会带来很多缺陷,为了使 IBGP 工作正常,可以采用 2 种方法:

1. 将外部路由重分发到 IGP 中从而保证 IGP 和 BGP 同步,但对于 IGP 路由器一般无法承受巨大的 internet 路由条目,在几个大型的中断事故中,多数是 BGP 无意中被分发到了 OSPF 或者 IS-IS 中
2. IBGP 为逻辑上全网状连接,且关闭同步功能,所有路由器通过 BGP 知道外部路由,且不用通知 IGP 就能将路由加入到路由表中,但如果一个 AS 中有多个 IBGP 路由器,一个路由器将于其他每个路由器建立对等,十分耗时,故常采用联盟或者路由反射器来控制 IBGP 逻辑上的全互连.

同步是一个较老的 BGP 特性，在新版的 IOS 中，同步默认被关闭

timers bgp keepalive holdtime

修改 BGP 的 keepalive 和 holdtime，默认为 60/180 改小后可以加快汇聚速度

show ip bgp | {begin |include|exclude} line

察看 BGP 表的信息，可以使用 begin include exclude 等关键字过滤

```
rtrA#show ip bgp | include 199
BGP table version is 90, local router ID is 199.172.15.1
*> 199.172.1.0      0.0.0.0                  0         32768 i
s> 199.172.2.0      0.0.0.0                  0         32768 i
*> 199.172.2.0/23   0.0.0.0                            32768 i
s> 199.172.3.0      0.0.0.0                  0         32768 i
*> 199.172.4.0      0.0.0.0                  0         32768 I
```

s--被抑制的路由条目

*--可用的路由

d--被惩罚(dampening)的路由

h—被惩罚(dampening)的路由

>--BGP 算法选出的最优路由

i--从 IBGP 学到的路由前缀

show ip bgp prefix

```
rtrA#show ip bgp 199.172.1.0
```

```
BGP routing table entry for 199.172.1.0/24, version 6
Paths: (1 available, best #1)
  Advertised to non peer-group peers:
    172.17.1.1
  Local
    0.0.0.0 from 0.0.0.0 (199.172.15.1)
     Origin IGP, metric 0, localpref 100, weight 32768, valid, sourced, local,best, ref 2
```

也可以使用

rtrA#show ip bgp 199.172.1.0 255.255.255.0

rtrA#show ip bgp 199.172.1.0/22

```
rtrA#show ip bgp 199.172.2.0/23 longer-prefixes
BGP table version is 90, local router ID is 199.172.15.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
s> 199.172.2.0      0.0.0.0                  0         32768 i
*> 199.172.2.0/23   0.0.0.0                            32768 i
s> 199.172.3.0      0.0.0.0                  0         32768 i
```

同时，如上命令也可以使用| begin include exclude 过滤显示

```
rtrA#show ip bgp 199.172.2.0 255.255.254.0 longer-prefixes | include s
BGP table version is 90, local router ID is 199.172.15.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
s> 199.172.2.0      0.0.0.0                  0         32768 i
s> 199.172.3.0      0.0.0.0                  0         32768 I

rtrA#show ip bgp 199.172.2.0 255.255.254.0 longer-prefixes | exclude s
   Network          Next Hop            Metric LocPrf Weight Path
*> 199.172.2.0/23   0.0.0.0                            32768 I
```

show ip bgp cidr-only | include line

显示具有非正常 mask 的路由条目，正常的 mask 为 A 类 8bit B 类 16bit C 类 24bit

show ip bgp community community-number(s) | include line exact-match

```
rtrA#show ip bgp community no-export
BGP table version is 41, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.1.0/24    10.1.1.1                 0             0 65530 i
```

show ip bgp community-list community-list-number exact-match | include line

```
ip community-list 1 permit no-export
rtrA#show ip bgp community 1
BGP table version is 41, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
```

```
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.1.0/24    10.1.1.1                 0             0 65530 I
```

show ip bgp dampened-paths | include line

```
rtrA#show ip bgp dampened-paths
BGP table version is 7, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          From            Reuse    Path
*d 172.16.2.0/24    10.1.1.1        00:19:30 1 i
```

show ip bgp filter-list as-path-access-list | include line

```
ip as-path access-list 1 permit ^2$

rtrA#show ip bgp filter-list 1
BGP table version is 90, local router ID is 199.172.15.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 200.1.1.0        172.17.1.1               0             0 2 i
*> 200.1.2.0        172.17.1.1               0             0 2 i
*> 200.1.3.0        172.17.1.1               0             0 2 i
*> 200.1.4.0        172.17.1.1               0             0 2 I
```

show ip bgp flap-statistics | include regular-expression

```
rtrA#show ip bgp flap-statistics
BGP table version is 9, local router ID is 10.1.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          From            Flaps Duration Reuse    Path
*d 172.16.2.0/24    10.1.1.1        3     00:02:55 00:28:10 1
```

show ip bgp inconsistent-as | include line

来自相同下一跳地址的路由需要来自相同的 AS，此条命令显示来自不同 AS 的相同下一跳地址的路由

```
rtrA#show ip bgp inconsistent-as
BGP table version is 3, local router ID is 172.17.1.1
Status codes: s suppressed, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
```

```
   Network          Next Hop            Metric LocPrf Weight Path
*  199.1.1.0        172.17.1.2               0             0 1 2 ?
*>                  172.17.1.2               0             0 3 ?
```

show ip bgp neighbors ip-address routes

```
rtrA#show ip bgp neighbors 172.17.1.2
BGP neighbor is 172.17.1.2,  remote AS 1, external link
  BGP version 4, remote router ID 199.172.15.1
  BGP state = Established, up for 1w1d
  Last read 00:00:43, hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised
    Address family IPv4 Unicast: advertised and received
  Received 13174 messages, 0 notifications, 0 in queue
  Sent 13168 messages, 0 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 14, neighbor version 14
  Index 1, Offset 0, Mask 0x2
  3 accepted prefixes consume 108 bytes
  Prefix advertised 40, suppressed 0, withdrawn 0

  Connections established 10; dropped 9
  Last reset 1w1d, due to User reset
Connection state is ESTAB, I/O status: 1, unread input bytes: 0
Local host: 172.17.1.1, Local port: 11106
Foreign host: 172.17.1.2, Foreign port: 179

Enqueued packets for retransmit: 0, input: 0  mis-ordered: 0 (0 bytes)

Event Timers (current time is 0x357A31F8):
Timer          Starts    Wakeups            Next
Retrans         11859          0             0x0
TimeWait            0          0             0x0
AckHold         11862      11505             0x0
SendWnd             0          0             0x0
KeepAlive           0          0             0x0
GiveUp              0          0             0x0
PmtuAger            0          0             0x0
DeadWait            0          0             0x0

iss: 2220955579  snduna: 2221180949  sndnxt: 2221180949     sndwnd:  16327
irs: 1147082461  rcvnxt: 1147307991  rcvwnd:      16156  delrcvwnd:    228
```

```
SRTT: 300 ms, RTTO: 607 ms, RTV: 3 ms, KRTT: 0 ms
minRTT: 0 ms, maxRTT: 376 ms, ACK hold: 200 ms
Flags: higher precedence, nagle
Datagrams (max data segment is 1460 bytes):
Rcvd: 23247 (out of order: 0), with data: 11862, total data bytes: 225529
Sent: 23520 (retransmit: 0), with data: 11859, total data bytes: 225388
```

show ip bgp paths | include line

```
rtrA#show ip bgp paths
Address    Hash Refcount Metric Path
0x125E94      0        4      0 i
0x996F4       2        1      0 1 i
0x125D6C      2        2      0 1 i
```

show ip bgp peer-group peer-group-name **summary**

```
rtrA#show ip bgp peer-group
BGP neighbor is demo, peer-group leader
  BGP version 4
  Minimum time between advertisement runs is 5 seconds
  Incoming update AS path filter list is 30
  Outgoing update AS path filter list is 40
  Route map for outgoing advertisements is adjust-weight
```

show ip bgp summary | include line

```
rtrA#show ip bgp summary
BGP router identifier 200.1.4.1, local AS number 2
BGP table version is 14, main routing table version 14
7 network entries and 7 paths using 931 bytes of memory
3 BGP path attribute entries using 156 bytes of memory
1 BGP AS-PATH entries using 24 bytes of memory
0 BGP route-map cache entries using 0 bytes of memory
0 BGP filter-list cache entries using 0 bytes of memory
BGP activity 193/657 prefixes, 193/186 paths, scan interval 15 secs

Neighbor        V    AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd

10.1.1.2        4     2    7584    7590       14    0    0 5d06h           0
172.17.1.2      4     1   13359   13353       14    0    0 1w1d            3
```

Clear ip bgp

```
clear ip bgp *
clear ip bgp * soft
clear ip bgp * soft in
clear ip bgp * soft out
clear ip bgp * soft in out
clear ip bgp AS-number
```

clear ip bgp AS-number soft
clear ip bgp AS-number soft in
clear ip bgp AS-number soft out
clear ip bgp AS-number soft in out
clear ip bgp neighbor-ip-address
clear ip bgp neighbor-ip-address soft
clear ip bgp neighbor-ip-address soft in
clear ip bgp neighbor-ip-address soft out
clear ip bgp neighbor-ip-address soft in out
clear ip bgp peer-group peer-group-name
clear ip bgp peer-group peer-group-name soft
clear ip bgp peer-group peer-group-name soft in
clear ip bgp peer-group peer-group-name soft out
clear ip bgp peer-group peer-group-name soft in out

可以对特定的 AS 或者对等体组或者特定的邻居 clear bgp 消息

clear ip bgp dampening prefix mask

重置 BGP 惩罚信息

clear ip bgp flap-statistics prefix mask

重置路由摆动的统计信息

debug ip bgp

debug ip bgp
debug ip bgp neighbor-ip-address updates
debug ip bgp neighbor-ip-address updates access-list-number
debug ip bgp dampening
debug ip bgp dampening access-list-number
debug ip bgp events
debug ip bgp keepalives
debug ip bgp in neighbor-ip-address updates
debug ip bgp in neighbor-ip-address updates access-list-number
debug ip bgp out neighbor-ip-address updates
debug ip bgp out neighbor-ip-address updates access-list-number
debug ip bgp updates
debug ip bgp updates in
debug ip bgp updates out
debug ip bgp updates access-list-number
debug ip bgp updates access-list-number in
debug ip bgp updates access-list-number out

Campus Network

园区网特点

1. 在一个固定地理区域内的一个公司或一个公司的一部分。
2. 拥有该园区网的公司通常也拥有该园区内所用的物理线路。

传统园区网的主要问题

1. 可用性
2. 性能

在传统园区网中，通常用多端口网桥将一个局域网分段成隔离的冲突域。这样可解决两个问题：

1. 冲突域（Collision Domain）
2. 距离限制

网络中通信的三种形式：

单播（Unitcast）
组播（Multicast）
广播（Broadcast）

1. 多点广播实例：Cisco IP/TV 分发多媒体数据、定位 IP 服务上的 Novell 5。
2. 提出请求的广播：IP 的地址解析协议（ARP）、NetBIOS 的名字请求、网间包交换协议（IPX）寻找最近服务器（Get Nearest Server，GNS）请求。
3. 发布通告的广播：IPX 服务通告协议（SAP）数据包、路由信息协议（RIP）、内部网关路由选择协议（IGRP）。

遏制广播的两种方法：

1. 使用路由器生成多个子网；
2. 利用交换机实施 VLAN。

当前园区网由两部分组成：

1. 局域网交换机
2. 路由器

导致流量模式的改变有两个因素：

1. 基于 Web 应用的计算普遍，很多 PC 既是信息的接受者，也是信息的发布者；
2. 企业部署集中式的服务器群（既降低成本、提高安全、便于管理）。

新的园区网模型中的 3 类服务

1. 本地服务：本地数据流不进入网络主干或通过路由器
2. 远程服务：远程服务数据流穿过广播域边界，但可能也可不通过网络主干
3. 企业级服务：放在距离网络主干很近的一个独立的子网上

与 OSI 分层相应的 PDU 和设备类型

模型层	PDU 类型	设备类型
数据链路层（第 2 层）	数据帧	交换机/网桥
网络层（第 3 层）	数据包	路由器
传输层（第 4 层）	TCP 数据分段	TCP 端口

多层交换机

多层交换基于单独的流，MLS-SE 为 MLS 流维护一个缓存条目并为每个流存储统计信息。流中的所有数据包都与缓存中的信息进行比较。

缺省情况下，256 秒之后，如果没有任何流利用到一个 MLS 缓存项（Cache Entry），那么这个缓存项将从缓存中删除。

Layer-2：支持 MAC 转发
Layer-3：支持 BGP OSPF RIP 等路由协议，PIM 多播，HSRP VRRP
Layer-4：支持服务器负载均衡，区分不同类型数据流 流量过滤 Qos 等
Layer-7：内容智能，区分 voip 长途 or local 等

路由器的优势

决定转发路径
验证 3 层包头的完整性、有效期(on header only)
修改 TTL
处理并响应任何选项信息
MIB 中更新转发统计数据
安全控制
第 3 层交换的优势（路由器没有）
低成本
低延时 //基于硬件的转发

交换机和网桥

第二层交换机由于采用 ASIC（专用集成电路，Application-Specified Integrated Circuits）硬件处理技术，所以交换机可比以太网桥低得多的成本提供高达 Gbit 速率的可扩展性和低时延。

第三层交换机主要有两种产品

多层交换
Cisco 快速转发（CEF）

Cisco 分层模型中各层使用的主要设备

层次	层次名	设备
第一层	访问层	Catalyst 1900，2820，2900，4000，5000
第二层	分布层	Catalyst 5000（支持多层交换，带路由模块），2926G（需要外部路由支持）6000（密集 Fast 或 Gigabit 以太网口，如 120 个 Gigabit 端口）
第三层	核心层	Catalyst 6500，8500（multicast routing，支持 PIM 协议）

接入层交换机应用

接入端口数	交换机
Less than 50	19xx, 2820, 29xx（如 CAD/CAM 和 IC 设计环境），35xx
Less than 100	4xxx（可提供多达 36Gbit 以太网端口，96 个用户接入）
More than 100	5xxx（Multigigabit 10/100/1000Mbps）

园区两个基本元素：
1. 交换区块（Switching Block）
2. 核心区块（Core Block）

影响交换区大小的主要因素：
1. 数据类型和行为；
2. 工作组的大小和数量（一般不超过 2000 个用户）

说明交换区过大：
1. 分布层路由上出现流量瓶颈；
2. 广播和 Multicast 降低了 Switch 和 Router 的处理速度。

分割交换区的原则

1. 应基于网络上通过的流量（Traffic Flow），而不是 Blocking 中的节点数；
2. 为了进行分割，需要定期进行流量采集。

有两种基本的核心层设计：

1. 紧缩核心（Collapsed）
   - ◎ 分布层和核心层功能由同一个设备执行；
   - ◎ 每台接入层交换机到分布层交换机都有一条冗余链路；
   - ◎ 第三层冗余是由运行 HSRP 的两台分布层交换机提供的。
2. 双核心（Dual）：在核心层至少有两个设备提供冗余。但他们之间没有连接，以防止生成树循环。

路由选择协议所支持 Blocking 的最大数量

协议	支持路由对等的最大数量	核心层子网数	Blocking 数
OSPF	50	2	25
EIGRP	50	2	25
RIP	30	2	15

实施第三层核心的好处：

很多设计采用第二层——第三层——第二层的模型，取得了成功，但有些情况下需要使用第三层核心，主要好处：

1. 快速收敛：路由协议收敛时间 5s~10s，而生成树收敛时间在 50s；
2. 自动负载均衡：路由协议可在多条等成本路径间均衡负载；
3. 消除对等问题：可以支持更多的 Switching Blocking，达 100 个。

坏处：费用和性能。

传统路由器功能：

_ Determine paths based on logical addressing
_ Run layer 3 checksums (on header only)
_ Use Time to Live (TTL)
_ Process and responds to any option information
_ Can update Simple Network Management Protocol (SNMP) managers with Management Information Base (MIB)
_ Provide Security

第三层交换机优点：

_ Hardware-based packet forwarding
_ High-performance packet switching
_ High-speed scalability
_ Low latency
_ Lower per-port cost
_ Flow accounting
_ Security
_ Quality of service (QoS)

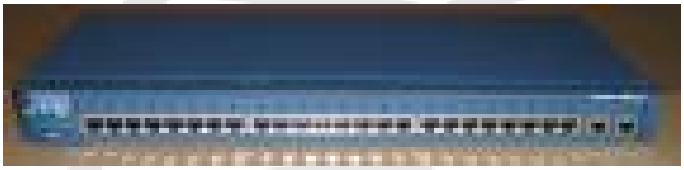
Switching hardware

Catalyst 1900

这种系列的交换机是能够在以太网使用的交换机中最小的一种。由于交换机所使用的软件的版本与交换机的性能有关，所以，按照交换机所使用的软件版本，这种系列的交换机的型号又分为3种。Catalyst 1900交换机的软件版本主要包括两种，分别是常规版和企业版，其中常规版C a t a l y s t可以提供小型交换机所具有的所有基本功能，如虚拟局域网和直通交换技术。企业版增加了一些高级功能，如快速以太通道（ Fast EtherChannel）和快速以太网端口上的I S L中继。企业交换机型号的最后标有“ E N”的字样。

Catalyst 1900的物理版本有3种。每种版本都有一个A U I（attachment user interface，连接用户接口）端口和两个快速

以太网端口。Catalyst 1912有1 2个带宽为1 0 M b / s的以太网端口，Catalyst 1924有2 4个1 0 M b / s的端口。Catalyst 1924C和Catalyst 1924一样，也有2 4个1 0 M b / s的端口，但是，在它的两个快速以太网端口中使用了一个光纤连接器。上面所述的这些交换机对于小型的配线柜来说是非常理想的，而且当其软件为企业版时，还允许实现中继和使快速以太通道返回主配线柜。Catalyst 1924C允许将小的配线柜放置在远至2 k m的地方



Catalyst 1900系列交换机有以下特性：

- 其中的标准软件可以升级成企业版。
- 在所有的以太网和快速以太网端口上进行全双工操作。
- 控制拥塞，包括基于IEEE 802.3x的流控制器。
- 基于We b进行网络管理。
- 控制端口广播的风暴，防止由于广播风暴而造成工作站的系统崩溃。
- 支持C i s c o的6 0 0 W冗余交流电源系统，提供备用电源，以改进误差容许度和增加网络的正常工作时间。
- 在网桥表中允许存储1 0 2 4个M A C地址。
- 支持S N M P（简单网络管理协议）和Te l n e t。
- 支持四种嵌入式的R M O N（远程监视），即历史、统计、警告和事件的R O N M。
- 通过使用交换机探测（ s w i t c h - p r o b e）分析器端口支持所有的9种R M O N，其中，这个分析器端口允许对单一端口、一组端口，或整个交换机进行监视（仅企业版）。

Catalyst 2900XL

C i s c o在推出了Catalyst 5000的交换机之后，又推出了Catalyst 2900XL系列，这两种系列的交换机很难区分。在Catalyst 5000系列的交换机中，有一种型号为Catalyst 2900的交换机，需要注意的是，这种型号的交换机并不与2 9 0 0 X L系列有任何关系，虽然其型号看起来与Catalyst 2900XL有点相同。Catalyst 2900XL运行真正的Cisco IOS，而不像Catalyst 5000系列那样使用它自己的操作系统。Catalyst 2900XL 的底板（ b a c k p l a n e）为3 . 2 G b / s，而C a t a l y s t5 0 0 0系列的底板只有1 . 2 G b / s。Catalyst 2900XL看起来很像Catalyst 1900或Catalyst 2820，所以，在确定交换机的型号之前，必须对交换机进行仔细的检查。

Catalyst 2900XL 的功能要比Catalyst 1900 强大得多。例如，它所有端口的带宽都为1 0 / 1 0 0 M b / s，而且所有的端口都与快速以太通道兼容。不过， Catalyst 2900XL的价格要比Catalyst 1900贵得多，但比Catalyst 5000便宜。Catalyst 2900XL有几种不同的类型，在这里将它们分为两组：即模块化的和非模块化的。在型号中有“ M”字

样的所有Catalyst 2900XL交换机中都有两个模块化的机架，这两个机架看起来与Catalyst 2820中的机架非常相似。不过，这些放进机架的模块都是不可互换的。

Catalyst 2900XL系列交换机的一些重要特性如下：

- 10baseT/100baseTX的端口。
- 3.2 Gb/s的交换构造和超过每秒1.19百万个信息包的转发速率。
- 交换100baseT端口上的全双工操作。
- 4MB的共享内存结构。
- 快速以太通道的拓扑结构。
- 端口广播风暴的控制。
- 使用和配置简单。
- 基于Web的接口。
- 存在于闪存中的默认配置。
- 简单的SNMP（网络管理协议）和Telnet支持。
- 基于Cisco IOS CLI的管理。
- 通过Cisco Work Windows网络管理软件支持4组RMON，加强网络的通信量管理、通信量监视和通信量分析。
- 通过使用交换机-端口的分析器端口支持所有9个RMON组，这将允许从单一的网络分析器或RMON探测进行单端口、1组端口和整个端口高性能的通信量监视。

Catalyst 3750

采用思科StackWiseT 技术的Catalyst 3750系列交换机，从而进一步提高了堆叠交换的标准。这个新的产品线将为客户带来前所未有的可堆叠弹性和极为方便的管理，并可以提供千兆位以太网性能。它可以帮助中型机构和企业分支机构大幅度提高他们的局域网（LAN）的运营效率，并让他们可以适应不断变化的业务需求。

Catalyst 3750系列的核心是思科StackWise技术，它是一种创新性的堆叠架构，提供了一个32Gbps的堆叠互联，连

接多达9台交换机，并将它们整合为一个统一的、逻辑的、针对融合而优化的设备，从而让客户可以更加放心地部署语音、视频和数据应用。思科的这种创新技术可以为可堆叠交换机的客户带来下列优势：

通过基于硬件和软件的高级故障转移功能实现可堆叠弹性

基于分布式转发和服务质量（QoS）的统一服务

通过自动配置简化管理

性能能够满足千兆位以太网连接的要求

可以支持IPv6

Catalyst 3750G-24TS，配有24 个10/100/1000端口和4个小型可插拔（SFP）上行链路。

Catalyst 3750G-24T，配有24 个10/100/1000端口。

Catalyst 3750-24TS，配有24个10/100端口和2个SFP上行链路。

Catalyst 3750-48TS，配有48个10/100端口和4个SFP上行链路。

上图即为GBIC和SFP模块

Catalyst 4500

Cisco Catalyst 4500系列能够为无阻碍的第2/3/4层交换提供集成式弹性，因而能进一步加强对融合网络的控制。可用性高的融合语音/视频/数据网络能够为正在部署基于互联网企业应用的企业和城域以太网客户提供业务弹性。

作为新一代Cisco Catalyst 4000系列平台，Cisco Catalyst 4500系列包括三种新型Cisco Catalyst 机箱：Cisco Catalyst 4507R（七个插槽）、Cisco Catalyst 4506（六个插槽）和Cisco Catalyst 4503（三个插槽）。Cisco Catalyst 4500系列中提供的集成式弹性增强包括1＋1超级引擎冗余（只对Cisco Catalyst 4507R）、集成式IP电话电源、基于软件的容错以及1+1电源冗余。硬件和软件中的集成式冗余性能够缩短停机时间，从而提高生产率、利润率和客户成功率。

作为Cisco AVVID（集成语音、视频和融合数据体系结构）的关键组件，Cisco Catalyst 4500能够通过智能网络服务将控制扩展到网络边缘，包括高级服务质量（QoS）、可预测性能、高级安全性、全面管理和集成式弹性。由于Cisco Catalyst 4500系列提供与Cisco Catalyst 4000系列线卡和超级引擎的兼容性，因而能够在融合网络中延长

Cisco Catalyst 4000系列的部署窗口。

Cisco Catalyst 4000 和 4500 系列交换管理引擎

- Supervisor Engine II-Plus -- 以入门级价格提供增强的第二层特性，适用于小型第二层配线间以及简单的第三层 QoS 和安全性。
- Supervisor Engine II-Plus-TS -- 在 Supervisor Engine II-Plus 的基础上增加了 20 个线速千兆以太网（GE）端口（12 个千兆以太网 PoE 铜线端口、8 个千兆以太网 SFP 光端口）。只在 Catalyst 4503 机箱中支持。
- Supervisor Engine IV -- 中等配线间和第三层网络、提供增强的安全特性和第三层路由（EIGRP，OSPF，IS-IS 协议，BGP）。
- Supervisor Engine V -- 高级性能和先进特性，在 Catalyst 4510R 机箱中支持 242 个端口。
- Supervisor Engine V-10GE -- 在 Supervisor Engine V 的基础上添加了两条万兆以太网上行链路和 NetFlow。在 Catalyst 4510R 机箱中最多支持 384 个端口。

线卡

- 百兆以太网铜线 -- 百兆以太网线卡包括 24 端口和 48 端口连接类型，都带可任选 PoE。
- 百兆以太网光纤 -- 支持多模光纤和单模光纤，以及 FX、LX 和 BX 收发器。
- 千兆以太网铜线 -- 提供 24 端口或 48 端口，带或不带 PoE。
- 千兆以太网光纤 -- 千兆以太网光纤卡提供高性能千兆以太网上行链路和服务器群连接。提供多种端口数和光接口类型（千兆位接口转换器[GBIC]和 SFP 光接口）。

集中式架构

为实现特性丰富的线速性能，明智的以太网交换机厂商采用了三重内容可定址内存（TCAM）。Cisco Catalyst 4500 系列交换机使用独立 TCAM 资源，以独立于线速智能服务处理的方式处理第三层路由。其架构让每个功能和特性都拥有大量专用 TCAM 空间，以保证获得出色性能。

Catalyst 4500 Power Supplies

Cisco Catalyst 4500 系列的优点

- 安全、强大 -- 通过冗余组件提供高可靠性
- 服务中升级 -- 热插拔和删除组件
- 千兆以太网的价格 -- 比可堆叠设备更为经济有效（大于 48 个端口）
- 自适应性 -- 不需要大规模升级就能提供万兆以太网上行链路
- 极高的安全性 -- 利用 Cisco Catalyst 集成式安全特性，能够提供思科最全面的局域网接入保护
- 战略性思科平台 -- 已安装了 400,000 多个机箱
- 集中式体系结构 -- 简洁、扩展能力强、性能高
- 支持 IP 语音
- 投资保护
- 是目前部署最广泛的模块化交换机

Catalyst 6500

Catalyst 6500 系列为企业园区网和电信运营商网络设立了新的 IP 通信和应用支持标准，它不但能提高用户的生产率，增强操作控制，还能提供无与伦比的投资保护。作为思科重要的智能多层模块化交换机，Catalyst® 6500 系列能够提供安全的端到端融合网络服务，其使用范围从布线室到核心，再到数据中心和广域网边缘。

Cisco Catalyst 6500 系列能够通过多种机箱配置和 LAN/WAN/MAN 接口提供可扩展的性能和端口密度，因而能帮助企业和电信运营商降低总体拥有成本。

Cisco Catalyst 6500 系列交换机提供 3 插槽、6 插槽、9 插槽和 13 插槽的机箱，以及多种集成式服务模块，包括数千兆位网络安全性、内容交换、语音和网络分析模块。 Catalyst 6500 系列中的所有型号都使用了统一的模块和操作系统软件，形成了能够适应未来发展的体系结构，由于能提供操作一致性，因而能提高 IT 基础设施的利用率，并增加投资回报。从 48 端口

到 576 端口的 10/100/1000 以太网布线室到能够支持 192 个 1Gbps 或 32 个 10Gbps 骨干端口，提供每秒数亿个数据包处理能力的网络核心，Cisco Catalyst 6500 系列能够借助冗余路由与转发引擎之间的故障切换功能提高网络正常运行时间。

Cisco Catalyst 6500 系列的优点

Cisco Catalyst 6500 系列不但能为企业和电信运营商提供市场领先的服务、性能、端口密度和可用性，还能提供无与伦比的投资保护能力，包括：

- 最长的网络正常运行时间--利用平台、电源、控制引擎、交换矩阵和集成网络服务冗余性提供 1～3 秒的状态故障切换，提供应用和服务连续性统一在一起的融合网络环境，减少关键业务数据和服务的中断。
- 全面的网络安全性--将切实可行的数千兆位级思科安全解决方案集成到现有网络中，包括入侵检测、防火墙、VPN 和 SSL。
- 可扩展性能--利用分布式转发体系结构提供高达 400Mpps 的转发性能。
- 能够适应未来发展并保护投资的体系结构--在同一种机箱中支持三代可互换、可热插拔的模块，以提高 IT 基础设施利用率，增大投资回报，并降低总体拥有成本；
- 操作一致性--3 插槽、6 插槽、9 插槽和 13 插槽机箱配置使用相同的模块、Cisco IOS Software、Cisco Catalyst Operating System Software 以及可以部署在网络任意地方的网络管理工具。
- 卓越的服务集成和灵活性--将安全和内容等高级服务与融合网络集成在一起，提供从 10/100 和 10/100/1000 以太网到万兆以太网，从 DS0 到 OC-48 的各种接口和密度，并能够在任何部署项目中端到端地执行。

在端到端 Cisco Catalyst6500 系列部署中的操作一致性

- 3 插槽、6 插槽、9 插槽和 13 插槽机箱配置使用相同的模块、软件和网络管理工具
- 可部署在网络的任意地方--从布线室到核心、数据中心和广域网边缘
- 为降低备件和培训成本，与 Cisco 7xxx 路由器系统使用相同的广域网端口适配器(PA)
- 用户可以自行选择所有控制引擎上支持的 Cisco IOS Software 和 Cisco Catalyst Operating System，确保顺利地从 Cisco Catalyst 5000 系列、Cisco 7500 系列移植到 Cisco Catalyst 6500 系列。

提高网络正常运行时间，提高网络弹性

- 提供数据包丢失保护，能够从网络故障中快速恢复
- 能够在冗余控制引擎间实现快速的 1～3 秒状态故障切换
- 提供可选的高性能 Cisco Catalyst 6500 系列 Supervisor Engine 720、无源背板、多引擎的冗余；并可利用 Cisco EtherChannel?技术、IEEE 802.3ad 链路汇聚、IEEE 802.1s/w 和热备份路由器协议/虚拟路由器冗余协议（HSRP/VRRP）达到高可用性

集成式高性能的网络安全性和网络管理

不需要部署外部设备，直接在 6500 机箱内部署集成式的千兆位的网络服务模块，以简化网络管理，降低网络的总体成本。这些网络服务模块包括：

- 数千兆位防火墙模块--提供接入保护
- 高性能入侵检测系统（IDS）模块--提供入侵检测保护
- 千兆位网络分析模块--提供可管理性更高的基础设施和全面的远程超级（RMON）支持
- 高性能 SSL 模块--提供安全的高性能电子商务流量终结
- 千兆位 VPN 和基于标准的 IP Security（IPSec）模块--降低的互联网和内部专网的连接成本。

能感知网络内容和网络应用的第 2～7 层交换服务

- 集成式内容交换模块（CSM）能够为 Cisco Catalyst 6500 系列提供功能丰富的高性能的服务器和防火墙网络负载平衡连接，以提高网络基础设施的安全性、可管理性和强大控制
- 集成式数千兆位的 SSL 加速模块与 CSM 结合在一起，能提供高性能的电子商务解决方案
- 集成式数千兆位的防火墙和 CSM 能提供安全的高性能数据中心解决方案
- 基于网络的应用识别（NBAR）等软件特性可提供增强网络管理和 QoS 控制机制。

可扩展的性能

- 利用分布式 Cisco Express Forwarding dCEF720 平台提供业界最高的交换机性能--400Mpps
- 支持多种 Cisco Express Forwarding（CEF）实现方式和交换矩阵速率。在布线室、核心、数据中心、广域网边缘部署以及电信运营商网络均可提供最优配置。

丰富的第 3 层网络服务

- 多协议第 3 层路由支持满足了传统的网络要求，并能够为企业网络提供平滑的过渡机制
- 硬件支持的从企业级到电信运营商级的大规模路由表
- 硬件支持 IPv6，并提供高性能的 IPv6 服务
- 在硬件中提供 MPLS 及 MPLS/VPN 的支持，可应用在高速电信运营商的网络核心和城域以太网，提供丰富的 MPLS 服务。

增强的数据、语音和视频服务

- 在所有 Cisco Catalyst 6500 系列平台上提供集成式 IP 通信
- 提供 10/100 和 10/100/1000 接口模块，借助在接口模块内增加电源子卡就可让这些接口模块提供在线的电源，提供 IEEE 802.3af 的支持，保护今天的投资
- 提供高密度的 T1/E1 和 FXS 的 VoIP 语音网关接口，可与公共电话网（PSTN）、传统的电话、传真和 PBX 连接，提供 VoIP 服务。
- 支持高性能的 IP 组播视频和音频应用
- 提供一个统一管理的、经济的、可灵活扩展的网络。

最高的接口灵活性、可扩展性和端口密度

- 满足大型关键业务布线室、企业核心层和分布层需要的端口密度和接口类型
- 每台设备可提供 576 个支持语音的，具有在线电源的 10/100/1000M 铜线接口
- 提供 192 个 GBIC 千兆位以太网接口
- 率先推出业界第一个万兆以太网接口，和可提供高密度的 OC-3 POS 接口的通道化的 OC-48 接口。
- 通过系列 FlexWAN 模块上使用 Cisco 7xxx 系列端口适配器(PA)，Cisco Catalyst 6500 可支持 T1/E1～OC-48 广域网接口，具有很高的投资保护能力
- 机箱从 3 插槽（6503）、6 插槽（6506）、9 插槽（6509）到 13 插槽（6513）。

高速广域网接口

- 提供与其它核心路由器兼容的高速广域网、ATM 和 SONET 接口
- 单一平台实现广域网汇聚以及园区网和城域连接管理 最高投资保护
- 高度灵活的模块化体系结构，在同一机箱内支持多代模块互操作
- 所有引擎上都支持 Cisco IOS Software 和 Cisco Catalyst Operating System Software
- 10/100Mbps 和 10/100/1000Mbps 以太网模块可现场升级为随线供电的模块，可让用户在需要时升级实现 IP 电话技术和无线局域网连接
- 可在各种应用场合中添加的不断推出的网络服务模块
- 包括 Cisco Catalyst 6500 系列网络安全性、内容交换和语音功能
- 未来的模块将进一步提高性能、端口密度，并推出其它服务

适用于城域以太网广域网服务

- 802.1Q 和 802.1Q 隧道（QinQ）提供点到点和点到多点以太网服务
- EoMPLS 的功能提供了 VLAN 的透传功能，大幅提升 MPLS 骨干网中的以太网服务扩展能力
- 通过在第 2 层和第 3 层 QoS 功能中提供速率限制和流量整形，可在城域以太网服务中提供分级的带宽服务
- 增强的生成树协议、IEEE 802.1s、IEEE 802.1w 和 Cisco EtherChannel IEEE 802.3ad 链路汇聚功能提供了网络超高的可用性。

Switching Block

FastEternet 的距离限制

技术	线缆分类	线缆长度
100BaseTX	EIA/TIA 类型 5（UTP） 非屏蔽双绞线 2 对	100m
100BaseT4	EIA/TIA 类型 3,4,5（UTP） 非屏蔽双绞线 4 对	100m
100BaseFX	多模光纤 MMF 缆线 62.5um 光纤核心 ，125um 外层包装（62.5/125）	400m

Gbit 以太网距离限制

技术	线缆分类	线缆长度
1000BaseCX	铜质屏蔽双绞线	25m
1000BaseT	铜质 EIA/TIA 类型 5（UTP） 非屏蔽双绞线 4 对	100m
1000BaseFX	多模光纤 62.5um 光纤核心和 50um 光纤芯，使用波长为 780nm	260m
1000BaseLX	单模光纤 9um 光纤芯，使用波长为 1300nm	3km（ Cisco 最长支持 10km）

Catalyst 两种 OS

OS 类型	交换机
Cisco IOS	Catalyst 1900/2800，2900XL
Set 命令集	Catalyst 2926，2926G，1948G，4000，5000，6000

802.3ab 规范要求铜线上的 Gi-Ethernet 必须使用自动协商

FastEthernet and GbitEthernet 的自动协商 802.3u

NIC	port	Nic-work	Switch-Work
Auto	Auto	100full	100full
100Full	Auto	100full	100half
Auto	100full	100half	100full
100half	auto	100half	100half
10half	auto	10half	10half
10half	100full/half	down	down
Auto	10full/half	10half	10half

Cisco Long-reach Ethernet :

用于扩展以太网 实现 5~15Mbps 连接，工作距离 5kinch

可以在 POTS 数字电话和 ISDN 相同的线路上提供宽带服务，同时和 adsl 相兼容的模式

UTP-cable

1. 从交换机到配线架（Patch Panel）为 5m；//过长线路容易产生串扰
2. 从 Patch Panel 到办公室模块（Punch-down Block）为 90m；
3. 从 Punch-down Block 到 Desktop 为 5m。

CDP

是 Cisco 的专有协议，用来发现邻居设备，Cisco 设备每 60s 发送基于第二层的 Multicast，目的 MAC 地址是 0100.0ccc.cccc。

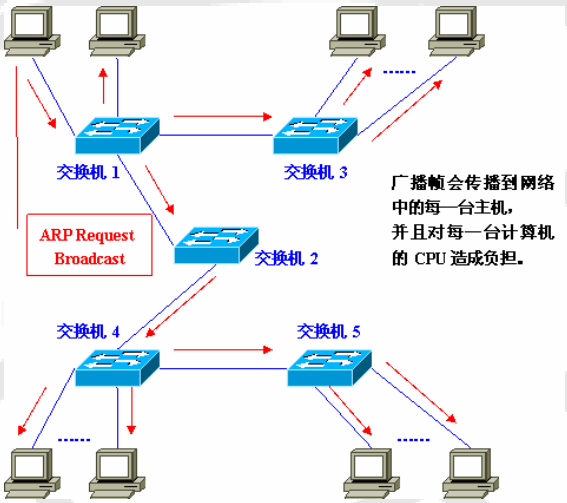
VLAN

What is vlan

广播域，指的是广播帧（目标 MAC 地址全部为 1）所能传递到的范围，亦即能够直接通信的范围。严格地说，并不仅仅是广播帧，多播帧（Multicast Frame）和目标不明的单播帧（Unknown Unicast Frame）也能在同一个广播域中畅行无阻。

未分割广播域时……

如果仅有一个广播域，有可能会影响到网络整体的传输性能。



图中，是一个由 5 台二层交换机（交换机 1～5）连接了大量客户机构成的网络。假设这时，计算机 A 需要与计算机 B 通信。在基于以太网的通信中，必须在数据帧中指定目标 MAC 地址才能正常通信，因此计算机 A 必须先广播“ARP 请求（ARP Request）信息”，来尝试获取计算机 B 的 MAC 地址。

交换机 1 收到广播帧（ARP 请求）后，会将它转发给除接收端口外的其他所有端口，也就是 Flooding 了。接着，交换机 2 收到广播帧后也会 Flooding。交换机 3、4、5 也还会 Flooding。最终 ARP 请求会被转发到同一网络中的所有客户机上。

1. 广播信息消耗了网络整体的带宽，
2. 收到广播信息的计算机还要消耗一部分 CPU 时间来对它进行处理。造成了网络带宽和 CPU 运算能力的大量无谓消耗。

广播信息是那么经常发出的吗？

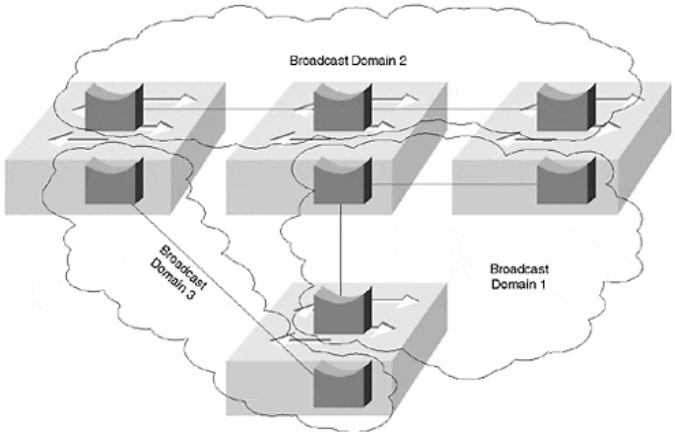
利用 TCP/IP 协议栈通信时，除了前面出现的 ARP 外，还有可能需要发出 DHCP、RIP 等很多其他类型的广播信息。

如果整个网络只有一个广播域，那么一旦发出广播信息，就会传遍整个网络，并且对网络中的主机带来额外的负担。因此，在设计 LAN 时，需要注意如何才能有效地分割广播域。

分割广播域时，一般都必须使用到路由器。使用路由器后，可以以路由器上的接口为单位分割广播域。
通常情况下路由器上不会有太多的网络接口，其数目多在 1～4 个左右。

与路由器相比，二层交换机一般带有多个网络接口。因此如果能使用它分割广播域，那么无疑运用上的灵活性会大大提高。

用于在二层交换机上分割广播域的技术，就是 VLAN。通过利用 VLAN，我们可以自由设计广播域的构成，提高网络设计的自由度。



Vlan in campus network

在园区网中实现 vlan 可以有如下几个优点

1. 有效的利用带宽
2. 安全性
3. 多路径负载均衡
4. 隔离故障域

Vlan type

（1）本地 VLAN（local vlan） -- 一套设备群中的 VLAN，跨越交换机少，一栋楼或一房间

（2）端到端 VLAN（end-to-end vlan） -- 跨越多套设备群的 VLAN，不受位置局限

静态 VLAN – 先定义一个 VLAN，然后手工的把交换机的一个接口划到一个 VLAN 中去

动态 VLAN – 基于 MAC 或者子网或者用户的方式设置 vlan，交换机查询 VMPS（VLAN 管理策略服务器），查询 MAC 或者子网或者用户信息，就把该接口划入哪个 VLAN，而不管是哪个具体的接口

启用 VLAN：

方法一，全局下用 **vlan 3** 定义一个 VLAN，然后在 VLAN 模式中用 **name kaka** 定义该 VLAN 的名字，删除 VLAN 在全局中用 **no vlan 3**，也只有在退出后才生效

方法二，特权下用 **vlan database** 进入 VLAN 数据库，然后用 **vlan 3 (name kaka)**定义一个 VLAN，交换机会自动生成一个名字或手工指定一个名字，最后一定要用 **exit** 或 **apply** 使其生效，用 **abort** 使其不生效。要删除，在进入 VLAN 数据库后用 **no vlan 3**，同样要使其生效

第二种只能配置 vlan2~1002，而且在稍后的 IOS 版本中这种方式可能会被 delete

静态方式下把一个接口划入某 VLAN：

在接口下用 **switchport mode access** 指明该接口是接入（主机）端口

然后接口下用 **switchport access vlan 3** 把该端口划入 VLAN 3

用 **show vlan** 查看 VLAN 信息

用 **show interface f0/1 switchport** 查看该端口的二层状态

用 **show mac-address-table interface f0/1** 查看与该端口相关的 MAC 地址

用 **show mac-address-table interface f0/1 vlan 3** 查看从该端口学到的属于 VLAN 3 的 MAC 地址

pVlan

私有 vlan，企业中常用来防止连接到某些接口或者某些接口组的网络设备之间相互通讯，但是却允许和默认网关通讯.

每个 pvlan 包括 2 种 vlan，主 vlan 和辅助 vlan，辅助 vlan 是主 vlan 的子代，辅助 vlan 包括真实的端口，

Pvlan 定义使用混杂(promiscuous)端口

Promiscuous port

能够与 pvlan 中的全部设备进行通讯混杂端口只是主 vlan，每个混杂端口可以映射到多个辅助 vlan

通常混杂端口都是路由器的端口，备份服务器端口，或者 vlan 接口

Aux-vlan type

隔离 vlan：如果端口属于隔离 vlan 则只能和混杂端口通讯。

团体 vlan：可以和团体内的其他端口通讯，同时也可以和混杂端口通讯

配置 pvlan 的方法

```
Router(config)#vlan vlan-id
Router(config-vlan)#private-vlan {community | isolated | primary}
Router(config-vlan)#exit

Router(config)#vlan primary-vlan-id
Router(config-vlan)#private-vlan association {2nd-vlan-list | add 2nd-vlan-lsit | remove 2nd-vlan-lst}

Router(config)#interface vlan primary-vlan-id
Router(config-if)#private-vlan mapping {2nd-vlan-list | add 2nd-vlan-lsit | remove 2nd-vlan-lst}

Router(config)#interfacet type slot/port
Router(config-if)#switchport

Router(config-if)#switchport mode private-vlan {host | promiscuous}
Router(config-if)#switchport private-vlan host-association primary-vlan-id 2nd-vlan-id
Router(config-if)#switchport private-vlan mapping prmary-vlan-id
```

Trunk Link

汇聚链接（Trunk Link）指的是能够转发多个不同 VLAN 的通信的端口。

汇聚链路上流通的数据帧，都被附加了用于识别分属于哪个 VLAN 的特殊信息。

通过汇聚链路时附加的 VLAN 识别信息，有可能支持标准的“IEEE 802.1Q”协议，也可能是 Cisco 产品独有的“ISL（Inter Switch Link）”。如果交换机支持这些规格，那么用户就能够高效率地构筑横跨多台交换机的 VLAN。

另外，汇聚链路上流通着多个 VLAN 的数据，自然负载较重。因此，在设定汇聚链接时，有一个前提就是必须支持 100Mbps

以上的传输速度。

另外，默认条件下，汇聚链接会转发交换机上存在的所有 VLAN 的数据。换一个角度看，可以认为汇聚链接（端口）同时属于交换机上所有的 VLAN。由于实际应用中很可能并不需要转发所有 VLAN 的数据，因此为了减轻交换机的负载、也为了减少对带宽的浪费，我们可以通过用户设定限制能够经由汇聚链路互联的 VLAN。

Trunk Type

Cisco 支持多种 Trunk 方式（即对 VLAN 帧标识）：

1. ISL——Cisco 专有封装协议，也是默认的。前面加 26 字节，后面加 4 字节 FCS。
2. IEEE 802.1Q——IEEE 标准方法，在帧头写入 VLAN 信息，后面只增加 4 字节 FCS。
3. 802.10——FDDI 上传输 VLAN 信息的 Cisco 专有协议，把 VLAN 信息写入 SAID 安全关联标识符部分
4. LANE——基于 ATM 上传输 VLAN 信息的一种 IEEE 标准方法。

帧标记和封装方法

表示方法	封装	标记（插入帧内）	介质	帧长度
ISL	是	否	以太网	1518/1548
802.1Q	否	是	以太网	1518/1522
802.10	否	否	FDDI	
LANE	否	否	ATM	

Baby Giant Frame（小巨人帧）

原始以太网帧大小不超过 1518 字节，如果一个最大长度的帧是通过 802.1Q 来标记得，那么这个帧变成 1522 字节，这种帧被成为小巨人帧。

IEEE802.1Q

IEEE802.1Q，俗称“Dot One Q”，是经过 IEEE 认证的对数据帧附加 VLAN 识别信息的协议。

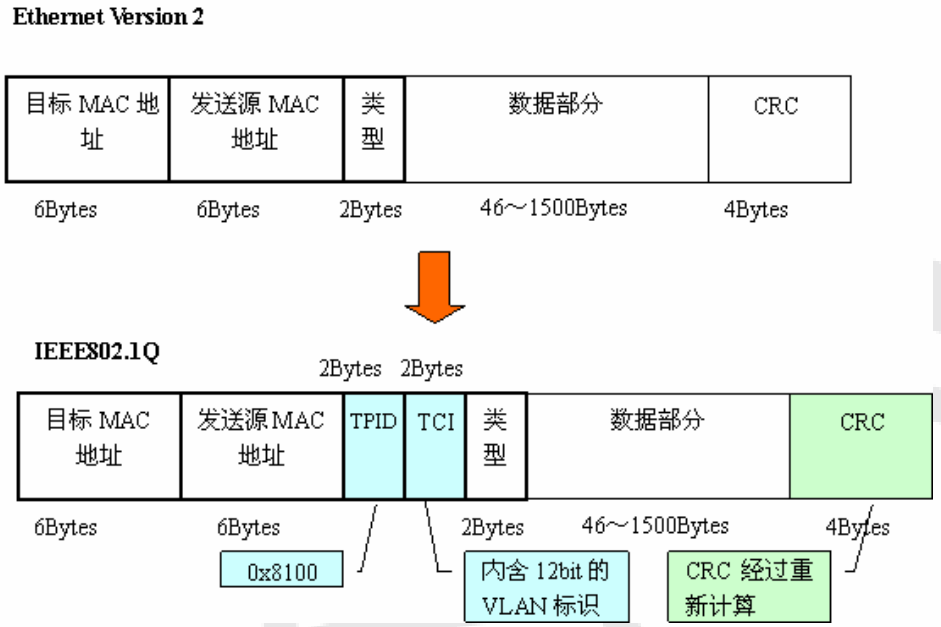
IEEE802.1Q 所附加的 VLAN 识别信息，位于数据帧中“发送源 MAC 地址”与“类别域（Type Field）”之间。具体内容为 2 字节的 TPID 和 2 字节的 TCI，共计 4 字节。

在数据帧中添加了 4 字节的内容，那么 CRC 值自然也会有所变化。这时数据帧上的 CRC 是插入 TPID、TCI 后，对包括它们在内的整个数据帧重新计算后所得的值。

而当数据帧离开汇聚链路时，TPID 和 TCI 会被去除，这时还会进行一次 CRC 的重新计算。

TPID 的值，固定为 0x8100。交换机通过 TPID，来确定数据帧内附加了基于 IEEE802.1Q 的 VLAN 信息。而实质上的 VLAN ID，是 TCI 中的 12 位元。由于总共有 12 位，因此最多可供识别 4096 个 VLAN。

基于 IEEE802.1Q 附加的 VLAN 信息，就像在传递物品时附加的标签。因此，它也被称作“标签型 VLAN（Tagging VLAN）”。



Nativevlan

802.1Q-trunk 为转发未被标记的 frame 而定义了 native VLAN。交换机能够从未被标记的 trunk 上的 nativeVLAN 转发 2 层 frame

接受方将把所有的未标记 frame 转发到 nativeVLAN 中 。

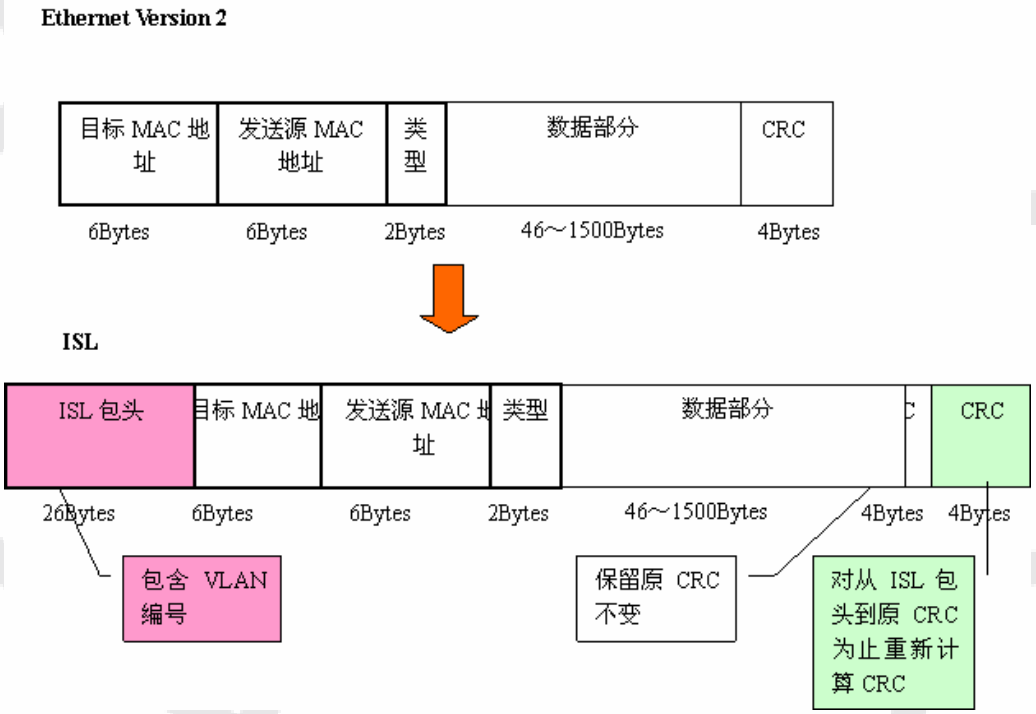
如果是 ISL，则没被封装的 frame 将会丢弃，包括 nativeVLAN，所有的数据将被封装

ISL（Inter Switch Link）

ISL，是 Cisco 产品支持的一种与 IEEE802.1Q 类似的、用于在汇聚链路上附加 VLAN 信息的协议。

使用 ISL 后，每个数据帧头部都会被附加 26 字节的“ISL 包头（ISL Header）”，并且在帧尾带上通过对包括 ISL 包头在内的整个数据帧进行计算后得到的 4 字节 CRC 值。换而言之，就是总共增加了 30 字节的信息。

在使用 ISL 的环境下，当数据帧离开汇聚链路时，只要简单地去除 ISL 包头和新 CRC 就可以了。由于原先的数据帧及其 CRC 都被完整保留，因此无需重新计算 CRC。



ISL 有如用 ISL 包头和新 CRC 将原数据帧整个包裹起来，因此也被称为“封装型 VLAN（Encapsulated VLAN）”。

并且由于 ISL 是 Cisco 独有的协议，因此只能用于 Cisco 网络设备之间的互联。

交换环境中的两种 link：

1. Access link（接入）：一单个 VLAN 的成员（A member of only one vlan）。

2. Trunk link（干道）：Capable of carring multiple vlans。

3..Dynamic（动态），即该链路既是 Trunk link 又是 Access link，它可传输两种帧：标记帧（带 VLAN 信息）和非标记帧

端口下用 **switchport mode access/trunk/dynamic** 指定

Vlan Range

VLAN 1 是本地 VLAN，所有未被划分的接口都属于该 VLAN

VLAN 号码：一共 4096 个可用 VLAN

0，4095 - 保留

1 - 本地 VLAN

2~~1001 - 用于以太网

VLAN1002：FDDI-Default

VLAN1003：Token-Ring-Default

VLAN1004：FDDInet-Default

VLAN1005：TRnet-default

1024~~4094 - 用于扩展的以太网

ISL 封装只支持到 1005；802.1Q 支持全部

802.1Q-in-Q

802.1Q 隧道：（Q in Q）(VLAN 里面包 VLAN)

为了避免在 VLAN 穿越 ISP 时有重名的 VLAN 号码，就需要进入 ISP 网络时把原来的 VLAN 包当成一个数据包打上一个标记在 ISP 的交换机上重新再封装进一个在 ISP 网络中独一无二的 VLAN 中，到了对端再解封装，在 ISP 边界交换机所连外部端口上用 **switchport mode dot1qtunnel** 进入隧道模式，在全局中用 **vlan dot1q tag native** 强制 ISP 边界交换机对 802.1Q 中继线上所有本地帧要求标记，没有标记的就丢弃。同时第二层协议（如 STP）一起被封装传输，ISP 边界交换机的边界接口上用 **l2protocol-tunnel cdp/stp/vtp** 使其能被传输

所以本地交换机和 ISP 中直连的交换机不是对称连接，即，本地连接端口是干线端口，ISP 中连接端口是接入端口

DTP

动态干线协商协议，cisco 专有，用于协商交换机 2 层端口为 **trunk** 或者 **access** 类型，每 30s 发送一次 DTP 的 frame

该协议仅在交换机间协商

协商方式有 5 种：

1. **acess** – 不启用 DTP，不协商
2. **trunk** – 会与对方协商，但是不论结果如果，它都是干线端口
3. **nonegotiate** – 指明它是干线端口，但是不主动向对方发 DTP 包，即不主动协商
4. **dynamic desirable** – 主动向对方发 DTP 包，以确定是否是干线或是接入端口
5. **dynmic auto** – 不主动向对方发 DTP 包，对方发送过来会回应

其中 1 属于 access 状态；2，3 属于 trunk 状态；4，5 属于 dynamic 状态

端口下用 switchport nonegotiate 指定

Trunk Configruation

```
Switch(config)# interface type mod/port
Switch(config-if)# switchport trunk encapsulation {isl | dot1q | negotiate}
Switch(config-if)# switchport trunk native vlan vlan-id
Switch(config-if)# switchport trunk allowed vlan {vlan-list | all | {add | except | remove} vlan-list}
Switch(config-if)# switchport mode {trunk | dynamic {desirable | auto}}
```

VTP

VTP（VLAN Trunk Protocol）管理交换机中配置的所有 VLAN，在一个管理域中向外分发 VLAN 的配置信息，并维持整个网络的一致性，是 cisco 的私有协议，它只在干线上以组播帧传播，此协议工作在第 2 层

1. 服务器：缺省模式，可建立、修改和删除 VLAN，向同一域中的交换机通告它的 VLAN 配置，并接受从 Trunk 链路上收到的通告与其它交换机进行 VLAN 配置的同步。

2. 客户机：行为同服务器模式，但不能建立、改变或删除 VLAN；倾听 vlan 信息，使得 z 自己的 vlan 配置信息保持与 vtp 服务器同步；也可以把 vlan 信息转发给其它交换机。

3. 透明：不参与 VTP。在 vtp v2 中，配置为透明模式的交换机将在 Trunk 端口上转发 VTP 信息以保证其他交换机接收到更新信息，但这些交换机将不修改自己的数据库，也不发送指示 VLAN 状态发生变化的更新信息。Vtp v1 中，透明模式的交换机也不转发 vtp 信息到其它交换机。需要注意的是透明模式下的交换机可以在本地创建 vlan，但这些 vlan 的变化信息不会扩散到其它交换机。

三种形式的 vtp 通告：

Summary advertisements—vtp 服务器发送，每隔 300s。

Subset advertisements—vtp 服务器发送。如 vlan 增删、vlan 的激活和挂起。

Advertisement requests from clients—vtp 客户发送，vtp 服务器回复 Summary advertisements 和 Subset advertisements。

两种原因促使 vtp 客户要发送请求

一种是从 Subset advertisements 了解到 vtp 状态发生变化；

另一种是从 Summary advertisements 获悉有更高的 vtp version number，

VTP 有 3 个版本

Vtp v2 有别于 v1 的一些特性：

1. Version-dependent transparent mode（与版本相关的透明模式）：v1 要先检查域名和版本，如果相同在转发；v2 则不检查版本。
2. Consistency checks（一致性检查）
3. Token Ring support（令牌环支持）：只有 v2 支持。
4. Unrecognized Type-Length-Value (TLV) support（不认识类型长度值的支持）

VTP3 支持创建和通告 pvlan 同时支持扩展 vlan

VTP pruning (0 裁减)：减少不必要的流量

裁减对透明模式的交换机不起作用，VLAN 1 和 1002~~1005 永远不做裁减

在一个域里加入一台交换机时，以客户机的模式加入，防止它传播不正确的 VLAN 信息

全局下用 **vtp version** 配置 VTP 版本

全局下用 **vtp mode server** 指定 VTP 模式

全局下用 **vtp domain kaka** 指定 VTP 域名，同一域里域名要一致

全局下用 **vtp password kaka** 指定 VTP 密码，同一个域里必须要有一致的密码

全局下用 **vtp pruning** 启用 VTP 裁减（即使被裁减掉的 VLAN，STP 还是会为其运行，所以最好把允许通过的 VLAN 手工指定，接口下用 **switchport trunk allowed vlan** …）

用 **show vtp status** 查看 VTP 配置

用 **show vtp counters** 查看 VTP 统计

用 **show vlan brief** 查看定义的 VLAN

用 **show interface f0/1 pruning** 查看 VTP 裁减状态

Vlan Routing

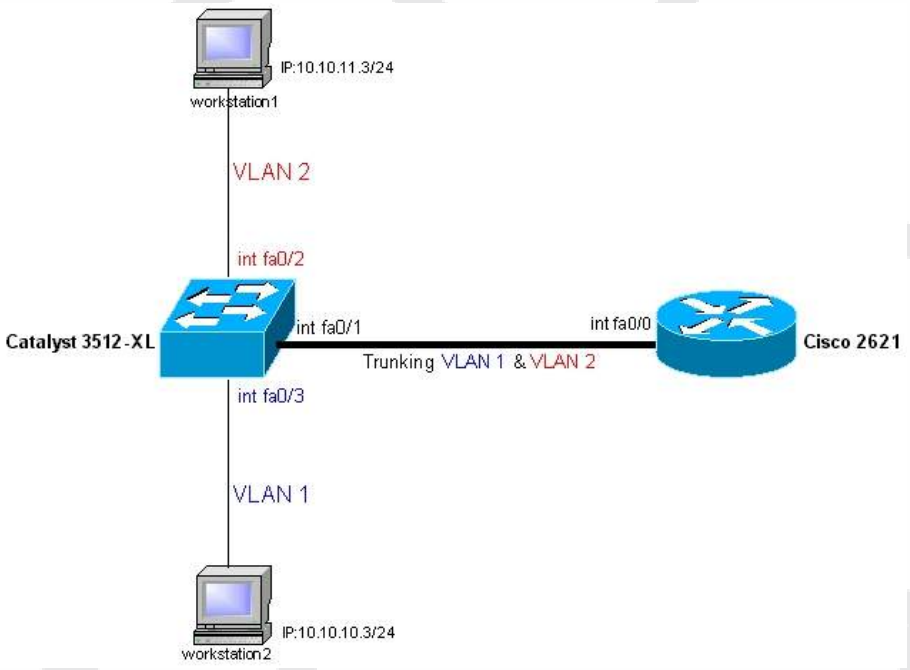
Vlan 间的通讯可以采用 3 种方式

1. 3 层交换机
2. 路由器上每个 vlan 分配一个独立的接口
3. 子接口做单臂路由

由于路由器是基于软件的转发，数据包仅 100，000pps 到 1，000，000pps。而 3 层交换机可以达到数百万 pps 的数据包吞吐

所以园区网中 3 层交换机使用较多。

单臂路由



交换机配置!

```
interface FastEthernet0/1
switchport mode trunk

!-- If 802.1Q is configured,
!-- you will instead see the following output
!-- under interface FastEthernet0/1:
!-- interface FastEthernet0/1
!-- switchport trunk encapsulation dot1q
!-- switchport mode trunk
!
interface FastEthernet0/2
switchport access vlan 2
spanning-tree portfast
!
interface FastEthernet0/3
spanning-tree portfast
!
interface VLAN1
ip address 10.10.10.2 255.255.255.0
no ip directed-broadcast
no ip route-cache
!
ip default-gateway 10.10.10.1
!
```

```
interface FastEthernet0/0
no ip address
duplex auto
speed auto
!
interface FastEthernet0/0.1
encapsulation isl 1
ip address 10.10.10.1 255.255.255.0
no ip redirects
!
!-- If 802.1Q is configured,
!-- you will instead see the following output
!-- under interface FastEthernet0/0.1:
!-- interface FastEthernet0/0.1
!-- encapsulation dot1Q 1 native
!-- ip address 10.10.10.1 255.255.255.0
!
interface FastEthernet0/0.2
encapsulation isl 2
ip address 10.10.11.1 255.255.255.0
no ip redirects
!
!-- If 802.1Q is configured,
!-- you will instead see the following output
!-- under interface FastEthernet0/0.2:
!-- interface FastEthernet0/0.2
!-- encapsulation dot1Q 2
!-- ip address 10.10.11.1 255.255.255.0
```

路由器配置

Layer3 switching

```
interface FastEthernet0/1
switchport access vlan 2

interface FastEthernet0/2
switchport access vlan 3

interface VLAN2
ip address 10.10.2.1 255.255.255.0

interface VLAN3
ip address 10.10.3.1 255.255.255.0

ip routing
router rip
network 10.0.0.0 255.0.0.0
```

IP foeward

例如 vlan 1 内有一台 DHCP 服务器，Vlan2 内为 DHCP 客户端，需要 DHCP 中继代理

此时需要在 vlan2 的 SVI 口上配置 ip helper-address dhcp-server-ip

Ip herlper-address 还可以转发 TFTP DNS NetBIOS 等，需要加入 ip forward-protocol udp netbios-ns

Spanning-Tree Protocols

一个交换机就是一个透明网桥，它对用户终端是透明的，它转发帧和广播，以及学习终端的 MAC 地址

交换机实际使用的三种交换技术：

存储转发(Store-and-forward) – 接收到整个数据帧并效验后转发

切头转发(Cut-through) – 接收到目的 MAC 地址就开始转发

无碎片(FragmentFree) – 接收到数据帧的前 64 位，只要不是碎片就转发

如果一台交换机收到一个单播帧，如果这个帧的目的地址未知，则该交换机广播这个帧

为了二层交换网络的稳定，通常采用冗余，而冗余会产生桥接环路（广播风暴），STP（生成树协议）就是用来发现和消除环路，它用标准的 802.1D 协议，阻碍冗余链路

Bridge-ID

网桥 ID 共 8 个字节，有两部分组成：

1. 2 个字节的优先级域：优先级低的为根桥，缺省优先级为 32768，即 0x8000。缺省地，所有 Cisco 交换机地优先级是相同的。

2. 6 个字节 MAC 地址域：即交换机或网桥的 MAC 地址。所以缺省情况下，具有最低 MAC 地址的交换机将成为根桥。全局下用 **spanning-tree vlan 200 priority 23423** 修改该交换机对一个特定 VLAN 的优先级

选择根桥（Root Bridge）

1. 自动选择：STP 自动选择具有最低 BID 的交换机为根桥。

2. 手工确定：原则是靠近网络的中心。所以一般根桥设在一台分布层的交换机而不是接入层交换机。建议手工设置来确定根桥。//手工修改 priority

确定到根桥的最佳路径

STP 协议利用 BPDU 中三个 Field——路径开销、网桥 ID、端口优先级/端口 ID 来确定到根桥的最佳路径顺序：

1. 路径开销：所有端口开销的综合为路径开销，路径开销低的端口为转发端口。

2. 网桥 ID：同一个交换机上有两条链路达到根桥（如平行链路），那么最佳路径就由下面的端口优先级或端口 ID 决定了。

3. 端口优先级/端口 ID：端口优先级范围 0～63，缺省值 32，具有低优先级的端口将转发数据。如果端口优先级相同，端口 ID 则是决定因素，低端口 ID 将转发数据。

BPDU：Bridge Protocol Data Unit

生成树协议用 BPDU (Bridge Protocol Data Units)进行信息的传递，它是一个二层的帧，用组播帧发送配置信息，选举根交换机，查找环路存在的位置

BPDU 有 2 种类型

Configuration-BPDU 配置 BPDU，包括 STP 参数

TCN-BPDU 拓扑变更通告 BPDU

BPDU 有三个计时器用来避免整个网络 BPDU 信息不同步而发生环路

Hello Time（默认 2s） - 多少时间发送一次

Max Age（默认 20s） - 当交换机收到一个 BPDU 后，多长时间内有效

Forward Delay（默认 15s） - 端口状态改变后，在监听和学习状态下分别停留的时间

当网络稳定下来后，所有 BPDU 计时器都是从根交换机学来的，整个网络计时器一致

全局下用 **spanning-tree vlan 2 hello-time/forward-time/max-age** 改变特定 VLAN 中 STP 计时器

BPDU 的偏移指 BPDU 应该收到的时间和实际收到的时间不一致，这时交换机会产生日志消息，如果很多 BPDU 都偏移，那么日志消息也会很多，会消耗 CPU 和内存

如果一个稳定的网络加入一台交换机，该交换机的优先级很小，那么新加入的交换机就会抢掉根交换机的位置，那么在连接新加入的交换机的交换机端口上用 **spanning-tree guard root** 保护原有的根交换机，把这个端口自动阻塞，用 **show spanning-tree inconsistentports** 查看因为根保护而被阻塞的端口

STP 默认的路径代价：10Mbps >> 100； 100Mbps >> 19； 1Gbps >> 4； 10Gbps >> 2

STP 用目标交换机到根交换机的所有经过链路代价之和

端口下用 **spanning-tree cost 20** 改变

STP 的决策：

1，先选出根交换机

2，其他交换机计算到达根交换机路径最短的端口是根端口，如果最短路经不止一条，就用端口编号小的(例：F0/1 > F0/2)

3，每一个网段有且只有一个指定端口，即离根交换机路径最小的端口，如果多个端口路径一样，则用 BID 最小的交换机上的端口

4，其他既不是根端口也不是指定端口的端口阻塞

STP 端口状态：

Disable - 被关闭的端口

Blocking - 初始状态或是被 STP 阻塞状态，该状态停留 Max Age，始终只能收到 BPDU，如果 Max Age 都没有收到 BPDU，就转到 Listening 状态

Listening - 收发 BPDU，以确信在传输数据帧时网络无环路，停留 Forward Delay

Learning - 收发 BPDU，学习 MAC 地址，但不转发数据帧，停留 Forward Delay

Forwarding - 除非不存在环路，并且确定它是到达根交换机的最佳路径，不然永远不会进入该状态

全局用 **spanning-tree vlan 200** 对 VLAN 200 启动生成树，前面加 no 就是关闭

该命名要在所有 VLAN 200 参与的交换机上激活

用 **show spanning-tree (vlan 200)** 查看（默认下，后面不加具体的 VLAN 号，只提供 VLAN1）

逐个虚拟网生成树（PVST）是 cisco 专有的 STP 实现技术，每一个 VLAN 都有一个单独的 STP，太多的 STP 会消耗掉大量的设备可用资源

IEEE 使用普通生成树(CST)技术，所有的 VLAN 都只用一个 STP，用 802.1Q 定义，这样收敛的问题很严重

Cisco 专有 PVST+让 PVST 和 CST 共存

多生成树（MST）是 802.1s 标准，**spanning-tree mode mst** 激活 MST

然后用 **spanning-tree mst configuration** 进入配置模式

用 **name kaka** 指定名字

用 **revision 12** 指定版本号

用 **instance 2 vlan 2-5,9** 关联某个 MST 实例的 VLAN 范围，缺省下所有 VLAN 都映射到 instance 0

用 **show spanning-tree mst configuration** 查看所有实例

用 **show spanning-tree mst 1** 查看具体的实例

一般建议把分布层中心的交换机用作根交换机，并且启用备份根交换机

EtherChannel

把冗余的并行链路捆绑的方法就形成以太网通道（EtherChannel）,能提供高带宽，负载分担和冗余的低收敛时间

要配置以太通道，被捆绑的链路端口必须遵循以下原则：

1，如果是 access 端口，必须在同一个 VLAN

2，如果是干线端口，必须使用同一种封装，相同的 VLAN 范围

3，要有相同的速率和双工模式

PagP

PAgP 端口聚合协议（Port Aggregation Protocol）

给 EtherChannel 增添了新的功能，有利于以太通道的自动建立。但也有些限制：

1. PAgP 不会在动态 VLAN 端口上建立聚合。因为动态 VLAN 可以强迫端口改为另一个 VLAN。
2. PAgP 要求通道中所有的端口同属于一个 VLAN，或者配置为 Trunk 端口。
3. 如果改变了通道中一个端口的速率或者单双工方式，那么通道中所有端口都设为这一速率或单双工方式。

这里默认采用 cisco 专用的端口集合协议（PagP）,有四个选项 on，off，auto（不希望成为通道，但如果必要，会成为通道），desirable（希望成为通道）。另外一个标准协议是 LACP（Link Aggregation Control Protocol），

端口下用 **channel-protocol lacp/pagp** 选择，

如果采用 LACP，则在端口下可用 **lacp port-priority 28** 指定端口优先级，

全局下用 **lacp system-priority 24** 指定交换机优先级

配置之前先要确定上述原则匹配，如不匹配，重置 STP

然后在每一个要捆绑得端口上用 **channel-group 1 mode on/auto/desirable**（auto 和 desirable 不能形成通道）配置二层通道，其中 1 是通道编号

如果要配置三层通道，要给这个捆绑后的逻辑端口（port-chanel）指明一个 IP 地址

用 **port-channel load-balance** 基于负载的方法指明负载分担的条件，用 **show ehterchannel load-balance** 查看

用 **show etherchannel** 查看

用 **show interface f0 etherchannel** 查看该端口的通道状态

PortFast（快速端口）：

用于 access 端口，能绕过监听和学习状态直接进入转发状态

端口下用 **spanning-tree portfast** （cisco 专有）

由于是接入端口，正常情况下是不会接受到 BPDU 的，如果接受到了 BPDU，STP 就要把这个端口转到阻塞状态，端口下用 **spanning-tree portfast bpduguard**，当收到 BPDU，STP 把这个端口关闭掉，更好的保护端口，如果端口下用 **spanning-tree portfast bpdufiltering default**，则当收到 BPDU 时，STP 把该端口变为普通端口，不再是 portfast 端口，用 **show spanning-tree summay totals** 查看

UplinkFast（快速上行链路）：

在有冗余的链路中使用，当启用的那条链路断掉，另一条备份链路绕过监听和学习状态直接进入转发状态

全局下用 **spanning-tree uplinkfast** （cisco 专有）

BackboneFast（快速骨干）：

在一个 STP 实例的整个网路交换机上使用，当一个交换机从被阻塞的端口收到一个劣等 BPDU 时，它查询根交换机后发现网络改变，就把原来那个阻塞端口马上转到监听状态，而不用等待 Max Age 的时间

在全局下用 **spanning-tree backbonefast** （cisco 专有）

RSTP

快速生成树（RSTP）是 802.1w 的标准与 802.1D 相比较：

802.1w 在拓扑改变时直接扩散，不用先传给根交换机，再有根交换机扩散

802.1d 在拓扑改变时先把改变传给根交换机，再由根交换机重新计算生成树，再扩散出去

RSTP 把 802.1d 中的 disable, blocking, listening 三个端口状态定义成同样的 discarding 状态

端口类型除了 802.1d 中的根端口，指定端口和非指定端口外，还定义了预备端口（Alternate Port）：同一网段上不是指定交换机的交换机的端口，备份端口（backup Port）：一个交换机在同一网段上有冗余，那么不是指定端口的那个端口

单向链路：两个相连的交换机不能互相发送数据，只能从一边发给另一边，那么启用单向链路检测协议，把这类端口关闭，全局下用 **udld enable** 激活所有光纤端口，端口下 **udld enable** 激活该端口，端口下用 **udld disable** 关闭光纤端口，端口下用 **no udld enable** 关闭一般端口，特权下用 **udld reset** 复位因为 udld 被关闭的端口，用 **show udld f0** 查看特定端口的 udld 情况

MLS and CEF

多层交换是通过在硬件中处理第三层数据包和重写功能来增强 IP 路由选择性能。

什么叫“流”，流是在一定的时间内，对某个网络源和目的之间，由多个数据包组成的特定的会话。多层交换机识别流中的信息，并将它和去往目的地所用地端口缓存在交换机中，当数据进入交换机时，它们被根据所缓存地流信息转发到正确的端口。

Cisco CEF

Cisco 快速转发（CEF）是一个高级三层交换技术。CEF 交换可以优化带有大规模的、动态数据流的网络（例如：Internet、具有增强的基于 Web 应用的网络或者交互式的业务）的性能和可扩展性。

CEF 的优点：

1. 改善网络性能－－CEF 和典型的快速交换路由相比较，CEF 可以使用较少的内存容量来实现数据包的转发。这样可以使得更多的处理器资源用于第三层的服务，比如服务质量（QoS）和加密功能等。

2. 提高网络的可扩展性－－当启用分散式 CEF(dCEF)模式时,CEF 的每个线路卡上维护着一个与转发信息库(FIB)和邻接表相同的拷贝，它能独自提供完全的交换能力。

3. 提高网络的收缩性－－在大规模的动态网络中，CEF 能提供了一种具有更好的可靠性和稳定性的交换。在动态网络中，路由的改变会导致快速交换高速缓存条目频繁地失效。这些变化可能导制：数据要通过路由表进行过程交换，而不是通过路由高速缓存进行快速交换。CEF 的转发信息库（FIB）中包含所有路由表中存在的路由，换句话说：FIB 维护着一个 IP 路由表中包含的转发信息的镜像，因此 CEF 免去了维护路由高速缓存、采用快速交换和过程交换相互转换的过程。CEF 比典型的高速缓存技术更能有效地交换数据流。

尽管 CEF 是一种高级 IP 交换技术,支持很多通讯媒体,但它并不是支持所有的通讯方式,目前 CEF 支持 ATM/AAL5snap、ATM/AAL5mux、ATM/AAL5nlpid、帧中继、以太网、FDDI、PPP、HDLC 和隧道（tunnel）等。

CEF 操作使用的部件

为了实现 CEF 的交换功能，CEF 将常规路由器中存储在路由高速缓存中的信息转而存储到几种为 CEF 专门设计的数据结构中。为了有效地进行数据包转发，这种数据结构要保证能进行优化的查询。CEF 的 2 种主要部件如下。

1. 转发信息库

CEF 利用转发信息库（FIB）来进行基于 IP 目的地前缀的交换决策。FIB 从概念上讲类似于路由表或信息库，它维护着一个 IP 路由表中包含的转发信息的镜像。当网络中路由或拓扑结构发生了变化时，IP 路由表就被更新，而这些变化也将反映在 FIB 中。FIB 基于 IP 路由表中的信息，维护着下一网络段的地址信息。

因为在 FIB 条目和路由表条目之间有一一对应的关系，所以 FIB 中包含了所有已知的路由，这样就不用维护路由高速缓存了，而先前的交换路径（比如快速交换和最优交换）都要维护路由高速缓存。

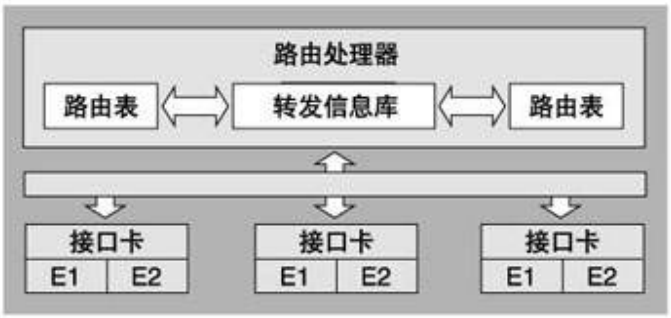
2. 邻接表(Adjacency Table)

如果网络中的网络节点只通过单独一个网络段就可以穿越链路层而彼此到达对方，那么它们是邻接的。除了 FIB 外，CEF 还利用邻接表来提供第二层的寻址信息。邻接表为所有 FIB 条目维护第二层的下一网段地址。当路由器发现存在邻接时就增加在邻接表中，每次生成一个邻接条目，CEF 会为那个邻接节点预先计算一个链路层头标信息，并把这个头标信息存储在邻接表中。当决定路由时，它就会指向下一网络段以及相应的邻接条目，随后即在对数据包进行 CEF 交换时用它来进行封装。

CEF 操作模式

1．集中 CEF 模式

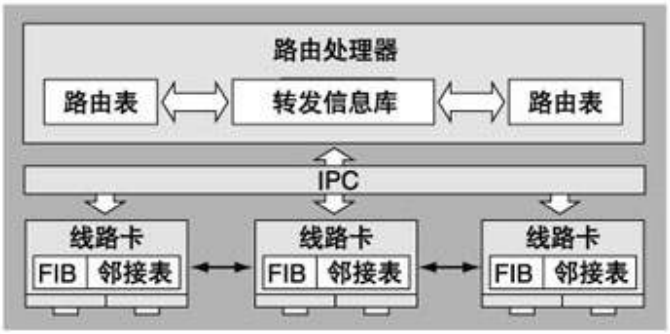
当启用集中 CEF 模式时，CEF 的 FIB 和邻接表驻留在路由处理器中，路由处理器来执行快速转发，见图 1。对于 CEF 交换



来说，当线路卡不可用时，或者需要使用的功能与分散 CEF 交换不兼容时，就可以使用 CEF 模式。

2．分散 CEF 模式

当启用分散式 CEF(dCEF)时，线路卡（例如 VIP 线路卡或者 GSR 线路卡）维护着一个与 FIB 和邻接表相同的拷贝。线路卡在端口适配器之间执行快速转发，这样，在进行交换操作时就无须 RSP 的参与了。dCEF 使用一个“内部过程通信”（Inter Process Communication，IPC）机制，在路由处理器和线路卡上，保证 FIB 和邻接表的同步，如图 2 所示。



CEF 的应用

CEF 在 Cisco 路由器中改善了路由器的性能，优化了路由交换，但它的使用却并不复杂，用户只需要配置启用或禁止 CEF/dCEF 即可实现 Cisco 快速转发。当然，为了更好地使用 CEF，需要对它做进一步的配置，如负载均衡功能等。

如果用户的 Cisco 路由器中有接口处理器支持 CEF 时，就可以启用 CEF。为了启用或禁止 CEF，在全局配置模式下，利用 **ip cef**

当用户想让其线路卡执行快速转发时，则启用 dCEF，这样，路由处理器就可以处理路由协议。为了启用或禁止 dCEF 操作，在全局配置模式下 **ip cef distributed**

当用户按全局模式启用 CEF 或 dCEF 时，所有支持 CEF 的接口都被默认地启用了。

在特定的接口上启用或禁止 CEF/ dCEF

在某个接口配置了一项功能，而 CEF 或 dCEF 并不支持该功能。这时用户就可能需要在这个特定的接口上禁止 CEF 或 dCEF。例如，策略路由和 CEF 就不能一起使用。用户可能想让一个接口支持策略路由，而让其他的接口支持 CEF。在这种情况下，可以按全局模式启用 CEF，而在那个打算配置策略路由的接口上禁用 CEF。这样，除了那一个接口外，在其他所有接口上都启用快速转发。为了在某个接口上禁止 CEF 或 dCEF，可以在接口配置模式下，使用 **no ip route-cache cef**

为 CEF 配置负载均衡功能

负载均衡要依据源头数据包和目的地数据包信息的组合来进行。为了把数据传送到一个目的地，Cisco 可以把数据分配到多条路径中，从而优化资源的使用。用户可以以目的地为单位，也可以以数据包为单位，来配置负载均衡。负载均衡决策机制要在数据出发的接口上做出，它分为以下 2 种方式。

(1) 按目的地配置负载均衡

配置按目的地进行负载均衡功能后，路由器将使用多条路径来均衡负载，对于某一源头/目的地主机，数据包转发采用同一路径，即使有多个路径可用，也将这样处理，对于到达不同目的地的数据包则可以采用不同的路径。当启用 CEF 时，按目的地配置负载均衡的功能默认被启用。大多数情况下，都采用这种负载均衡方法。按目的地配置负载均衡时，要依赖于数据流的统计分布信息，所以随着源头/目的地对个数的增加，负载的平分会变得更为有效。你可以采用按目的地负载均衡的办法，来保证针对某个给定的源头/目的地主机对的数据包依一定的次序到达。

由于启用 CEF 后，也就默认启用了按目的地进行负载均衡功能，为了利用按目的地进行负载均衡的功能，用户就不需要再执行任何其他操作了。

为了禁止按目的地进行负载均衡的功能，在接口配置模式下，利用 **no ip load-sharing per-destination**

(2) 按数据包配置负载均衡

通过按数据包进行负载均衡，使得路由器可以在路径上连续发送数据包，而不用考虑具体的主机或用户情况。这种负载均衡机制采用轮转的办法确定每个数据包采用哪条路径到达目的地。这种机制可以保证在多个连接上进行负载均衡，有助于保证任何单个源头/目的地对的路径都不会变得负担过重。如果有大量的、通过并行链路的数据是针对某单个源头/目的地主机对，那么如果按目的地进行负载均衡，将会使那个链路负担过重，而其他链路上的数据流却很少。启用按数据包进行负载均衡，用户就可以利用不同的路径到达同一个繁忙的目的地。

利用按数据包进行负载均衡使得路径的使用情况变得合理。但是这种机制的直接后果就是对于一对给定的源头/目的地主机对的数据包可能会采用不同的路径到达，使得在目的地一端要对数据包重新排序，正是因为这个原因，这种类型的负载均衡对有些类型的数据流可能就不适应了。比如通过 IP 进行语音传送，由于这种类型的传送要求数据包按照顺序依次到达目的地。

为了启用按数据包进行负载均衡功能，在接口配置模式下，利用 **ip load-sharing(per-packet)**

当然，针对某个特定目的地启用按数据包进行负载均衡，则在所有可以向该目的地转发数据包的接口上，都必须启用按数据包进行负载均衡的功能。

为 CEF 配置网络记账功能

用户可能需要收集统计信息，以便更好地理解和使用网络中的 CEF 模式的功能。例如可能想收集这样的信息：交换到某个目的地的数据包的个数和字节数，或者通过某个目的地交换的数据包的个数。为了给 CEF 收集网络记账信息，可以在全局配置模式中，利用下面的命令进行。

(1) 开始收集被快速转发到某个目的地的数据包个数和字节数

ip cef accounting per-prefix

(2) 开始收集通过某个目的地被快速转发的数据包的个数

ip cef accounting non-recursive

当用户为 CEF 启用网络记账功能后，就在相应的路由处理器中收集记账信息。当用户为 dCEF 启用网络记账功能时，就在线路卡上收集信息。

用户可以查看被收集的记账信息。为此在 EXEC 模式下，使用 show ip cef

记账信息中详细描述了路由器转发数据包的情况，可以由此了解路由器的负载情况，从而决定如何优化路由器的配置，最大限度地发挥路由器的性能。

MLS basic

1.多层交换机可以通过第三层交换机或者是外部路由器来实现。

当使用第三层交换机时的要求：

A. CATALYST5000、6000 系列交换机，监控引擎的软件版本在 4.1 以上

B. CISCO IOS 版本的 11.3 WA4（4）或更新。

C. 监控引擎 III 或是 IIIF 配置 NFFCII，或是监控引擎 IIG 或是 IIIG。

D. 路由交换特征卡（RSFC）

CATALYST 5000 系列交换机：RSFC ，RSM

RSM：路由交换模块

5000/5500 交换机上与监控引擎接口的线路卡，用于提供 VLAN 间路由

RSFC：路由交换功能卡

5000/5500 监控引擎 IIG 和 IIIG 的附属卡，用于提供 VLAN 间路由和 MLS

当使用一个外部路由器和 CATALYST 交换机组合来实现 MLS。它的需求是：

A. CATALYST5000、6000 系列交换机，监控引擎的软件版本在 4.1 以上

B. CISCO IOS 版本的 11.3 WA4（4）或更新。

C. 监控引擎 III 或是 IIIF 配置 NFFCII，或是监控引擎 IIG 或是 IIIG。

D. CISCO 高端路由器，如 CISCO7500,7200,4500,4700 或是 8000 系列

MLS 的组件：

MLS-SE：它是负责转移和重写数据包功能的交换实体。MLS-SE 是位于 CATALYST 交换机监控引擎 III 上的一个 NETFLOW 特征卡。

MLS-RP：它是外接的、安装有支持多层交换软件的路由器上的一个路由模块。

MLSP：它是用来建立和管理 MLS-SE 和 MLS-RP 之间会话的一个协议。MLSP 是 RSM 或是路由器通告路由变化和 VLANS 或是参与 MLS 的接口的 MAC 地址的方法。

MLS 工作原理

MLS-RP 通告：

当 MLS-RP 在园区网被激活后，它会每隔 15 秒钟发出一个多点广播消息，这个消息是用 CGMP 消息，但是由于使用了不同的协议，因此交换机可以将这些消息和其他的多点广播消息区别开来。运用 CGMP 的原因是为了使路由器和交换机可以相互操作。

在这些 HELLO 消息当中有如下的信息：

A. MLS-RP 用来在参与 MLS 的接口的 MAC 地址：交换机运用这个 MAC 地址来决定源数据包是否是发送往第三层设备的。同时让交换机知道路由器，因为多层交换引擎通过 MLS-RP 的 MAC 地址来分配 XTAG 来区分不同的 MLS-RP。

B. 访问控制列表

C. 路由的增加和删除

MLSP HELLO 的消息：

因为 MLS-RP 会发送 CGMP 消息给所有的交换机，这个 HELLO 消息只有具有第三层交换功能的才能处理，第二层交换交换机将将这个消息传向下游。当交换机收到 HELLO 消息时，它将提出该所含的所有的 MAC 地址以及接口信息存到 CAM 中

分配 XTAG：

如果交换机连接了多个 MLS-RP，那么交换机将用 MLS-RP 的 MAC 地址配备 XTAG 的方式来区分它们。当 MLS-RP 失效或是退出网络而要从第三层表格中删除的时候很有用.

建立 MLS 缓存条目：

多层交换基于单独的流，MLS-SE 为 MLS 流维护一个缓存条目并为每个数据流存储统计信息，流中的所有数据包都会于缓存中的缓存条目进行比较。

建立缓存条目是一个较为复杂的过程，当数据流经过多层交换机时，交换机会检查包中目的地 MAC 地址，如果该地址是第三层路由设备地地址，那么它就会比较 MLS 缓存确定时候存在为该流建立的 MLS 缓存条目，如果该数据包是该流当中的第一数据包，那么 MLS 缓存肯定就不会存在缓存条目，这时，交换机还是没有关于这个流的三层交换所需消息，因此它会将这个数据包转发到第三层路由器上去。这时这个数据流会在多层交换中生成一个候选的条目。

当路由器接受到这个数据包时，它会将读取地址，并在路由表中看是不是有存在转发的路由表，如果有的话，它就会使用自己的 MAC 地址作为该帧的源地址，并将该帧发往 MLS-SE。这时交换机根据 CAM 表，交换机知道了它将从什么地方转发该数据，这时他也会识别出源地址是 RP 的地址，这个识别就会触发交换机检查 MLS 缓存中是否存在关于该路由器的条目。交换机比较 MLS 候选条目和返回数据帧的 XTAG，如果相同的话，就说明，该数据流就是来自同一路由处理器。

当建立了完整的 MLS 条目的话，多层交换机在处理一个流下续的数据流，将要进行第二层交换。

当交换机接受到后续流后，交换机将要重写这个数据帧的帧头：将源 MAC 地址换成路由器的 MAC 地址，但是在第三层的 IP 地址是没有变化的，TTL 会减去 1，这样就可以看作是被路由器处理过的数据帧一样。

MLS 工作原理：

1. MLS-SE 接收到初始化的 MLSP Hello 包时，MLS-SE 会针对 MLS-RP 生成本地唯一的 XTAG，并记录每个 Vlan 关联

的 MLS-RP 的 MAC 地址。MLS-SE 为通过 MLSP 从同一个 MLS-RP 学到的所有 MAC 地址附加相同的 XTAG。

CAM		
XTAG	MAC	VLAN
1	00-00-0C-11-11-11	1
1	00-00-0C-22-22-22	2

2. 主机 A 与主机 B 位于不同的 VLAN 中。主机 A 向主机 B 发起数据传送。当 A 发送首个数据包给 MLS-SE 时，MLS-SE 将此数据包看作 3 层交换的“候选数据包”，因为 MLS-SE 通过 MLSP 学到了 MLS-RP 的各个 VLAN 和 MAC 地址信息，所以它能正确转发数据包，并在 MLS 缓存中创建关于此 3 层数据流的不完整的 MLS 项。MLS-RP 接收到此数据包，将目的 MAC 地址改成主机 B 的 MAC 地址，并以自己的主机 B 所属的 VLAN 相关联的 MAC 地址作为源 MAC 转发数据包。

3. MLS-RP 将数据包路由到主机 B。当 MLS-SE 接收到此数据包时，MLS-SE 发现源 MAC 地址属于 MLS-RP，XTAG 与候选数据包匹配。MLS-SE 将此数据包作为“使能数据包”，并将 MLS 缓存中的 MLS 项补充完整。

4. MLS 项被补充完整后，相同流内的所有从源主机到目的主机的 3 层数据包都直接由交换机进行第 3/4 层交换，不经过路由器。IP MLS 是单向的：对于从主机 B 到主机 A 的通信，需要创建另一条 3 层交换路径。

使 MLS 失效的命令

有些 IP 命令会要求路由器对每个数据包都要进行处理时

任何要求每个数据包都要进行处理的命令都会阻碍多层交换功能。

A. **no ip routing**：清除所有的 MLS 缓存条目，并在 MLS-RP 关闭 MLS 功能。
B. **ip security**：关闭在接口上的 MLS 功能
C. **ip tcp compression-connections** 在接口上关闭 MLS 功能
D. **clear ip-route**:清除所有为这个 MLS-RP 执行第三层的交换机中的 MLS 缓存条目

以下几点与 mls 不兼容：

IP accounting, encryption, compression, IP security,

Network Address Translation (NAT),Committed Access Rate (CAR).

配置多层交换路由处理器

1. 在路由处理器上全局性的为某个路由选择协议启用 MLS 功能

router(config)#mls rp ip

2. 在接口上分配一个 MLS VTP 域：

我们决定哪些路由处理器将作为 MLS 接口之后，我们必须将它们加入到于交换机处于相同的 VTP 域中。交换机和 MLS 接口必须处在相同的 VTP 域中。

router(config)#interface vlan vlan-num

router(config-if)#mls rp vtp-domain domain-name

使用 show mls rp vtp-domain 查看相关信息

3. 在接口上启用 MLS：

将一个 MLS 接口放入 VTP 域并不能激活接口上的 MLS 功能。MLS 功能必须明确的在接口上启用。因为 VLAN 与子网

是一一对应关系，参与第三层交换的每个接口都必须为指定启用了多层交换的功能。

router(config-if)#mls rp ip

4. 创建一个空域

这种情况出现在当路由器所属的 VTP 域与交换机所属的 VTP 不同时，因为我们前面已经知道路由器接口和交换机所属的 VTP 相同的时候，才能正常的进行多层交换。

有目的地将两台设备放在不同的域中

当配置交换机或是路由器处理器时，拼错了 VTP 名字

当把接口放在 VTP 域之前输入了 MLS 接口命令。

5. 指定一个 MLS 管理接口

在 RSM 或是路由器配置参与多层交换时，该设备将通过 MLSP 发送 HELLO 消息、通告路由变化，并且公布该设备上参与的 MLS 的那些接口的 MAC 地址。在 MLS-RP 上的一个接口必须指定为管理接口，通过该接口进行 MLSP 数据包进行转发和接受

router(config)#interface vlan 1

router(config-if)#mls rp management-interface

6. 为外部路由器上的接口分配一个 VLAN ID

多层交换是 VLAN 间的路由选择。多层交换机根据哪个端口是哪个 VLAN 配置的来作转发决定。内部路由处理器和 ISL 配置的脸露自然用 VLAN id 来识别接口。外部路由处理器接口知道有关子网的信息，但没有 VLAN 的信息。因此 MLS 要求每台外部路由器接口都有一个分配给它的 Vlan id

router(config)#INTERFACE interface number

router(config-if)#MLS RP VLAN-ID vlan-id-number

7. Show mls rp 来显示 MLS 的配置

Show mls rp interface interface number 来显示某个接口上的 MLS 配置。

MLS-RP 配置如下：

1. 需要启用 MLS：mls rp ip
2. 加入 VTP 域：mls rp vtp-domain www（主接口配置）
3. mls rp vlan-id xx
4. 在子接口启用 MLS：mls rp ip
5. 配置管理接口

1. **router(config)# mls rp ip** //全局启动 MLS

2. 确定哪些路由器接口作为 MLS 接口，并将这些接加入与 MLS-SE 相同的 VTP 域中。

Router(config)# inter f2/0

Router(config-if)# mls rp vtp-domain name //在主接口上配置 VTP 域名，对子接口有效

Router(config-if)# inter f2/0.1

Router(config-subif)# encapsulation dotlq 20 //为子接口配置封装

Router(config-subif)# mls rp ip //启动 Ip MLS

应用流掩码

1.用来决定将数据包中多少信息放入 MLS 缓存中，而不是用来将数据包与 MLS 缓存中现有条目进行比较的。

MLS-SE 支持三种流掩码模式：

目的 IP（没有访问列表，缺省）：最不具体的流掩码(The least specific flow mask mode)

源-目的 IP（标准访问列表）

IP 流（扩展访问列表）：最具体的流掩码(The most specific flow mask mode)

对于 MLS-SE 索服务的所有的 MLS-RP 来说，它只支持一种流掩码。当检测到不同的流掩码是，它将使用最具体的流掩码来实现。当 MLS-SE 流掩码改变时，整个 MLS 缓存将重清。

2.输入访问控制列表和 IP 流掩码和使用输出访问控制列表

在启用了 MLS 接口上设置一个输入访问控制列表将重清所有去往这个接口流的 MLS 缓存，然而输入访问控制列表的缺省行为是对所有进入数据包进行检查和路由选择。在 CISCO IOS 11.3 或是更迟的版本是不支持在配置了 MLS 的接口上自动支持输入控制列表。这样的话，就会对每个数据都是要通过路由器。

为了能让多层交换能与输入访问控制列表协同工作，可以使用 **router(config)#mls rp ip input-acl**

配置多层交换机的交换引擎

MLS 在支持三层交换的 CATALYST 的交换机上是缺省使用的，如果 MLS-RP 是内部的 RSM，那么该交换机不必要其他的额外的配置。但是在如下的情况下，要进行配置。

检验配置

要在 MLS-SE 显示多层交换的信息，我们可以在特权的模式下 **switch (enable)#SHOW MLS**

它显示的内容包括：

是否在交换机上启用多层交换功能；(SET MLS ENABLE)

以秒为单位的 MLS 的缓存条目更新时间;(SET MLS AGININGTIME 304)

以秒为单位的 MLS 的快速更新时间和门限值;(SET MLS AGININGTIME FAST 64 7)

流掩码（SET MLS FILOW [destination|destination-source|full]

所交换的数据包的流的总数；

缓存中活跃的 MLS 条目的总和；

是否启用了 NETFLOW 的数据输出，如果有的话，是针对哪个端口和哪台主机；

MLS-RP 的 IP 地址、MAC 地址、XTAG 和支持的 VLAN；

要显示有关具体的 MLS-RP 的信息，可以使用 **SHOW MLS RP ip－address**

命令	使用场合	功能
Show mls rp	Mls－rp	Show which switchs an mls-rp is supporting
Show mls rp *ip－address*	Mls－switch	Show detailed information about specific Mls-rp

要显示 MLS 的缓存条目，可以在特权的模式下执行

switch (enable)#SHOW MLS ENTRY

他也可以针对某个具体的条件来显示 MLS 条目中的

show mls entry destination ip-address（对某个具体的目的 IP 地址）

show mls entry source ip-address（对某个具体的源 IP 地址）

show mls entry rp ip-addrss（对某个具体的 MLSRP id）

show mls entry flow protocol source-port destination-port（对某个具体的 IP 流）

Configuring MLS on a Router

	Command	Purpose
Step 1	mls rp ip	Globally enables MLSP. MLSP is the protocol that runs between the MLS-SE and the MLS-RP.
Step 2	interface type number	Selects a router interface.
Step 3	mls rp vtp-domain [domain-name]	Selects the router interface to be Layer 3 switched and then adds that interface to the same VLAN Trunking Protocol (VTP) domain as the switch. This interface is referred to as the MLS interface. This command is required only if the Catalyst switch is in a VTP domain.
Step 4	mls rp vlan-id [vlan-id-num]	Assigns a VLAN ID to the MLS interface. MLS requires that each interface has a VLAN ID. This step is not required for RSM VLAN interfaces or ISL-encapsulated interfaces.
Step 5	mls rp ip	Enables each MLS interface.
Step 6	mls rp management-interface	Selects one MLS interface as a management interface. MLSP packets are sent and received through this interface. This can be any MLS interface connected to the switch.
	Repeat steps 2 through 5 for each interface that will support MLS.	

相关 MLS 的命令：

显示特定接口的 IP MLS 信息：**show mls rp interface vlan 10**

显示 MLS-RP 上关于特定 VTP 域的信息：**show mls rp vtp-domain domain-name**

显示 Catalyst 6000 MSFC 上的 MLS 信息：**show mls status**

显示 MLS-SE 上的 MLS 信息：**show mls**

显示 MLS-SE 上关于某个 MLS-RP 的 MLS 信息：**show mls rp 192.168.1.100**

显示 MLS-SE 上所有的 MLS 信息：**show mls entry**

基于目的 IP 地址显示 MLS 项：**show mls entry destination 192.168.1.100**

基于源 IP 地址显示 MLS 项：**show mls entry source 192.168.1.100**

基于特定的 MLS-RP 显示 IP MLS 项：**show mls entry rp 192.168.1.100**

基于特定流显示 IP MLS 项：**show mls entry flow tcp 23 3389**

清除特定流的 IP MLS 项：**clear mls entry destination 192.168.1.100 source 192.168.1.200 flow tcp 3389 23**

QoS

Basic Introduction

在传统的 IP 网络中，所有的报文都被无区别的等同对待，每个路由器对所有的报文均采用先入先(FIFO) 的策略进行处理. 它尽最大的努力(Best-Effort)将报文送到目的地，但对报文传送的可靠性，传送延迟等性能不提供任何保证.

网络发展日新月异，随着 IP 网络上新应用的不断出现，对 IP 网络的服务质量也提出了新的要求. 例如 VoIP 等实时业务就对

报文的传输延迟提出了较高要求. 如果报文传送延时太长, 将是用户所不能接受的(相对而言 E-Mail 和 FTP 业务对时间延迟并不敏感). 为了支持具有不同服务需求的语音, 视频以及数据等业务, 要求网络能够区分出不同的通信, 进而为之提供相应的服务. 传统 IP 网络的尽力服务不可能识别和区分出网络中的各种通信类别, 而具备通信类别的区分能力正是为不同的通信提供不同服务的前提. 所以说传统网络的尽力服务模式已不能满足应用的需要 QoS(Quality of Service ,服务质量技术)的出现便致力于解决这个问题

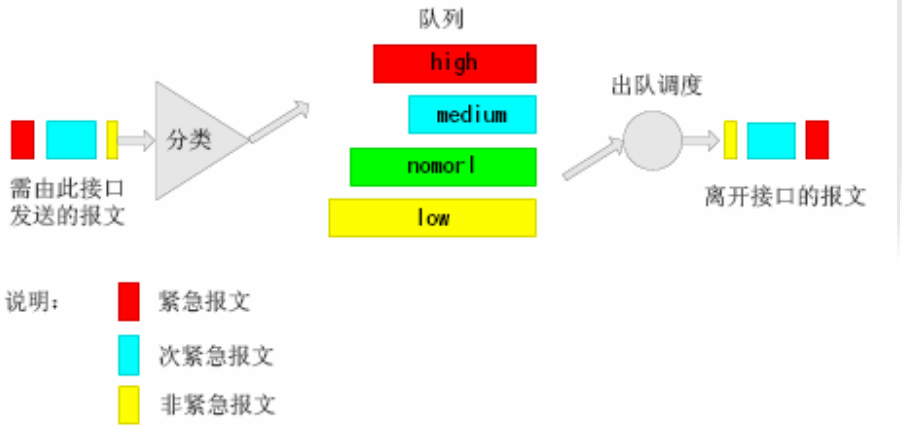
QoS 旨在针对各种应用的不同需求为其提供不同的服务质量例如提供专用带宽减少报文丢失率降低报文传送时延及时延抖动等为实现上述目的 QoS 提供了下述功能

a) 报文分类和着色
b) 网络拥塞管理
c) 网络拥塞避免
d) 流量监管和流量整形
e) QoS 信令协议

下面用简单的例子对比了在网络发生拥塞时报文在无 QoS 保证和有 QoS 保证网络中的不同处理过程

下图所示为发生拥塞时网络设备的一个接口在不支持 QoS 的情况下报文的发送情况

所有要从该接口输出的报文，按照到达的先后顺序进入接口的 FIFO 队列尾部。而接口在发送报文时从 FIFO 先入先出队列的头部开始依次发送报文。所有的报文在发送过程中没有任何区别也不对报文传送的质量提供任何保证



上图为一个 PQ 对列进行 Qos 处理的过程。

在报文到达接口后，首先对报文进行分类，然后按照报文所属的类别让报文进入所属队列的尾部。在报文发送时，按照优先级总是在所有优先级较高的队列中的报文，发送完毕后再发送低优先级队列中的报文。这样在每次发送报文时总是将优先级高的报文先发出去。保证了属于较高优先级队列的报文有较低的时延报文的丢失率和时延。这两个性能指标在网络拥塞时也可以有一定的保障。

QoS可以控制各种网络应用和满足多种网络应用要求如

控制资源。如可以限制骨干网上FTP使用的带宽，也可以给数据库访问以较高优先级

可裁剪的服务，对于ISP 其用户可能传送语音视频或其他实时业务QoS使ISP能区分这些不同的报文并提供不同服务

多种需求并存，可以为时间敏感的多媒体业务提供带宽和低时延保证，而其他业务在使用网络时也不会影响这些时间敏感的

业务

在一个网络中需要以下的三个部分来完成端到端的QoS

1. 各网络元件路由器以太网交换机等支持QoS 提供队列调度流量整形等功能
2. 信令技术来协调端到端之间的网络元件，为报文提供QoS
3. QoS技术控制和管理端到端之间的报文在一个网络上的发送

而每个网络元件提供如下功能

1. 报文分类，对不同类别的报文提供不同的处理
2. 队列管理和调度来满足不同应用要求的不同服务质量
3. 流量监管和流量整形限制和调整报文输出的速度
4. 接入控制来确定是否允许用户信息流使用网络资源

QoS service mode

网络应用是端到端的通讯结构，比如两个不同网络的主机进行通讯，中间可能跨越各种 router 和核心 switch，那么想整体的实现所谓的 QOS，就必须全局考虑，QOS 的服务模型的概念就是采用通过什么模式全局实现服务质量保证，一共分成三种。

Best-Effort service	尽力而为服务模型
Integrated service	综合服务模型 简称 Intserv
Differentiated service	区分服务模型 简称 Diffserv

Best-Effort

Best-Effort 是一个单一的服务模型，也是最简单的服务模型，应用程序可以在任何时候，发出任意数量的报文，而且不需要事先获得批准，也不需要通知网络，对 Best-Effort 服务，网络尽最大的可能性来发送报文，但对时延、可靠性等性能不提供任何保证 Best-Effort 服务是现在 Internet 的缺省服务模型，它适用于绝大多数网络应用，如 FTP、 E-Mail 等。其实 best-effort 并非是什么 QOS，就是互联网的简单数据传输方式而已，有什么传什么，阻塞也就阻塞了，丢且也就丢弃了。

Intserv

Intserv：集成服务模型，它可以满足多种 QoS 需求。这种服务模型在发送报文前，需要向网络申请特定的服务。应用程序首先通知网络它自己的流量参数和需要的特定服务质量

请求：包括带宽、时延等。应用程序一般在收到网络的确认信息，即确认网络已经为这个应用程序的报文预留了资源后，才开始发送报文，同时应用程序发出的报文应该控制在流量参数描述的范围以内。

网络在收到应用程序的资源请求后，执行资源分配检查 Admission control 即基于应用程序的资源申请和网络现有的资源情况，判断是否为应用程序分配资源，一旦网络确认为应用程序的报文分配了资源，则只要应用程序的报文控制在流量参数描述的范围内，网络将承诺满足应用程序的 QoS 需求。而网络将为每个流 flow 由两端的 IP 地址、端口号、协议号确定、维护一个状态，并基于这个状态执行报文的分类、流量监管、policing、排队及其调度来实现对应用程序的承诺。

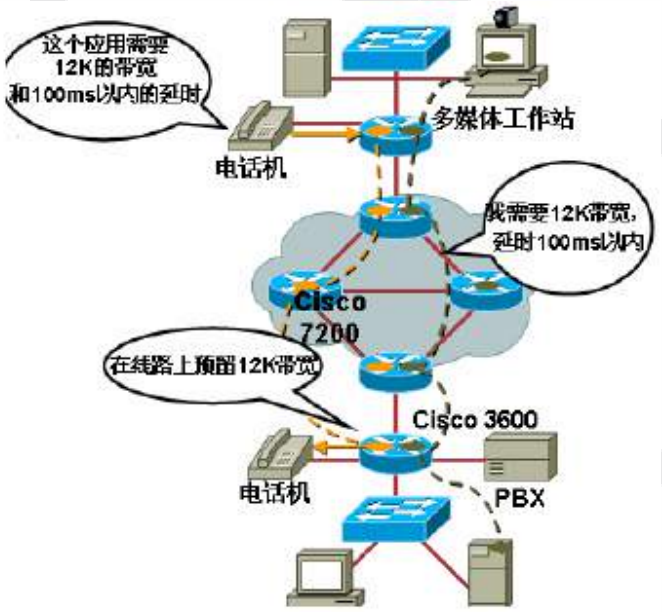
在 IntServ 服务模型中，负责传送 QoS 请求的信令是 RSVP（Resource Reservation Protocol）资源预留协议，它通知路由器应用程序的 QoS 需求。RSVP 是在应用程序开始发送报文之前来为该应用申请网络资源的。 Intserv 实际上是一种对服务的预定机制，通过申请来获取相应得服务，这里面主要依靠的就是 RSVP——资源预留协议。

RSVP

RSVP 是第一个标准 QoS 信令协议，它用来动态地建立端到端的 QoS，它允许应用程序动态地申请网络带宽等。RSVP 协议不是一个路由协议，相反，它按照路由协议规定的报文流的路径为报文申请预留资源，在路由发生变化后，它会按照新路由进行调整，并在新的路径上申请预留资源。RSVP 只是在网络节点之间传递 QoS 请求，它本身不完成这些 QoS 的要求实现，而是通过其他技术来完成这些要求的实现。（**RSVP** 只是一种用来预定的协议）

RSVP 的处理是接收方发出资源请求，按照报文发送的反向路径发送资源请求，所以它可以满足非常大的多播组，多播组的成员也可以动态变化，RSVP 协议是针对多播设计的 单播可以看作是多播的一个特例。

由于 RSVP 在 Internet 上还没有得到广泛的推广，在主机不支持 RSVP 的情况下，我们可以通过配置 RSVP 代理，即代替不支持 RSVP 的主机发送 RSVP 报文来获得这种服务，对报文流路径上不支持 RSVP 的路由器，它只需要简单的转发 RSVP 报文 所以对 RSVP 协议不会有太大影响，但这些节点不会对报文提供所要求的 QoS 。（这是 **RSVP** 的一个缺点）



RSVP 信令在网络节点之间传送资源请求，而网络节点在收到这些请求后，需要为这些请求分配资源，这就是资源预留。网络节点比较资源请求和网络现有的资源，确定是否接受请求，在资源不够的情况下，这个请求可以被拒绝，可以对每个资源请求设置不同的优先级。这样，优先级较高的资源请求可以在网络资源不够的情况下，抢占较低优先级的预留资源，来优先满足高优先级的资源请求。

资源预留判断是否接受资源请求，并承诺对接受了的资源请求提供请求的服务，但资源预留本身不实现承诺的服务，需要通过队列等其他技术来实现。

Intserv 有它的好处，但是也有严重缺点，首先就是 RSVP 协议数据太多，而且不断刷新，并且这种给单一数据流的

路径进行带宽预留的解决思路在浩瀚的 Internet 上实现简直是不可能的，而且 RSVP 的部署，厂商之间设备的互联，业务管理方面 等存在着种种问题，所以这么模型在 1994 年推出之后就没有获得任何规模的商业应用。

DiffServ

DiffServ 是一个多服务模型，它可以满足不同的 QoS 需求，与 IntServ 不同，它不需要使用 RSVP 即应用程序在发出报文前，不需要通知路由器为其预留资源，对 DiffServ 服务模型，网络不需要为每个流维护状态 ，它根据每个报文指定的 QoS 来提供特定的服务 可以用不同的方法来指定报文的 QoS,如 IP 报文的优先级位 IP Precedence），报文的源地址和

目的地址等，网络通过这些信息来进行报文的分类、流量整形、流量监管和队列调度。

DiffServ 一般用来为一些重要的应用提供端到端的 QoS 它通过下列技术来实现

CAR：它根据报文的 ToS 或 CoS 值（对于 IP 报文是指 IP 优先级或者 DSCP 等等）IP 报文的五元组（指源地址 目的地址协议端口号）等信息进行报文分类，完成报文的标记和流量监管。

队列技术：WRED、PQ、CQ、WFQ、CBWFQ 等队列技术对拥塞的报文进行缓存和调度，实现拥塞管理。

通常在配置DiffServ时，边界路由器通过报文的源地址和目的地址等对报文进行分类，对不同的报文设置不同的CoS值，而其他路由器只需要用CoS值来进行报文的分类

在MPLS上应用DiffServ用以下两种方法来解决：

在以太网等网络中,MPLS报文在二层链路层和三层网络层之间有一个薄层(shim). 我们扩展薄层中未用的字段---EXP(包含三个位).由这几个位来决定报文的队列调度及丢弃的优先级.

在ATM FR等网络中,其MPLS报文没有薄层(shim), 可针对FEC ForwardingEquivalance Class 转发等价类和QoS请求的组合来分配标签,而不同于以前仅针对FEC分配标签.这样在收到一个MPLS报文后,根据收到报文的标签就可以确定发出报文的标签及报文所要求的服务.

Intserv与Diffserv之间的互通

一般来讲,在提供IP网络的QoS时,为了实现规模适应性,在IP骨干网往往需要采用Diffserv体系结构,在IP边缘网可以有两种选择:采用Diffserv体系结构或采用Intserv体系结构.目前在IP边缘网络采用哪一种QoS体系结构还没有定论,也许这两种会同时并存于IP边缘网中.在IP边缘网采用Diffserv体系结构的情况下,IP骨干网与IP边缘网之间的互通没有问题.在IP边缘网采用Intserv体系结构的情况下,需要解决Intserv与Diffserv之间的互通问题,包括RSVP在Diffserv域的处理方式,Intserv支持的业务与Diffserv支持的PHB(Per-Hop Behavior, 单中继段行为)之间的映射.

RSVP在Diffserv域的处理可以有多种可选择的方式.例如

1. 一种方式为RSVP对Diffserv域透明,RSVP在Intserv域边界路由器终结,Diffserv域对Intserv域采用静态资源提供方式:
2. 一种方式为Diffserv域参与RSVP协议处理,Diffserv域对Intserv域采用动态资源提供方式.

前一种互通方式实现相对简单,可能造成Diffserv域资源的浪费.后一种互通方式实现相对复杂,可以优化Diffserv域资源的使用.

除此以外,还需要解决Intserv支持的业务与Diffserv支持的PHB之间的映射问题.映射标准为两者支持的应用是否相同或相近.为了说明这个问题我们首先回顾一下Intserv支持的业务,它支持的业务包括保证服务(Guaranteed Service)负载控制服务(Controlled-Load Service).前者可以为用户应用提供严格的端到端时延及带宽保证,适用于实时应用.后者在网络负荷较重的情况下为用户应用提供与网络轻负荷情况下相近似的性能,不能保证端到端的时延.

Diffserv提供的PHB包括EF (Expedited Forwarding 加速转发)AF(Assured Forwarding 确保转发)

EF用于支持低丢失率,低时延,确保带宽的应用AF可以保证在应用向网络发送的业务流量没有超过约定值的情况

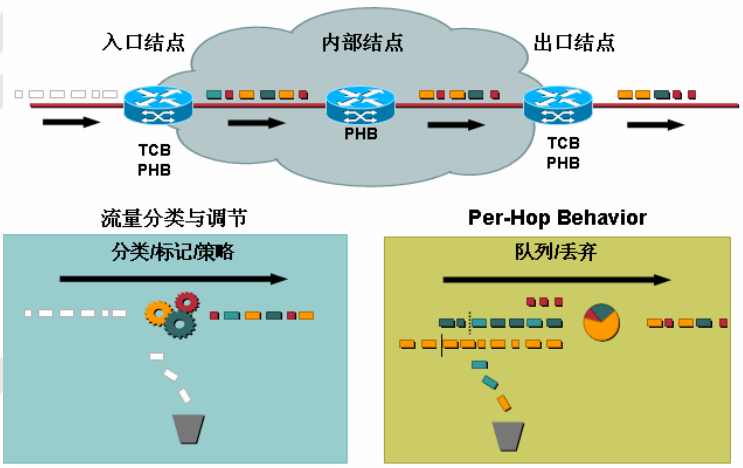
下,该应用的报文丢失概率非常低AF有4类每一类可以设置3个不同的丢弃优先级.从上面的叙述易于获得Diffserv与Intserv 之间的映射关系

将Intserv中的保证服务映射为Diffserv中的EF
将Intserv中的负载控制服务映射为Diffserv中的AF

QoS packet and Frame

Basic introduction

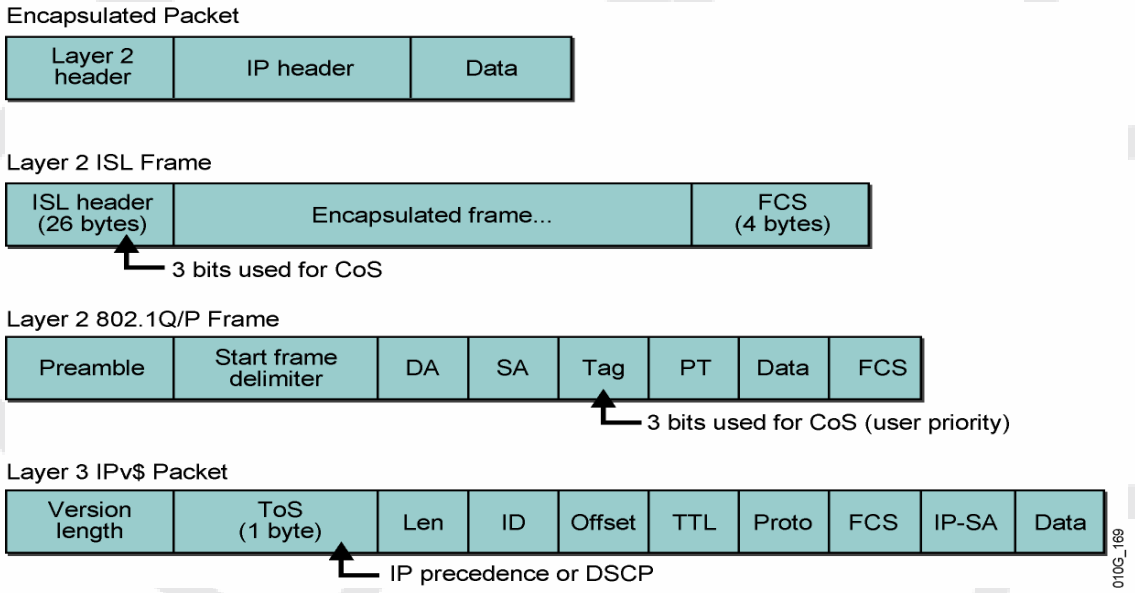
通常在配置DiffServ 时，边界路由器通过报文的源地址和目的地址等对报文进行分类， 对不同的报文设置不同的CoS值，而其他路由器只需要用CoS 值来进行报文的分类。



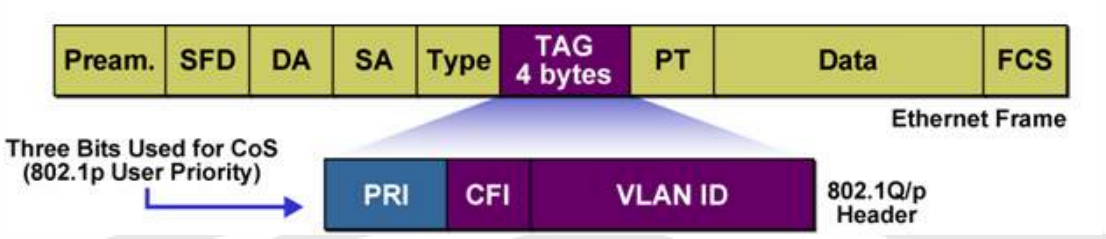
Diffserv 服务模型

DSCP

这种标记在 frame 中和 packet 中本身就存在，frame 中存在 cos 字段，packet 中有 tos 字段，只不过在正常的传输模型中（例如 best-effort)并不使用，而在 Diffserv 中，数据包 的标记作用就体现出来了，并且在 packet 中，Diffserv 提出了一个新的标记，就是 DSCP（Diffesentiated Services Code Point ）区分服务代码点



如上图所示,正常的以太网 frame 中是不存在标记的,但是 ISL 和 dot1q 的 frame 中有 三个 bit 定义服务级别,一共有 6 个服务级别可以使用



需要关注的是，Frame 中的 cos 只使用 0-5，6-7 并不使用。

而针对于数据包，却有两种标识服务类型的方法（如下图所示）分别是 IP precedence（ip 优先级）和 DSCP（区分服务代码点）

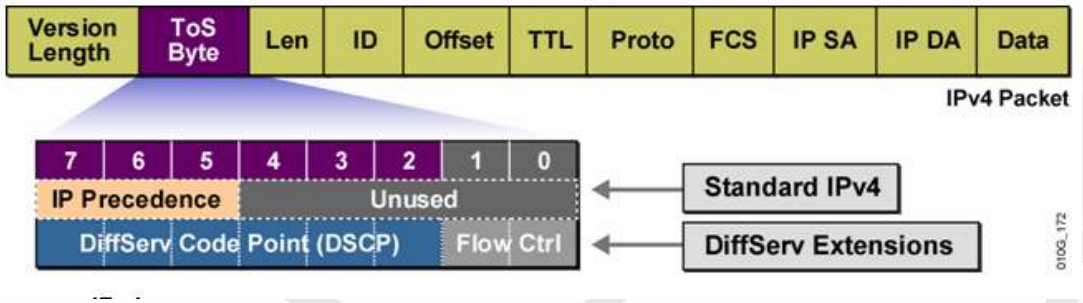


Table 3 IP Precedence Values

Number	Name
0	routine
1	priority
2	immediate
3	flash
4	flash-override
5	critical
6	internet
7	network

TOS 整个字段一共 8 位，图中上面部分是 ip precedence 的标识方法，使用前三位，也 就是 P0、P1、P2 一共三位，共 8 个 bit 位，也就是 8 个优先级，分别是 0-7，其中6和7一般保留，常用的是 0-5,图中下面部分介绍的事 IP 优先级的含义，提供这个表格的意义在于 进行更改数据包优先级等配置时，，我们既可以使用数字，也可以使用名称。
在来看看 DSCP，它使用 tos 中的前 6 个字节，即 DS1-DS5，（如上图所示）定义了 0-63,一共 64 个优先级。 分类是没有范围限制的，也就是说我们可以对数据进行灵活的分类，比如说把某一个源ip 到目的 ip 的流量优先级进行更改 IP 优先级或者 DSCP 的操作。也可以定义去更改某一个 特定的流量（扩展控制列表）的优先级。
通常于网络边界处对报文进行分类时，同时标记 IP 优先级或 DSCP。这样，在网络的 内部就可以简单的使用 IP 优先级或 DSCP 作为分类的标准，而队列技术如 WFQ 等 CBWFQ 就可以使用这个优先级来对报文进行不同的处理。
什么是标识,实际上就是通过更改这些优先级字段将这些数据分出种类来,即便是上面 的图标中有所谓的什么 0-7 优先级，好似 7 要比 0 就会优先级大一些，但是一定要清楚这只 是区分，执行的策略要靠后面的队列机制来解决，实际上之所以这样定义优先级我认为只是 为了制定一个共同遵守的类别优先标准，没有实际的意义，真正的操作是在配置上针对于不 同优先级采用的措施——例如在队列里面使用什么标识的数据包属于什么队列等等。
但是在进行配置时，由于存在二三层优先级的对应关系

DCSP-to-Transmit queue map	DSCP 0-15 Queue 1 DSCP 16-31 Queue 2 DSCP 32-47 Queue 3 DSCP 48-63 Queue 4

上表是 DSCP 与 WRR 中应用的队列匹配。

CoS to DSCP map (DSCP set from CoS values)	CoS 0 = DSCP 0 CoS 1 = DSCP 8 CoS 2 = DSCP 16 CoS 3 = DSCP 24 CoS 4 = DSCP 32 CoS 5 = DSCP 40 CoS 6 = DSCP 48 CoS 7 = DSCP 56
DSCP to CoS map (CoS set from DSCP values)	DSCP 0-7 = CoS 0 DSCP 8-15 = CoS 1 DSCP 16-23 = CoS 2 DSCP 24-31 = CoS 3 DSCP 32-39 = CoS 4 DSCP 40-47 = CoS 5 DSCP 48-55 = CoS 6 DSCP 56-63 = CoS 7

上表是 COS 和 DSCP 彼此匹配时的对应关系。

Marking

Basic introduction

marking就是修改IP优先级或者DSCP，但是由于IP优先级和DSCP都 是占用TOS字段,后者相当于前者的扩展,所以不能同时设置这两种值,如果同时设置了这两 种值,那么只有IP DSCP 的值生效。
标记是后续很多QOS策略应用的基本，使用的是**policy map**。

Configurations

1.定义class map

Class map是一个匹配表，类似于ACL。所有的policy map实质上是对class map进行操作的

nimokaka(config)#class-map [match-all|match-any] {map-name}

参数中match-all 表示匹配所有条件，match-any表示至少符合一个条件

2. class map的匹配

nimokaka(config-cmap)#

match access-group {ACL}	匹配IP ACL （主要就是对应数据包了）
match protocol {protocol}	匹配协议（这个在NBAR—基于网络应用中使用）
match input-interface {interface}	匹配进站接口
match qos-group {Group ID}	匹配组ID（不知道干啥的）
match destination-address {mac mac-address}	匹配目标MAC 地址
match source-address {mac mac-address}	匹配源MAC 地址
match ip {dscp dscp}	匹配IP DSCP 值
match ip {precedence precedence}	匹配IP 优先级
match class-map {map-name}	匹配class map（class map嵌套）
match vlan {vlan-id}	匹配VLAN

3. 设置policy map

nimokaka(config)#policy-map {policy-name}

nimokaka(config-pmap)#class {class-map}

4. 配置优先级和DSCP值

nimokaka(config-pmap-c)#

一些用于标记的动作选项:

set ip {precedence precedence}	设置IP优先级
set ip {dscp dscp}	设置IP DSCP 值
set qos-group {Group ID}	设置组ID
set cos {cos}	设置CoS 值
priority {kbps|percent percent} [Bc]	定义优先级流量的保留的带宽以及突发流量
bandwidth {kbps|percent percent}	定义保留的带宽

police {CIR Bc Be} conform-action {action} exceed-action {action} [violate-action{action}]

使用令牌桶算法进行限速

random-detect 启用WRED

queue-limit {packets}	定义队列中数据包的最大个数
service-policy {policy-map}	使用别的策略进行嵌套,做为match语句匹配的标准
shape {average|peak} {CIR [Bc] [Be]}	定义CIR,Bc以及Be进行整形

5. 将配置挂接到接口上

nimokaka（config-if）service-policy [input|output] policy-name

6. 检查配置

nimokaka#show policy-map [policy-name]

查看接口的policy map 信息:

nimokaka#show policy-map interface [interface]

Case

把来自192.168.10.0/24 的出站telnet 流量的IP优先级设置为5,其他的出站流量的IP 优先级设置为1:

access-list 100 permit tcp 192.168.10.0 0.0.0.255 any eq telnet

class-map match-all telnet

match access-group 100

```
policy-map nimokaka
class telnet
    set ip precedence 5
classclass-default
    set ip precedence 1
interface Serial1
clock rate 100
no shut
ip address 1.1.1.1 255.255.255.252
service-policy output nimokaka
```

PS

Class-map嵌套有两点理由：在创建class map的时候去调用一个已有的class map

1、管理方便,在已有的基础上增加一个修改进行平滑的过度。

2、允许用户在同一个class map里分别使用匹配所有(match-all)和匹配任何(match-any)。

比如4个匹配标准：A、B、C和D。现在想让class map 匹配A，或匹配B，或同时匹配C和D，就可以使用class map的嵌套：创建一个新的class map，定义为匹配所有(match-all)新标 准为匹配E即同时匹配C和D；然后定义另一个匹配任何(match-any)的class map，去匹配A， 或B，或E(即同时匹配C 和D)。

NBAR

Basic introduction

我们来看在做了分类之后的一个常规应用,就是NBAR.基于网络的应用程序识别(NBAR) 可以对使用动态分配TCP/UDP 端口号的应用程序和HTTP 流量等进行分类。在使用NBAR的时 候要先启用CEF特性。而且可以使用数据包描述语言模块(PDLM)从路由器的闪存里加载，用于在不使用新的Cisco IOS 软件，或重启路由器的情况下对新的协议或应用程序进行识别。

Notice

NBAR不能在以下几种逻辑接口上使用：

1、快速以太网信道

2、使用了隧道或加密技术的接口

3、SVI

4、拨号接口

5、多链路PPP(MLP)

Limited

1、不支持多于24个的并发URL，HOST或MINE的匹配类型

2、不支持超过400字节的URL 匹配

3、不支持非IP流量

4、不支持组播或其他非CEF的交换模式

5、不支持被分片的数据包

6、不支持源自或去往运行NBAR 的路由器的IP流

Configurations

1、启用CEF 特性:

```
nimokaka(config)#ip cef
```

2、把流量分类,定义class map:

```
nimokaka(config)#class-map [match-all|match-any] {map-name}
```

3、定义NBAR 要匹配的协议:

```
nimokaka(config-cmap)#match protocol {protocol}
```

4、设置policy map:

```
nimokaka(config)#policy-map {policy-name}
```

5、调用class map:

```
nimokaka(config-pmap)#class {class-map}
```

6、设置策略:

```
nimokaka(config-pmap-c)#
```

7、把策略应用在接口上:

```
nimokaka(config-if)#service-policy {input|output} {policy-map}
```

8、查看相关配置

1、查看流量分类信息: nimokaka#show class-map [map-name]

2、查看policy map: nimokaka#show policy-map [policy-name]

3、查看接口的policy map 信息: nimokaka#show policy-map interface [interface]

4、显示NBAR所使用的PDLM: nimokaka#show ip nbar pdlm

5、显示NBAR使用的协议到端口号的映射信息: nimokaka#show ip nbar port-map

Case

使用NBAR识别BitTorrent程序流量

1加载bittorrent.pdlm 到路由器闪存里

```
nimokaka(config)#ip nbar pdlm flash://bittorrent.pdlm
```

2、定义class map,识别BitTorrent程序流量,并将进站的BitTorrent程序流量丢弃

```
ip cef
!
class-map kaka
match protocol bittorrent
!
policy-map drop-bittorrent
class kaka
drop
!
interface Serial0
ip address 192.168.0.1 255.255.255.0
service-policy input drop-bittorrent
```

用NBAR对进站的HTTP 流量下载进行限速,其中凡是下载的图象格式包括jpg,jpeg 和gif的, 速率限制为100kbps

```
ip cef
!
class-map match-any HTTP
match procotol  http url   "*.jpeg|*.jpg"
match procotol  http url   "*.gif"
!
policy-map nimokaka
class HTTP
police 100000 conform-action transmit exceed-action drop
!
interface Serial0
ip address 10.0.0.1 255.255.255.252service-policy input nimokaka
```

使用NBAR来防止红色代码(Code-Red)和尼姆达(Nimda)蠕虫病毒, 配置如下

```
ip cef
!
class-map match-all DENY-ATTACK
match protocol http url "*.ida*"
match protocol http url "*cmd.exe*"
match protocol http url "*root.exe*"
match protocol http url "*readme.eml*"
!
```

```
policy-map nimokaka class DENY-ATTACK drop
interface Serial0
ip address 10.0.0.1 255.255.255.252
service-policy input nimokaka
```

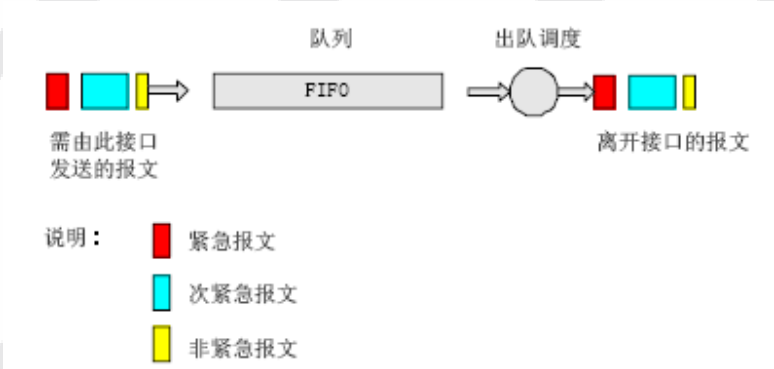
Congestion administration

Basic introduction

在网络设备产生拥塞得时候，常常采用各种队列更改包转发的顺序，对于每个接口只能使用一种队列

First In First Out Queueing

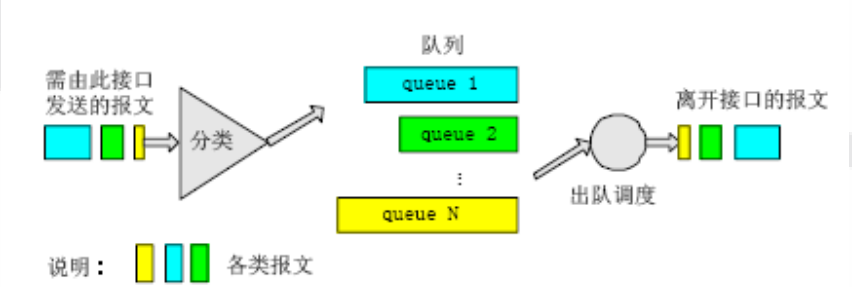
先进先出队列（First In First Out Queueing, FIFO）



如图所示FIFO队列,不对报文进行分类,当报文进入接口的速度大于接口能发送的速度 时,FIFO按报文到达接口的先后顺序让报文进入队列,同时FIFO在队列的出口让报文按进队 的顺序出队先进的报文将先出队后进的报文将后出队。注意：当没有使用其他的队列机制时，除了传输速率小于2.048Mbps 的串行接口以外的所有接口，默认都使用这种队列机制， 比如以太网口等。

Weighted Fair Queueing

加权公平队列（Weighted Fair Queueing, WFQ）



如图所示， WFQ对报文按流进行分类（对于IP网络相同源IP地址、目的IP地址、源端口 号、目的端口号、协议号、IP优先级的报文属于同一个流）每一个流被分配到一个队列，该过程称为散列，采用HASH算法来自动完成。尽量将不同的流分入不同的队列,WFQ的队列数目N可以配置在出队的时候WFQ按流的IP优先级来分配每个流应占有出口的带宽优先级的数 值越小所得的带宽越少,优先级的数值越大所得的带宽越多,这样就保证了相同优先级业务 之间的公平，体现了不同优先级业务之间的权值。例如接口中当前有8个流它们的优先级分 别为0、1、2、3、4、5、6、7、则带宽的总配额将是所有流的优先级 + 1 之和即1 + 2 + 3+ 4 + 5 + 6 + 7 + 8 = 36。

每个流所占带宽比例（为自己的优先级数 + 1）/ （所有（流的优先级 + 1)之和）即 每个流可得的带宽比例分别为1/36、2/36、3/36、4/36、5/36、6/36、7 /36、8/36。

又比如当前共4个流，3个流的优先级为4，1个流的优先级为5，则带宽的总配额将是(4 + 1) * 3 + (5 + 1)

=21 那么3个优先级为4的流获得的带宽比例均为5/21，优先级为5的流获得的带宽比例为6/21。 由此可见WFQ在保证公平的基础上对不同优先级的业务体现权值，而权值依赖于IP报文头中 所携带的IP优先级。注意： WFQ是传输速率低于2.048Mbps的串行接口默认的队列机制。但是WFQ存在一些限制:第一个是WFQ不支持隧道或采用了加密技术的接口,因为这些技术要修改数据包中WFQ 用于分类的信息第二个WFQ提供的带宽控制的精确度不如CBWFQ和CQ等队列 机制。

Configuration

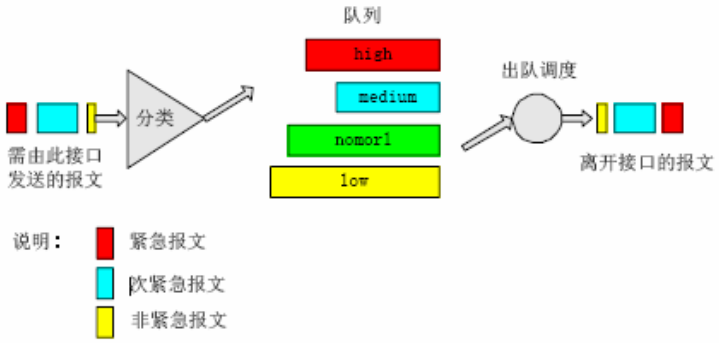
接口下启用WFQ: **nimokaka(config-if)#fair-queue**

显示公平队列的配置状态: **nimokaka#show queueing fair**

显示接口的队列信息: **nimokaka#show queue [interface]**

Priority Queueing

优先级队列（Priority Queueing, PQ）



如 图所示，PQ对报文进行分类，对 于 IP 网络可以根据IP报文的优先级/DSCP等条件进行分类，将所有报文分成最多至4类，分别属于PQ的4个队列中的一个，然后按报文的类别将报文送入相应的队列。PQ的4个队列分别为高优先队列、中优先队列、正常优先队列和低优先队列、它们的优先级依次降低。在报文出队的时候,PQ首先让高优先队列中的报文出队并发送，直到高优先队列中的报文发送完，然后发送中优先队列中的报文，同样直到发送完，然后是正常优先队列和低优先队列， 这样分类使属于较高优先级队列的报文将会得到优先发送,而较低优先级的报文将会在发生拥塞时被较高优先级的报文抢先,使得高优先级业务如 VoIP 的报文能够得到优先处理较低优先级业务如E-Mail 的报文在网络处理完关键业务后的空闲中得到处理既保证了高优先级业务的优先又充分利用了网络资源。

PQ使用的限制和缺点：

第一， 由于PQ是静态配置的,因此它不能适应网络结构的改变。

第二， 由于数据包要经过处理器卡的分类， 因此PQ对数据包转发的速度要比FIFO慢。

第三， PQ 不支持隧道接口

另外PQ一个非常显著的缺点就是如果高优先级的队列没有发送完成， 低优先级的数据将永远不会发送，造成两级分化，使较低优先级的数据转发困难。

Configuration

1、定义优先级列表,可以基于协议或基于进站接口：

基于协议：

nimokaka(config)#priority-list list-number protocol protocol-name {high|medium|normal|low} queue-keyword keyword-value

priority-list号码为1-16，关注一下红色标记部分，其实我们可以添加一些扩充选项 来准确定位流量比如

fragment	(IP packets with non-zero fragment offset) gt/lt
<size>	based on packet size (including L2 frame) list
<acl>	ACL classification

tcp/udp <port> TCP or UDP port number

前面是配置参数，后面是解释，常用的是最后三个参数。

基于进站接口：

nimokaka(config)# priority-list list-number interface interface-type interface-number {high | medium | normal | low}

2、定义默认的优先级队列,未分类的流量默认被分配进该队列,优先级默认为normal:

nimokaka(config)#priority-list {list} default {high|medium|normal|low}

3、定义每个队列中数据包的最大个数,由高到低,默认为20,40,60 和80.可以更改:

nimokaka(config)# priority-list {list} queue-limit {high-limit medium-limit normal-limit low-limit}

4、把优先级列表应用在接口上:

nimokaka(config-if)#priority-group {list}

检查配置

1、显示接口队列信息： **nimokaka#show queue [interface]**

2、显示PQ 列表信息： **nimokaka#show queueing priority**

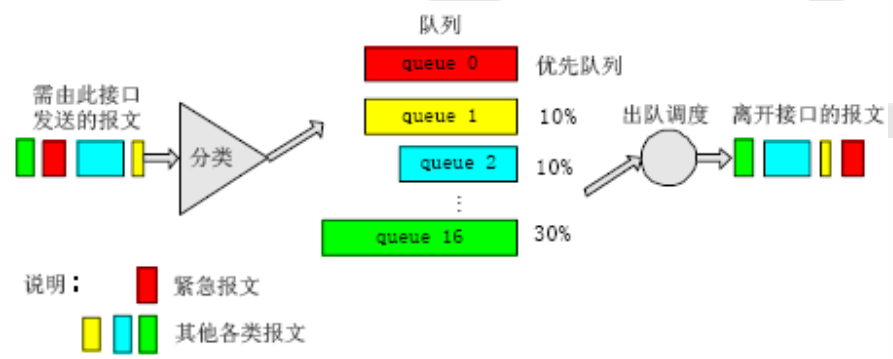
Case

要求sna流量高优先级，www 的ip 流量低优先级别，其他流量中等优先级别，给sna流 量设置队列深度为50消息

```
interface serial 0                          //在接口下应用队列
priority-group 1
priority-list 1 protocol sna high           //sna是高优先级
priority-list 1 protocol ip low tcp 80      //www是低优先级
priority-list 1 protocol ip medium              //其他ip流量是中等优先级
priority-list 1 queue-limit 50 40 60 80     //高优先级队列，队列深度50
```

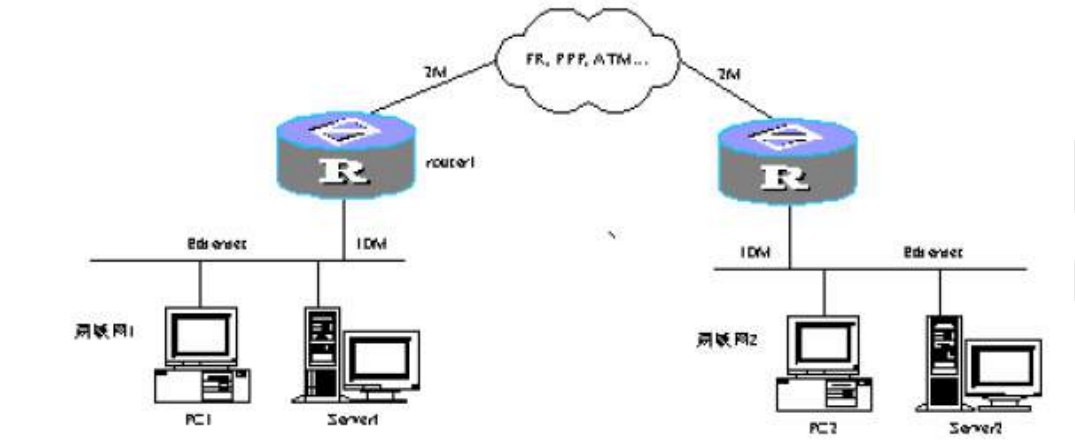
Custom Queueing

自定义队列（Custom Queueing, CQ）



如图所示CQ对报文进行分类，报文分成最多16类，分别属于CQ的16个队列中的一个， 然后按报文的类别将报文送入相应的队列，实际上队列的号码是0-16一共17个，但是0号队列是超级优先队列，路由器总是先把0号队列中的报文发送完然后才处理1到16号队列中的数 据包，所以0号队列一般作为系统队列，通常把实时性要求高的交互式协议和链路层协议报文放到0号队列中。1到16号队列可以按用户的定义分配它们能占用接口带宽的比例，在报文出队的时候，CQ按定义的带宽比例分别从1到16号队列中取一定量的报文在接口上发送出去。 可以将CQ和PQ做个比较，PQ赋予较高优先级的报文绝对的优先权，这样虽然可以保证关键业务的优先，但在较高优先级的报文的速度总

是大于接口的速度时，将会使较低优先级的 报文始终得不到发送的机会。采用CQ则可以避免这种情况的发生，CQ可以把报文分类然后按类别将报文分配到CQ的一个队列中去，而对每个队列又可以规定队列中的报文所占接口带宽的比例，这样就可以让不同业务的报文获得合理的带宽，从而既保证关键业务能获得较多的带宽，又不至于使非关键业务得不到带宽。但是由于CQ轮循调度，各个队列它对高优先级尤其是实时业务的时延保证不如PQ。



在如图所示的网络图中，假设局域网1的服务器向局域网2的服务器发送关键业务的数 据，局域网1的PC向局域网2的PC发送非关键业务的数据，如果对路由器1的串口1配置CQ进行拥塞管理，同时配置服务器间的数据流的进入队列1，队列1中的报文占有60%的带宽，例如 每次出队6000个字节的报文，PC间的数据流进入队列2，队列2中的报文占有20%的带宽，例 如每次出队2000个字节的报文，则CQ对这两种不同业务的报文将做区别，对待报文的发送采用轮询调度的方式，首先让队列1中的报文出队，并发送直到此队列中的报文被发送的字节数不少于6000字节，然后才开始发送队列2中的报文，直到此队列中的报文被发送的字节数 不少于2000字节，然后是其他队列，如果路由器1的串口1的物理带宽是2M，则局域网1的服 务器向局域网2的服务器发送关键业务的数据所能占的带宽将至少为1.2M（2*0.6），局域网1的PC向局域网2的PC发送非关键业务的数据所能占的带宽将至少为0.4M（2*0.2），当路由 器1的串口1中除了上述两个数据流外没有其他数据要发送时这两种数据流将按比例分享接 口的剩余空闲带宽即局域网1的服务器向局域网2的服务器发送关键业务的数据所能占的带 宽将为1.5M［2 * 0.6 / (0.2 + 0.6)］。局域网1的PC向局域网2的PC发送非关键业务的数据 所能占的带宽为0.5M［2 * 0.2 / (0.2 + 0.6)］，当局域网1的服务器向局域网2的服务器不 发送关键业务的数据时并且除了局域网1的PC向局域网2的PC发送非关键业务的数据外没有 其他的数据流则局域网1的PC向局域网2的PC发送非关键业务的数据所能占的带宽将可以为2M。

为了能够为每个队列分配一定的带宽，必须为每个队列定义一定字节数的数据包。自定 义队列中的数据包也按照队列号顺序被转发当队列为空或超出本次队列允许发送的数据包 时，接下来会轮到下一个队列。但是假如定义的字节数为100字节，而某个数据包的大小为1024字节，那么该队列每次将转发的数据包的大小即为1024字节，而不是100字节。假如有3 个队列，每个队列中的数据包大小分别为500 字节，300字节和200字节如果想 让这3个队列平均的占用带宽，为这3个队列定义的字节数分别为200字节，200字节和200字 节，但是实际上生效的带宽占用比为5/3/2，因此如果把队列中数据包的字节数定义的过小 的话，将导致带宽分配的不尽如人意。但是如果把队列中数据包的字节数定义的过大，那么 将导致下一个队列中的数据包被转发的等待时间过长。

CQ使用中存在一些限制：

1， 由于CQ是静态配置的，因此它不能适应网络结构的改变。

2， 由于数据包要经过处理器卡的分类，因此CQ对数据包转发的速度要比FIFO慢。

Configuration

1、定义CQ列表：

```
nimokaka(config-if)#custom-queue-list {list}
```
（后面是列表的号码）

2、定义队列中数据包的字节数或最大个数：

定义最大个数

nimokaka(config)# queue-list list-number queue queue-number limit limit-number

（queue的号码是队列的号码，limit的后面接的是队列里面的数量，默认为20个，范围是0 到32767）

定义队列大小（我觉得就是定义队列的带宽）

nimokaka(config)#queue-list {list} queue {queue-number} byte-count bytes

默认为1500字节

3、把数据包分配进特定的CQ 中,可以基于协议或基于进站接口：

基于协议

nimokaka(config)#queue-list {list} protocol protocol {queue-number} queue-keyword keyword-value

同样的道理，也可以加入相应的参数，参看PQ配置的参数

基于接口

nimokaka(config)#queue-list {list} interface interface {queue-number}

4、定义默认的CQ 队列,未分类的流量默认被分配进该队列:

nimokaka(config)#queue-list {list} default {queue-number}

检查配置

1、显示接口队列信息： **nimokaka#show queue [interface]**

2、显示CQ列表信息： **nimokaka#show queueing custom**

Case

```
interface serial 0
custom-queue-list 1                          //在接口应用队列

queue-list 1 protocol sna 1                  //在queue-list 1中定义sna,ip,ipx队列。
queue-list 1 protocol ip 2
queue-list 1 protocol ipx 3

queue-list1queue1byte-count15000             //queue 1,sna被设为3*(1500+3500)=15000
queue-list 1 queue 2 byte-conut 3500         //queue 2,ipx,被设为3500字节
queue-list 1 queue 3 byte-conut 1500         //默认1500
```

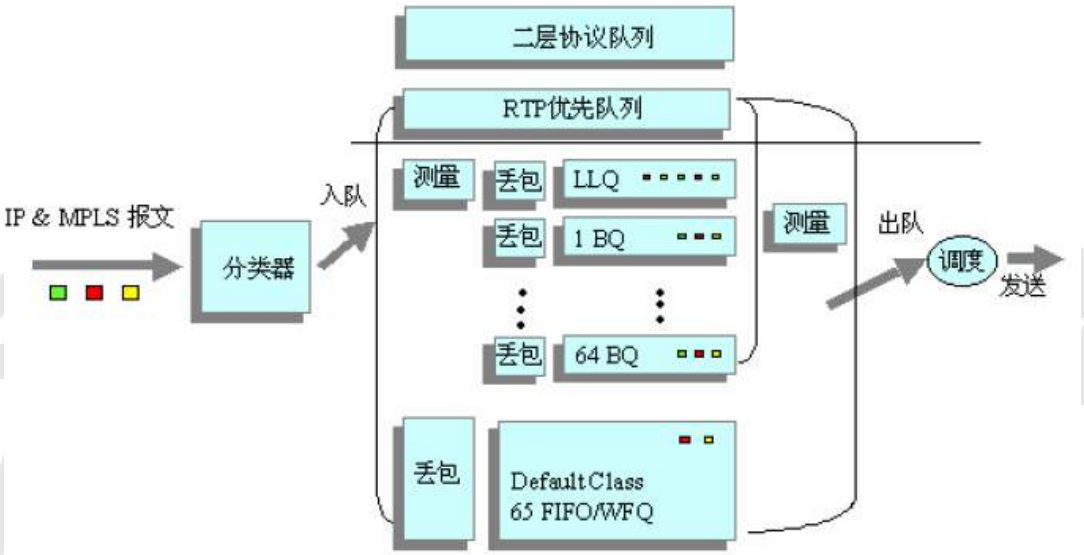
Class Based Weighted Fair Queueing

基于类的加权公平队列（Class Based Weighted Fair Queueing, CBWFQ）

CBWFQ，首先根据IP优先级、DSCP或者输入接口的IP数据流等规则来对报文进行分类，让不同类别的报文进入不同的队列对于不匹配任何类别的报文报文被送入系统定义的缺省类。

图中所示LLQ（Low Latency Queueing，低延迟队列）是一个具有较高优先级的队列，它的优先级仅次于二层协议队列（如同CQ中的0号队列），与RTP优先队列（RTP优先队列的参见后文介绍）一个或多个类的报文可以被设定进入LLQ，队列不同类别的报文可设定占用不同的带宽，在调度出队的时候，若LLQ中有报文则总是优先，发送LLQ中的报文直到LLQ中没有报文时或者超过为LLQ配置的最大预留带宽时，才调度发送其他队列中的报文。 进入LLQ的报文在接口没有发生拥塞的时候,此时所有队列中都没有报文,所有属于LLQ的报文都可以被发送在接口，发生拥塞的时候队列中有报文时，进入LLQ的报文被限速超出 规定流量的报文将被丢弃，这样在接口不发生拥塞的情况下可以使属于LLQ的报文能获得空 闲的带宽在，接口拥塞的情况下又可以保证属于LLQ的报文不会占用超出规定的带宽，保护了其他报文的应得带宽，另外由于只要LLQ中有报文系统就会发送LLQ中的报文，所以LLQ中的报文被发送的延迟最多是接口发送一个最大长度报文的时间，无论是延时还是延时抖动 LLQ都可以将之降低为最低限度，这为对延时敏感的应用如VoIP业务提供了良好的服务质量保证

下图中1到64的队列为各类报文的队列每类报文占一个队列，我们称它们为BQ(Bandwidth Queueing)



在系统调度报文出队的时候，按用户为各类报文设定的带宽将报文出队发送这种队列技术，应用了先进的队列调度算法可以实现各个类的队列的公平调度属于1到N1号BQ队列的报文可以被确保得到用户设定的带宽，当接口中某些类别的报文没有时,BQ 队列的报文还可以公平地得到空闲的带宽,大大提高了线路的利用率,同时在接口拥塞的时候仍然能保证各类报文得到用户设定的最小带宽。

当报文不匹配用户设定的所有类别时，报文被送入系统定义的缺省类，虽然允许为缺省类配置带宽使其作为BQ类进行基于类的队列调度，但是更多的情况是为缺省类配置WFQ，使所有进入缺省类的报文进行基于流的队列调度。

CBWFQ最多允许配置64个BQ类， 缺省类的WFQ的队列个数N可以由用户设定。 对于缺省类的WFQ和BQ，当队列的长度达到队列的最大长度时,缺省采用尾丢弃的策略。但用户还可以选择用加权随机预检测(Weighted Random Early Detection, WRED)的丢弃策略（加权随机预检测的丢弃策略请参见后面加权随机预检测WRED的描述）。

对于LLQ,由于在接口拥塞的时候流量限制开始起作用,所以用户不必设置队列的长度， 由于优先队列中的报文一般是语音报文Voice over IP, VoIP 采用的是UDP报文所以没有必要采用WRED的丢弃策略。

综上所述CBWFQ有一个低时延队列 - LLQ 用来支撑EF（加速转发）类业务， 被绝对优先发送，另外有64个BQ用来支撑AF（确保转发）类业务，可以保证每一个队列的带宽及可控的时延，还有一个WFQ对应BE（尽力而为传输）业务使用接口剩余带宽进行发送。CBWFQ可根 据报文的输入接口ACL、IP优先级/DSCP等规则对报文进行分类，进入相应队列规则可以是通手工配置， 对于进入LLQ和BQ的报文要进行测量， 考虑到链路层控制报文的发送链路层封装开销及物理层开销

建议RTP优先队列LLQ与BQ占用接口的总带宽不要超过接口带宽的75%（默认就是只占用75％带宽）LLQ只采用尾丢弃，BQ可采用尾丢弃或者WRED（基于IP优先级DSCP） WFQ可采用尾丢弃和WRED。CBWFQ可为不同的业务定义不同的调度策略，如带宽时延等。由于 涉及到复杂的流分类， 对于高速接口GE以上启用CBWFQ特性系统资源存在一定的开销。

RSVP也可以和CBWFQ 协同工作.当一个接口同时配置了CBWFQ 和RSVP,它们之间的工作 是独立的.并且当CBWFQ 不存在的时候RSVP 还是会继续工作.

CBWFQ使用存在的一些限制第

一、目前流量和整形不支持CBWFQ。

二、CBWFQ 不支持以太网子接口

Configuration

一、定义分类的策略，即class map
二、设置策略，即定义policy map
三、把policy map 应用在相关接口上

定义class map 步骤如下：

1、定义class map

nimokaka(config)#class-map [match-all|match-any] {map-name}

2、定义匹配语句

nimokaka(config-cmap)#

一些匹配条件选项：

match access-group {ACL} 匹配IP ACL
match protocol {protocol} 匹配协议
match input-interface {interface} 匹配进站接口
match qos-group {Group ID} 匹配组ID
match destination-address {mac mac-address} 匹配目标MAC 地址
match source-address {mac mac-address} 匹配源MAC 地址
match ip {dscp dscp} 匹配IP DSCP 值
match ip {precedence precedence} 匹配IP 优先级
match class-map {map-name} 匹配
class map match vlan {vlan-id} 匹配VLAN

定义分类的策略,即policy map 的步骤如下:

1. 设置policy map:

nimokaka(config)#policy-map {policy-name}

2. 调用class map 或默认的class map(所有未分类的流量默认都属于该分类,否则未分类的流量将以尽力而为的方式被处理):

nimokaka(config-pmap)#class {class-map|class-default}

3. 设置策略:

nimokaka(config-pmap-c)#bandwidth {kbps|percent percent}

4. 定义尾丢弃机制允许的队列中数据包个数的上限,默认值为64:

nimokaka(config-pmap-c)#queue-limit {packets}

其他配置参数

nimokaka(config-pmap-c)#random-detect 用于WRED
nimokaka(config-pmap-c)#shape 令牌桶参数
nimokaka(config-pmap-c)#police (car限速)
nimokaka(config-pmap-c)#priority 优先级，低延迟队列(LLQ).

在出站接口上应用policy map:

nimokaka(config-if)#service-policy output {policy-name}

更改用于RSVP 和CBWFQ 等队列机制保留的最大带宽值,默认为75%:

nimokaka(config-if)#max-reserved-bandwidth {percent}

Check Configuration

1、查看policy map 信息： **nimokaka#show policy-map [policy-name]**
2、查看接口的policy map 信息： **nimokaka#show policy-map interface [interface]**

3. 显示接口的队列信息： nimokaka#show queue [interface]

Case

限制源自192.168.10.0/24 的流量的带宽为1000kbps：

```
!
class-map match-all nimokaka
!
policy-map kaka
bandwidth 1000              限制带宽1000k
queue-limit 30              限制队列数据包上限30个包
class class-default         其他的放置到默认队列
!
interface Serial1
ip address 172.16.10.1 255.255.255.252
service-policy output kaka
!
access-list 1 permit 192.168.10.0 0.0.0.255
```

多class-map联合使用

```
class-map match-any nimokaka1
match protocol sqlnet
match protocol ipsec
match access-group 100
match ip precedence 4 5
!
class-map match-all nimokaka2
match access-group 101
match access-group 102
!
class-map nimokaka3
match access-group 103
policy-map kaka
class nimokaka1 bandwidth 6000 class nimokaka2 bandwidth 3000 class nimokaka3
bandwidth 700 class class-default bandwidth 200
!
interface ethernet 1/1
service-policy output kiss
```

Low Latency Queueing

低延迟队列（Low Latency Queueing，LLQ）

Basic introduction

低延迟队列(LLQ)把优先级队列的特性加入到了CBWFQ中，这点和IP RTP优先级特性类似。如果没有LLQ，对于一些实时的数据流量，比如语音数据流量，CBWFQ对于每个定义好的分类的操作是基于WFQ的。采用了LLQ 之后，该分类的操作将优先于别的分类。LLQ 减少了语音会话的抖动。LLQ 和IP RTP优先级特性的区别在于，它不受UDP端口号的限制。

Configuration

nimokaka(config-pmap-c)#priority {bandwidth}

在CBWFQ中使用bandwidth命令是用于定义普通队列的，

但是如果改用priority,配置的就是低延时队列，凌驾于CBWFQ的上面

nimokaka（config-if）#max-reserved-bandwidth percent 为LLQ和IP RTP设置占用带宽的百分比

Check Configuration

1、显示接口队列信息： nimokaka#show queue [interface]

2、调试优先级队列： nimokaka#debug priority

Case

使用LLQ给视频流量提供保留带宽

```
class-map match-all nimokaka1
match access-group 100
class-map match-all nimokaka2
match access-group 101
```

```
!
policy-map video1            //定义LLQ策略
class nimokaka1
priority 1518                //使用priority定义为Video预留的带宽
set ip precedence 5              //更改优先级
class class-default fair-queue
policy-map video class nimokaka2 priority 1518
set ip precedence 5 class class-default fair-queue
!
interface FastEthernet1/0/0
no ip address
no ip route-cache distributed full-duplex
!
interface FastEthernet1/0/0.1 description HQ1 LAN encapsulation dot1Q 1
ip address 129.2.80.21 255.255.0.0
service-policy output video1        //在接口下应用策略
!
interface Serial1/0/0
description Router 1 to ROUTER 2
ip address 206.141.24.1 255.255.255.0
ip route-cache policy
no ip route-cache distributed service-policy output video
!
access-list 100 permit ip host 141.2.20.20 host 129.2.20.34 precedence critical
//通过ACL匹配
access-list 101 permit ip host 129.2.20.34 host 141.2.20.20 precedence critical
```

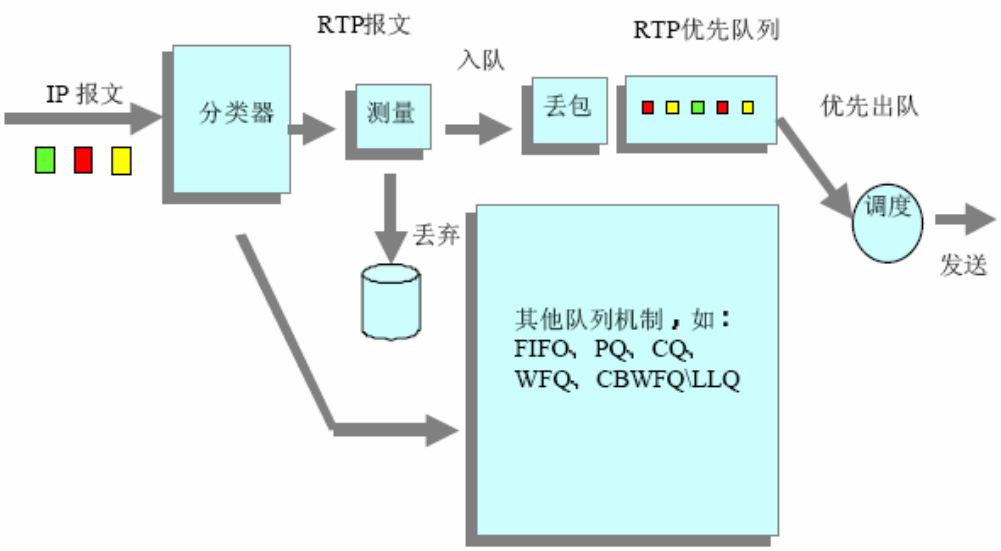
在配置之前肯定事先已经将视频流量的数据包优先级更改成5

给ACL100和101定义的数据包加入class map，

然后使用policy map中的priority命令来保证1518k的带宽，并且把优先级set到5，交给下一跳路由器

Real Time Protocol Priority Queueing

Basic introduction



如图所示：RTP优先队列是一种解决实时业务包括语音与视频业务服务质量的简单队列技术。其原理就是将承

载语音或视频的RTP报文送入高优先级队列使其得到优先发送，保证延时和抖动降低为最低限度，从而保证了语音或视频这种对时延敏感业务的服务质量，RTP 优先队列将RTP报文送入一个具有较高优先级的队列RTP报文是端口号在一定范围内为偶数的UDP报文，端口号的范围可以配置一般为16384~32767，RTP优先队列可以同前面所述的任何一种队列包括FIFO、PQ、CQ、WFQ与CBWFQ 结合使用。它的优先级是最高的。

一般语音数据包的体积较小，如果有体积较大的数据包要从该接口被转发出去，该接口应配置链路分片和交叉（LFI）特性.体积较大的数据包被分片为体积较小的数据包。该特性 防止语音数据包要等待到体积较大的数据包被转发完毕之后才能被转发这样语音数据包可 以和被分片的数据包交叉被转发出去。从而减少了语音数据包转发消耗的时间。

由于对进入RTP优先队列的报文进行了限速，超出规定流量的报文将被丢弃这样，在接口拥塞的情况下可以保证属于RTP优先队列的报文不会占用超出规定的带宽，保护了其他报文的应得带宽解决了PQ的高优先级队列的流量可能丢弃低优先级流量的问题。

Configuration

nimokaka(config-if)#ip rtp priority {starting-rtp-port-number port-number-range}{bandwidth}

添加起始port和port范围,后面的bandwidth填写的是预留的带宽

nimokaka（config-if）#max-reserved-bandwidth percent 为LLQ和IP RTP设置占用带宽的百分比

Check Configuration

1、显示接口队列信息： **nimokaka#show queue [interface]**

2、调试优先级队列： **nimokaka#debug priority**

Case

RTP同CBWFQ结合使用

定义class map

```
nimokaka(config)# class-map nimokaka
nimokaka(config-cmap)# match access-group 101
nimokaka(config-cmap)# exit
```

定义和使用policy map nimokaka(config)# **policy-map kiss**

```
nimokaka(config-pmap)# class nimokaka
nimokaka(config-pmap-c)# bandwidth 3000
nimokaka(config-pmap-c)# queue-limit 30 nimokaka(config-pmap-c)# random-detect
nimokaka(config-pmap-c)# random-detect precedence 0 32 256 100
nimokaka(config-pmap-c)# exit nimokaka(config)# interface Serial1
nimokaka(config-if)# service-policy output nimokaka
```

设置RTP

nimokaka(config-if)# ip rtp priority 16384 16383 40

RTP是直接应用到接口上的，所以优先级超级超级高，另外看看端口范围，起始端口是16384，范围是从16384加上后面的range=16383，应该是从16384--32767这个范围，存在一个加法公式。

Compression Real Time Protocol

Basic introduction

实时传输协议(RTP)是一种用于传输实时数据流量的协议。RTP包含数据部分和包头部分。数据部分用于支持实时传输应用程序的各种属性。包头部分的体积比较大，最少包括了12 字节的RTP包头,20字节的IP 包头(IPH)和8字节的UDP包头,因此IP/UDP/RTP包头总长度 至少为40字节。根据IP/UDP/RTP包头长度的不同,RTP数据包的长度为20字节到160字节不等。如果不对IP/UDP/RTP包头进行压缩的话,对于RTP数据包的传输是很没效率的。因此后来出现了压缩IP/UDP/RTP包头的压缩式实时传输协议(CRTP)。

CRTP是一种逐跳的压缩机制。CRTP可以把IP/UDP/RTP包头从40字节压缩为2到5字节。当 广域网接口带宽不高,并且RTP数据流量过大的话，应该考虑使用CRTP；但是对于高于T1线 路速率的接口，无需使用CRTP。CRTP支持ISDN,

支持使用PPP，HDLC或FR封装的接口.但是FR 的封装格式只能使用Cisco特有的格式。

Configuration

1、启用CRTP：如果不指定关键字passive，将对所有IP/UDP/RTP包头进行压缩；如果指定 passive关键字，当进站RTP数据包的IP/UDP/RTP包头被压缩，才相应的压缩出站的RTP数据 包的IP/UDP/RTP包头：

nimokaka(config-if)#ip rtp header-compression [passive]

2、更改IP/UDP/RTP包头压缩的连接数,默认为16条.可选：

nimokaka(config-if)#ip rtp compression-connections {number}

RTP头压缩，默认16条，最大500条

nimokaka(config-if)#ip tcp compression-connections {number}

TCP头压缩，默认16条，最大128条

Config CRTP over FR connections

1、在物理接口上启用CRTP，那么该物理接口相关的子接口将继承CRTP的配置信息。如果不指定关键字passive，将对所有IP/UDP/RTP包头进行压缩；如果指定passive关键字，当进站RTP数据包的IP/UDP/RTP包头被压缩，才相应的压缩出站的RTP数据包的IP/UDP/RTP包头：

nimokaka(config-if)#frame-relay ip rtp header-compression [passive]

2、更改IP/UDP/RTP包头压缩的连接数，默认为16条，可调整：

nimokaka(config-if)#frame-relay ip rtp compression-connections {number}

3、只针对特定的PVC启用CRTP。如果指定关键字active，将对所有IP/UDP/RTP包头进行压缩；如果指定passive关键字，当进站RTP数据包的IP/UDP/RTP包头被压缩，才相应的压缩出站的RTP数据包的IP/UDP/RTP包头。还可以指定最大的IP/UDP/RTP包头压缩的连接数,默认为16条，可调整：

nimokaka(config-if)#frame-relay map ip {ip-address} {dlci} [broadcast] rtp header-compression [active|passive] [connections number]

4、在特定PVC上同时启用CRTP 和TCP 头部压缩。

nimokaka(config-if)#frame-relay map ip {ip-address} {dlci} [broadcast] compress

Check Configuration

1、显示IP/UDP/RTP包头的压缩统计信息：

nimokaka#show ip rtp header-compression [interface] [detail]

2、显示FR 的IP/UDP/RTP包头的压缩统计信息：

nimokaka#show frame-relay ip rtp header-compression [interface]

Weighted Round-Robin

(1) WRR调度算法

交换机的端口支持4 个输出队列，WRR 队列调度算法在队列之间进行轮流调度，保证每个队列都得到一定的服务时间。WRR 可为每个队列配置一个加权值（依次为w3、w2、w1、w0），加权值表示获取资源的比重。如一个100M的端口，配置它的WRR 队列调度算法的加权值为50、30、10、10（依次对应w3、w2、w1、w0），这样可以保证最低优先级队列至少获得10Mbit/s 带宽，避免了低优先级队列中的报文可能长时间得不到服务的缺点。WRR 队列还有一个优点是，虽然多个队列的调度是轮循进行的，但对每个队列不是固定地分配服务时间片——如果某个队列为空，那么马上换到下一个队列调度，这样带宽资源可以得到充分的利用。

(2) HQ-WRR 调度算法

HQ-WRR 调度模式在WRR 的基础上，在4 个输出队列中选择队列3 为高优先级队列。如果4 个队列的占用的带宽超过了端口的能力，交换机首先保证队列3 的报文优先发送出去，然后对其余3 个队列实行WRR 调度。

Cisco Catalyst 3550 交换机 QoS 时序及队列

3550 交换机有两种不同类型的端口：千兆端口和非千兆端口（10/100M 端口）每个 3550 的端口上都有4个不同的输出队列。这些队列中的一个可以被配置为优先级队列。余下的几个端口被配置为非绝对的优先级队列，

并使用 Weighted Round Robin (WRR)。 所有的端口上，数据包根据各自的服务类别（CoS）被分配为四中可能的类别之一。

其中千兆端口还能支持每个队列的管理机制。每个队列可以使用Weighted Random Early Discard (WRED)或者双线程的tail drop 。队列大小可调（每个队列均分配相应的 缓冲区）。非千兆端口不支持任何队列管理机制，例如WRED或者双线程tail drop

10/100M端口支持FIFO 队列。每个端口队列的大小都不可改变。但是可以为每个队列分配最小的保留带宽。

WRR Configuration

CoS 到队列映射

本节讨论3550如何决定将每个数据包放置到队列中去。数据包队列取决于服务类别（CoS）。通过使用CoS 到队列的接口映射命令，每个八种可能的Cos数值将被映射到相应 的四个队列。下面是该命令的示例：

```
nimokaka(config-if)# wrr-queue cos-map queue-id cos1... cos8
nimokaka(config-if)# wrr-queue    cos-map   1 0   1
nimokaka(config-if)# wrr-queue    cos-map   2 2   3
nimokaka(config-if)# wrr-queue    cos-map   3 4   5
nimokaka(config-if)# wrr-queue    cos-map   4 6   7
```

该示例将 CoS 0和1映射到 Q1, CoS 2 和3 映射到 Q2, CoS 4和5映射到 Q3, CoS 6和7映射到Q4。

每个端口的CoS到队列的映射情况可以通过使用下面的命令来进行验证：

```
nimokaka# sh mls qos int gig 0/1 queueing
GigabitEthernet0/1
...Cos-queue map:
cos-qid
0 - 1
1 - 1
2 - 2
3 - 2
4 - 3
5 - 3
6 - 4
7 - 4.
```

当然，DSCP也可以有默认的映射，如下图

DCSP-to-Transmit queue map	DSCP 0-15 Queue 1 DSCP 16-31 Queue 2 DSCP 32-47 Queue 3 DSCP 48-63 Queue 4

绝对的优先级队列

绝对的优先级队列在初始状态下通常是空的。这就意味着一旦有数据包进入队列，该包将马上被转发。当 WRR 队列中所有的数据包都被转发后，优先级队列根据需要关闭并清空。 绝对的优先级队列被特别设计来处理对延迟/抖动比较敏感的数据流，例如语音。绝对的优先级队列将导致其他队列严重滞后。在其他三个 WRR 中的数据包在绝对的优先级队列中数据传输完成之前，将不会被转发。注意：要避免其他队列的严重滞后，要特别注意放到优先级 队列中的流量。

该队列通常用于语音数据流，而此类型应用并不占用很高的带宽。但是若有人将一些占 用带宽较多的应用（例如数据转移或备份）放到绝对的优先级队列。这将引起其他流量的严重滞后。要避免该问题，特殊的数据流应被放置在分类/准入，并在网络中标记该数据流。 例如，你可能需要采取一下预防措施：

1、在非可信的源端口使用非可信的端口 QoS 状态；

2、在使用 Cisco IP 电话端口可靠的边界特性时，确信 IP 电话配置于其它应用是可信的

3、修正进入绝对优先级队列的数据流。在千兆端口上修正数据流的流量限制为 100M。

在 3550 上，可以配置一个队列为优先队列，（总是 Q4），在端口模式下使用如下命令：

```
nimokaka(config-if)# priority-queue out
```

（加载这个命令之后，Q4 就会成为优先队列）

如果某个端口没有配置优先队列，则 Q4 被当做标准的 WRR 队列（下节将详细描述） 你可以通过输入和下面一样

的 IOS 命令来验证某端口是否被配置为绝对优先级队列，

```
nimokaka#sh mls qos interface gig 0/1 queueing
GigabitEthernet0/1
Egress expedite queue: ena
```

WRR ovet Catalyst 3550

在3550上,WRR是一个对输出时间序列进行管理的机制。WRR在三个或四个队列(如果 没有绝对优先级队列)之间工作。使用WRR模式的队列在循环方式下是置空的，可以为每个 队列配置相应的权值。

例如，配置了不同的权值，不同的队列将提供不同的服务，如下所示：

```
Serving WRR Q1 : 10% of time
Serving WRR Q2 : 20% of time
Serving WRR Q3 : 60% of time
Serving WRR Q4 : 10% of time
```

对每个队列，你可以在端口模式使用以下命令来配置四个权值（各自相对于一个队列）：

```
nimokaka(config-f)#wrr-queue bandwidth weight1 weight2 weight3 weight4
```

Eaxmple

```
nimokaka(config)# interface gigabitethernet0/1
nimokaka(config-if)# wrr-queue bandwidth 1 2 3 4
```

注意：权值是相对的，下面是计算方式

```
Q1 = weight 1 /(weight1 + weight2 + weight3 + weight4) = 1/(1+2+3+4) = 1/10
Q2 = 2/10
Q3 = 3/10
Q4 = 4/10
```

WRR可通过以下两种方式执行：

1.、WRR per bandwidth: 每个权值描述了可以用于发送的特别带宽。权Q1允许使用大约 10%的带宽，Q2 将获得大约20%的带宽，以此类推。

2、 WRR per packet: 该算法在3550交换机上实现。 这表示每个权值表示了某个数量的数据包将被发送，而不管包的大小如何。

3550上实现WRR per packet表现为如下形式：

Q1 传输 1/10 的数据包
Q2 传输 2/10 的数据包
Q3 传输 3/10 的数据包
Q4 传输 4/10 的数据包

如果被传送的包是同样大小则是最理想的情况.在4个队列中你依然能够获得理想的共享带宽。然而，如果队列间的平均包大小有差异，则会在拥塞事件发生时对传输产生巨大的影响。

假设当前交换机只有两个数据流， 同时假设处于以下的情形：

一个千兆口的队列2（Q2）以Cos 3类别方式每秒传输少量的交互应用数据流（80字节/帧）

一个千兆口的队列1（Q1）以Cos 0类别方式每秒传输大型文件数据流（1518字节/帧）

两个队列都将以传输1Gbps 的速率传输数据。两个数据流需要共享同一个输出的千兆口。假设我们已经为Q1 和 Q2 设置了同样的权值，WRR 应用到每个数据包，并且每个队列 内传输的数据量不同于两个队列之间的数据量。每个队列都转发了同样数量的数据包,然而 交换机实际上发送了下面数量的数据：

77700 包/秒由Q2输出 = (77700 x 8 x 64) bits/sec (大约 52 Mbps)

77700 包/秒由Q1输出= (77700 x 8 x 1500) bits/sec (大约 948 Mbps) 注意：

如果你想要每个队列都公平的接入网络，需要考虑每个数据包的平均值。每个数据包都被假设放置在同一个队列，因而权值得到改善。

例如：如果你想要为四个队列赋予相同的接入（每个队列各自分配到1/4的带宽），流量表现为如下形式：

Q1：最佳的互联网数据流量。假定数据流的平均包大小为256字节。

Q2 ： 文件备份形成的文件传输，主要由1500字节构成的数据包。

Q3 ： 视频流，每个包被分成192字节。

Q4 ： 交互应用，主要由64字节构成的数据包。 这就产生了以下的情形：

Q1消耗 4 倍于Q4的带宽 Q2消耗 24 倍于Q4的带宽 Q3消耗 3 倍于Q4的带宽

若要以同样的带宽接入网络，采用如下的配置：

Q1 权值设为6

Q2 权值设为1

Q3 权值设为8

Q4 权值设为24

如果分配了以上的权值，则在拥塞事件发生时，四个队列将分享到同样的带宽。

如果设置了绝对优先级队列，WR权值将在其余三个队列中重新分配。下面是一个设置了绝对优先级，而Q4没有进行配置的情况下，队列1、2、3。

Q1 = 1 / (1+2+3) = 1/6数据包输出

Q2 = 2/6数据包输出

Q3 = 3/6数据包输出

队列的权值可以通过IOS show 命令进行验证：

```
nimokaka#sh mls qos interface gig 0/1 queueing GigabitEthernet0/1
QoS is disabled. Only one queue is used
When QoS is enabled, following settings will be applied
Egress expedite queue: dis wrr bandwidth weights:
qid-weights
1 - 25
2 - 25
3 - 25
4 - 25
```

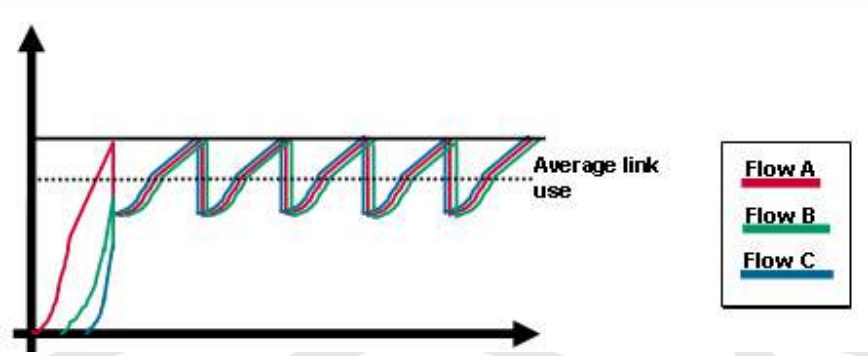
如果启用了快速优先级队列，Q4的权值仅在快速队列失效时使用。 看下面的示例：

```
nimokaka#sh mls qos interface gig 0/1 queueing
GigabitEthernet0/1
Egress expedite queue: ena wrr bandwidth weights:
qid-weights
1 - 25
2 - 25
3 - 25
4 - 25
```

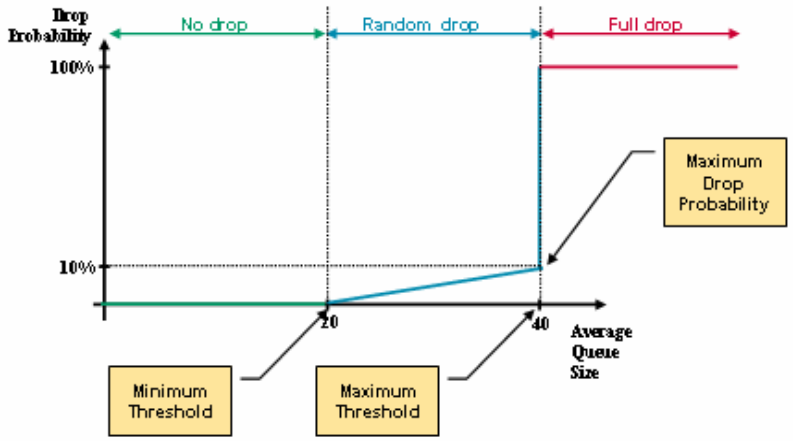
Congestion avoidance

Basic introduction

由于内存资源的有限按照传统的处理方法，当队列的长度达到规定的最大长度时，所有到来的报文都被丢弃，对于TCP报文如果大量的报文被丢弃将造成TCP超时， 从而引发TCP的慢启动和拥塞避免机制。使TCP减少报文的发送。当队列同时丢弃多个TCP连接的报文时将造 成多个TCP连接同时进入慢启动和拥塞避免。称之为TCP全局同步。这样多个TCP连接发向队列的报文将同时减少，使得发向队列的报文的量不及线路发送的速度，减少了线路带宽的利用，并且发向队列的报文的流量总是忽大忽小，使线路的上的流量总在极少和饱满之间波动如下图所示

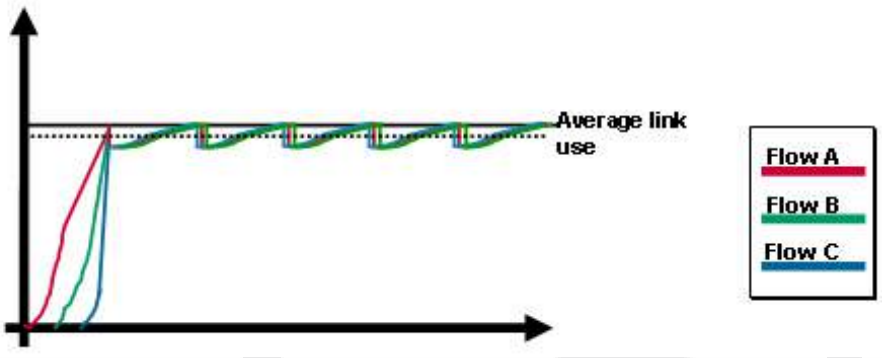


为了避免这种情况的发生引入了RED技术（Random Early Detection）随机预检测，如图

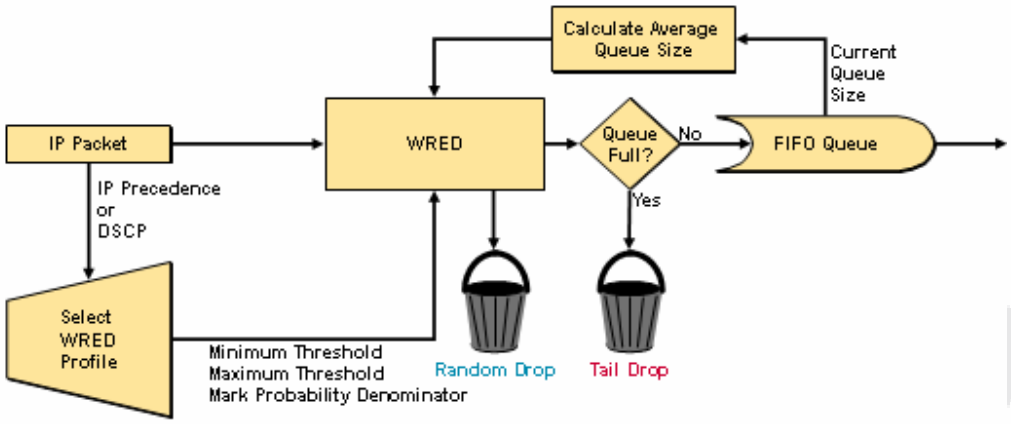


采用RED时用户可以设定队列的阈值threshold,当队列的长度小于低阈值时,不丢弃报文;当队列的长度在低阈值和高阈值之间时，WRED开始随机丢弃报文，队列的长度越长，丢弃的概率越高；当队列的长度大于高阈值时，丢弃所有的报文。

由于RED随机地丢弃报文，将避免使多个TCP连接同时降低发送速度，从而避免了TCP的全局同步现象,当某个TCP连接的报文被丢弃开始减速发送的时候,其他的TCP连接仍然有较高的发送速度，这样无论什么时候总有TCP连接在进行较快的发送，提高了线路带宽的利用率。



如上图所示，图线平稳了很多，但是RED只是随机进行丢弃，所以灵活性要差一点，现在所采用的基本上都是WRED（Weighted Random Early Detection）。原理都是一样的。只不过 WRED与RED的区别在于前者引入IP优先权DSCP值来区别丢弃策略，可以为不同IP优先级DSCP 设定不同的队列长度、队列阈值、丢弃概率,从而对不同优先级的报文提供不同的丢弃特性。 这是WRED的重要特点。下图介绍了WRED和队列之间的关系



在设置时如果直接采用队列的长度与用户设定的阈值比较并进行丢弃（这是设置队列门 限的绝对长度），将会对突发性的数据流造成不公正的待遇，不利于数据流的传输，所以在 与设定的阈值比较并进行丢弃时采用队列的平均长度（这是设置队列门限与平均长度比较的 相对值）队列的平均长度既反映了队列的变化趋势又对队列长度的突发变化不敏感避免了对 突发性的数据流造成不公正的待遇。另外还要注意WRED不能配置在使用了基于路由交换处理 器(RSP)的CQ、PQ和WFQ队列机制的接口上。

Configuration

WRED可以在接口上进行配置，也可以在policy上进行配置，可以针对于precedence进行 RED，也可以针对于DSCP值进行RED，当然，两者之间只能选择一个。

DSCP-Based

1. 使用IP DSCP 来配置WRED:

```
nimokaka(config-if)#random-detect dscp-based
```

2. 设置丢弃数据包的最小值,最大值和丢弃数据包的轮循间隔:

```
nimokaka(config-if)#random-detect dscp {dscp} {min max mark}
```

IP precedence-Based

1、启用WRED:

```
nimokaka(config-if)#random-detect
```

2、设置WRED 丢弃数据包的最小值,最大值和丢弃数据包的轮循间隔:

```
nimokaka(config-if)#random-detect precedence {precedence|rsvp} {min max mark}
```

show commands

1、显示接口队列信息： nimokaka#show queue [interface]

2、显示WRED信息： nimokaka#show queueing random-detect

Case

1.Base on Interface

```
interface Serial 0/1/0
ip address 200.200.14.250 255.255.255.252
random-detect
random-detect precedence  0 10 25 10
random-detect precedence  1 20 35 10
random-detect precedence  2 15 25 10
random-detect precedence  3 25 35 10
random-detect precedence  4 1  2    1
random-detect precedence  5 35 40 10
random-detect precedence  6 30 40 10
random-detect precedence  7 30 40 10
```

2.Base on policy-map

```
Switch(config)#class-map kaka
Switch(config-cmap)#match access-group 101
Switch(config)#policy-map kaka1
Switch(config-pmap)#class kaka
Switch(config-pmap-c)#bandwidth 48
Switch(config-pmap-c)#random-detect dscp-based
Switch(config-pmap-c)#random-detect dscp 8 24 40
Switch(config-pmap-c)#interface S1/0
Switch(config-if)#service-policy output kaka1
```

WRED by Netflow

当WRED和WFQ配合使用时还可以实现基于流的WRED，这是因为在进行分类的时候，不同 的流有自己的队列，对于流量小的流，由于其队列长度总是比较小，所以丢弃的概率将比较 小，而流量大的流将会有较大的队列长度，从而丢弃较多的报文，保护了流量较小的流的利 益。即使WRED和其他的队列机制配合使用，对于流量小的流，由于其报文的个数较少，所以 从统计概率来说被丢弃的概率也会较小， 也可以保护流量较小的流的利益。

Configuration

在配置基于流的WRED 之前， 必须先启用WRED。步骤如下：

1、启用基于流的WRED：

```
nimokaka(config-if)#random-detect flow
```

2、设置平均深度因素(average depth factor)的值,值必须为2的幂,默认值为4.可选：

```
nimokaka(config-if)#random-detect flow average-depth-factor {scaling-factor}
```

这个参数是改变一个乘法的比例因数.从而改变对列的大小， 其实就是改变队列的长度， 我们可以不去研究

3、设置基于流的WRED 的数据流数目,默认值为256.可选：

```
nimokaka(config-if)#random-detect flow count {number}
```

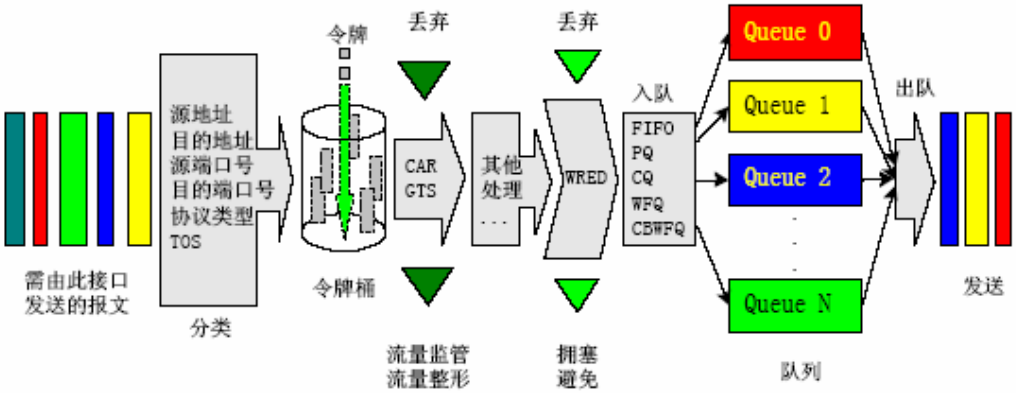
show command

1、显示接口队列信息： `nimokaka#show queue [interface]`

2、显示WRED 信息： `nimokaka#show queueing random-detect`

Traffic-Policing

Basic introduction



在WRED和队列技术之前，还存在着一个流量管理和流量整形技术，通过这种 技术来管理流量，在结合后面的WRED和各种队列技术，才能整体实现了QOS。

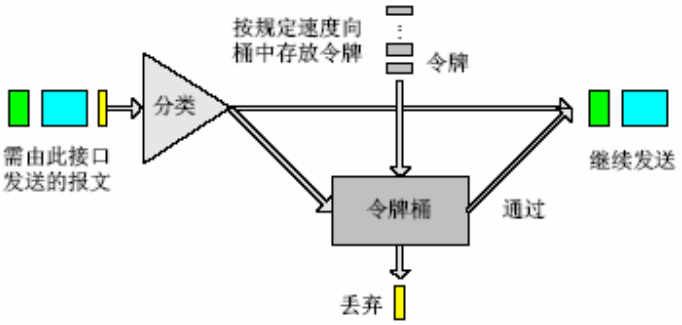
流量策略（traffic policing ）的典型作用是限制进入某一网络的某一连接的流量与突发（当然，限制发出也是可以的），在报文满足一定的条件时，如某个连接的报文流量过大流量，监管就可以对该报文采取不同的处理动作例如丢弃报文或重新设置报文的优先级等，通常的用法是使用CAR来限制某类报文的流量例如限制HTTP报文不能占用超过50%的网络带宽。

Committed Access Rate

承诺访问速率-CAR

Basic introduction

对于ISP来说对用户送入网络中的流量进行控制是十分必要的。对于企业网，对某些应 用的流量进行控制也是一个有力的控制网络状况的工具，网络管理者可以使用约定访问速率 CAR来对流量进行控制。CAR利用令牌桶（Token Bucket, TB）进行流量控制。如下图：



上图所显示的是利用CAR进行流量控制的基本处理过程,首先根据预先设置的匹配规则 来对报文进行分类，如果是没有规定流量特性的报文就直接继续发送，并不需要经过令牌桶的处理；如果是需要进行流量控制的报文，则会进入令牌桶中进行处理，如果令牌桶中有足 够的令牌可以用来发送报文，则允许报文通过，报文可以被继续发送下去；如果令牌桶中的令牌不满足报文的发送条件则报文被丢弃，这样就可以对某类报文的流量进行控制。

令牌桶按用户设定的速度向桶中放置令牌，并且用户可以设置令牌桶的容量，当桶中令牌的量超出桶的容量的时候，令牌的量不再增加；当报文被令牌桶处理的时候，如果令牌桶中有足够的令牌可以用来发送报文，则报文可以通过可以被继续发送下去，同时令牌桶中的令牌量按报文的长度做相应的减少，当令牌桶中的令牌少到报文不能再发送时，报文被丢弃。 令牌桶是一个控制数据流量的很好的工具，当令牌桶中充满令牌的时候，桶中所有的令牌代 表的报文都可以被发送，这样可以允许数据的突发性传输，当令牌桶中没有令牌的时候报文将不能被发送，只有等到桶中生成了新的令牌报文才可以发送，这就可以限制报文的流量只 能是小于等于令牌生成的速度，达到限制流量的目的。

在实际应用中CAR不仅可以用来进行流量控制，还可以进行报文的标记（mark）或重新标记（re-mark），具体来讲就是CAR可以设置IP报文的优先级或修改IP报文的优先级，达到 标记报文的目的。例如当报文符合流量特性的时候可以设置报文的优先级为5，当报文不符 合流量特性的时候可以丢弃，也可以设置报文的优先级为1并继续进行发送，这样后续的处 理可以尽量保证不丢弃优先级为5的报文。在网络不拥塞的情况下也发送优先级为1的报文。 当网络拥塞时首先丢弃优先级为1的报文。然后才丢弃优先级为5的报文。

CAR可以为不同类别的报文设置不同的流量特性和标记特性，即首先对报文进行分类， 然后不同类别的报文有不同的流量特性和标记特性，此外CAR的策略还可以进行串联处理。 例如可以对所有的报文限制一个总的流量，然后在总的流量中再限制部分报文的流量符合某 个流量特性。

CAR通常使用在网络边界路由器的接口上，用来限制进入或离开该网络的流量速率。每个接口可以配置多个CAR策略，当数据包进入使用了多个策略的接口时，路由器将检查每个策略，直到数据包和某个策略相匹配；如果没有找到匹配的策略,默认操作是转发该数据包。

CAR的使用限制：

第一、CAR只能对IP 流量限速。

第二、CAR不支持快速以太网信道(FastEtherChannel)

第三、CAR不支持隧道接口

第四、CAR不支持ISDN PRI 接口。

Configuration

nimokaka(config-if)#rate-limit {input|output} {CIR Bc Be} conform-action {action} exceed-action {action}

output|input指输出或者输入的流量。CIR配置的是承诺接入速率，它的值的范围是在8000-2000000000 bit每秒。Bc是普通突发，它的值应在1000-512000000byte，Be是最大突发，其值范围为2000-1024000000bytes.conform-action后面规定的是遵从条件时候的动作，exceed-action 后面规定的是超出时的动作，遵从的条件说得就是当要发的数据小于正常突发(bc)的时候。最大条件说得是要发的数据大于普通突发，小于最大突发(be)的时候。违章条件是说得是要发的数据大于最大突发(be)的时候。

{action}

continue	继续执行下一条CAR 语句
drop	丢弃该数据包
set-prec-continue {precedence}	设置IP 优先级并继续执行下一条CAR 语句
set-prec-trasnmit {precedence}	设置IP 优先级并转发该数据包
set-dscp-continue {dscp}	设置IP DSCP 值并继续执行下一条CAR 语句
set-dscp-trasnmit {dscp}	设置IP DSCP 值并转发该数据包
set-qos-continue {group ID}	设置QoS 组ID 并继续执行下一条CAR 语句
set-qos-transmit {group ID}	设置QoS 组ID 并发送该数据包
transmit	转发该数据包

基本上就是这些，但是我们发现好像这样做是管理整个接口的流量，也可以对某一个流量进行CAR管理。或者针对IP优先级或者根据DSCP进行管理。

针对于DSCP值进行CAR

nimokaka(config-if)#rate-limit {input|output} [dscp dscp] {CIRBc Be} conform-action{action} exceed-action {action}

针对于ACL进行CAR

nimokaka(config-if)#rate-limit {input|output}access-group {ACL} {CIRBc Be}conform-action {action} exceed-action {action}

针对于限速ACL进行CAR

nimokaka(config-if)#rate-limit {input|output} access-group rate-limit {ACL} {CIR Bc Be} conform-action {action} exceed-action {action}

限速ACL是一种特殊的ACL，其实也没啥特殊的，就是一个调用关系

nimokaka(config)#access-list rate-limit {ACL} {precedence|mac-address}

ACL是限速ACL的号码，可以匹配优先级，也可以匹配源mac地址

Show command

1、查看限速ACL： **nimokaka#show access-lists rate-limit [ACL]**

2、查看接口的限速信息： **nimokaka#show interfaces [interface] rate-limit**

Case

Interface-based

```
interface Hssi0/0/0
description 45Mbps to R1
rate-limit input 15000000 2812500 2812500 conform-action transmit exceed-action drop
ip address 200.200.14.250 255.255.255.252
rate-limit output 15000000 2812500 2812500 conform-action transmit exceed-action drop
```

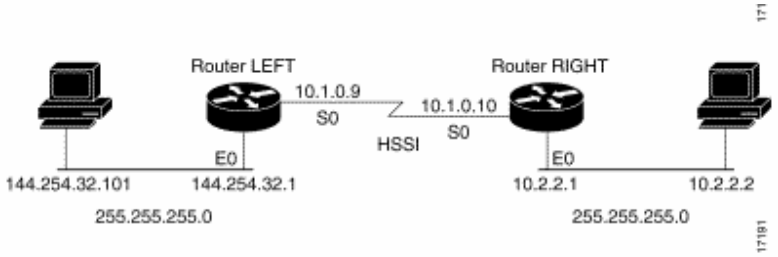
hssi高速串口是45M的带宽，但是ISP的接入承诺信息速率为15M，

并且限定普通突发大小为2812500，最大突发大小也是2812500

DSCP-Based

```
interface Serial1
ip address 10.0.0.1 255.255.255.252
rate-limit output dscp 1 20000000 24000 32000 conform-action transmit exceed-action drop
```

ACL-Based



a) 所有的www流量都得发出，而且web中遵从第一个速率策略的流量设置ip优先级为5,不遵从的就把ip precedence设为0(尽力而为的传输)。

b) ftp流量遵从第 二个速率策略的ip precedence设置为5,如果ftp超出速率策略就扔包。

c) 其他剩余流量 限制到8m，普通突发大小为16000byte,最大突发大小为24000byte；遵从策略的流量设ip precedence为5,超出的流量扔包。

```
interface hssi0/0/0
desription 45mbps to r2
rate-limit out put accees-group 101 200000000 24000 32000 conform-action set
prec-transmit 5 exceed-action set-prec-transmit 0
rate-limit output access-group 102 10000000 24000 32000 conform-action set-prec-tranmit 5
exceed-action drop
rate-limit output 8000000 16000 24000 conform-action set-prec-transmit 5
exceed-action drop
ip address 10.1.0.9 255.255.255.0
!
access-li 101 per tcp any any eq www
access-li 102 per tcp any any eq ftp
```

match-ip-prefix

```
interface Serial1
ip address 10.0.0.1 255.255.255.252
rate-limit output access-group 1 20000000 24000 32000 conform-action transmit
exceed-action drop
!
access-list 1 permit 192.168.0.0 0.0.0.255
```

match-ip-priority

```
interface Serial1
ip address 10.0.0.1 255.255.255.252
rate-limit output access-group rate-limit 1 20000000 24000 32000 conform-action transmit
exceed-action drop
!
access-list rate-limit 1 3
```

match source mac-address

```
interface Fddi2/1/0
rate-limit input access-group rate-limit 100 80000000 64000 80000
conform-action transmit exceed-action drop ip address 200.200.6.1 255.255.255.0
!
access-list rate-limit 100 00e0.34b0.7777
```

policy-map in CAR

如上操作是针对接口设置的CAR，CAR同时也可以作用在policy-map上

Configuration

```
nimokaka(config-pmap-c)#police {CIR Bc Be} conform-action {action}exceed-action
{action}[violate-action {action}]
```

把rate-limit改成了police，后面增加了一个violate-action，违规操作，也就是超过了Be流量之后的操作 Action的操作命令

continue 继续执行下一条CAR 语句

drop	丢弃该数据包
set-prec-continue {precedence}	设置IP 优先级并继续执行下一条CAR 语句
set-prec-trasnmit {precedence}	设置IP 优先级并转发该数据包
set-dscp-continue {dscp}	设置IP DSCP 值并继续执行下一条CAR 语句
set-dscp-trasnmit {dscp}	设置IP DSCP 值并转发该数据包
set-qos-continue {group ID}	设置QoS 组ID 并继续执行下一条CAR 语句
set-qos-transmit {group ID}	设置QoS 组ID 并发送该数据包
transmit	转发该数据包

show command

1. 查看policy map: nimokaka#show policy-map [policy-name]

2. 查看接口的policy map信息: nimokaka#show policy-map interface [interface]

CASE

Case1

限制来自192.168.0.0/24的进站数据包的平均速率为8000bps，突发流量(Bc)为2000 字 节，额外突发流量(Be)为4000 字节。对突发流量和额外突发流量分别采取转发和设置QoS 组ID为25的策略;对违反突发流量和额外突发流量的数据流量采取丢弃的策略:

```
!
class-map match-all nimokaka match access-group 1
!
policy-map kiss class nimokaka
police 8000 2000 4000 conform-action transmit exceed-action set-qos-transmit 25
violate-action drop
!
interface Serial1
ip address 172.16.0.1 255.255.255.252
service-policy input kiss
!
access-list 1 permit 192.168.0.0 0.0.0.255
```

case 2

```
nimokaka(config)# class-map acgroup2
nimokaka(config-cmap)# match access-group 2
nimokaka(config-cmap)# exit
nimokaka(config)# policy-map police
nimokaka(config-pmap)# class acgroup2
nimokaka(config-pmap-c)# police 8000 2000 4000 conform-action transmit exceed-action
set-qos-transmit 4 violate-action drop
nimokaka(config-pmap-c)# exit
nimokaka(config-pmap)# exit
nimokaka(config)# interface fastethernet 0/0
nimokaka(config-if)# service-policy input police
```

traffic shaping

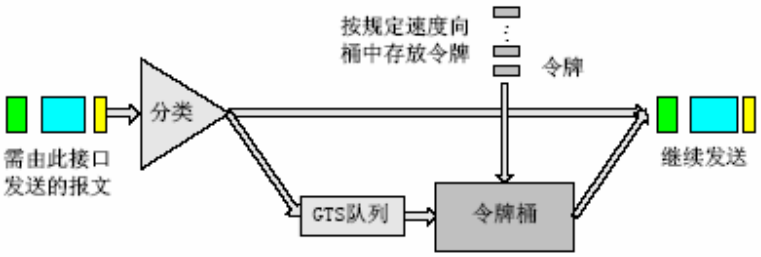
basic introduction

流量整形（traffic shaping）典型作用是限制流出某一网络的某一连接的流量与突发， 使这类报文以比较均匀的速度向外发送流量整形通常使用缓冲区和令牌桶来完成，当报文的发送速度过快时，首先在缓冲区进行缓存，在令牌桶的控制下再均匀地发送这些被缓冲的报文。

流量整形采用的技术叫做Generic Traffic Shaping（通用流量整形，简称GTS），可以对不规则或不符合预定流量特性的流量进行整形，以利于网络上下游之间的带宽匹配

GTS与CAR一样均采用了令牌桶技术来控制流量，GTS与CAR的主要区别在于：利用CAR进行报文流量控制时对不符合流量特性的报文进行丢弃，而GTS对于不符合流量特性的报文则 是进行缓冲减少了报文的丢弃，同时满足报文的流量特性。 GTS

的基本处理过程如下图所示，其中用于缓存报文的队列称为GTS队列



GTS可以对接口上指定的报文流或所有报文进行整形当报文到来的时候，首先对报文进 行分类如果报文不需要进行GTS处理，就继续发送不经过令牌桶的处理；如果报文需要进行 GTS处理，则与令牌桶中的令牌进行比较，令牌桶按用户设定的速度向桶中放置令牌，如果令牌桶中有足够的令牌可以用来发送报文，则报文直接被继续发送下去，同时令牌桶中的令牌量按报文的长度做相应的减少，当令牌桶中的令牌少到报文不能再发送时，报文将被缓存 入GTS队列中。当GTS队列中有报文的时候。GTS按一定的周期从队列中取出报文进行发送。 每次发送都会与令牌桶中的令牌数作比较。直到令牌桶中的令牌数减少到队列中的报文不能 再发送或是队列中的报文全部发送完毕为止.

例如路由器A和路由器B相连为了减少报文的丢失，可以在路由器A的出口对报文进行GTS 处理，对于超出GTS流量特性的报文将在路由器A中缓冲，当可以继续发送下一批报文时，GTS 再从缓冲队列中取出报文进行发送，这样发往路由器B的报文将都符合路由器B的流量规定， 从而减少报文在路由器B上的丢弃，相反如果不在路由器A的出口做GTS处理，则所有超出路由器2的CAR流量特性的报文将被路由器B丢弃。

Configuration

配置流量整形GTS有两种方式，一种是基本的GTS

启用GTS：

```
nimokaka(config-if)#traffic-shape rate {CIR [Bc [Be]]}
```

一种是基于ACL 的GTS，这是可选配置：

```
nimokaka(config-if)#traffic-shape group {ACL} {CIR [Bc [Be]]}
```

show command

1. 查看GTS 的配置信息： `nimokaka#show traffic-shape [interface]`
2. 查看GTS 的统计信息： `nimokaka#show traffic-shape statistics [interface]`

case

```
access-list 101 permit udp any any interface Ethernet0
traffic-shape group 101 1000000 125000 125000          //对于udp的数据，限制其流量稳定在5M
!
interface Ethernet1
traffic-shape rate 5000000 625000 625000               //整体接口流量保持在5M
```

GTS over FR

1. 启用GTS:

```
nimokaka(config-if)#traffic-shape rate {CIR [Bc [Be]]}
```

2. 当接口收到向后显性拥塞通知(BECN)时，估算流量速率的最低值:

```
nimokaka(config-if)#traffic-shape adaptive {CIR}
```

3. 以向前显性拥塞通知(FECN)做为BECN 的响应. 可选:

```
nimokaka(config-if)#traffic-shape fecn-adapt
```

case

限制接口流量传输速率的上限为128kbps，下限为64kbps，并以FECN 做为BECN 的响应:

```
interface Serial1.1 point-to-point
ip address 172.16.0.1 255.255.255.252 traffic-shape rate 128000 7936 1000 traffic-shape
```

```
adaptive 64000
traffic-shape fecn-adapt
```

GTS over policy map

GTS也可以应用在policy map上，基于分类的流量 整形可以启用在支持GTS 的任何接口上。基于分类的流量整形可以打破普通GTS仅仅以ACL 分类的限制。它还可以定义平均值和峰值的流量整形。并且可以在配置GTS 的时候采用 CBWFQ。

Configuration

配置基于分类的流量整形的步骤如下：

1、定义平均值和峰值的CIR,Bc 和Be:

```
nimokaka(config-pmap-c)#shape {average|peak} {CIR [Bc] [Be]}
```

average指的是平均值，peak说得是峰值

2.定义缓冲区上限,默认值为1000.可选:

```
nimokaka(config-pmap-c)#shape max-buffers {number-of-buffers}
```

3、在策略上应用CBWFQ。可选:

```
nimokaka(config-if)#service-policy output {policy-name}
```

show command

1、查看基于分类的流量整形的配置信息:

```
nimokaka#show traffic-shape [interface]
```

2、查看基于分类的流量整形的统计信息:

```
nimokaka#show traffic-shape statistics [interface]
```

3、查看policy map:

```
nimokaka#show policy-map [policy-name]
```

4、查看接口的policy map 信息:

```
nimokaka#show policy-map interface [interface]
```

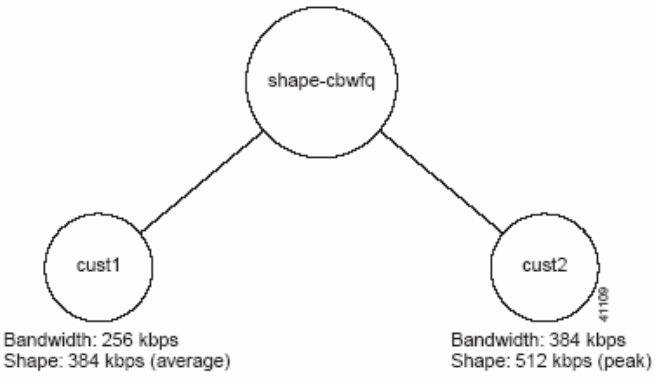
Case

简单的分类GTS

```
nimokaka(config)# policy-map shape nimokaka(config-pmap)# class c1
nimokaka(config-pmap-c)# shape average 38400 15440
nimokaka(config)# interface Serial 3/3
nimokaka(config-if)# service out shape
```

流量整形平均速率稳定在38400字节，突发流量15440字节

GTS和CBWFQ关联使用

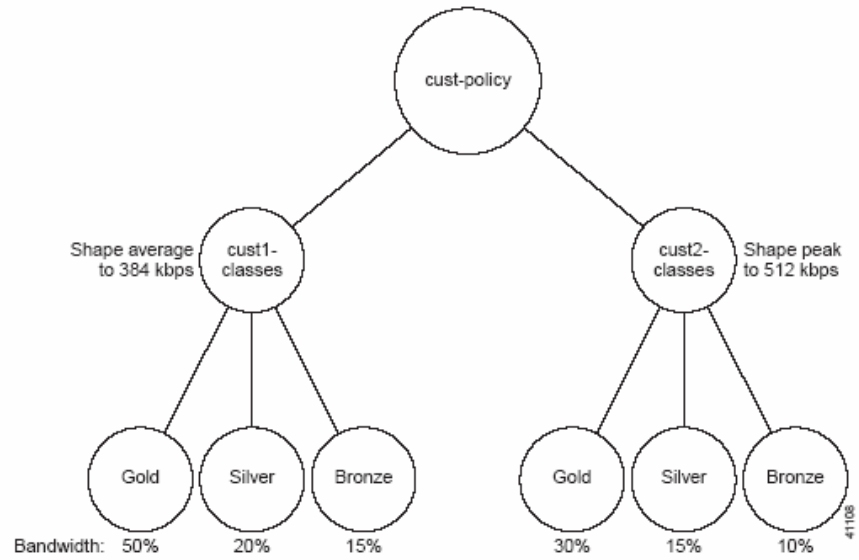



```
nimokaka(config)# policy-map shape-cbwfq
nimokaka(config-pmap)# class cust1
nimokaka(config-pmap-c)# shape average 384000
```

```
nimokaka(config-pmap-c)# bandwidth 256
nimokaka(config-pmap)# class cust2
nimokaka(config-pmap-c)# shape peak 512000
nimokaka(config-pmap-c)# bandwidth 384
nimokaka(config-pmap-c)# configure terminal
nimokaka(config)# interface Serial 3/3
nimokaka(config-if)# service out shape-cbwfq
```

上面的案例中，cust1和cust2是两个队列，当出现拥塞时，使用CBWFQ使其带宽分别限制带宽为256k和384k，但是如果链路的带宽足够用，可以通过GTS把cust1的平均带宽整形为384k，同时也把cust2的最大带宽限定在384k，来保证他们两个队列无休止的占用流量。

GTS和policy map的复杂结合




```
cust1-classes Configuration
nimokaka(config)# policy-map cust1-classes
nimokaka(config-pmap)# class gold
nimokaka(config-pmap-c)# bandwidth percent 50
nimokaka(config-pmap)# class silver
nimokaka(config-pmap-c)# bandwidth percent 20
nimokaka(config-pmap)# class bronze
nimokaka(config-pmap-c)# bandwidth percent 15
cust2-classes Configuration
nimokaka(config)# policy-map cust2-classes
nimokaka(config-pmap)# class gold
nimokaka(config-pmap-c)# bandwidth percent 30
nimokaka(config-pmap)# class silver
nimokaka(config-pmap-c)# bandwidth percent 15
nimokaka(config-pmap)# class bronze
nimokaka(config-pmap-c)# bandwidth percent 10
Customer Policy and QoS Features Configuration

nimokaka(config)# policy-map cust-policy
nimokaka(config-pmap)# class cust1
nimokaka(config-pmap-c)# shape average 384000
nimokaka(config-pmap-c)# service-policy cust1-classes
nimokaka(config-pmap)# class cust2
nimokaka(config-pmap-c)# shape peak 512000
nimokaka(config-pmap-c)# service-policy cust2-classes
nimokaka(config-pmap-c)# interface Serial 3/2
nimokaka(config-if)# service out cust-policy
```

其实这就是一个policy map的调用。首先需要注意的是，创建了两个子policy map，里面针对于不同的队列做了带宽 的

调整，然后创建了一个母policy map，名称是cust-policy，对这两个子policy做了流量整形

Frame Relay Traffic Shaping

FR中的FECN和BECN用于暗示网络上发生了拥塞，当收到带有BECN标记的数据包时，FR 流量整形(FRTS)将动态的对流量进行整形。注意：FRTS只能使用在FR的PVC和SVC上。

Configuration

配置FRTS 的步骤如下：

1、启用FRTS：

nimokaka(config-if)#frame-relay traffic-shaping

2、全局定义map class。当定义了map class 之后，所有VC将继承该map class的FRTS参数

nimokaka(config)#map-class frame-relay {name}

3、基于接口的定义map class。当基于接口的定义了map class之后，所有该接口的子接口的VC将继承该map class的FRTS参数。可选：

nimokaka(config-if)#frame-relay class {name}

4、定义该map class的速率的平均值和峰值：

nimokaka(config-map-class)#frame-relay traffic-rate {average [peak]}

5、定义CIR,Bc和Be，如果不指定方向，则定义为双向。可选：

nimokaka(config-map-class)#frame-relay {cir [in|out] CIR|bc [in|out] Bc|be [in|out] Be}

6、定义CIR 的最低值。可选：

nimokaka(config-map-class)#frame-relay mincir [in|out] {min-CIR}

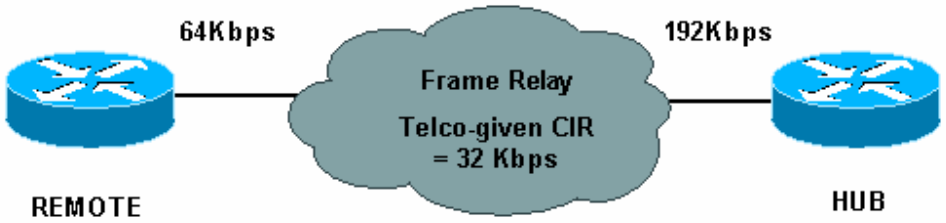
7、定义以BECN 做为拥塞通知符。可选：

nimokaka(config-map-class)#frame-relay adaptive-shaping becn

8、定义CQ 列表和PQ 列表。可选：

nimokaka(config-map-class)#frame-relay {custom-queue-list|priority-group} {list}

Case



HUB - access rate = 192 Kbps, guaranteed rate = 32Kbps

REMOTE - access rate = 64Kbps, guaranteed rate = 32Kbps

HUB 配置

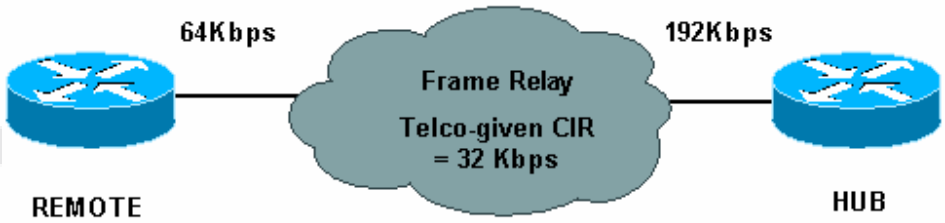
```
interface Serial0/0
 no ip address encapsulation frame-relay no fair-queue
frame-relay traffic-shaping

!— Apply traffic shaping to main interface
interface Serial0/0.1 point-to-point ip address 10.1.1.1 255.255.255.0 frame-relay interface-dlci 16
frame-relay class cisco

!— Apply map class to the DLCI / subinterface
!

!— Configure map class parameters
```

```
map-class frame-relay cisco frame-relay cir 64000
frame-relay mincir 32000
frame-relay adaptive-shaping becn frame-relay bc 8000
frame-relay be 16000
!
```

Remote配置

```
interface Serial0/0
no ip address encapsulation frame-relay no fair-queue
frame-relay traffic-shaping
!
interface Serial0/0.1 point-to-point ip address 10.1.1.2 255.255.255.0 frame-relay interface-dlci 16
frame-relay class cisco
!
map-class frame-relay cisco frame-relay cir 64000
frame-relay mincir 32000
frame-relay adaptive-shaping becn frame-relay bc 8000
```

GTS and CBWFQ over FR



HUB - physical rate = 192 Kbps, guaranteed rate = 32 Kbps

REMOTE - physical rate = 64 Kbps, guaranteed rate = 32 Kbps

Hub 配置

```
!
class-map match-all YYY
match access-group 101
!
!
policy-map ZZZ
class YYY
bandwidth percent 50
REMOTE
interface Serial0/0
no ip address encapsulation frame-relay no fair-queue
frame-relay traffic-shaping


interface Serial0/0.1 point-to-point ip address 10.1.1.1 255.255.255.0 frame-relay interface-dlci 16
frame-relay class XXX
!
map-class frame-relay XXX frame-relay cir 64000 frame-relay mincir 32000
frame-relay adaptive-shaping becn frame-relay bc 8000
service-policy output ZZZ
```

REMOTE 配置

```
!
access-list 101 permit ip host 10.0.0.1 host 11.0.0.1
interface Serial0/0

no ip address encapsulation frame-relay no fair-queue
frame-relay traffic-shaping
```

```
!
interface Serial0/0.1 point-to-point ip address 10.1.1.2 255.255.255.0 frame-relay interface-dlci 16
frame-relay class XXX
!
map-class frame-relay XXX frame-relay cir 64000 frame-relay mincir 32000
frame-relay adaptive-shaping becn frame-relay bc 8000
```

RSVP

SVP是第一个在大型网络中动态建立端到端QoS服务模型的工业标准的信令协议。RSVP 运行在IP之上,它可以让应用程序在网络上预留带宽。主机和路由器使用RSVP沿着数据流传 输的线路进行传输相应的QoS请求信息，比如带宽和延迟。RSVP本身不实行路由决策，相反带宽预留的请求由下层的路由协议执行，因此RSVP 能够适应网络拓扑的变化。RSVP的操作对于不支持RSVP的路由器是透明的。RSVP和现有的一些队列机制协同工作,而不是替代现有的队列机制，并且RSVP 支持组播，RSVP 目前通常为组播应用程序，比如视频会议,进行资源的预留。

主机使用RSVP请求特定的QoS服务来为它的应用程序预留带宽。只要带宽足够，应用程 序能够以超过请求预留带宽的速率进行数据的传输;如果带宽不足,那么这些超过请求预留 带宽的部分将被丢弃。网络资源预留的要求对于数据流量和实时传输流量是不同的,前者对 资源预留的要求很小；后者反之。

资源预留和队列机制的结合使用两个关键点：

1、端到端的RSVP数据流:数据流从单一或多个源地址向单一或多个目标地址进行单向传输。

2、路由器到路由器的WFQ会话：穿越特定接口的单一传输层会话或网络层数据流，WFQ会话通过源地址和目标地址，端口号或协议号等属性进行区分。

Configuration

配置RSVP步骤如下：

1、启用RSVP,默认带宽预留上限为接口带宽的75%。可以指定RSVP数据流带宽总量，也可以指定每个RSVP数据流的带宽：

```
nimokaka(config-if)#ip rsvp bandwidth [interface-kbps [single-flow-kbps]]
```

2、指定只接收符合特定条件的邻居路由器的RSVP请求。可选：

```
nimokaka(config)#ip rsvp neighbor {ACL}
```

3、对于符合RSVP所定义的带宽和超出RSVP所定义的带宽的数据包分配IP优先级。可选：

```
nimokaka(config-if)#ip rsvp precedence {[conform precedence] [exceed precedence]}
```

show command

1、允许远程管理工作站监视RSVP相关的信息：

```
nimokaka(config)#snmp-server enable traps rsvp
```

2、显示接口的RSVP信息：

```
nimokaka#show ip rsvp interface [interface]
```

3、显示接口的RSVP过滤和带宽信息：

```
nimokaka#show ip rsvp installed [interface]
```

4、显示当前的RSVP邻居信息：

```
nimokaka#show ip rsvp neighbor [interface]
```

5、显示RSVP发送方,接收方以及请求信息：

```
nimokaka#show ip rsvp {sender|reservation|request} [interface]
```

HSRP

Basic Introduction

随着Internet的日益普及，人们对网络的依赖性也越来越强。这同时对网络的稳定性提出了更高的要求，人们自然想到了基于设备的备份结构，就像在服务器中为提高数据的安全性而采用双硬盘结构一样。 路由器是整个网络的核心和心脏，如果路由器发生致命性的故障，将导致本地网络的瘫痪，如果是骨干路由器，影响的范围将更大，所造成的损失也是难以估计的。因此，对路由器采用热备份是提高网络可靠性的必然选择。在一个路由器完全不能工作的情况下，它的全部功能便被系统中的

另一个备份路由器完全接管，直至出现问题的路由器恢复正常，这就是热备份路由协议（HotStandbyRouterProtocal），HSRP RFC2281技术要解决的问题。

冗余性网络中的路由问题

1.缺省网关：当设置默认的网关失效后，数据就不能到别的网段；即使存在作为网关使用的冗余路由器，也不能动态的方法将这些设备切换到新的网关地址。

2.代理ARP：如果要获得切换路由器的MAC地址，源端工作站必须要么发起另一个ARP请求，要么重新启动，那么就可能在一段时间通信无法实现。导致PC机的ARP缓存可能有几个小时不能刷新ARP缓存。

注：在win95、98环境下使用代理ARP，缺省网关必须设置为该主机设备自己IP地址

3.路由选择协议：采用RIP协议，如果源工作站被配置为使用RIP，那么在RIP可以选用另一路由器之前，就会浪费很长的时间去更新。

4.IRDP （ICMP Router Discovery Protocol）

①. 用icmp，每一个路由每7分钟发router advertisment， host就可以自动获得gateway

②. 有支持IRDP的PC，开机发处请求

HSRP协议概述

实现HSRP的条件是系统中有多台路由器，它们组成一个“热备份组”，这个组形成一个虚拟路由器。在任一时刻，一个组内只有一个路由器是活动的，并由它来转发数据包，如果活动路由器发生了故障，将选择一个备份路由器来替代活动路由器，但是在本网络内的主机看来，虚拟路由器没有改变。所以主机仍然保持连接，没有受到故障的影响，这样就较好地解决了路由器切换的问题。

为了减少网络的数据流量，在设置完活动路由器和备份路由器之后，只有活动路由器和备份路由器定时发送HSRP报文。如果活动路由器失效，备份路由器将接管成为活动路由器。如果备份路由器失效或者变成了活动路由器，将有另外的路由器被选为备份路由器。

在实际的一个特定的局域网中，可能有多个热备份组并存或重叠。每个热备份组模仿一个虚拟路由器工作，它有一个Well－known－MAC地址和一个IP地址。该IP地址、组内路由器的接口地址、主机在同一个子网内，但是不能一样。当在一个局域网上有多个热备份组存在时，把主机分布到不同的热备份组，可以使负载得到分担。

HSRP的工作原理

HSRP协议利用一个优先级方案来决定哪个配置了HSRP协议的路由器成为默认的主动路由器。如果一个路由器的优先级设置的比所有其他路由器的优先级高，则该路由器成为主动路由器。路由器的缺省优先级是100，所以如果只设置一个路由器的优先级高于100，则该路由器将成为主动路由器。

通过在设置了HSRP协议的路由器之间广播HSRP优先级，HSRP协议选出当前的主动路由器。当在预先设定的一段时间内主动路由器不能发送hello消息时，优先级最高的备用路由器变为主动路由器。路由器之间的包传输对网络上的所有主机来说都是透明的。

配置了HSRP协议的路由器交换以下三种多点广播消息：

Hello——— hello消息通知其他路由器发送路由器的HSRP优先级和状态信息，HSRP路由器默认为每3秒钟发送一个hello消息；

Coup——— 当一个备用路由器变为一个主动路由器时发送一个coup消息；

Resign——— 当主动路由器要宕机或者当有优先级更高的路由器发送hello消息时，主动路由器发送一个resign消息。

在任一时刻，配置了HSRP协议的路由器都将处于以下六种状态之一：

Initial—— HSRP启动时的状态,HSRP还没有运行,一般是在改变配置或端口刚刚启动时进入该状态。

learn——— 路由器已经得到了虚拟IP地址，但是它既不是活动路由器也不是等待路由器。

它一直监听从活动路由器和等待路由器发来的HELLO报文。

Listen————路由器正在监听hello消息。

Speak————在该状态下，路由器定期发送HELLO报文，并且积极参加活动路由器或等待路由器的竞选。

Standby————当主动路由器失效时路由器准备接包传输功能。

Active————路由器执行包传输功能。

使用HSRP时，当末端工作站使用的缺省网关不可用时，可以继续在网络上进行通信。

1. 在HSRP中，是采用有一套的路由器，分为活跃、备份、虚拟、其他路由器等体系，在外部看来，它是一台拥有IP和MAC地址的目标路由器。
2. 活跃路由器的功能是负责转发发送到虚拟路由器的数据。它通过发送HELLO消息（基于UDP广播）来通告它的活跃状态
3. 组中会有另外的一台路由器来作为备份路由器。它的功能是监视HSRP组中的运行状态，并且在当前活跃路由器不可用时，迅速承担起负责数据转发的任务。备份路由器也发送HELLO消息来通告组中其他的路由器它备份路由器的角色。
4. 虚拟路由器的功能是向最终的用户来代表一台能持续工作的路由器设备。它有自己的MAC和IP地址。但是实际上它是不用来转发数据包，它的作用仅仅是代表一台可用的路由设备。
5. 其他路由器也监听HELLO消息，但是不作应答，这样它就不会在备份组有身份的概念，同时它也不参与发送到虚拟路由器的数据包，但是还是转发其他路由器发来的数据包。

注：在每个VLAN子网配置一个单独的HSRP组。

活跃路由器的竞争机制

1. 默认状态的情况下，MAC地址最小的路由器将成为活跃的路由器。
2. 当活跃路由器失效的情况下，备份路由器将成为活跃路由器。但是如果在活跃和备份路由器都失效的情况

下，路由器身份的竞争依照如下的原则

①. 在HSRP中有最高备份优先级的路由器将成为活跃/备份路由器，缺省的优先级是100，优先级从0～255。

②. 当优先级相同的情况下，具有最高IP地址的路由器将成为活跃的路由器

③. 默认还是MAC地址最小的成为活跃路由器

Configuration

1. 设置路由器的优先级：优先级最高的将成为活跃/备份路由器,缺省为100

Router (config-if) standby group-number priority priority-value

2. 设置HSRP备份占先权：它应用在当活跃路由器失效或是从服务中移除时，会导致重新选主新的活跃路由器。但是即使原来具有更高的优先级、原来活跃的路由器重新恢复工作时，它还是不能恢复它活跃路由器的身份。如果要重新恢复它的活跃路由器的身份，就可以在此路由器上进行如下的设置

Router (config-if) standby group-number preempt

HSRP的运行

前面讲过在HSRP中各种路由器的身份及其它们的职责。活跃路由器对发送到虚拟路由器的数据流进行响应。即当一个末端站点发送一个数据包到虚拟路由器的MAC地址，那么由活跃路由器来处理这个数据包。如果一个末端站点发送一个ARP请求，这时活跃路由器用虚拟路由器的MAC地址来进行响应。

注意：在运行HSRP的同时，很重要的一点就是不能让最终用户知道路由器的实际MAC地址（活跃路由器地址），因此任何可能将路由器的实际的MAC地址告诉最终用户的协议都将被关闭。要确保实际路由器的MAC地址不被发现，启用了HSRP的路由器将会自动关闭ICMP重定向功能。

定位虚拟路由器的MAC地址的方法：

虚拟路由器的MAC地址由三部份组成：

厂商ID ＋ HSRP编码（固定为07.ac） + 组ID（换成十六进制，如47：2f）

可以用**show ip arp**来显示ARP缓存结果。

HSRP Configuration

1. 配置一个接口以参与HSRP备份组

Router (config –if)#standby group-number ip virtual-ip-address

如：

```
Interface vlan10
ip address 172.16.10.88 255.255.255.0
no ip redirects                          (关闭ICMP从定向)
Standby 47 ip 172.16.10.11
```

2. 配置HSRP的备份优先级，前面已经讲过。

Router (config –if) Standby group-number priority priority-number

3. 配置HSRP组中的备份占先权

Router (config –if) standby group-number preempt

4. 配置HELLO消息计时器：前面讲过活跃路由器和备份路由器通过发送HELLO消息来说明它现在的身份。HELLO消息当中包括了路由器的优先级、HELLO时间、保持时间等（默认的保持时间是HELLO消息的3倍）。

Router (config)#standby group-number timers hellotime holdtime

①. Hellotime:是以秒来计算的，来定义HELLO消息之间的间隔;默认是3秒

②. Holdtime:是以秒来计算的，来定义活跃或是备份路由器在宣布失效之前的时间，从1～255，缺省为10秒。也就是是说：路由器在3个HELLO消息的间隔内如果还是没有收到HELLO包，宣告现在这台路由器已经失效了。

要恢复到缺省的备份计时器值：**no standby group timers**命令。

HSRP接口跟踪：配置HSRP的功能是为了解决如下的问题：

在某些情况下，端口的状态直接的影响了路由器的身份即哪台路由器要变成活跃的路由器。尤其是在每台路由器有到某个目的地的不同路径时。因为当启用了HSRP的路由器的端口是关闭了端口的ICMP的重定向的功能的，因此，当有链路发生了错误时，活跃路由器将不会对数据包进行重定向，导致了数据包的人为不可到达。

通过启用接口跟踪时,可以根据接口的状态自动的调整路由器的优先级,当被跟踪的端口变成不可用时,将自动的降低其路由器的优先级。这样就降低了它继续成为活跃路由器的可能性,

接口配置的模式下作以下的配置：

Router(config-if)# standby group-number track type number interface-priority

type: 跟踪的接口的类型，与接口号一起使用如s0

number: 被跟踪的接口的号，与接口类型一起使用

interface-priority: 路由器要被降低的优先级的值，缺省为10

要关闭此功能的话使用：**no standby group track**命令。

在干道上配置HSRP

此项功能能通过在子网和VLAN间提供负载均衡和冗余能力提高整个网络的韧性。

我们可以实现通过两个路由器来实现在干道上的互为活跃/备份路由器。我们只要设置一下他们在HSRP中的优先级就可以实现。

以下有个简单的例子：

对router A

Interface f1/1.10

```
Encapsulation isl 10
Ip address 172.16.10.2 255.255.255.0
Standby 1 ip 172.16.10.110
Standby 1 priority 105
Standby 1 preempt
Interface f1/1.20
Encapsulation isl 20
ip address 172.16.20.2 255.255.255.0
Standby 2 ip 172.16.20.110
Standby 2 priority 50
```

```
Interface f1/1.10
Encapsulation isl 10
ip address 172.16.10.3 255.255.255.0
Standby 1 ip 172.16.10.110
Standby 1 priority 50
Interface f1/1.20
Encapsulation isl 20
Ip address 172.16.20.3 255.255.255.0
Standby 2 ip 172.16.20.110
Standby 2 priority 105
Standby 1 preempt
```

对router B

Show HSRP

要显示HSRP路由器的状态，可以在特权的状态下执行如下命令即可

router#show standby type-number group brief

type-number：说明要显示目标接口的类型和序号

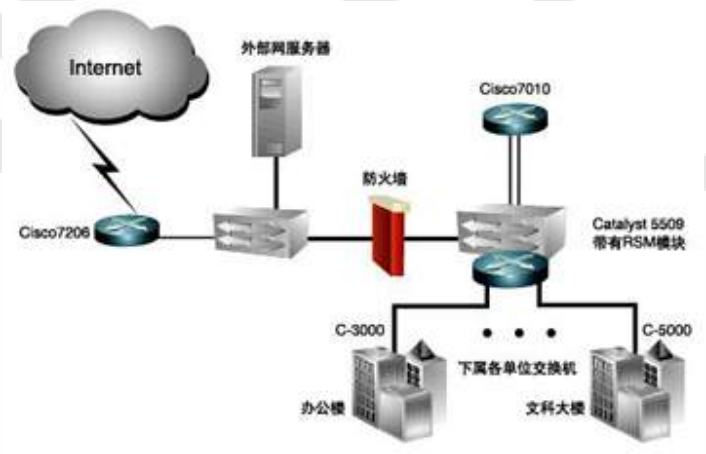
group：显示接口上某个具体的HSRP组

brief：每个备份组总结成一行来输出

如：Show standby vlan10 47/show standby brief/show standby.

HSRP Case

某校园网规模比较大，上网的主机相对比较多，共分配有16个C类地址。为了保证数据安全和广播风暴，提高网络性能，将校园网划分成60个子网。在网络中心采用Cisco系统公司的Catalyst 5509作为中心交换机，并且带有RSM作为VLAN间的路由器，另外使用一个Cisco 7000系列的路由器和RSM。它们都支持VLAN以及VLAN上的HSRP。如图所示。



在每一个虚拟局域网内都有一个HSRP组，从逻辑上讲，Cisco 7010和Cisco 5509的RSM在每个虚拟局域网上都有局域网接口，并且都配置有IP地址，同时配置一个虚拟地址，该地址作为在该虚拟局域网内所有主机的网关。下面以VLAN 9为例，RSM中VLAN 9的配置如下：

---- interface Vlan9

---- description surportcenter
---- ip address 202.120.95.66 255.255.255.224
该路由器在该VLAN9上的接口的IP地址以及掩码
no ip redirects
no ip directed-broadcast
no ip route-cache cef
standby 9 timers 3 250
定义热等待组号为9，每3秒交换一次hello信息，250没有收到hello信息就开切换
standby 150 priority 110
定义路由器的权值，值越大，成为活动路由器的希望越大
standby 9 preempt
Enable该组的HSRP抢占功能，谁的权值大就可以立即成为活动路由器
standby 9 ip 202.120.95.65
该组的虚拟IP地址，作为该VLAN中主机的网关地址
Cisco 7010路由器中接口的配置如下:
interface FastEthernet0/0.9
description surportcenter
ip address 202.120.95.67 255.255.255.224
cisco7010在VLAN9上的接口的IP地址以及掩码，该地址和RSM中的地址必须属于同一个子网，并且不同
no ip redirects
encapsulation isl 9
所使用的虚拟局域网协议
standby 9 timers 3 250
和在RSM中的含义一样，并且必须相同
standby 9 priority 100
比在RSM中的值小，所以RSM在该VLAN中为活动的
standby 9 preempt
和在RSM中含义一样
standby 9 ip 202.120.95.65
该组的虚拟IP地址，必须和RSM中一样

为了达到负载均衡的目的，应该使Cisco 5509 RSM和Cisco 7010承担大致相同的负载，我们的方法是，在RSM中，VLAN 1到VLAN 30的权值为110，VLAN 31到VLAN 60的权值为100；相反，在Cisco 7010中，VLAN 1到VLAN 30的权值为100，VLAN 31到VLAN 60的权值为100。这样，在正常情况下，Cisco 5509的RSM负责VLAN 1到VLAN 30的路由，Cisco 7010负责VLAN 31到VLAN 60的路由。如果有一方出现了故障，将由另一个来负载全部的路由工作。

HSRP存在的问题

对于在HSRP协议，最大的问题是没有提供安全防护，在一个局域网内部，通过发送虚假的UDP多播数据包很容易对局域网中的路由器实施攻击，导致数据包黑洞(Packet Black Hole)和拒绝服务攻击(Denial-of-Service Attack)。一般无法从一个局域网的外部实施攻击，因为大多数路由器都不转发目的地址为所有路由器的多播地址(224.0.0.2)。

HSRP只是实现了路由器的平滑切换，使用户感觉不到这种切换，保证了网络的稳定性。但是，一个HSRP组内的路由器不能互通它们的其他网络配置信息，例如访问控制列表等。所以在管理实施管理时，为了保证一致性，必须对它们进行相同的修改，增加了管理的复杂性，这也许是为了提高性能而付出的代价吧。

VRRP

Basic introduction

在VRRP协议中，有两组重要的概念：VRRP路由器和虚拟路由器，主控路由器和备份路由器。VRRP路由器是指运行VRRP的路由器，是物理实体，虚拟路由器是指VRRP协议创建的，是逻辑概念。一组VRRP路由器协同工作，共同构成一台虚拟路由器。该虚拟路由器对外表现为一个具有唯一固定IP地址和MAC地址的逻辑路由器。处于同一个VRRP组中的路由器具有两种互斥的角色：主控路由器和备份路由器，一个VRRP组中有且只有一台处于主控角色的路由器，可以有一个或者多个处于备

份角色的路由器。VRRP协议使用选择策略从路由器组中选出一台作为主控，负责ARP相应和转发IP数据包，组中的其它路由器作为备份的角色处于待命状态。当由于某种原因主控路由器发生故障时，备份路由器能在几秒钟的时延后升级为主路由器。由于此切换非常迅速而且不用改变IP地址和MAC地址，故对终端使用者系统是透明的。

工作原理

一个VRRP路由器有唯一的标识：VRID，范围为0—255。该路由器对外表现为唯一的虚拟MAC地址，地址的格式为00-00-5E-00-01-[VRID]。主控路由器负责对ARP请求用该MAC地址做应答。这样，无论如何切换，保证给终端设备的是唯一一致的IP和MAC地址，减少了切换对终端设备的影响。

VRRP控制报文只有一种：VRRP通告(advertisement)。它使用IP多播数据包进行封装，组地址为224.0.0.18，发布范围只限于同一局域网内。这保证了VRID在不同网络中可以重复使用。为了减少网络带宽消耗只有主控路由器才可以周期性的发送VRRP通告报文。备份路由器在连续三个通告间隔内收不到VRRP或收到优先级为0的通告后启动新的一轮VRRP选举。

在VRRP路由器组中，按优先级选举主控路由器，VRRP协议中优先级范围是0—255。若VRRP路由器的IP地址和虚拟路由器的接口IP地址相同，则称该虚拟路由器作VRRP组中的IP地址所有者；IP地址所有者自动具有最高优先级：255。优先级0一般用在IP地址所有者主动放弃主控者角色时使用。可配置的优先级范围为1—254。优先级的配置原则可以依据链路的速度和成本、路由器性能和可靠性以及其它管理策略设定。主控路由器的选举中，高优先级的虚拟路由器获胜，因此，如果在VRRP组中有IP地址所有者，则它总是作为主控路由的角色出现。对于相同优先级的候选路由器，按照IP地址大小顺序选举。VRRP还提供了优先级抢占策略，如果配置了该策略，高优先级的备份路由器便会剥夺当前低优先级的主控路由器而成为新的主控路由器。

为了保证VRRP协议的安全性，提供了两种安全认证措施：明文认证和IP头认证。明文认证方式要求：在加入一个VRRP路由器组时，必须同时提供相同的VRID和明文密码。适合于避免在局域网内的配置错误，但不能防止通过网络监听方式获得密码。IP头认证的方式提供了更高的安全性，能够防止报文重放和修改等攻击。

应用实例

最典型的VRRP应用：RTA、RTB组成一个VRRP路由器组，假设RTB的处理能力高于RTA，则将RTB配置成IP地址所有者，H1、H2、H3的默认网关设定为RTB。则RTB成为主控路由器，负责ICMP重定向、ARP应答和IP报文的转发；一旦RTB失败，RTA立即启动切换，成为主控，从而保证了对客户透明的安全切换。

在VRRP应用中，RTA在线时RTB只是作为后备，不参与转发工作，闲置了路由器RTA和链路L1。通过合理的网络设计，可以到达备份和负载分担双重效果。让RTA、RTB同时属于互为备份的两个VRRP组：在组1中RTA为IP地址所有者；组2中RTB为IP地址所有者。将H1的默认网关设定为RTA；H2、H3的默认网关设定为RTB。这样，既分担了设备负载和网络流量，又提高了网络可靠性。

HSRP and VRRP

1. 在功能上,VRRP和HSRP非常相似,但是就安全而言,VRRP对HSRP的一个主要优势:它允许参与VRRP组的设备间建立认证机制.并且,不像HSRP那样要求虚拟路由器不能是其中一个路由器的ip地址,但是VRRP允许这种情况发生(如果"拥有"虚拟路由器地址的路由器被建立并且正在运行,那么应该总是由这个虚拟路由器管理—等价于HSRP中的活动路由器),但是为了确保万一失效发生的时候终端主机不必重新学习MAC地址，它指定使用的MAC地址00-00-5e-00-01-VRID,这里的VRID是虚拟路由器的ID(等价于一个HSRP的组标识符).

2. 另外一个不同是VRRP不使用HSRP中的政变或者一个等价消息,VRRP的状态机比HSRP的要简单,HSRP有6个状态(初始(Initial)状态，学习(Learn)状态，监听(Listen)状态，对话(Speak)状态，备份(Standby)状态，活动(Active)状态)和8个事件, VRRP只有3个状态(初始状态(Initialize)、主状态(Master)、备份状态(Backup))和5个事件.

3. HSRP有三种报文，而且有三种状态可以发送报文 呼叫(Hello)报文 告辞(Resign)报文 突变(Coup)报文.

VRRP有一种报文

VRRP广播报文:由主路由器定时发出来通告它的存在，

使用这些报文可以检测虚拟路由器各种参数，还可以用于主路由器的选举。

4. HSRP将报文承载在UDP报文上，而VRRP承载在TCP报文上(HSRP 使用UDP 1985端口，向组播地址224.0.0.2 发送hello消息。)

5.VRRP的安全:VRRP协议包括三种主要的认证方式:无认证,简单的明文密码和使用 MD5 HMAC ip认证的强认证.

强认证方法使用IP认证头(AH)协议.AH是与用在IPSEC中相同的协议,AH为认证VRRP分组中的内容和分组头提供了一个方法. MD5 HMAC 的使用表明使用一个共享的密钥用于产生hash值.路由器发送一个VRRP分组产生MD5 hash值,并将它置于要发送的通告中,在接收时,接受方使用相同的密钥和MD5值,重新计算分组内容和分组头的hash值,如果结果相同,这个消息就是真正来自于一个可信赖的主机,如果不相同,它必须丢弃,这可以防止攻击者通过访问LAN而发出能影响选择过程的通告消息或者其他一些方法中断网络.

另外,VRRP包括一个保护VRRP分组不会被另外一个远程网络添加内容的机制(设置TTL值=255,并在接受时检查),这限制了可以进行本地攻击的大部分缺陷.而另一方面,HSRP在它的消息中使用的TTL值是1.

6.VRRP的崩溃间隔时间:3*通告间隔+时滞时间(skew-time)

Configuration

```
Router(config-if)#vrrp {group-number} ip virtual_IP_addr
Router(config-if)#vrrp {group-number} priority priority_value
```

查看vrrp配置

```
Show vrrp interface vlan 1
```

GLBP

Introduction

全称Gateway Load Banancing Protocol，和HRSP、VRRP不同的是，GLBP不仅提供冗余网关，还在各网关之间提供负载均衡，而HRSP、VRRP都必须选定一个活动路由器，而备用路由器则处于闲置状态。和HRSP不同的是，GLBP可以绑定多个MAC地址到虚拟IP，从而允许客户端选择不同的路由器作为其默认网关，而网关地址仍使用相同的虚拟IP，从而实现一定的冗余。

Choose AVG

使用类似于HRSP的机制选举活动网关，优先级最高的路由器成为活动落由器，称作Acitve Virtual Gateway，其他非AVG提供冗余。某路由器被推举为AVG后，和HRSP不同的工作开始了，AVG分配虚拟的MAC地址给其他GLBP组成员。所有的GLBP组中的路由器都转发包，但是各路由器只负责转发与自己的虚拟MAC地址的相关的数据包。

地址分配

每个GLBP组中最多有4个虚拟MAC地址，非AVG路由器有AVG按序分配虚拟MAC地址，非AVG也被称作Active Virtual Forwarder(AVF)。AVF分为两类：Primary Virtual Forwarder和Secondary Virtual Forwarder。直接由AVG分配虚拟MAC地址的路由器被称作Primary Virtual Forwarder，后续不知道AVG真实IP地址的组成员，只能使用hellos包来识别其身份，然后被分配虚拟MAC地址，此类被称作Secondary Virtual Forwarder。

LBP配置

如果AVG失效，则推举就会发生，决定哪个AVF替代AVG来分配MAC地址，推举机制依赖于优先级。最多可以配置1024个GLBP组，不同的用户组可以配置成使用不同的组AVG来作为其网关。

```
router#conf t
router(config)#int fastethernet 0/0
router(config-if)#ip address 10.1.1.1
router(config-if)#glbp 99 ip 10.1.1.254
```

```
router(config-if)#glbp 99 priority 105
router(config-if)#glbp 99 preempt delay 10
router(config-if)#glbp 99 weighting track int s0 10
router(config-if)#exit
router(config)#^Z
```

Cisco-SLB

Introduction

Cisco-SLB(Cisco Server Load Balancing)，用于提高冗余和提高性能，其实质是将服务器封装在一个虚拟ip中，对外界，仅显示为1台主机，实际上会有很多台进行负载均衡处理

Server Farms

Step 1 Name the server farm:

```
Switch(config)# ip slb serverfarm serverfarm-name
```

Step 2 Choose a load-balancing method.

```
Switch(config-slb-sfarm)# predictor {roundrobin | leastconns}
```

Step 3 Identify the real servers in the server farm:

```
Switch(config-slb-sfarm)# real ip-address
```

Step 4 Assign a weight for the relative server capacity:

```
Switch(config-slb-real)# weight weighting-value  //The weighting value (1 to 255, default 8)
```

Step 5 Put the real server into service:

```
Switch(config-slb-real)# inservice
```

Virtual Servers

Step 1 Name the virtual server:

```
Switch(config)# ip slb vserver virtual-server-name
```

Step 2 Assign the virtual server to a server farm:

```
Switch(config-slb-vserver)# serverfarm serverfarm-name
```

Step 3 Assign an IP address to the virtual server:

```
Switch(config-slb-vserver)# virtual ip-address {tcp|udp}
```

Step 4 Control access to the virtual server:

```
Switch(config-slb-vserver)# client ip-address inverse-mask
```

Step 5 Put the virtual server into service:

```
Switch(config-slb-vserver)# inservice
```

Security

Basic security

1. Enable secret将特权模式加入口令
2. Service password-encrption
3. 使用ACL限制管理网段的访问
4. 确保VTY安全
5. 配置banner进行相应的警告
6. 关闭不用服务，例如udp/tcp small server PAS 等
7. 尽可能关闭CDP，防止信息泄露
8. 禁用集成的HTTP后台
9. 确保SNMP安全

AAA

验证，授权，统计（AAA）是一种有关如果配置不同的安全特性的框架体系。

1. 身份验证：
   身份验证提供了一种处理如下工作的方法
   - 用户身份
   - 登陆和口令对话
   - 挑战和响应
   - 消息接发
   - 加密

   除了本地，线路口令和特权口令身份验证，其他所有身份验证都要求使用AAA
2. 授权
   提供一种控制远程接入的方法，通常使用RADIUS或者TACACS+服务器保存数据库，通过AVP提供用户访问权限
3. 统计
   提供一种收集和发送安全服务器信息，以便用于记帐，审计和报告的方法。包括用户身份，访问时段，执行命令，数据流量等

AAA 的优点：

1， 通常需要一台或一组服务器（安全服务器[ACS]）来存储用户名和密码，而不用每台路由器上配置和更新。
2， 支持 TACACS+, RADIUS, Kerberos 标准安全协议
3， 允许配置多个备用系统，如：先访问安全服务器，如果报错，在查看本地数据库
4， 用户名和密码不以明文形式出现

接入服务器通常和主机用一种安全协议。主机使用安全协议来与安全服务器通信

1. Kerberos 不支持 AAA 中的授权和统计
2. TACACS+运行于 TCP 上，对有效负载进行加密，Cisco 专有，能控制用户权限等级，可将验证和授权分开，因此可使用 TACACS+进行授权和统计，而用其他方法进行验证
3. RADIUS 运行于 UDP 上，只对密码进行加密，开放式的协议，不能控制用户权限等级，不可将验证和授权分开

Configuring AAA

1. 配置身份认证

全局用**tacacs-server host secru-sv-IP single-connection**（使用单连接使接入服务器和安全服务器之间的会话始终保留一条TCP连接，而不会为每次会话打开和关闭TCP连接）或**radius-server host secru-sv-IP** 指定安全协议

全局用**tacacs-server key key**或**radius-server key key** 设置与安全服务器共享的密码

全局用**aaa authentication login default/list-name method1 method2 method3 method4**启用AAA身份验证进程

default列表自动用于所有线路和接口

list指定列表用于指定接口和线路，优先权大于default列表，在线路模式用**login authentication list-name**使其用于指定线路

全局用**aaa authentication enable default method1 method2 method3 method4**启用EXEC模式设分验证，用户在执行enable时验证身份

只有default列表一种无需应用于线路或接口

由于用PPP访问网络时绕开了命令行，所以全局用**aaa authentication ppp default/list-name method1 method2 method3 method4**来使用PPP中任何一个身份验证的方法(CHAP, PAP, …)

在运用PPP的接口下用**ppp authentication chap default/list-name**启用验证

全局用**aaa authotization type default/list-name method1 method2 method3 method4**启用AAA授权

用**show privilege**查看当前权限级别

全局用**privilege exec level 7 ping**指定一个具体命令的权限级别，

该例子说明将EXEC模式中位于权限1的命令PING的权限升到权限7

```
Kaka(config)#aaa new-model //将初始化一种AAA访问控制模型
Kaka(config)#aaa authentication login KAKA tacacs+ //配置login认证，及认证方式tacacs+
Kaka(config)#tacacs-server host 1.1.1.1 //配置tacacs+服务器地址
Kaka(config)#line vty 0 4
Kaka(config-line)#login authentication KAKA
```

2. 配置AAA授权

Auth-proxy
: 用户访问网络前，通过web浏览器向TACACS+或RADIUS服务器证明身份，通过身份认证后，向路由器或者交换机发送ACL条目或者配置文件信息

Commands
: 命令授予执行EXEC命令的权限，例如部分权限低的用户进入交换路由后，只能执行show

EXEC
: 与用户EXEC终端会话相关联的属性

Network
: 网络授权应用于网络接入类型 ppp slip arap等

Reverse-access
: 反向接入指反向telnet会话，用于从控制台访问各种线路

TACACS+ and RADIUS
: 一种C/S结构的认证授权模型

If-authenticated

使用这种方法，只要通过身份验证，便能访问任何功能

None

接口上禁止访问

Local

这种方法使用交换路由本地的用户-口令数据库。在cisco设备上使用username配置本地数据库指明本地用户权限，全局用**username kaka privilege 7 password nichole**指定

全局模式：**Aaa authorization {auth-proxy | network | exec | commands level | reverse-access | configuration | ipmobile } {default | list-name } { method 1[method 2....] }**

接口模式：**authorization {arp | commands level | exec | reverse-access} {default | listname}**

```
Kaka(config)#aaa new-model
Kaka(config)#aaa authorization commands 0 default if-authenticated group tacacs+
Kaka(config)#line vty 0 4
Kaka(config-line)# authorization commands 0 default
```

3. 配置统计

Network

提供所有ppp slip arap会话信息，包括数据包数和字节数

Connection

提供从网络中发起的所有外出连接(eg: telnet and rlogin)的信息

EXEC

提供接入服务器上的用户EXEC会话的信息，包括用户名，日期，起始和结束时间，接入服务器ip，主叫方电话等

system

提供所有的系统级事件

Commands

提供在网络接入服务器上执行的特定权限级别的EXEC外壳

资源统计

提供了用户身份验证的呼叫的起始和终止记录

Aaa accouting { system | network | exec | connection | command level } {default | listname |{start-stop | stop-only | none } method1 method2...

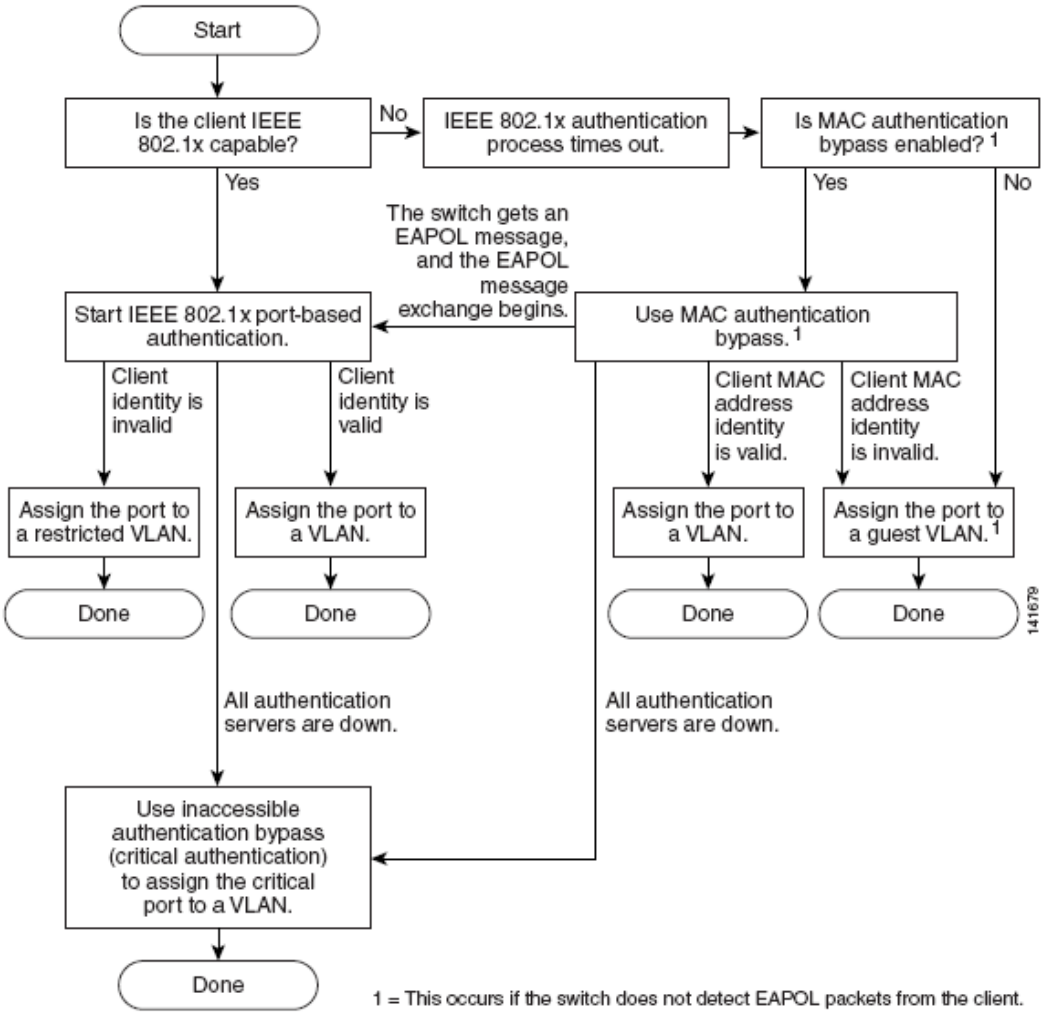
Accouting {arap | commands level | connection | exec {default | listname}

SSH-Connection

```
Kaka(config)#username kaka password nichole
Kaka(config)#ip domain-name kaka.com
Kaka(config)#crypto key generate rsa
Kaka(config)#line vty 0 4
Kaka(config-line)#transport input ssh
Kaka(config-line)#exit
```

IEEE 802.1X

下图为IEEE 802.1x认证的全过程



Port-control有3种类型:

force-authorized

force-unauthorized

auto

配置AAA客户端及802.1x

```
aaa new-model
aaa authentication dot1x default group radius
aaa authorization network default group radius
!---和802.1x相关的AAA设置
dot1x system-auth-control
!---打开802.1x功能
interface FastEthernet0/2
switchport mode access
dot1x port-control auto
spanning-tree portfast
!---在F0/2口上打开802.1x端口控制功能
radius-server host 192.168.168.155 key xxxxxx
!---定义RARIUS Server
```

SPAN and RSPAN

交换式端口分析器(SPAN)分析经过某个本地端口或VLAN的流量信息.SPAN发送一份流量的拷贝给连接安全设备的交换机端口.注意,一旦启用了 SPAN,VSPAN或RSPAN,目标端口的STP*作就被禁止,可能会造成环路.另外,配置RSPAN之前要先定义一个RSPAN专用的VLAN.如果在VTP服务器上配置了RSPAN VLAN,那么VTP服务器自动将正确的信息传播给其他中间交换机;否则要确保每台中间交换机都配置的有RSPAN VLAN.

SPAN的3种模式:

1. SPAN:源端口和目标端口都处于同一交换机,并且源端口可以是一个或多个交换机端口.
2. 基于VLAN的交换式端口分析器(VSPAN):SPAN的一种变体,源端口不是物理端口,而是VLAN.
3. 远程交换式端口分析器(RSPAN):源端口和目标端口处于不同的交换机.

Configuration:

配置SPAN和VSPAN的步骤如下:

Catalyst 3550仅支持2个monitor的会话

```
Kaka(config)#monitor session {session-number} source {interface interface|vlan vlan-id} [rx|tx|both]
Kaka(config)#monitor session {session-number} destination {interface interface} [encapsulation {dot1q|isl}]
Kaka(config)#monitor session {session-number} filter vlan {vlan-list}   //限制vlan
```

监控cpu

```
Monitor session 1 souce cpu {queue }
```

配置RSPAN的步骤如下:

1. 创建RSPAN专用VLAN:

```
Kaka(config)#vlan {vlan-id}
```

2. 定义RSPAN-VLAN:

```
Kaka(config-vlan)#remote-span
```

源交换机配置:

```
Kaka(config)#monitor session {session-number} source {interface interface|vlan vlan-id} [rx|tx|both]
Kaka(config)#monitor session {session-number} destination remote vlan {rspan-vlan-id} reflector-port
{interface}
```

目的交换机配置

```
Kaka(config)#monitor session {session-number} destination remote vlan {rspan-vlan-id} reflector-port
{interface}
Kaka(config)#monitor session {session-number} destination {interface interface|vlan vlan-id} [rx|tx|both]
```

Example: configure SPAN on 3550 SW1, source interface is fa0/8, destination port is fa0/9.no need config the port secure information.

```
Interface fastethernet0/8
Switchport mode access
Interface fastethernet0/9
Switchport mode access
!
No monitor session all
Monitor session 1 souce interface fastethernet0/8 both
Monitor session 1 destination interface fastethernet0/9
!
Source Switch
vlan 627
remote-span
monitor session 1 source interface FastEthernet0/1 both
monitor session 1 destination remote vlan 627 reflector-port fa0/2
!
中继switch
vlan 627
remote-span
!
监控switch
vlan 627
remote-span
monitor session 1 source remote vlan 627 reflector-port fa0/2
monitor session 1 destination interface Fastethernet 0/23
```

MultiCast

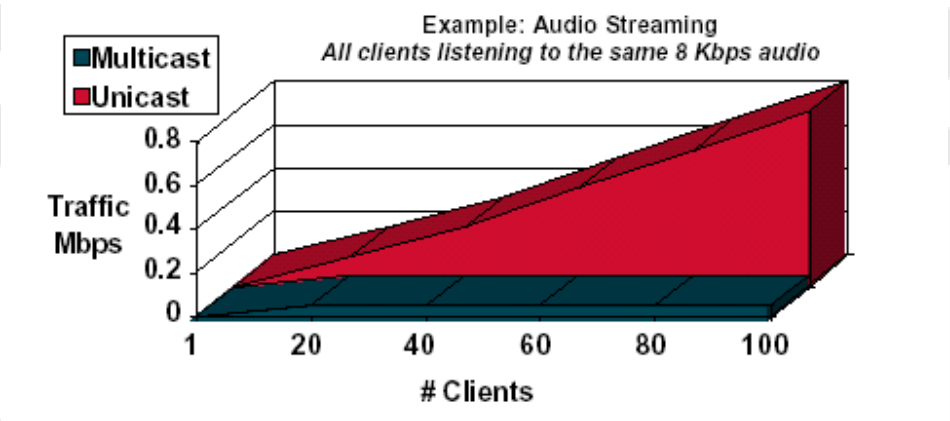
Introduction

Multicast Advantages

1、 当完全相同的数据要发送给多个接收者时（天气预报、股市行情、远程音/视频传输……）

2、 在前面所述情况下，相对于单播的广播传送技术，组播可以更有效的利用网络带宽，并在很大程度上降低对主机和网络设备系统资源的占用。

3、 发送者无需事先知道接收者的地址，接收者可自由加入或退出组播接收组，发送者无需为接收者的变化对系统设置做出任何改动。



图显示的是采用单播与组播方式传送相同数据时，随着客户机数量的增加网络流量变化的情况，在这一点上组播技术的好处非常明显，无需多说。

Multicast disadvantages

组播的缺点源于它是基于UDP技术的。

Best Effort Delivery	这就意味着组播不能保证每个数据包都被正确传送到目的地。中间可能会被丢弃。
No Congestion Avoidance	组播技术本身不提供流控机制。
Duplicates	组播可能导致重复数据包的产生，应用程序应该有能力处理这一情况。
Out-of-Sequence Packets	数据包的顺序可能会被打乱。接收方要有重新排序的能力。

Multicast Applications 组播的应用

基于远程网络的组播传送技术目前主要依赖于 MBONE。有很多基于 MBONE 的组播传送应用程序可以免费下载，很多程序的源代码也是公开的。 目前由于国内互联网骨干带宽有限，使用成本较高，加之信息化、数字化水平较低等原因， 基于远程组播传送技术的应用并不多，但是随着互联网应用水平的不断提高，在远程实时数 据传送、远程教学、远程等领域中的组播应用会越来越普遍。 这个我就不多说了，目前在国内大规模应用组播传送技术的案例真的是不多。

Multicast Service Model 组播服务模型

RFC 1112 即 IGMP（Internet Group Management Protocol）中定义了主机可以动态加入和退出组播组。组播组成员可以分布于Internet的任何地点。（当然了，前提是网络互联设备支持组播信息的传递）组播信息的传递依赖于分布树，有很多组播路由协议，如 MOSPF（Multicast Extension to Open Shortest Path First）、CBT（Core Based Tree）、

Multicast Address 组播地址

组播组采用D类 IP 地址（224.0.0.0-239.255.255.255）进行标识，这些地址用来表示整个接收组，而非单一接收者。

组播地址当中包含一些 Well Known 地址，它们都为特殊的应用而保留，范围是 224.0.0.0 至224.0.0.255，如 224.0.0.1 代表子网中所有支持组播技术的主机，224.0.0.2 表示子网中所有支持组播技术的路由器等。

组播地址中 224.0.1.0-238.255.255.255 为全球范围组播地址。

239.0.0.0-239.255.255.255 为有限范围组播地址，用于私有域内的组播信息传送。

组播地址可以动态分配，SDR 技术允许应用程序在建立新的会话时随意选用组播地址，通过冲突检测技术避免地址的重复使用，这种方法适用于初期应用较少的MBONE，MASC（Multicast Address Set-Claim）是 IETF 设计的新的组播分配方案，首先将组播地址段静态分配到不同的地区，在每个地区内仍然采用动态租用的方案使用组播地址，想法是好的，实现起来的难度较大，周期较长。

目前的解决方案多采用静态组地址分配技术，SGAA（Static Group Address Assignment）是在 MASC 方案得以实现之前的应急之道。采用 233/8 地址段用于静态地址分配，中间两节包含网络自治系统号，最后一节用于组分配。

Multicast Protocol Basics 组播协议基础

下面一些内容是我们研究组播技术时需要关注的：

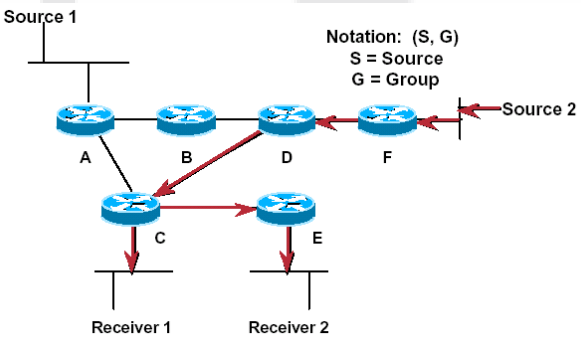
组播的分布树，定义了组播流量从源到目的地址的路径，我们有很多方法用于构建它。

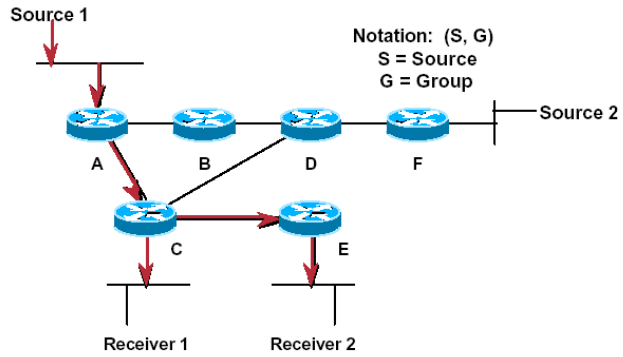
组播信息转发，这一点与单播有很大的不同，单播基于目的地址转发，组播基于源地址转发。

组播协议类型，密模式和疏模式

Multicast-Tree

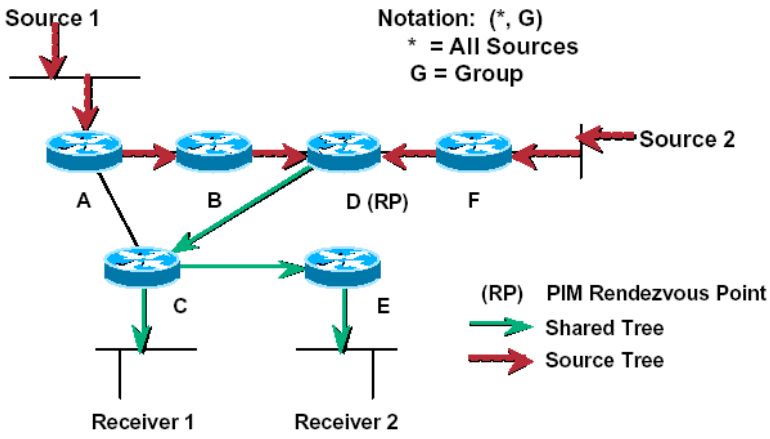
SPT（Shortest Path Trees）即（Source Distribution Tree 源分布树）采用最短路径优先算法， 中间的转发设备采用 Notation（S，G）的方法记录转发表，S 表示多播源 IP 地址，G 表示多播组地址。





在上面的两幅图中，中间的路由器基于源和组地址分别建立不同的转发条目。

SDT（Shared Distribution Tree 共享分布树）技术需要先指定 RP（Rendezvous Point 会聚点） 所有的多播源先把信息送到 RP，再由 RP 分发到各接收点，SP相当于树根，到达R之后的多播信息转发时与源无关，故采用 Notation（*，G）的方式记录转发表，



Source or Shortest Path trees 因为要为每个源和组记录转发条目，会耗费较多的路由器资源， 但是转发路径准确，Shared trees 记录条目少，但是因为多播信息的转发必须通过 RP，可能导致次佳转发路径的存在。

如何构建分布树

PIM基于单播路由表来构建树，路由器用 Join（加入）、Prune（修剪）和 Graft（嫁接）三种消息通知上游路由器是否需要向其转发某一多播组的消息。

DVMRP很像RIP，是一种距离向量型协议，采用Poison-Reverse（毒性反转）、Prune（修剪）和 Graft（嫁接）三种消息通知上游路由器是否需要向其转发某一多播组的消息。

MOSPF 利用单播 OSPF 协议的 LSP 来传递通告，构建（S，G）转发条目。

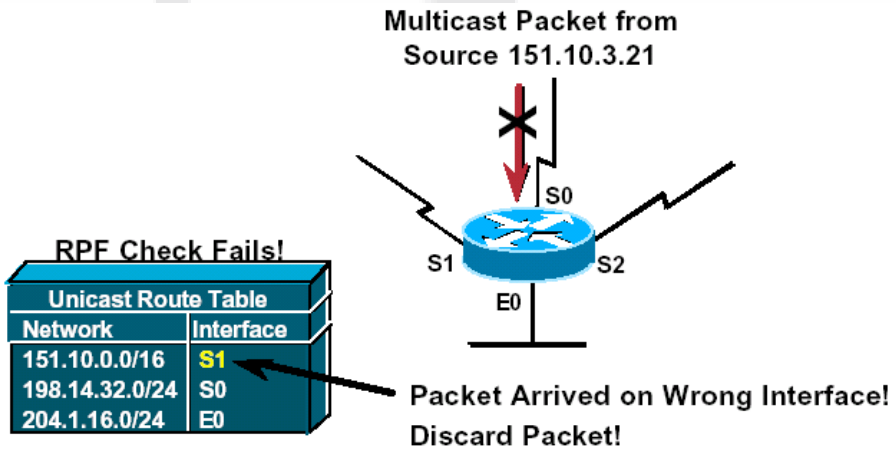
CBT 基于原理与 PIM 疏模式相同，路由器用 Join（加入）、Prune（修剪）和 Graft（嫁接）三种消息通知上游路由器是否需要向其转发某一多播组的消息。

多播信息的转发

基于发送源的 IP 地址（数据包中的源地址）而非接收者的 IP（数据包中的多播地址不代表单个接收者）来转发。路径的选择依靠 RPF（Reverse Path Forwarding 反向路径转发）技术。包括三个过程：Broadcast（广播）洪泛传递，假定网络上的每个主机都是多播组成员；Prune（修剪）停止向那些没有组成员存在的网络转发多播信息；Selective Forwarding（选择性转发）如果有多条转发路 径可用，则依靠单播路由表来帮助选择最佳路径。

RPF Check

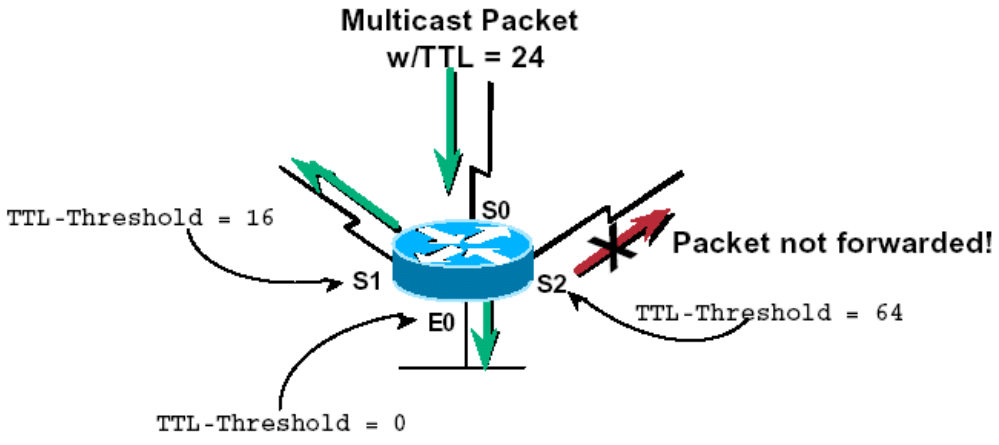
路由器只转发从可达源的上游路由器的接口（依靠单播路由表判断）收到的多播数据包。 RPF 检验成功则转发，RPF 检验失败则 Silently Dropped（静悄悄的丢弃）。这项技术主要防止多次转发多播数据包。



在这个图中路由器从 S0 口收到了一个来自 151.10.3.21 的多播包,可是检查路由表后发现， 由它到网络 151.10.0.0 网络的数据包应由 S1 口发出，RPF 检验失败。

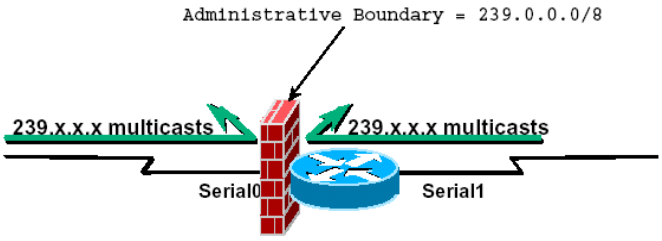
TTL Thresholds（TTL 阈值）

组播传送中的 TTL 类似于单播，但是它更灵活，可用于限制多播信息转发的范围。 一个多播包进入路由器后，它的 TTL 值首先减 1，在从出口送出去之前，如果出口上所设 置的 TTL 阈值非 0，则要进行 TTL 阈值检验，所有 TTL 值小于接口阈值的多播包将不会被 发送。用接口命令 **ip multicast ttl-threshold ttl** 来实现。



Administrative Boundary 管理边界

用于限定到达特定多播组的数据包的转发。用接口命令 ip multicast boundary <acl>来实现。



Dense Mode & Sparse mode（密模式与疏模式）

Dense mode 采用推方式传送多播信息，假定所有的网络节点都是接收者，然后再修剪掉向那些不需要多播信息的网络的转发，Flood & Prune 的行为每隔几分钟就要发生一次。

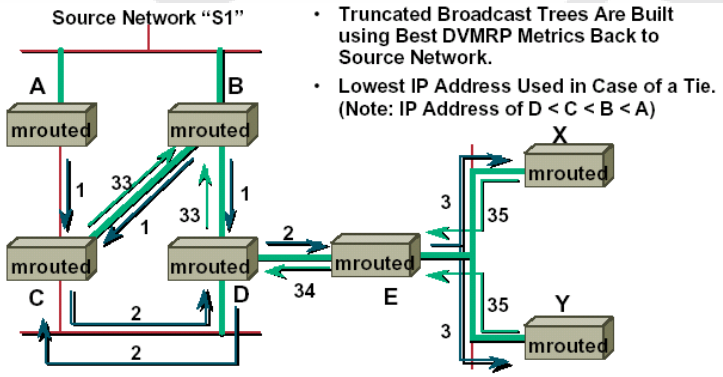
Sparse mode 采用拉方式传送多播信息，除非接收端显式的声明，否则将不会向其发送多播数据。

注意：鉴于Dense mode周期性的Flood行为对网络影响较大，实际的多播网络中只采用Sparse mode 来构建多播转发表。

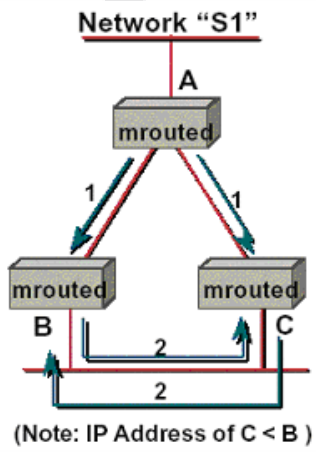
DVMRP

DVMRP（Distance Vector Multicast Routing Protocol）主要特点距离向量型协议，建立分布树时按照到达源的Metric（基于跳数-hop count）来选路。最大支持 32 跳，需要接收到达某个组播组的数据包的路由器向其上游发送毒性反转消息，Metric 为原 Metric+32，收到这一毒性反转消息的上游路由器会将收到该消息的接口加入多播消息转发列表。当下游路由器不再需要到达某个组播组的数据包时，它发送 Prune 消息通知上游 路由器停止向其转发到达该组播组的数据包。

TBT（Truncated Broadcast Tree）是 DVMRP 协议形成的多播转发树，基本到达源有最小Metric 原则构建，当多条链路有相同的 Metric 时，路由器选择 IP 地址最小的上游邻居路由器。在下图中，C 同时收到A和B发来的 Metric 相同的路由信息，它选择了B，因为B的IP地址小于A的IP地址。



相似的情况也出现在下面的多路访问网络中，B 和 C 都有关于网络 S1 的路由，有最小 Metric的路由器会被选为 Designated Forwarder，Metric 相同的情况下同样根据最小 IP 地址原则选定DF。本例中是C路由器。



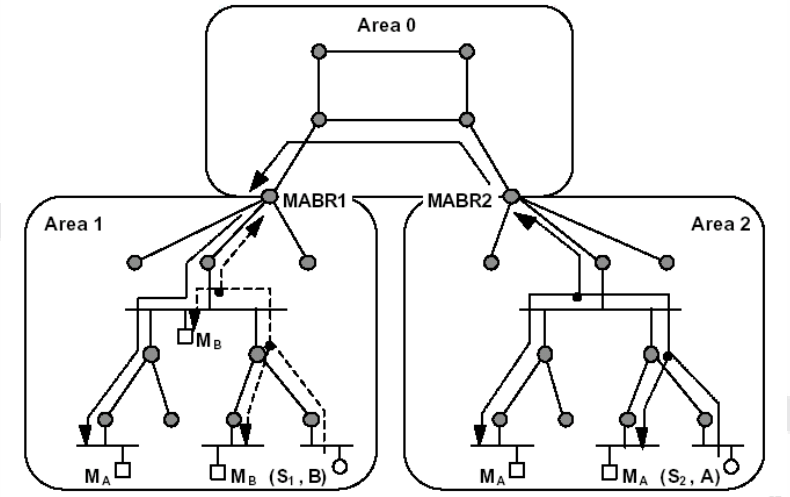
TBT 的优点是保证在有环路的网络中也不会有重复的组播数据包出现，缺点是需要在整个网络中交换DVMRP 路由信息，而且由于它被设计成与 RIP 相似的距离向量型路由协议，和 RIP等DV协议一样，存在计数到最大，保持时间和周期更新等问题。跟其它密模式多播协议一样，它适用于大量的组播接收者离组播源较近的情况。因为它是较早的多播路由协议，所以在 MBONE之上应用广泛，但扩展性较差，逐渐被淘汰出局。

Multicast OSPF

基于单播 OSPF路由协议，使用一种新的OSPF链路状态数据包 ，我们称之为Group-Membership LSA 来通告网络中组成员的存在情况。

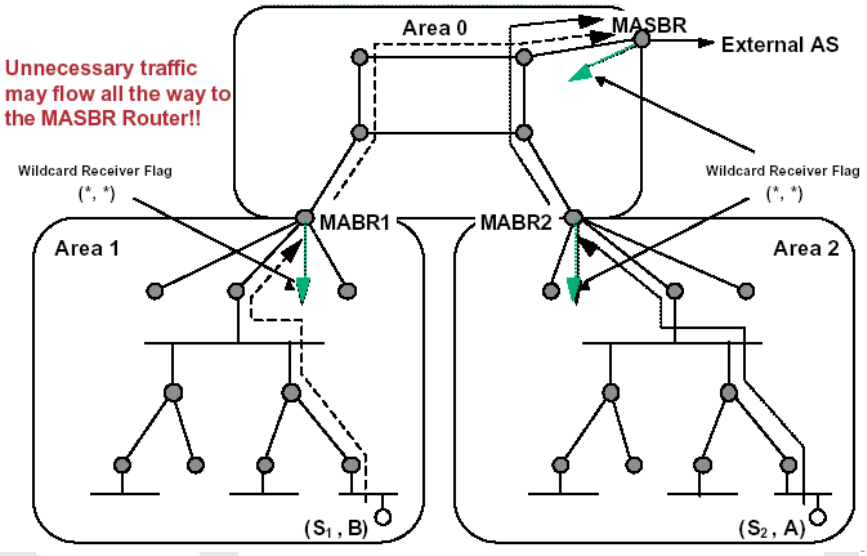
根据 OSPF 路由协议的基本原则，GMLSA 只能在OSPF区域内传递，这将导致多播信息不能跨越 OSPF 区域传递。解决的办法是 MABR（MOSPF Area Border Router）向区域内其它路由器发送带有“Wildcard Receiver”标志的 GMLSA，通知内部路由器它可以到达任何多播组。 所有的多播转发树都应有指向 MABR 的分支，所有多播组的多播信息都应被转发给 MABR。MABR 会向 Backbone 区域发送汇总的组成员关系 LSA。通知 Backbone 区域它需要接收发 往哪些多播组的信息。

在下图中，因为区域 1 中有源 S2 的接收者，所以源 S2 发出的多播信息经骨干区域传入了区域 1。而源 S1 所发出的多播信息则不需要传入区域 2，只在区域 1 内传递。



同样的机制用来解决域间路由，MASBR 与 MABR 一样，也发出带有 Wildcard Receiver 标志的LSA。

从下图中我们还可以看到，Wildcard Receiver 这种方法虽然解决了区域间和域间多播路由信息传递的问题，但是却可能引起许多发往 MABR 和 MASBR 以及那些中转路由器不必要的多播流量。这是MOSPF的一个问题。



Evaluation to MOSPF:使用链路状态数据包来传递多播路由信息，保证多播信息传递的正确性。协议相关，只能用于采用OSPF 路由协议的网络。 也存在可扩展性问题，在大的网络中使用可能存在过多的路由表条目，并可能由于组成员关系和链路状态的经常性变化而引发频繁的 SPF 计算。不适用于多播信息源较多的网络。

PIM-DM Protocol Independent Multicast Dense-Mode

协议无关多播密模式

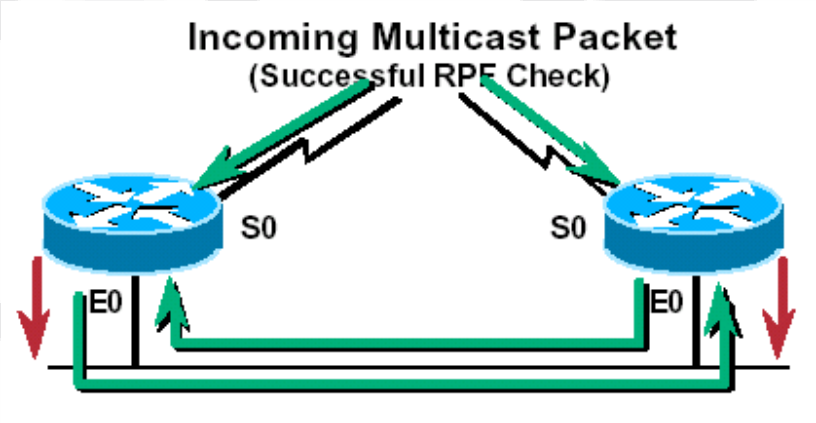
首先要说的是 PIM 分为两种模式，密模式和疏模式

密模式：

它最大的特点当然是它的协议无关性，可以与静态路由及各种动态路由协议很好的共存。它使用RPF（Reverse Path Forwarding 反向路径转发）技术构建转发树，密模式的 PIM最开始采用Flooding的方式向全网传送多播信息，然后再修剪掉不必要的和冗余的流量。这种 Flood&Prune 的行为每 3 分钟进行一次。适用于小型网络和实验性网络。

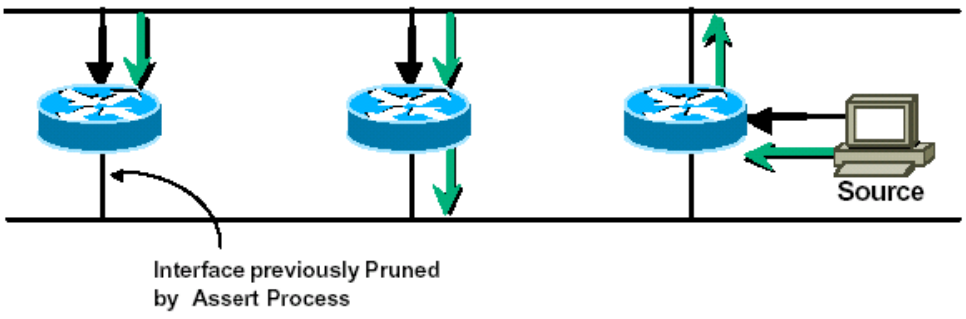
Assert Mechanism

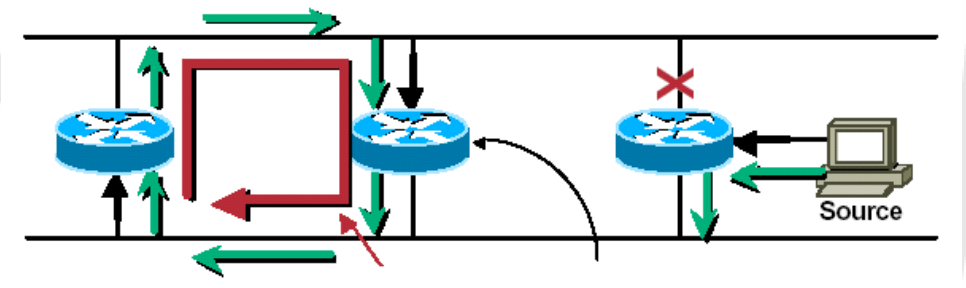
看下面的图，两条路径都将通过 RPF 检验，为了防止多播信息复制，必须加以处理。



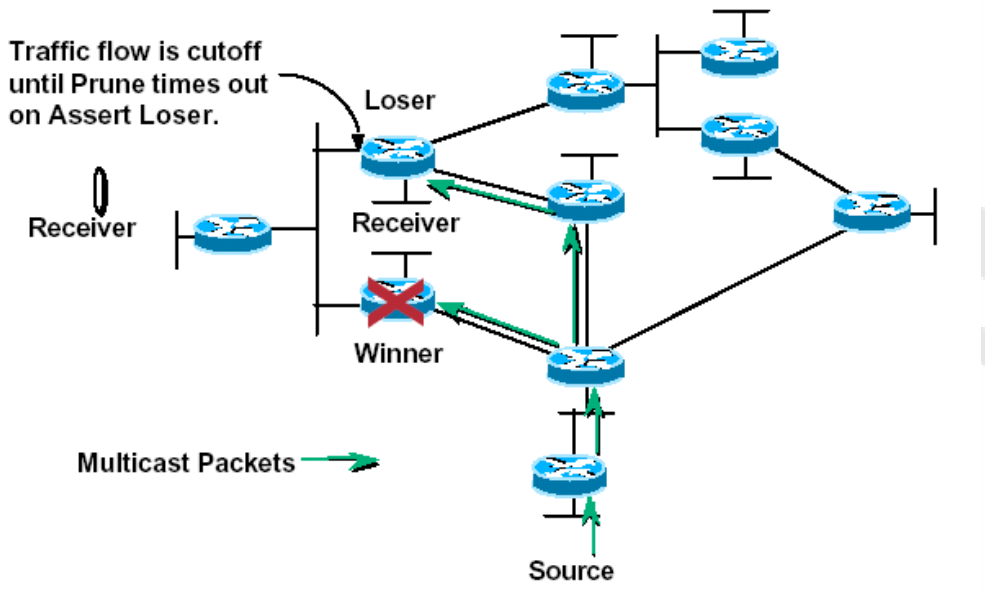
这种情况下两台路由器会发出 Assert 值（Distance+Metric）进行比较，有最好的到达源的路由的路由器胜出，相同的情况下比较 IP 地址，最大者胜出。败者 Prune 掉相应接口。

Assert 机制可能导致多播路由的环路，看下面两幅图，第一幅图是经 Assert 比较做出修剪之 后的稳定网络，当右边的路由器上端口断掉连接时，多播网络需要重新收敛，如果左边的路 由器收敛快，就可能出现图中所示的多播路由环路。





下图中显示的是由于 Assert 机制引起的另一问题，中间的两台路由器进行 Assert 比较的结 果是下面一台路由器胜出,当它的左侧接口断掉时,最左边的接收者将暂时收不到来自下面 多播源的信息，直到上面的 Loser 左侧接口的 Prune 超时。



Evaluation to PIM-DM在小型实验性网络中效果很好。 优点是容易配置（只需两条命令）洪泛和修剪的机制简单。缺点是洪泛和修剪行为影响较大（3 分钟一次）Assert 机制复杂。

PIM-SM（Protocol Independent Multicast Sparse-Mode）

采用 RP（rendezvous point）作为中继节点，组播源通过到达 RP 的最短路径向 RP 发送组播 数据，RP 再通过到达每个接收者的最短路径向接收者转发多播数据。从RP到接收者的转发路径信息与组播源的位置和数量无关。 适用于企业级大型网络环境,接收者稀疏或密集的情况均适用。最优路径的选择与网络尺寸 和成员密度无关。

组播接收者需知道 RP 的存在，通过向 RP 及途经路由器发送（*，G）的消息加入组播组。 该消息一跳一跳传至 RP 后，形成了一条从 RP 到接收者的 ST（Shared Tree） 组播发送者同样需知道 RP 的存在。组播源将多播数据封装成专用的 PIM SM 注册消息包， 并以单播的形式向RP发送注册消息，RP收到注册消息后，解封装这个注册包，将多播消 息发往接收者。同时也向通往组播源的各个路由器发送（S，G）的消息，构建从组播源到RP 的 SPT（Shortest Path Tree+）当 RP检测到从组播源发出的多播数据包经由 SPT 到达时，（表明 SPT 稳定，沿途各路由器均已有正确转发条目）RP 向组播发送“Register Stop（注册停止）”消息，至此，从组播源 到接收者的转发路径形成。下图显示全部处理过程节束后的多播转发路径。由组播源到 RP 的源树和 RP 到接收者的共享树两部分构成。

问题：接收者虽然收到了发自源的组播信息，但是经由 RP 转发的传送路径不是从源到接收 者的最佳路径，接下来介绍这个次佳路径是如何修正的。

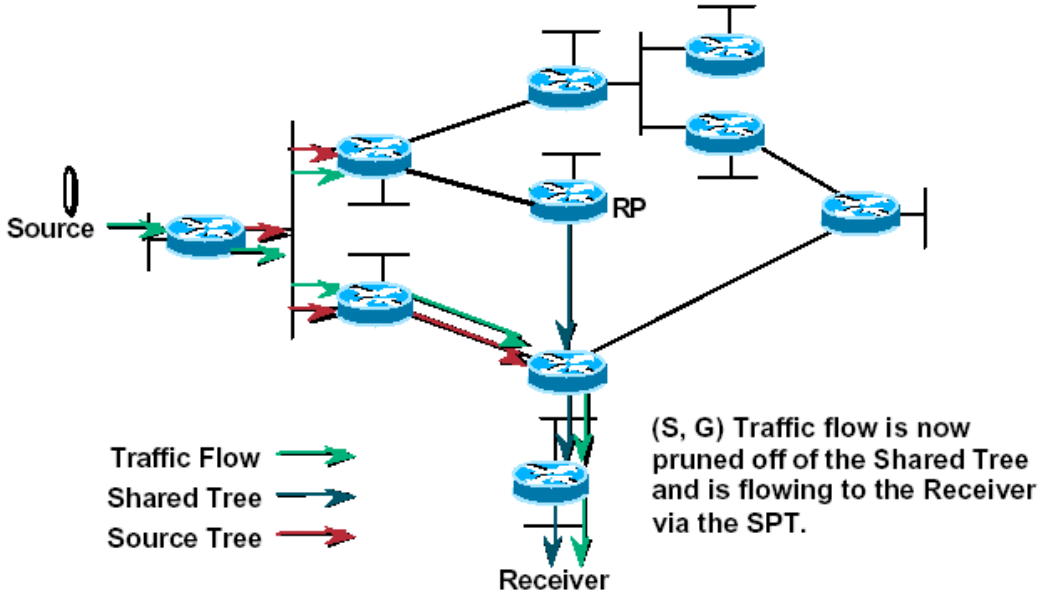
SPT Switchover（SPT 交换翻动，呵呵，Switchover 怎么翻译好？）

当 last-hop 路由器（即离接收者最近的路由器）收到经由（*，G）共享树由 RP 转发来的组 播消息时，它就向 SPT 路径中的上游路由器发送（S，G）加入消息（延迟由 SPT-Threshold 值决定，这个值缺省等于 0，表示立刻切换成经 SPT 的转发），这个加入过程一跳一跳的进 行直至 first-hop 路由器（即离组播源最近的路由器），从而形成另一条从源直接到达接收者的 SPT 转发路径。

最后，向 RP 发送修剪消息，通知它停止通过共享树向接收者转发特定源组（S，G）的消息， 在这一步完成之前，接收者可能收到重复组播数据。

另外，如果 RP 在所有属于共享树的接口上（注意：本例中只有一条共享树）均收到了关于（S，G）的修剪消息，它也会向组播源发送（S，G）修剪消息，取消（S，G）源组向它的不必要转发。通过这种机制，PIM SM 实现也与 PIM DM 一样的转发路径，因为没有最初的 Flood 过程， 对无关网络及设备影响较小。

下图是最终形成的从源到接收者的 SPT。



对 PIM-SM 的评价,对于稀疏和或密集的接收者分布同样有效。

优点：流量只会沿必要的分支路径转发，可动态切换到最优路径，与单播路由协议无关。可与DVMRP互操作。

缺点：需要配置稳定的RP路由器。

Core-Based Trees: CBT

CBT的基本目标是减少网络中路由器组播状态，以提供组播的可扩展性。为此，CBT被设计成稀疏模式（与PIM-SM相似）。CBT使用双向共享树，双向共享树以某个核心路由器为根，允许组播信息在两个方向流动。这一点与PIM-SM不同（PIM-SM中共享树是单向的，在RP与组播源之间使用SPT将组播数据转发到RP），所以CBT不能使用RPF检查，而使用IP包头的目标组地址作检查转发缓存。这就要求对CBT共享树的维护就需非常小心，以确保不会产生组播路由循环。

从路由器创建的组播状态的数量来看， CBT比支持SPT的协议效率高，在具有大量组播源和组的网络中，CBT能把组播状态优化到组的数量级。

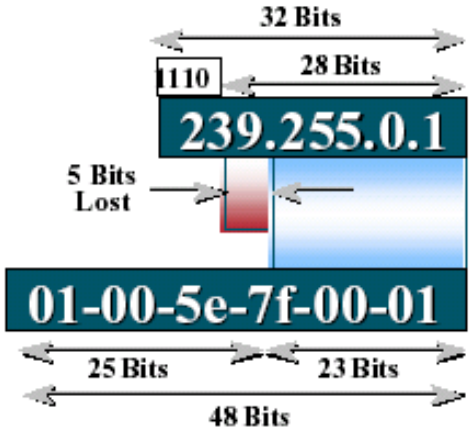
CBT为每个组播组建立一个生成树，所有组播源使用同一棵组播树。CBT工作过程大体如下：

1. 首先选择一个核，即网络中组播组的固定中心，来构造一棵CBT；
2. 主机向这个核发送join命令；
3. 所有中间路由器都接收到该命令，并把接收该命令的接口标记为属于这个组的树；
4. 如果接收到命令的路由器已是树中一个成员，那么只要再标记一次该接口属于该组；如果路由器第一次收到join命令，那么它就向核的方向进一步转发该命令，路由器就需要为每个组保留一份状态信息；
5. 当组播数据到达一个在CBT树上的组播路由器时，路由器组播数据到树的核。以保证数据能够发送到组的所有成员

CBT将组播扩张限制在接收者范围内，即使第一个数据包也无需在全网扩散，但CBT导致核周围的流量集中，网络性能下降。所以某些版本的CBT支持多个核心以平衡负载。

目前CBTv3草案已公布。该方案通过使用CBT边界路由器（BR）更好地处理域间组播的转发。CBTv3还引入新的状态及单向分支CBT概念。尽管CBT很有代表性，但至今却几乎没有已实现的CBT网络。

Layer 2 Multicast Addressing



三层组播地址前四个二进制位固定为1110，所以后面共有28 位可变。MAC 地址一共48 位,分6 节,前面三节为01-00-5E 的专门用于与三层多播地址建立映射。 且第四节的首位固定为0，所以MAC 地址中有23 位可变。我们就是把组播地址的后23 位映射到MAC 地址的后23 位上

在令牌环网中，三层组播地址被简单映射为功能地址

C0-00-00-04-00-00（在令牌环网中因为二进制位是反序的，实际是03-00-00-20-00-00）或全1 地址FF-FF-FF-FF-FF-FF。缺省使用全1地址，

可以用接口命令**ip multicast use-functional** 更改为使用功能地址。

IGMP

Introduction

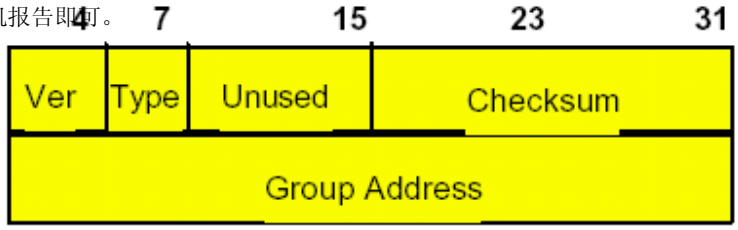
IGMP（Internet Group Management Protocol）主要用于主机向路由器通知其加入某个多播组 的行为，以便路由器向其转发到达特定多播组的数据包。IGMP 目前共有三个版本，版本1在RFC 1112 中阐述，版本2 在RFC 2236 中阐述，版本3目前仍在开发之中。

IGMP Version 1

版本1采用路由器查询和主机报告两种方法维护组成员关系。路由器向 224.0.0.1(all hosts)地址发送 TTL=1 的查询包，这种查询每 60-120s 发生一次， 如果在一个 LAN 上面存在多台路由器，则只有其中的一台作为 Designated/Elected 的路由器 发送查询信息。

主机可以向路由器发送 TTL=1 的组成员关系报告，在主机想要加入某个多播组或是收到来 自路由器的组成员关系查询消息时发出 ，因为同一网段上的主机只要有一台想要接收到达某个多播组的消息，

路由器就会将到达这一多播组的消息发至该网段，所以主机的报告采用压 制的方式，对某一个组信息的接收请求只需有一台主机报告即可。



上图是版本1 的消息包格式，Ver=1，Type=1 表示主机成员关系查询，Type=2 表示主机成员关系报告。Group Address 表示Report消息中主机要加入的多播组。

当一个主机要加入多播组时，它可以立即向路由器发送成员关系报告，不必等待查询，这样 就减少了端系统最初加入一个多播时的延时。路由器也会定期的发出组成员关系查询，同一网段中只需一台欲接收组播信息的主机响应查 询即可。其它同属于该组的主机的响应被压制。（Response Suppression：每台主机在响应发 自路由器的查询之前启动一个 count-down 计数器，计数器的值是一个组定范围内的随机数， 当这个数减到 0 时主机对查询发出响应，如果在这之前主机收到了其它主机发出的响应，则 它不再发出响应消息。）这个查询周期可以通过命令 **ip igmp query-interval** 来修改，缺省时间间隔为 60s。

一个网段只需一台路由器发出查询即可，但是 IGMPv1没有规范的 Query Router 的选举机制，选举过程依赖于多播路由协议，有时候会出现多次查询的情况。

当主机离开一个组后，它不再响应路由器发出的组成员关系查询，同一网段上的所有主机都离开一个组后，路由器将收不到任何报告，达到超时值后路由器停止向该网段转发相应组播 消息。这个 Passively Leave 的方式增加了注销延时，在版本 2 中得以改良。

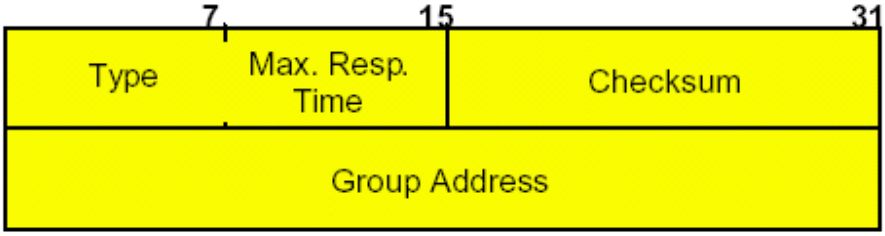
IGMP Version 2

对版本 1 的改良主要体现在以下几个方面：

在原有的发向 224.0.0.1 的针对所有组成员关系的 General Query 的基础上增加发向特定组地址的针对特定组的查询，以确定是否仍存在该组的接收者。 当主机离开一个组时，主动向路由器发送注销报告，当发送此报告的主机是最后一个组成员时，可以减少路由器停止特定组播消息前的延时。

版本 2 中关于 Designated Router 的选举有了固定的机制，单播地址最小的支持 IGMP 协议的路由器成为 Query Router。

此外，路由器在发出查询包时可以指定主机响应时间间隔。



上图是版本 2 所采用的信息包格式，Type=11 表示成员关系查询，=12 表示版本 1 成员关系报告，=16表示版本 2 成员关系报告，=17 表示退出组报告。 Max.Resp.Time 表示最大响应时延，以 1/10 秒为单位，缺省等于 10 秒。 组地址表示报告或查询的组地址，General Queries 用 0.0.0.0 表示。

主机欲加入一个多播组时可主动发出报告（与版本 1 相同）

```
rtr-a>show ip igmp group
IGMP Connected Group Membership
Group Address    Interface      Uptime    Expires   Last Reporter
224.1.1.1        Ethernet0      6d17h     00:02:31  1.1.1.11
```

上图显示的是路由器中记录的多播组成员关系记录。 路由器也可发出组成员关系查询，最初每个路由器都认为自己是 Query Router 并发出查询，

直到它收到来自比自己的 IP 地址更低的路由器所发出的查询。最后，具有最低 IP 地址的路由器成为最后的 Query Router。版本 2 中路由器还可以发出对特定组成员关系的查询。查询 消息缺省每 60 秒发送一次。

如下图所示，我们可以通过 **show ip igmp interface** 命令查看哪一台路由器是当前的 Query Router。

```
rtr-a>show ip igmp interface e0
Ethernet0 is up, line protocol is up
  Internet address is 1.1.1.1, subnet mask is 255.255.255.0
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is disabled on interface
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Inbound IGMP access group is not set
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 1.1.1.1 (this system)
  IGMP querying router is 1.1.1.1 (this system)
  Multicast groups joined: 224.0.1.40 224.2.127.254
```

主机对查询的响应同样采用压制机制。

主机退出多播组时主动向路由器发送退出消息，路由器收到该消息后会发出指向特定组的查询，以判断该网段上是否还有属于该多播组的成员，如果没有收到任何回应报告，将停止向 该网段转发该多播组数据。（从收到退出消息到停止转发大约经历 1-3 秒）

在版本 1 中，count-down 计数器的最大值固定为 10 秒，这样如果同一网段上的接收者分属 许多不同的多播组，路由器将同时收到大量的回应，版本 2 中可以由 Query Router 指定 count-down 计数器的上限，通过

使用命令 **ip igmp query-max-response-time** 来增加这个值（缺省为 10 秒）可以减少报告拥塞情况的发生。

版本 1 与版本 2 的互操作能力

如果 Query Router 运行版本 1，则支持版本 2 的主机必须发送版本 1 的报告消息。而且在这种情况下 Query Router 不提供对退出报告的支持。

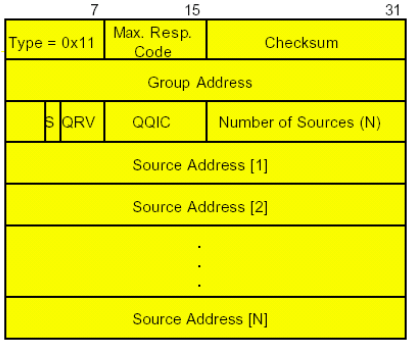
如果 Query Router 运行版本 2，它必须能够发现发送版本 1 的主机，因为版本 1 不提供对最大查询响应时间的支持（固定为 10 秒）。而且必须忽略版本 2 的退出消息，因为版本 1 不能识别 Group Specific Queries。

同一网段上的多台路由器所运行的 IGMP 版本必须相同，如果版本 1 和版本 2 路由器共存的 情况，版本 2 的路由器必须在该接口下配置为版本 1，命令为：**ip igmp version 1 | 2**。

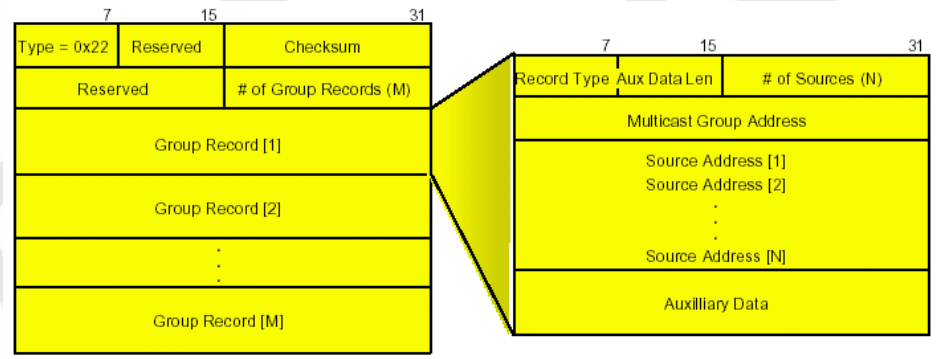
IGMP Version 3

IGMP 版本 3 仍在开发之中，主要的变化是在原有的组播记录中增加了针对组播源的列表， 用来记录接收者接受或拒绝的源。这样就决定了要想利用版本 3 的新特性，路由器和主机内的 IGMP 协议以及基于组播的应用程序都要更新。

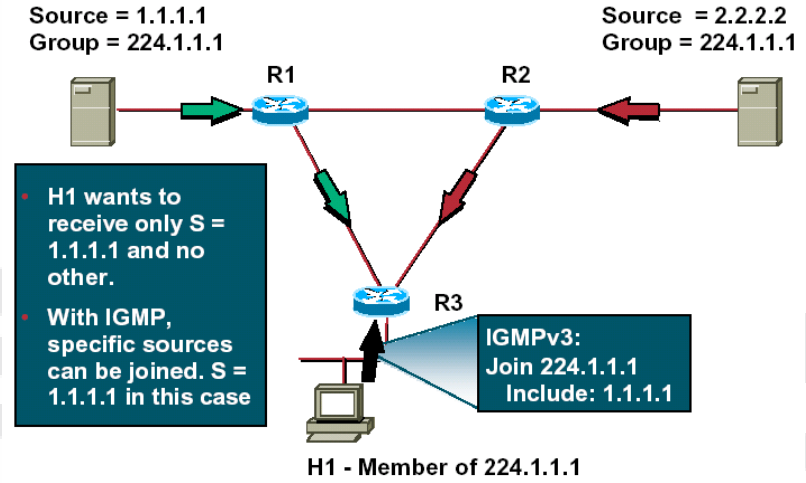
主机向路由器发送报告的地址改为 224.0.0.22，主机的报告不再受到压制，响应间隔仍可调节。



上图是 IGMP 版本 3 所采用的查询包格式，比前面两个版本复杂一些，最重要的是后面增加 了记录组播源的字段。更复杂的是下面的报告包。一个包之中以组记录的形式同时向路由器 通告主机想要接收的多播组和相应的源列表。



，有两个多播源都有数据发送至224.1.1.1 多播组，但是下面的主机只想接收来自源1.1.1.1 的多播数据流，这在以前两个版本的多播中是不能由主机自身实现的。而版本3 提供了对这一功能的支持。



版本 3 的组成员关系报告方式和前面两个版本一样。不过在主机加入一个多播组时可以指定它接收或拒绝（Include or Exclude）源自哪些多播源的数据流。

路由器周期性的向 224.0.0.1 发送查询消息，所有主机发回响应。（因为不同主机可能有不同的多播源要求，所以这里要求所有主机均要做出回应，不采用压制机制。）

Layer 2 Multicast Frame Switching 第二层组播帧交换

前面讲过，用于映射多播地址的 MAC 地址段 01-00-5E 是专用的，这样的 MAC 地址不会出现在源地址中，所以交换机不可能学习到关于组播的交换条目，默认的情况下，交换机处理 组播帧的方式和广播帧一样，在一个广播域内洪泛传播。当然，我们可以通过手工添加静态表项来解决这一问题，但是组播的接收者是动态变化的， 静态的映射方式往往不能反映主机的真实要求。

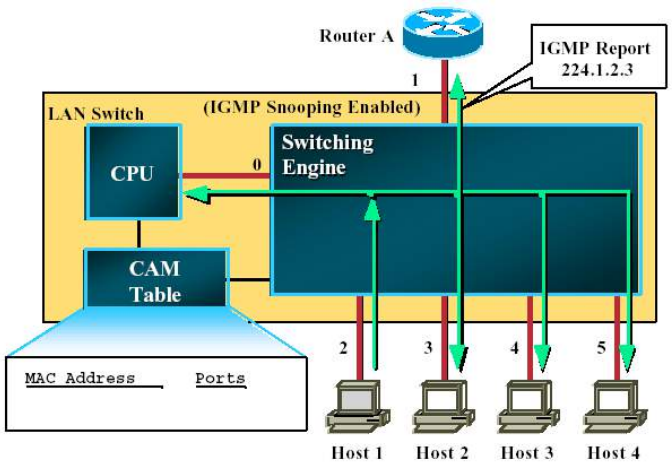
好的方法是动态建立转发映射，但是交换机没有能力知道第三层的事，无法对组播的转发做出准确判断。

IGMP Snooping

让交换机也懂 IGMP？交换机在监听主机也路由器之间的 IGMP 会话， 但这样做意味着交换机必须复制并分析所有的组播帧才能找寻出所有的 IGMP 数据包。可以想像这是一项繁重的任务，会严重影响交换机的性能，用专用芯片来实现这一处理过程可以 释解交换机的压力，但同时会增加交换机的成本。

二层交换机的内部由三部分组成：CPU，CAM Table 和 Switching Engine。CAM Table 是核心，它的内容由 CPU 来维护，Switching Engine 则依照 CAM Table 的内容和 CPU 的指令来 完成端口间的交换任务。

好了，现在让我们通过一个实例来讲述 IGMP Snooping 会给交换机的 CPU 带来多大的负担。 主要内容都基于下面这幅图。



首先，主机 1 发出报告，请求接收到达 224.1.2.3 的多播数据，CAM 表中没有相关条目，于是这条消息被洪泛传递到所有接口。交换机在分析这个 IGMP 包后知道，知道端口 0（CPU， 为了接收后续的组播包）端口 1（路由器）和端口 2（主机 1）与这个组播地址有关，于是在 CAM 表中增加了如下条目：

MAC Address	Ports
0100.5e01.0203	0,1,2

接下来是主机 4 也要加入这个多播组，根据 CAM 表，它发出的 IGMP 包只被转发到端口 0，1 和 2。但同时 CAM 表变成如下的形式：

MAC Address	Ports
0100.5e01.0203	0,1,2,5

接下来是主机 4 向 224.1.2.3 地址发送流量为 6Mbps 的视频信息，根据 CAM，这些数据包 将被转发到端口 0，1 和 2。所以这种方式只能用于多播数据流较小的情况下，实际网络中通常无法应用。

Layer 3 Aware Switch

这种方法是在交换机内部加上一块专门用来处理 IGMP 消息的三层感知芯片。同原来一样， 所有的多播数据包仍然要送至CPU，不过它们不再由CPU 而是由这块专用芯片来处理。为了实现这一点，最初的CAM 表中应该包含下面的项目：

MAC Address	L3	Ports
0100.5exx.xxxx	IGMP	0

IGMP Processing Entry

还是刚才的情况，主机 1 要加入多播组 224.1.2.3，专用芯片处理后 CAM 表变成下面这个样子：

MAC Address	L3	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1,2

接下来是主机 4 也加入，于是 CAM 表又变成如下的情况：

MAC Address	L3	Ports
0100.5exx.xxxx	IGMP	0
0100.5e01.0203	!IGMP	1,2,5

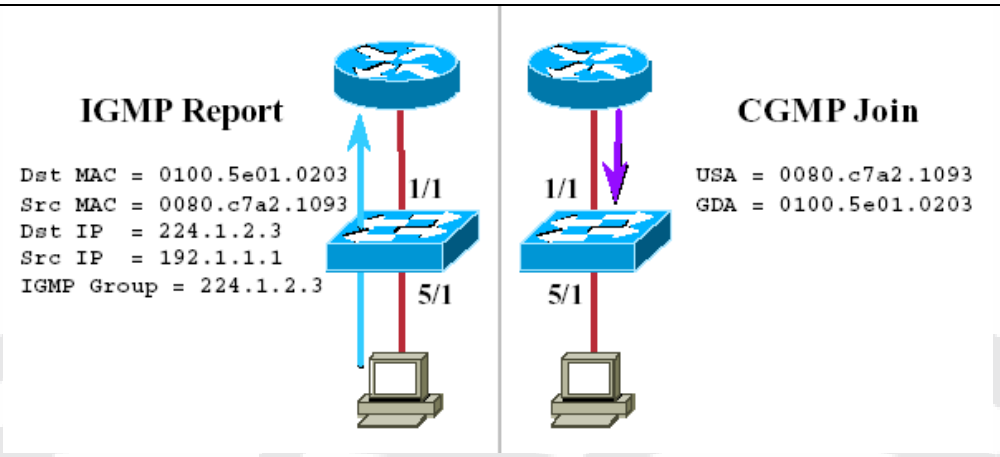
现在如果主机4再发送 6Mbps 的数据到地址 224.1.2.3，将不会对 CPU 产生任何影响。因为根据 CAM 表中的具体条目，该信息流只会被送至 1，2 和 5 号端口。这种方式很好的解决了IGMP Snooping带来的问题，但是却增加了交换机的成本。

CGMP（Cisco Group Multicast Protocol）

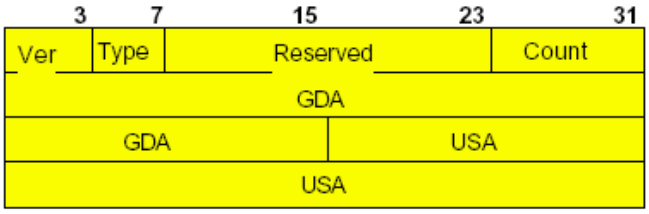
CGMP以Client/Server方式工作，路由器相当于服务器，交换机相当于客户机，路由器接收到IGMP 消息后，以CGMP指令通知交换机多播接收者加入或退出多播组的情况，指导交换机进行准确的多 播定向转发。

CGMP消息中包含以下信息：请求类型（加入或退出），二层多播地址，客户的实际MAC地址。

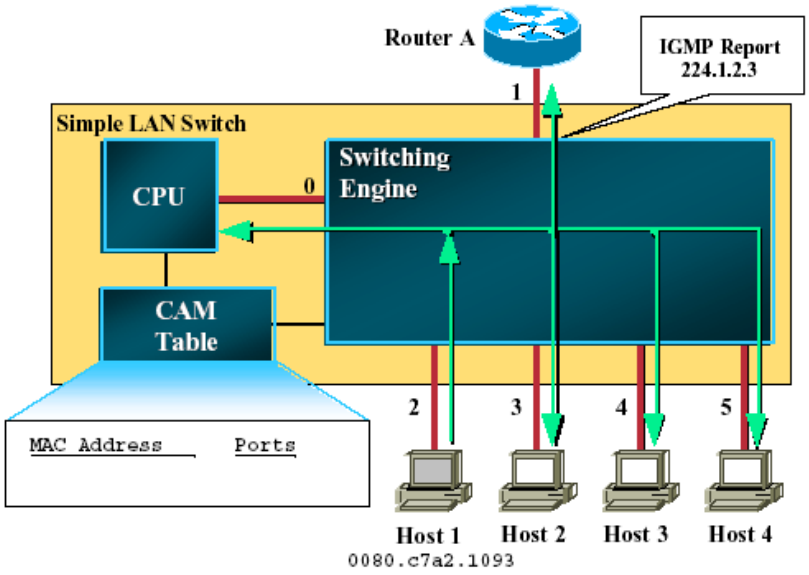
下图显示了全部过程：（其中的USA表示Unicast Source Address,GDA表示组播目的地址）



CGMP数据包的格式：



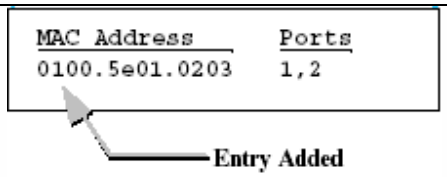
Type=0表示加入，=1表示退出；Count表示包中GDA/USA地址对的数量，后面可 以接任何多个GDA/USA地址对。



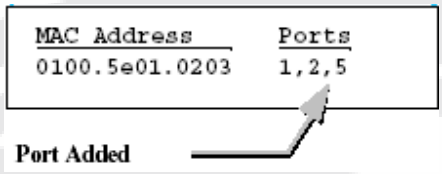
主机1欲加入多播组224.1.2.3，因为交换机中没有相关条目，于是该IGMP消息被洪泛传递。路由器收到这个IGMP加入消息后，会用CGMP消息通知交换机：

CGMP Join
USA 0080.c7a2.1093
GDA 0100.5e01.0204

交换机收到这一消息后，查CAM表找到与USA对应的端口，然后在CAM表中增加相应的多播转发条目。注意该条目中除了USA地址对应端口外，还包括路由器所在端口，交换机有很多方法可以用 来判断它的哪个端口连有路由器，比如：DVMRP Probes,PIM Hellos和IGMP Queries.



接下来是主机4发出加入多播组的消息，根据CAM中的条目，该消息被交换机定向发送至1，2端口，类似于上面的过程，路由器在收到来自主机4的IGMP消息后，向交换机发送相应的CGMP消息，交换机在原有条目中加入端口5。



CGMP的消息分为六类，见下图：

GDA	USA	Join/Leave	Meaning
Mcst MAC	Client MAC	Join	Add USA's port to the Group
Mcst MAC	Client MAC	Leave	Delete USA's port from Group
0000…0000	Router MAC	Join	Assign Port = Router Port
0000…0000	Router MAC	Leave	Deassign Port = Router Port
Mcst MAC	0000…0000	Leave	Delete Group from CAM
0000…0000	0000…0000	Leave	Delete ALL Groups from CAM

所有这些消息均由路由器发往交换机，交换机不产生CGMP消息，所有这些消息均被限制在某个的VLAN之内。GDA和USA都包含有具体值的加入或退出消息用来添加或删除特定组播内的端口项，当路由器收到 IGMP加入或退出消息时产生此类CGMP消息。 GDA全0的加入或退出包中USA是支持多播的路由器的地址，交换机通过这些信息包来判断它的哪 些端口连接的是发送CGMP消息的路由器设备，修改CAM，在所有组播转发条目中加入或删除与该 路由器相连接的端口。 USA全为0的退出消息用来通知交换机从它的CAM表中删掉关于特定组的条目。（当路由器收到来 自最后一个组成员的退出消息时） GDA和USA全都为0的退出消息通知交换机删掉CAM表中的所有多播条目。（出现在路由器关闭CGMP 支持或使用**clear ip cgmp**命令时）

IOS的CGMP相关命令

Router(config-if)#ip cgmp 在接口或子接口下启动CGMP协议

Router(config-if)#ip cgmp proxy 在接口或子接口下启动CGMP和DVMRP代理功能

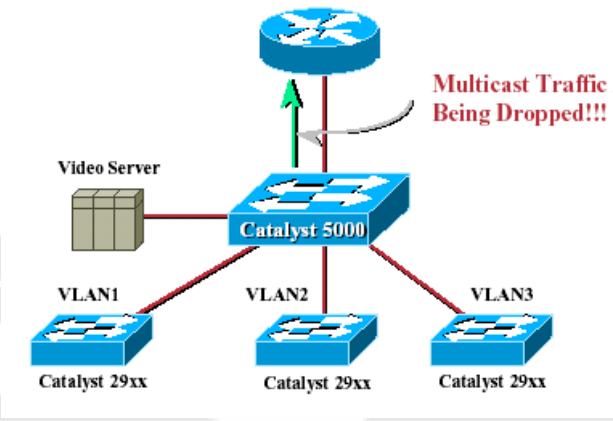
Router#debug ip cgmp 察看CGMP活动情况

Router#show ip igmp interface [int] 显示接口下的与多播相关的信息

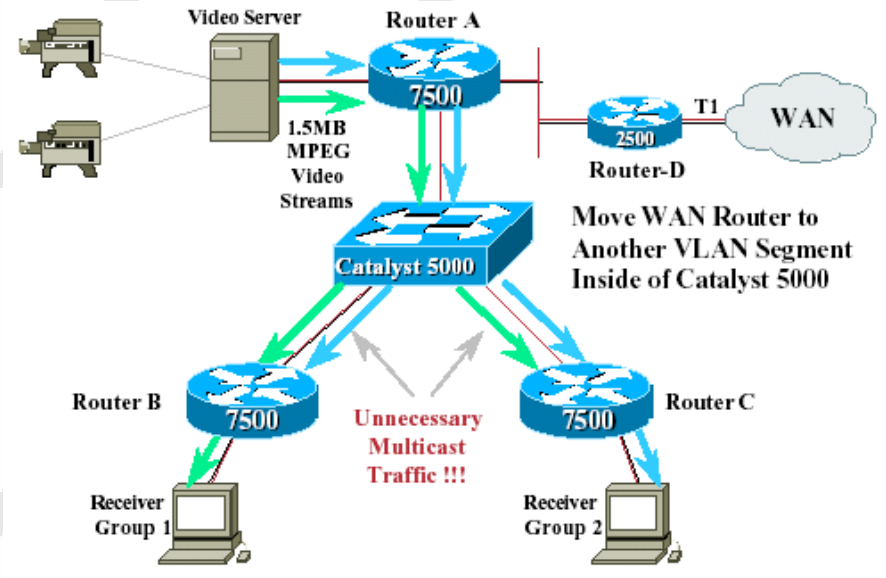
Router#clear ip cgmp [int] 清除所有的CGMP组信息

组播源的位置

看下图，Video Server 以组播形式向外发送组播信息，根据前面的介绍我们知道，收到路由 器的 CGMP 消息的交换机会将多播数据无条件的向支持 CGMP 协议的路由器转发，因为我 们应该尽量的让多播源靠近路由器，以减少对网络带宽的占用。



连接远程网络的 Router-D是一台性能较低的设备，如果我 们把它也同样接在核心交换机 Cat5000 上，大量的多播视频信息会被传递至 Router-D，尽管 远程网络中没有任何的组播信息接收者。图中的连接方法是合适的，如果远程网络上没有多播信息的接收者，Router-A 到 Router-D 链路上的多播传递会被修剪掉。



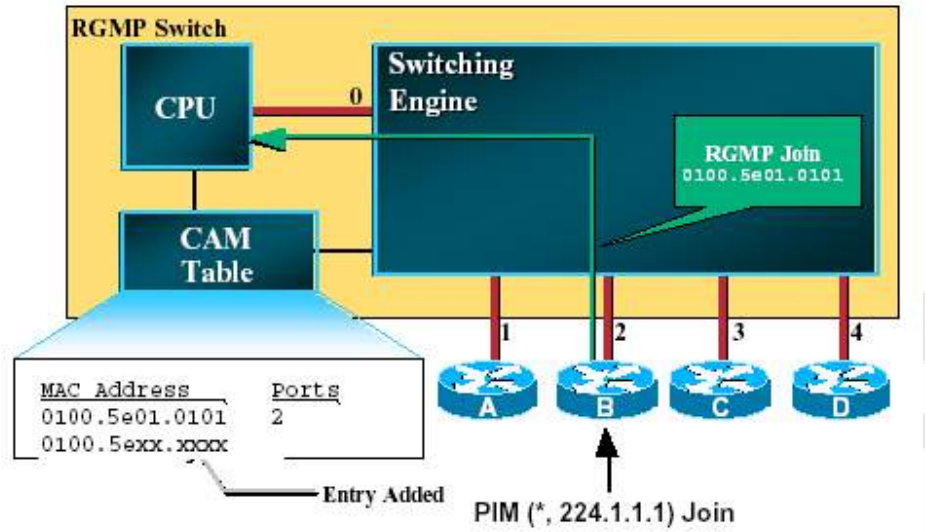
可是上面的图形中仍有问题没有解决，Router B 和 Router C 都收到了一组它们并不需要的多 播信息，因为路由器之间靠 PIM 而不是 IGMP 协议来传递多播转发信息，第二层的交换机不知晓PIM 包的含义，只能假定路由器需要所有多播组的数据流，为了让核心交换机能够

在路由器之间准确转发多播信息，我们必须增加第二层上的多播组转发信息通告方法。

RGMP（Router Group Management Protocol）路由器组管理协议

RGMP 运行于核心路由器和交换机之上，路由器之间通过 RGMP Hello/Bye 消息相互确认身份，并为交换机发送专用的二层加入/退出消息（*，G）。交换机缺省的情况下不向路由器接 口转发多播信息，除非路由器发出特定请求。

RGMP 只能与 PIM-SM 协议同时应用，且只能用在多播骨干路由器所在的 VLAN，该 VLAN内不能存在多播主机，因为多播的洪泛传递缺省的情况下是关闭的。

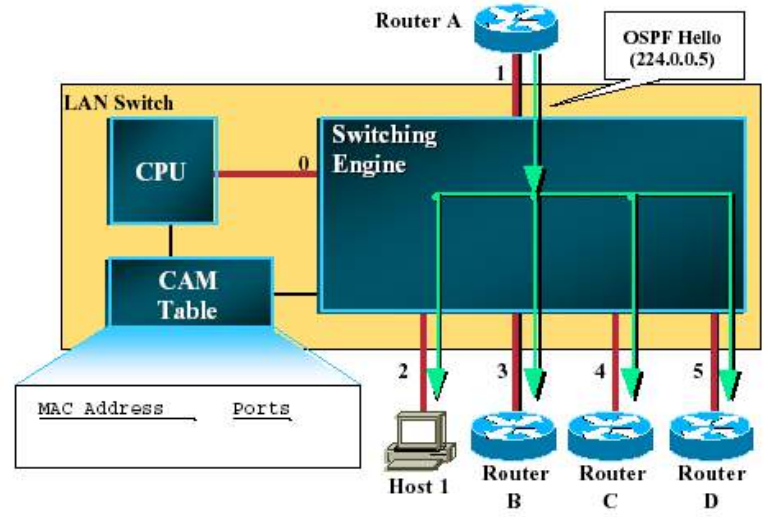


我们看上图中的例子，最开始的时候，多播转发表为空，当路由器B通过PIM协议发出一个加入多播组 224.1.1.1 的消息后，交换机才为其增加相应转发条目（如图中所示）。

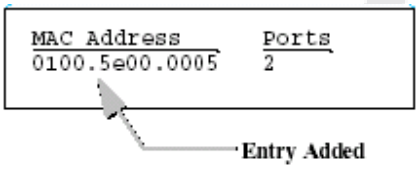
路由器 D 也发出加入消息后，CAM 表变成如下形式：

MAC Address	Ports
0100.5e01.0101	2,4
0100.5exx.xxxx	

依靠这种方法，交换机可以根据路由器的请求准确的在共享同一 VLAN 的路由器之间转发多播信息。



上图中的交换机连接了四台运行 OSPF 路由协议的路由器和一台主机，为了保证一些重要的多播流不被交换机阻断，思科交换机被设计成无条件洪泛传递发往224.0.0.x的多播数据， 此时如果Host 1的用户发送一个加入多播组 224.0.0.5 的IGMP报告，则交换机的CAM表中会增加相应的选项，变成下面的样子：



思科对这一问题的解决办法是强制交换机总是洪泛传递发往 0100.5E00.00xx的多播信息， 但是这可能会对低端交换机的性能产生较大影响。

最后提醒你的是要注意三层组播地址到二层组播地址映射时的 32：1 的重叠问题，在选择组播地址的时候防

止将那些会出现重叠的组播地址用在同一个二层网络中。

PIM-DM

Basic introductions

密模式假定组播接收者遍布整个网络且分布密度较大。

初始状态下，路由器洪泛传递多播流量到符合下述条件的接口：

——连接到另一个 PIM-DM 邻居；

——该接口下有直接相连的组成员；

——接口被手工配置为加入一个组。

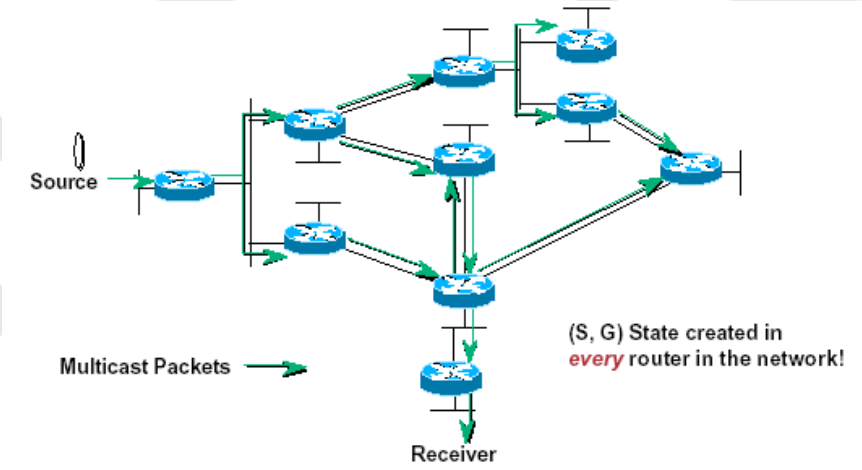
不需要该组播数据的分支可通过 Prune 消息裁剪掉，但是这种情况大约只能维持三分钟，因为三分钟后路由器将重新执行 Flood 过程。所以，在使用密模式的网络中，充足的带宽是必要的。在密模式中，多播状态的创建过程由多播数据的到达触发。当路由器收到由源 S 发往组 G 的多播消息时，关于源组（S，G）的转发树随即生成，连续的三分钟没有收到该多播消息， 则源组（S，G）的转发树被删除。密模式只使用源树，不使用共享树，源树的维持很简单，但是当网络结构变化时，可能会导 致多播信息转发不稳定，临时性丢失数据的情况也可能发生。

组播成员用 Graft 消息来通知路由器它对某个刚刚被裁剪掉的组播数据的请求，根据前面的 介绍，我们知道这不是必须的，因为成员只要耐心的等上至多三分钟，下一次 Flood 就可以 满足它的要求，但是 Graft 机制可以减少这一延迟。如果两台路由器都转发同样的（S，G）多播流到一个公共的多路访问局域网，重复流量将会出现， Assert 机制将最终决定其中一台路由器的转发行为停止以避免重复流量出现。

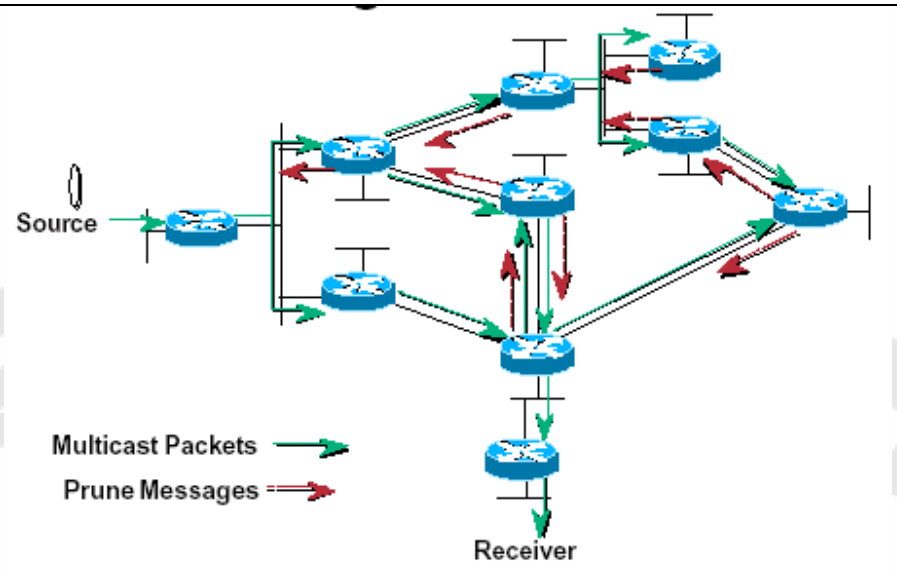
如果路由器从一个非RPF接口（到达该多播源的单播数据途经接口以外的所有其它接口） 收到了多播数据，且这个接口是点对点类型，那么 Prune 消息将立即发送给上游路由器。（如果不是 P2P 接口，将由 Assert 机制做出最终裁定。）

当一个非 RPF 点对点接口不断收到多播数据时，速率限制 Prune 消息将被发送，当一个 RPF 点对点接口不再需要一组多播数据时也会发送速率限制 Prune 消息。

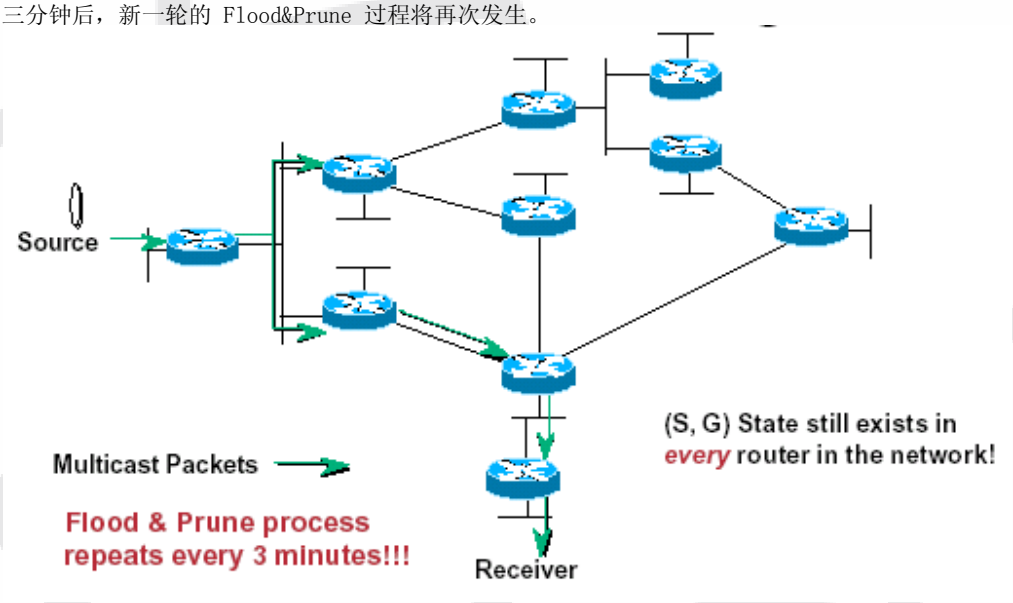
下图是最初的网络，所以的路由器都有（S，G）记录，多播数据被 Flood 传送。



接下来所有路由器通过其不需要该多播数据的接口向上游路由器发送 Prune 消息。



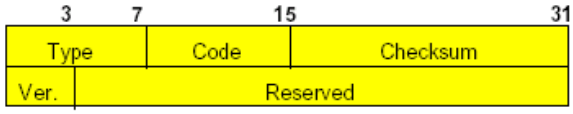
下图是最终形成的转发树,因为只有图中最下方的网络中存在多播的接收者,但是这种局面只能维持三分钟，三分钟后，新一轮的 Flood&Prune 过程将再次发生。



PIM Packet Formats

PIM 各种数据包格式，其中包括：

- Packet Headers（包头）
- Hello Messages（问候消息）
- Join/Prunes（加入/剪裁）
- Grafts/Graft Acks（嫁接/嫁接应答）
- Asserts（裁定）
- PIM Registers（PIM 注册） //仅在PIM-SM中使用
- PIM Register-Stop（PIM 注册停止） //仅在PIM-SM中使用



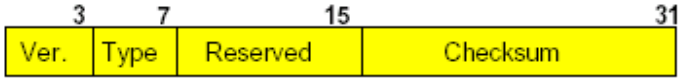
PIMv1 used IGMP Packets

Type:
0x14 = PIM Message

Code:
0 = Router-Query
1 = Register (SM only)
2 = Register-Stop (SM only)
3 = Join/Prune
4 = RP-Reachibility (SM only)
5 = Assert
6 = Graft
7 = Graft-ACK

Ver:
PIM Version = 1

上图中显示的是 PIM 版本 1 中的数据包头格式及主要字段的含义，PIM 版本 1 使用 IGMP协议向 224.0.0.2（All Routers）发送 TTL=1 的数据包，这意味着控制信息不会跨越路由器传递。

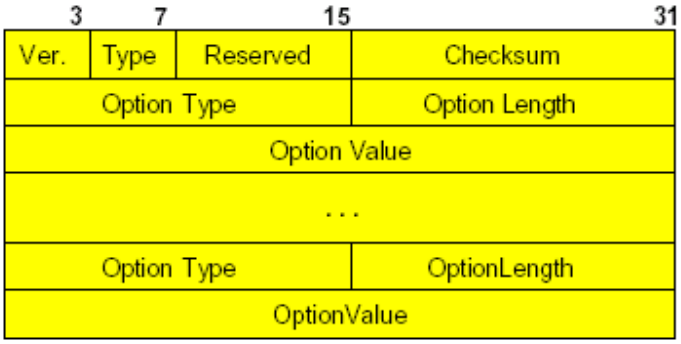


Ver:
PIM Version = 2

Type:
0 = Hello
1 = Register (SM only)
2 = Register-Stop (SM only)
3 = Join/Prune
4 = Bootstrap (SM BSR only)
5 = Assert
6 = Graft (DM only)
7 = Graft-Ack (DM only)
8 = C-RP-Announcement (SM BSR only)

上图是PIM版本2的包头，它与版本1最大的区别在于它不再依靠IGMP协议传送，而使用协议号为 103 的 PIM 专用协议，TTL=1 的 PIM 协议包发往 224.0.0.13（All PIM Routers）。

下面的各种数据包我们将只介绍 PIM 版本 2 的格式。



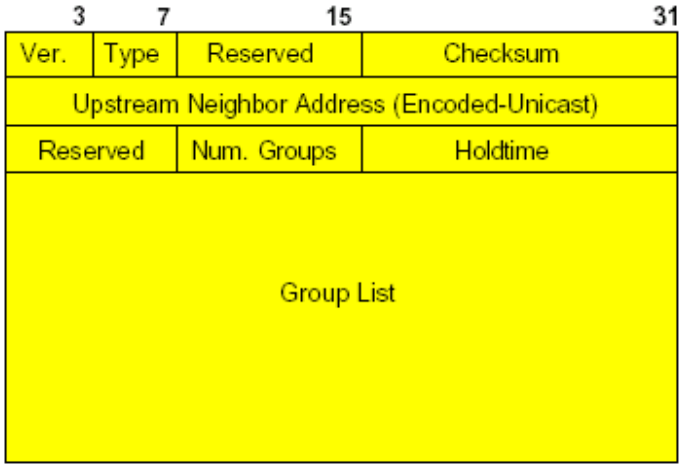
上图是版本 **2** 的 **Hello** 包格式。这个 **Hello** 包用来建立和维持邻居关系，每个 **PIM** 路由器都要周期性的发送这个包向邻居 PIM 路由器通告自己的存在。

Hello 包当中有很多 TLV（Type、Length、Vaule），常用的有以下几种：

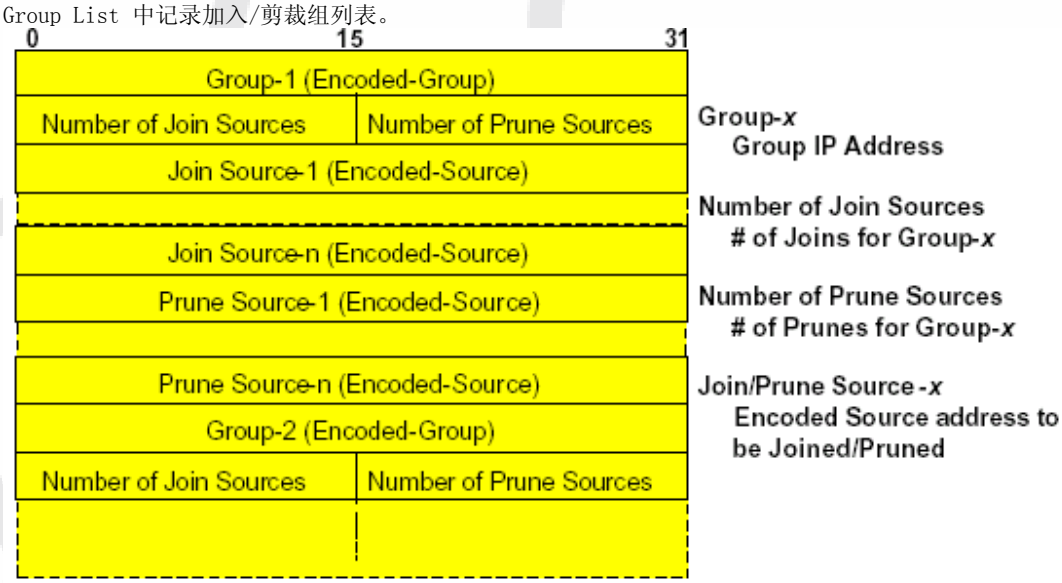
Holdtime（保持时间）收不到来自邻居设备的 Hello 包时，认为邻居不存在之前可以等待的时间。

DR Prioirty（DR 优先级）在子网中选择 DR 时使用。

Generation ID（衍生 ID）一个随机的 32 位的值，用于决定检测到失败后多长时间需要重新激活邻居关系。



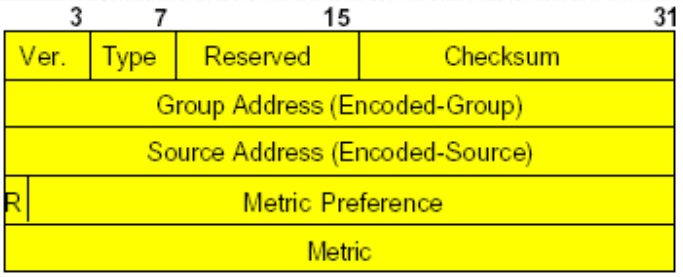
上图是 PIM 加入/剪裁消息包，Upstream Neighbor Address（上游邻居地址）用来记录 RPF路径中上游路由器的单播地址，Holdtime 表示此信息包的失效时间，Num.Groups 用来表示此包中包含的组的数量，Group List 中记录加入/剪裁组列表。



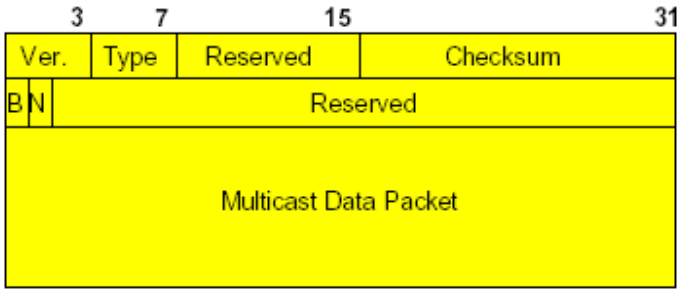
上图中显示的是 Group List 部分的内容，每一个多播组一个条目，其中详细记录了欲加入或剪裁的多播源列表。后面要介绍的 Graft/Graft-ACK 消息包中也包含这一部分。图中的组地 址和源地址都采用了特殊的编码形式，其中可以包含多种地址类型，虽然我们只介绍基于

IP 地址的 PIM 协议。

Graft/Graft-ACK 消息包格式与前面介绍的 Join/Prune 消息包的格式完全相同，不再列出。 与其它包不同的是，它是所有 PIM 消息包中惟一要求可靠传输的消息包（需要确认）。

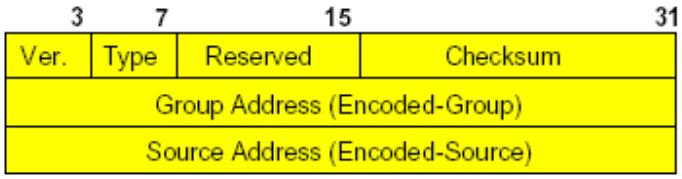


上图是 Assert 包的格式，当网络中存在多播消息转发的冗余路径时，Assert 机制用来裁定哪一个路由器保持转发，比较的原则是：路由协议不同时比较 Metric Preference（通常是管理距离），路由协议相同时比较 Metric，度量值相同时则比较IP地址（IP地址高者胜出）。



上面是仅在 PIM SM 中采用的 Register 消息包格式，DR（多播源的下一跳路由器）用它来向 RP 注册多播的发送，以便通过它将多播消息沿共享树传递到每一个多播接收者，DR 会 持续发送这个消息直到收到来自 RP 的 Register-Stop 消息。

下面图中显示的就是 Register-Stop 消息包的格式，当 RP 已经加入到达 DR 的源树并从 SPF所确定的路径接收到来自特定源的多播数据时，用它来通知 DR 停止发送注册消息。



Neighbor Discovery 邻居发现

路由器通过周期性的查询（缺省 30 秒）来发现 PIM 邻居路由器的存在，对于多路访问网络， 这种查询消息发现 224.0.0.2（All-Routers）。对于多路访问网络，还要选举 DR，IP 地址最高的路由器成为 DR。在疏模式中，DR负责为多路访问网段上的主机向 RP 发送 Join 消息，在密模式中，DR 的选举没有意义。

```
wan-gw8>show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface          Uptime    Expires   Mode
171.68.0.70       FastEthernet0      2w1d      00:01:24  Dense
171.68.0.91       FastEthernet0      2w6d      00:01:01  Dense  (DR)
171.68.0.82       FastEthernet0      7w0d      00:01:14  Dense
171.68.0.86       FastEthernet0      7w0d      00:01:13  Dense
171.68.0.80       FastEthernet0      7w0d      00:01:02  Dense
171.68.28.70      Serial2.31         22:47:11  00:01:16  Dense
171.68.28.50      Serial2.33         22:47:22  00:01:08  Dense
171.68.27.74      Serial2.36         22:47:07  00:01:21  Dense
171.68.28.170     Serial0.70         1d04h     00:01:06  Dense
171.68.27.2       Serial1.51         1w4d      00:01:25  Dense
171.68.28.110     Serial3.56         1d04h     00:01:20  Dense
171.68.28.58      Serial3.102        12:53:25  00:01:03  Dense
```

上图中显示的是 **show ip pim neighbor** 命令的输出，我们可以清楚的看到上面五条记录中的路由器和这台路由器同属于一个多路访问网络，171.68.0.91 因为有最高的 IP 地址而成为DR。

PIM State

从单个路由器的角度看，PIM的状态表现为它对多播分布树的理解，从整个多播网络来看，PIM的状态则表现为整个多播流量在网络中的传递情况。对于每个路由器，多播的传递情况通过多播路由表来体现，我们可能通过命令 **show ip mroute**查看。

```
sj-mbone> show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:00:10/00:00:00, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0, Forward/Dense, 00:00:10/00:00:00
    Serial1, Forward/Dense, 00:00:10/00:00:00
    Serial3, Forward/Dense, 00:00:10/00:00:00

(128.9.160.43/32, 224.1.1.1), 00:00:10/00:02:49, flags: T
  Incoming interface: Serial0, RPF nbr 198.92.1.129
  Outgoing interface list:
    Serial1, Forward/Dense, 00:00:10/00:00:00
    Serial3, Prune/Dense, 00:00:05/00:02:55
```

所有条目以（*，G）或（S，G）的形式存在。 每个条目中包含 RPF、OIL 等信息。源接口为通过 RPF 确定的到达多播源的最近路径的出口，地址为上游路由器的接口地址。OIL（Outgoing Interface List 流出接口列表）记录从流入接口进来的多播数据需要转送的所有接口。

（*，G）条目是多播路由器里的缺省条目，通常指向所有运行 PIM 的接口。做为所有（S， G）条目的父条目，当直连接口下的一台主机发出 Join 消息而又没有相应的多播源时，（*， G）条目被创建。因为这种条目并不反映真正的多播消息转发，所以 Incoming interface 为空， OIL 列表中包含所有连有PIM－DM邻居路由器或多播组成员的接口，我们也可以使用命令**ip igmp static-group <group>**静态指定一个接口连于某多播组成员。（S，G）条目与具体的多播源组有关，用于指导路由器如何转发始自特定源的多播数据。IIF 为 RPF 接口或与多播源相连接口，OIL 最初与（*，G）条目的 OIL 相同，并根据 Flood&Prune过程中收到的 Join/Prune 消息做适当的改变。

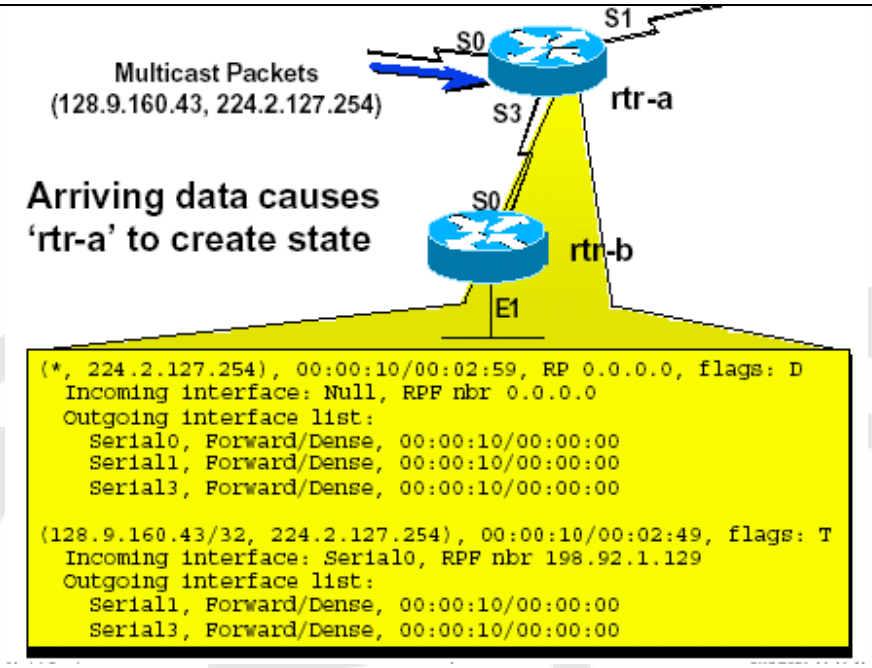
```
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
```

上图中显示的是多播路由中的状态标志：

- **D** 表示密模式，只出现在（*，G）条目中。
- **C** 表示该接口下直接连有多播组的成员。
- **L** 表示路由器自身是一个多播组的成员。
  （如所有的路由器都会自动加入 Auto-RP 发现组224.0.1.40）
- **P** 表示所有 OIL 中的接口都收到了修剪消息，这也意味此路由器不再需要该组多播消息，接下来它会向上游路由器发送 Prune 消息。
- **T** 表示从 SPT 接收到了至少一个数据包。

PIM DM Flooding

多播数据包（128.9.160.43，224.2.127.254）经路由器 rtr-a 的 S0 口到达，路由器在产生（128.9.160.43，224.2.127.254）转发条目前会先产生（*，224.2.127.254）条目，其OIL包括所有支持多播的端口，（ 128.9.160.43 ， 224.2.127.254 ）条目的OIL复制于（*，224.2.127.254）（除去入口 S0），于是128.9.160.43发出的所有到达 224.2.127.254 的多播数据经S0进入后洪泛传递到S1和S3端口。

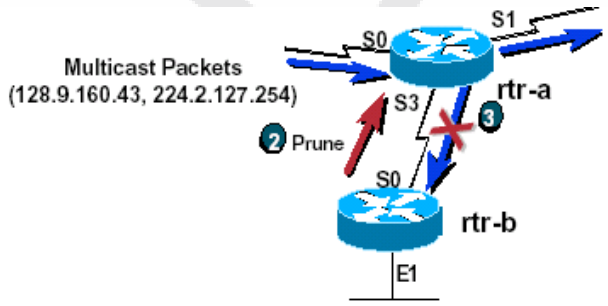


路由器 rtr-b 收到此类多播数据包后其多播路由表如下：

```
(*, 224.2.127.254), 00:00:12/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0, Forward/Dense, 00:00:12/00:00:00

(128.9.160.43/32, 224.2.127.254), 00:00:12/00:02:48, flags: PT
  Incoming interface: Serial0, RPF nbr 198.92.2.31
  Outgoing interface list: Null
```

对于路由器rtr-b，只有S0口参与多播活动，所以其转发表中（128.9.160.43，224.2.127.254）条目的OIL为空，于是该记录带有P标志，rtr-b向rtr-a发送针对该源组的Prune消息.

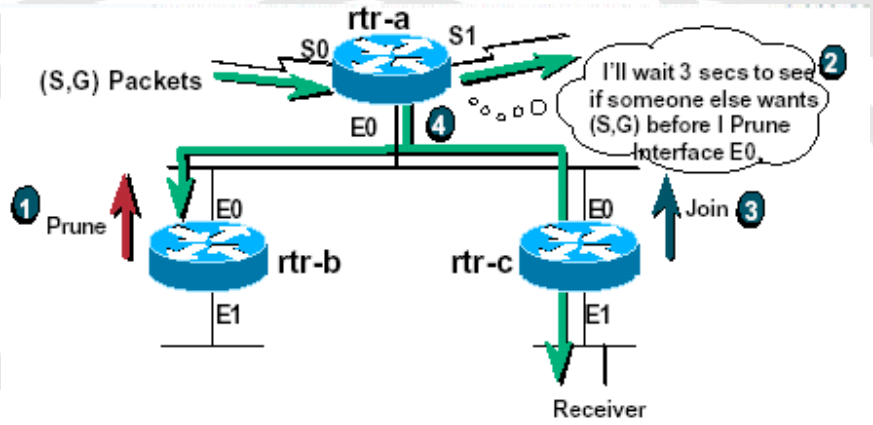
PIM DM Pruning



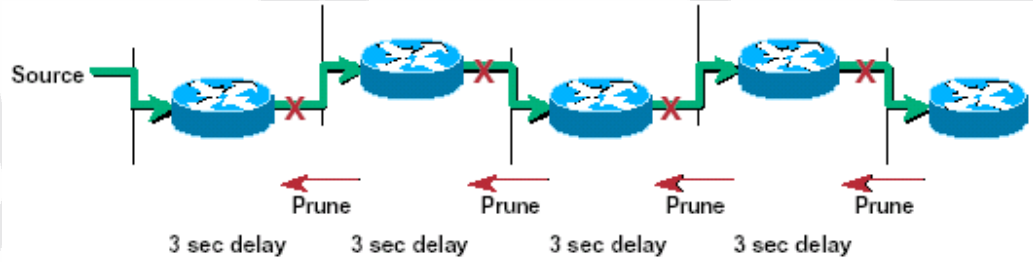
对于多播源组（128.9.160.43，224.2.127.254），rtr-b是叶节点，在它的下面没有任何接收者，于是 rtr-b 通过 S0口向rtr-a发送针对多播源组（128.9.160.43，224.2.127.254）的 Prune 消息，rtr-a 收到此消息后从多播源组（128.9.160.43，224.2.127.254）的 OIL中修剪掉了 S3 端口。下图是修剪之后的rtr-a的多播路由表，S3口已经被标志为 Prune/Dense。

```
(*, 224.2.127.254), 00:00:12/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Serial0, Forward/Dense, 00:00:12/00:00:00
    Serial1, Forward/Dense, 00:00:12/00:00:00
    Serial3, Forward/Dense, 00:00:12/00:00:00

(128.9.160.43/32, 224.2.127.254), 00:00:12/00:02:48, flags: T
  Incoming interface: Serial0, RPF nbr 198.92.1.129
  Outgoing interface list:
    Serial1, Forward/Dense, 00:00:12/00:00:00
    Serial3, Prune/Dense, 00:00:12/00:02:56
```

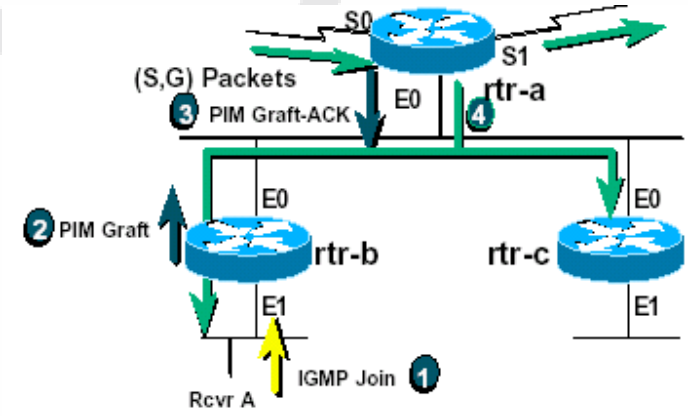
上述过程只适用于 P2P 链路，在多路访问网络中，修剪过程稍有变化，我们看下图：



叶节点rtr-b不需要（S，G）源组的数据包，向rtr-a发送Prune消息，rtr-a在做修剪之前会等待一小段时间，以确定该网段上是否还有其它的接收者存在，rtr-c 收到来自rtr-b的修剪消息后马上会向 rtr-a 发送 Join消息，rtr-a 收到 Join 消息后中止了 Prune 行为。 这种机制可以保证多播接收者对多播数据的接收不被中断，但是在下图所示的网络中却给我们带来麻烦，因为修剪行为在12秒才会发生。



PIM DM Grafting

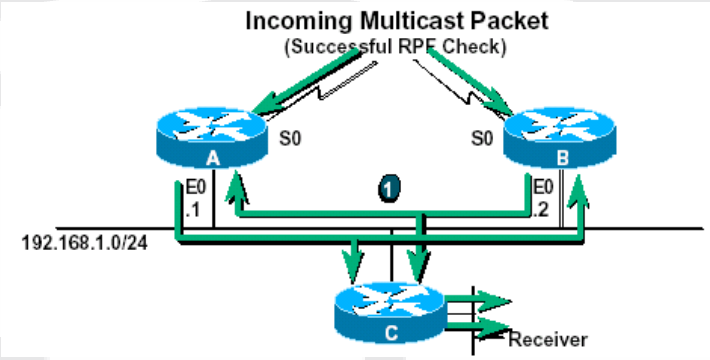


上图中，最开始的时候rtr-b和rtr-c的下面都没有任何源组（S，G）的接收者，于是它们发送Prune消息到rtr-a，在这一多播流被修剪之后，rtr-b 的下面的 Rcvr A 发出了针对源组（S，G）的Join消息。如果rtr-b不发送 Graft 消息，那么Rcvr A 要等 3 分钟之后 rtr-a 再次Flood的时候才能收到 这组多播数据，为了减少这一延

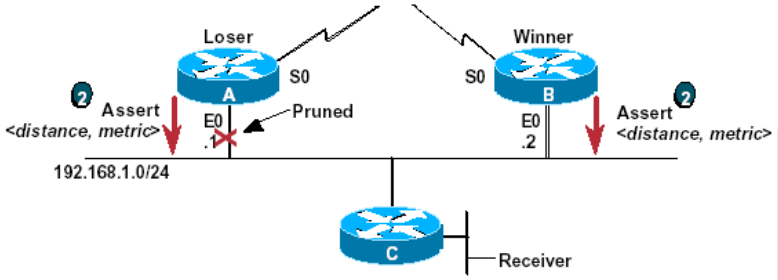
时，rtr-b 向 rtr-a 发送Graft消息，请求重新向其发送源组（S， G）的数据。

看起来似乎可以了，但问题是rtr-c会不会发送Prune消息到rtr-a，基于一个源组的Prune消息只会在该条目创建且 OIL为空的时候发送一次，所以rtr-c只会在3分钟之后，rtr-a 再次 向其发送Flood的多播消息时发送一次，但这不会中断rtr-b的接收，因为rtr-b在收到rtr-c的Prune消息后会立刻向rtr-a 发送 Join 消息覆盖掉 rtr-c 的 Prune 消息。

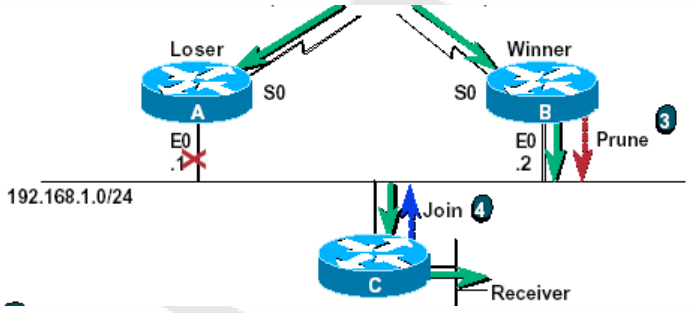
PIM Assert Mechanism



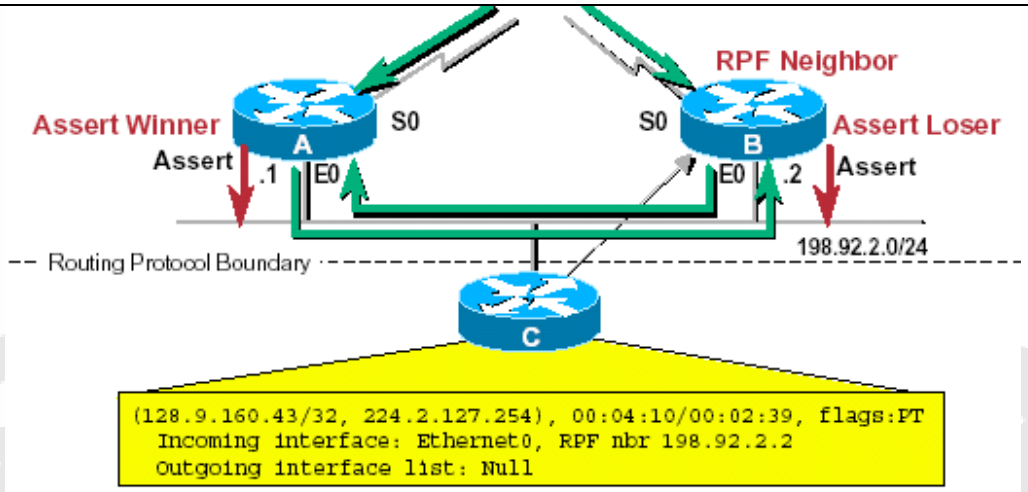
如果不采取措施，C 所连接的接收者会收到两份重复的多播数据。路由器A和B都可以发现这个异常情况，因为它们从一个属于（S，G）的 OIL 的接口收到了该源组的数据！这将触发仲裁过程的发生。



两台路由器会发出仲裁包，经过对管理距离，度量值以及 IP 地址的比较后，路由器 B 胜出， 路由器 A 将其 E0 口从源组（S，G）的路由条目中裁减掉。

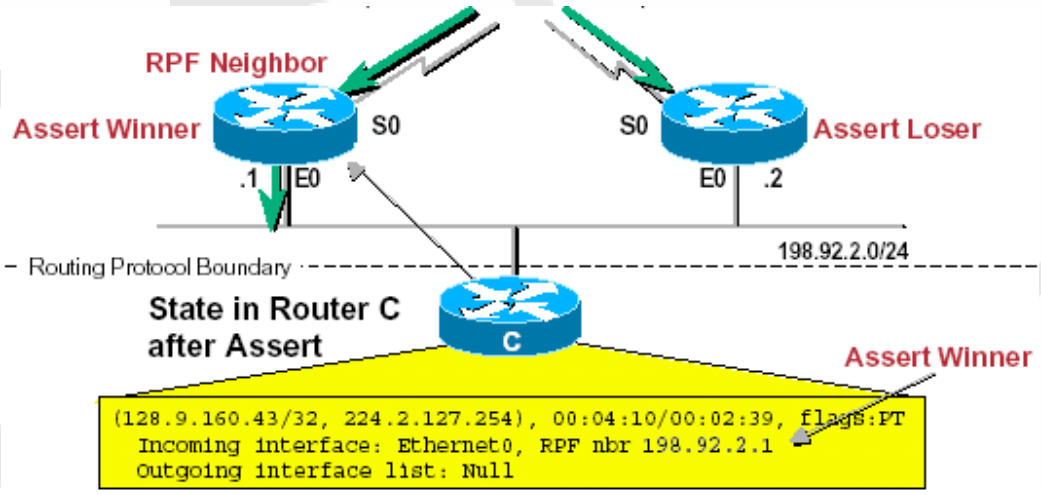


如果在多路访问网段上没有直接的多播接收者，仲裁的胜出者会向多路访问网段发出Prune消息以检测该网段上是否还有其它路由器需要这一多播数据流，因为路由器 C 连有接收者，它会发出 Join 消息覆盖掉 Prune 消息，转发得以保持。 关于仲裁我们还有需要注意的地方，看下图：

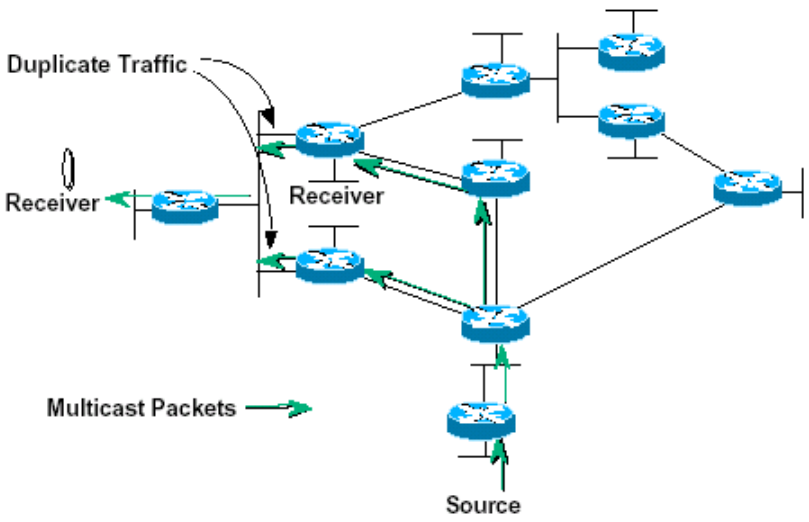


路由器 C 的上游路由器指向了路由器B，但是路由器B在仲裁中失败了。

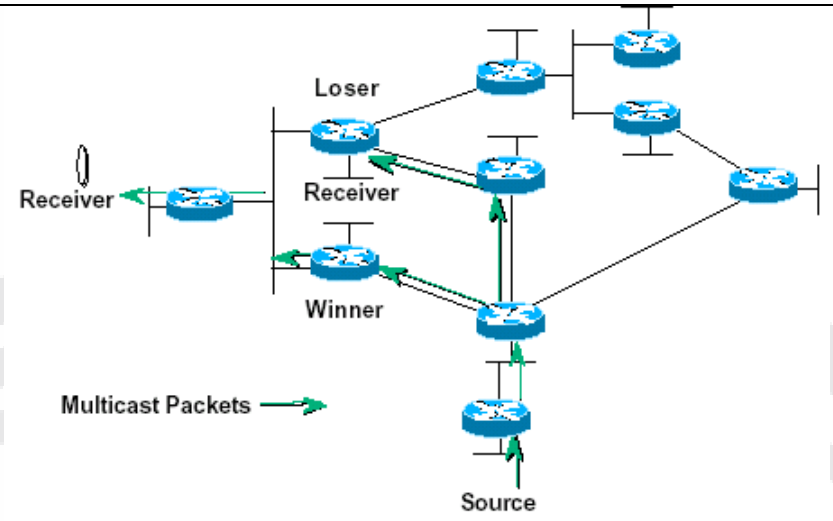
如果C不修改它的记录将上游路由器指向 A 的话，接下来的一段时间里它向上游路由器发出的 Joins，Prunes 或 Grafts 消息都不能对多播消息转发进行有效的控制。好在路由器 C 可 以听到 A 和 B 之间的仲裁信息并能就此作出正确的响应，如下图：



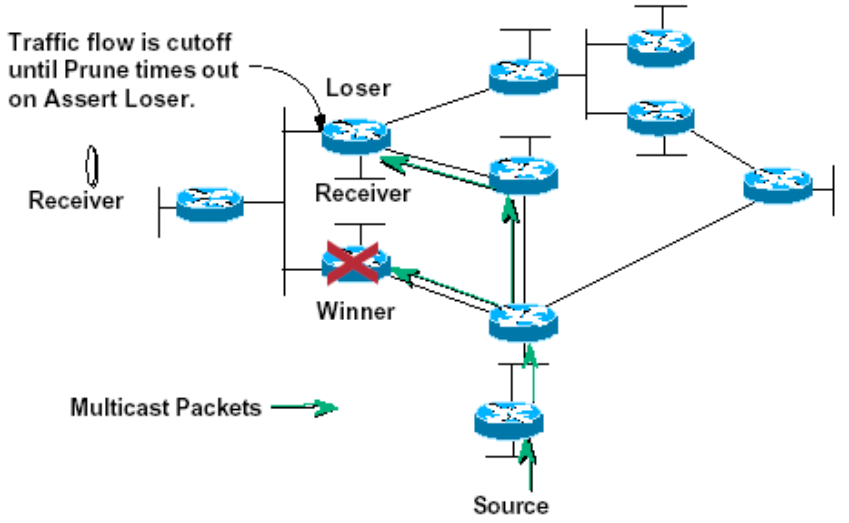
仲裁机制也会带来一些暂时不能很好解决的问题，看下面这幅图：



重复流量引发仲裁，结果下面的路由器胜出，于是新的多播流量以下图所示方式传递：



但是如果那个 Winner 在胜出后马上坏掉了会如何呢？

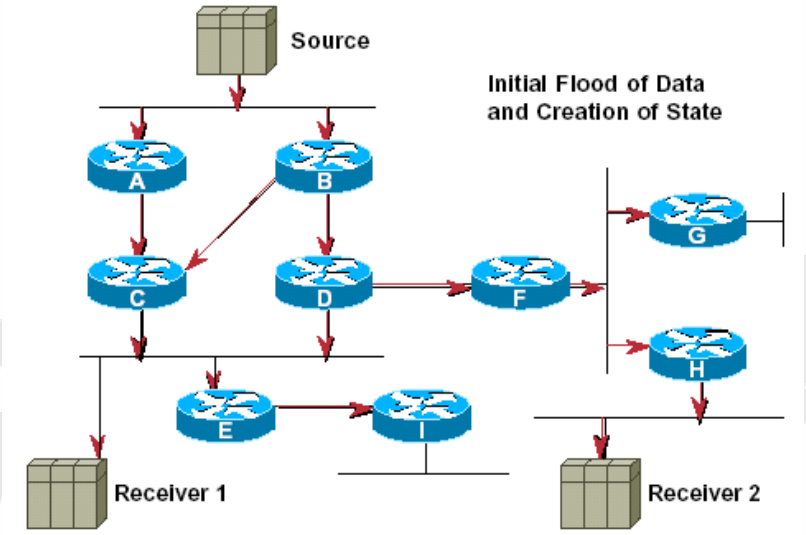


the Loser不能得知这一情况，直到3分钟之后才会重新打开Prune的端口，也就是说，图左侧的接收者将有近3分钟的时间接收不到该多播数据流。

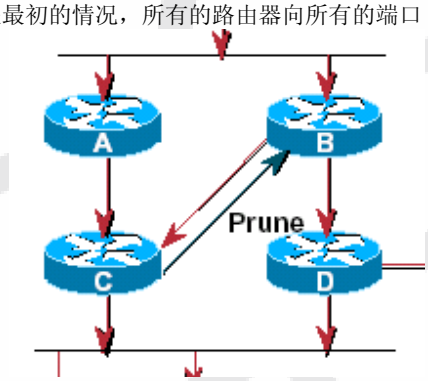
PIM DM State Maintenance

密模式PIM的多播转发信息靠反复的 Flood&Prune 过程来维护，接收到多播数据包会复位（S，G）超时计时器，超时计时器如果减到0，则相应条目会被删除，所有被 Prune 掉的端口在3分钟后会被Flood过程重新启动，不需要的转发会再次被Prune掉。

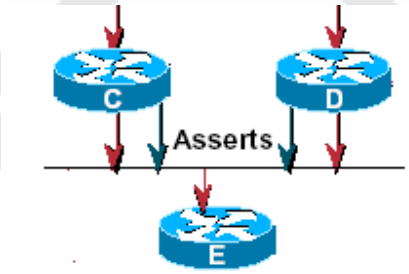
PIM DM review



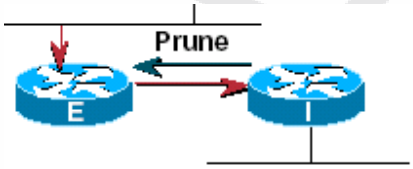
这是最初的情况，所有的路由器向所有的端口 Flood 多播流量。



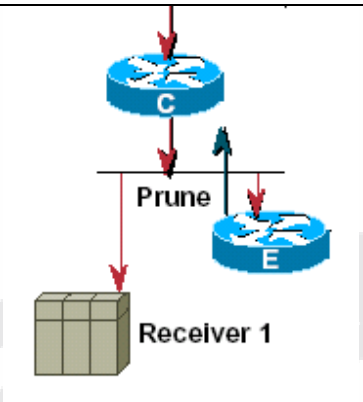
第一个修剪发生在上图所示部分，路由器C与B之间的多播信息没有经过RPF检验，因为C 到达上面的多播源所走的路径是经由 A，C与B之间的流量被减裁掉。



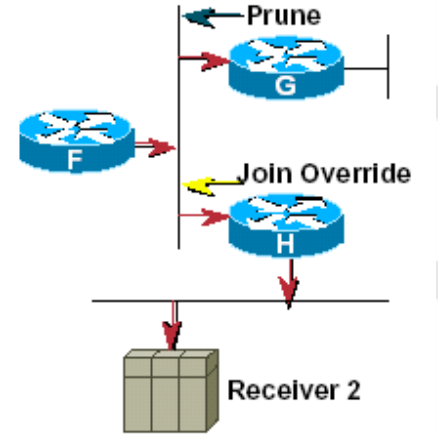
第二个修剪发生在上图所示部分，C和D形成了多播转发的冗余路径，经过仲裁过程，其中的一个会成为胜利者。



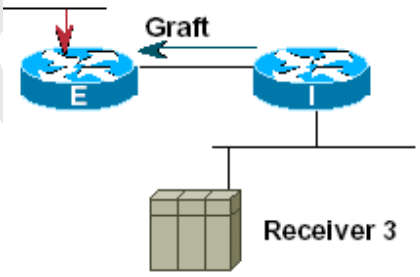
第三个修剪发生在上图所示部分，原因很简单，因为路由器 I 没有与任何接收者相连。



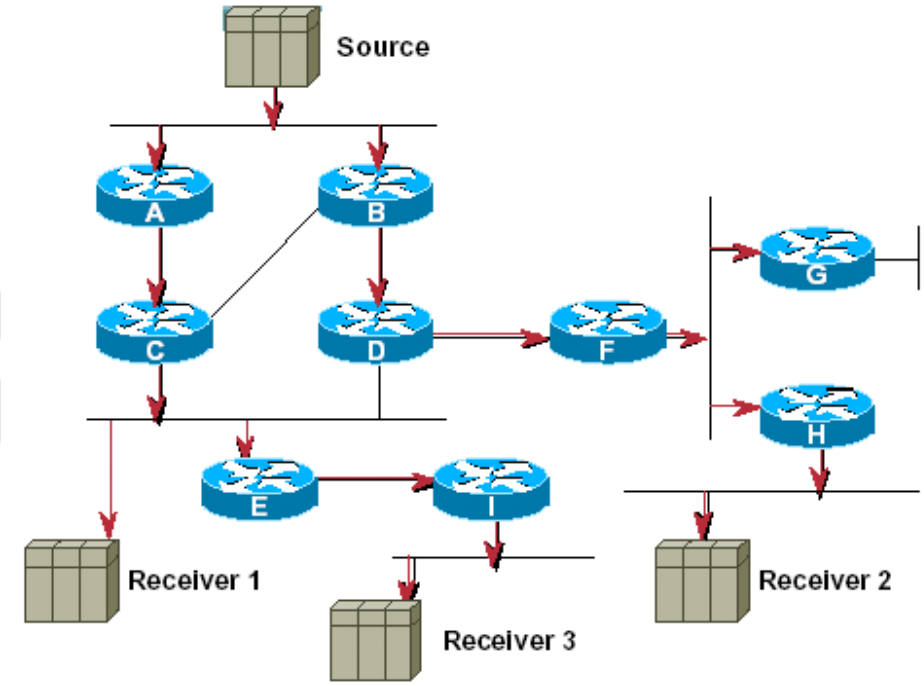
E在收到它右侧的路由器的修剪消息后，向C发出修剪消息，但C不会做出修剪，因为该网段上直接连有一个接收者。



类似的情况出现在上图中，G不需要该组播消息，向F发出修剪消息，但是H需要，所以它在收到了G的 Prune 消息后向F发出的 Join 消息，覆盖掉了G的 Prune 消息。



路由器I下面网段出现了新的接收者，于是路由器I向路由器E发出Grant消息，E重新开始向路由器I的转发。



相关的配置如下

```
ip multicast-routing

interface Ethernet 0
ip address 1.1.1.1 255.255.0.0
ip pim dense-mode

interface Ethernet 1
ip address 2.1.1.1 255.255.0.0
ip pim dense-mode

interface Serial 0
ip address 199.1.1.1 255.255.0.0
ip pim dense-mode
```

Basic Multicast Debugging

Show Commands

show ip igmp groups

```
R4#show ip igmp group
IGMP Connected Group Membership
Group Address    Interface              Uptime    Expires   Last Reporter
224.1.1.1        Ethernet1              3d16h     00:01:59  172.16.7.2
224.0.1.40       Ethernet0              4d15h     never     172.16.6.2
```

Uptime 显示这个成员关系在该接口上存在的时间
Expires 显示这条记录的失效时间,来自客户机的周期性的 IGMP 报告会在记录过期前刷新这个记时器。
Last Reporter 因为报告压制的原因，实际上不管有多少接收者，路由器都将只收到来自一个接收者的报告，当然，如果此报告者停止接收该多播组的消息，会有另一个接收者代替它向 路由器报告这一请求。

show ip igmp interface

```
R4#show ip igmp interface
Ethernet1 is up, line protocol is up
  Internet address is 172.16.7.1, subnet mask is 255.255.255.0
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is disabled on interface
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Inbound IGMP access group is not set
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 172.16.7.1 (this system)
  IGMP querying router is 172.16.7.1 (this system)
  No multicast groups joined
```

用来检验 IGMP 和 CGMP 在接口下是否启用。
用来检验 IGMP 版本，我们前面已经讨论过，使用不同版本的IGMP的路由器混合在一起的时候我们可能需要进行一些针对 IGMP 版本的特殊设置。
在这里还可以看到关于 IGMP 的各种计时器，我们可以基于性能的原因来调整它。另外，我们还可以用这条命令找出某一子网中的 DR 和 IGMP 查询者。

show ip pim neighbor

```
R6#show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface          Uptime    Expires   Mode
172.16.10.2       Serial0            4d15h     00:01:19  Dense
172.16.11.2       Serial1            4d15h     00:01:00  Dense
172.16.9.1        Ethernet0          4d15h     00:01:00  Dense
```

Uptime 显示毗邻关系存在的时间
Expires 显示条目过期前的剩余时间，周期性的 PIM Hello 消息用来刷新这条记录。
Mode 显示接口下 PIM 的模式：Dense，Sparse or Sparse/Dense。

show ip pim interface

```
R6#show ip pim interface
Address          Interface          Mode          Nbr   Query     DR
                                                  Count Intvl
172.16.10.1      Serial0            Dense         1     30      0.0.0.0
172.16.11.1      Serial1            Dense         1     30      0.0.0.0
172.16.9.2       Ethernet0          Dense         1     30      172.16.9.2
```

Nbr Count 显示该接口连接网络中邻居的数量。
DR=0.0.0.0 说明该接口所连接的是 P2P 网络，不存在DR。

show ip rpf

```
R4#show ip rpf 172.16.8.1
RPF information for R1 (172.16.8.1)
  RPF interface: Ethernet0
  RPF neighbor: R3 (172.16.6.1)
  RPF route/mask: 172.16.8.0/255.255.255.0
  RPF type: unicast

R4#sh ip rpf 172.16.12.2
RPF information for Source1 (172.16.12.2)
  RPF interface: Ethernet0
  RPF neighbor: R6 (172.16.11.1)
  RPF route/mask: 172.16.12.0/255.255.255.0
  RPF type: unicast
```

上图中显示了两条 RPF 信息，这些信息都是基于单播路由表形成的。

关于每一个多播源和 RP 路由器都有相关条目，记录 RPF 接口，上游路由器，相关路由等信 息。

show ip route

```
R4#show ip route
Gateway of last resort is not set

     172.16.0.0/24 is subnetted, 7 subnets
D       172.16.2.0 [90/2354611] via 172.16.6.1, 4d15h, Ethernet0
D       172.16.3.0 [90/2354611] via 172.16.6.1, 4d15h, Ethernet0
D       172.16.4.0 [90/2221056] via 172.16.6.1, 4d15h, Ethernet0
D       172.16.5.0 [90/2221056] via 172.16.6.1, 4d15h, Ethernet0
C       172.16.6.0 [90/2281542] via 172.16.6.1, 4d15h, Ethernet0
D       172.16.10.0 [90/2281542] via 172.16.6.1, 4d15h, Ethernet0
D       172.16.8.0 [90/2221056] via 172.16.6.1, 4d15h, Ethernet0
     192.169.1.0/24 is subnetted, 1 subnets
D       192.169.1.0 [90/2349056] via 172.16.6.1, 3d15h, Ethernet0
```

不要忘了，多播的转发决断依赖于单播路由表（如RPF检验），所以在我们调试多播路由时经常检验单播路由的正确性同样重要。

show ip mroute

```
barrnet-gw>show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
Timers: Uptime/Expires
Interface state: Interface, Next-Hop, State/Mode
(*, 224.2.130.100), 00:18:53/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Fddi1/0, Forward/Dense, 00:09:20/00:02:38
    Hssi3/0, Forward/Dense, 00:18:53/00:00:00
(208.197.169.209/32, 224.2.130.100), 00:18:53/00:02:27, flags: T
  Incoming interface: Hssi3/0, RPF nbr 131.119.26.9
  Outgoing interface list:
    Fddi1/0, Forward/Dense, 00:16:16/00:02:38
(*, 239.100.111.224), 05:35:08/00:02:58, RP 171.69.10.13, flags: DP
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list: Null
```

这是一个多播路由表的一部分，同其它的多播路由表一样，其中包含（*，G）和（S，G） 条目，每个条目下会有流入接口，OIL 流出接口列表，RP（如果存在的话），标志位和Uptime/Expires计时器。

此外还有 **show ip mroute summary** 和 **show ip mroute count** 两条命令没有列出示例。 分别用于查看多播路由表每个条目的汇总信息和统计信息。

show ip mroute active

```
barrnet-gw>show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps

Group: 224.2.154.118, Radio Bandit
   Source: 192.36.125.68 (falcon.pilsnet.sunet.se)
     Rate: 11 pps/30 kbps(1sec), 30 kbps(last 33 secs), 23 kbps(life avg)

Group: 224.2.246.13, UO Presents KWAX Classical Radio
   Source: 128.223.83.204 (d83-204.uoregon.edu)
     Rate: 24 pps/69 kbps(1sec), 72 kbps(last 2 secs), 70 kbps(life avg)

Group: 224.2.180.115, ANL TelePresence Microscopy Site
   Source: 146.139.72.5 (aem005.amc.anl.gov)
     Rate: 1 pps/5 kbps(1sec), 9 kbps(last 52 secs), 12 kbps(life avg)
...
```

显示当前活动的源（流量大于某个门限值，缺省为 4K） 其中包含组地址，会话名称，组播源地址，域名称以及带宽占用情况。

show ip pim rp mapping

```
sjck-rp1>show ip pim rp mapping
PIM Group-to-RP Mappings
This system is an RP (Auto-RP)
This system is an RP-mapping agent (Loopback1)

Group(s) 224.0.0.0/4
  RP 171.69.10.13 (sj-mbone-loopback0.cisco.com), v2v1
    Info source: 171.69.10.13 (sj-mbone-loopback0.cisco.com), via Auto-RP
         Uptime: 4w4d, expires: 00:02:55
Group(s) 239.192.111.0/24
  RP 192.168.165.15 (sjc25b-00rp-gw1-loop1.cisco.com), v2v1
    Info source: 192.168.165.15 (sjc25b-00rp-gw1-loop1.cisco.com), via Auto-RP
         Uptime: 1d18h, expires: 00:02:35
```

这条命令是关于 PIM Sparse Mode 的。
RP 可以被自动发现，也可以被手工设置。
RP 在一个多播网络中可以存在多个，并且可以基于组播地址范围进行转发分工。

Debug Commands

debug ip igmp

```
R4# debug ip igmp
IGMP: Send v2 Query on Ethernet1 to 224.0.0.1
IGMP: Received v2 Report from 172.16.7.2 (Ethernet1) for 224.1.1.1
IGMP: Received v2 Query from 172.16.6.1 (Ethernet0)
IGMP: Set report delay time to 2.2 seconds for 224.0.1.40 on Ethernet0
IGMP: Send v2 Report for 224.0.1.40 on Ethernet0
IGMP: Received v2 Report from 172.16.6.1 (Ethernet0) for 224.0.1.40
IGMP: Received v2 Report from 172.16.6.1 (Ethernet0) for 224.0.1.40
```

这条命令可以帮助我们了解当前路由器是否发出查询信息，发查询信息的间隔以及是否收到来自接收者的对查询的响应。

debug ip mpacket

```
R6#  debug ip mpacket 224.1.1.1 detail
IP: MAC sa=00e0.b063.cf4b (Ethernet1), IP last-hop=172.16.12.2
IP: IP tos=0x0, len=100, id=0x175, ttl=254, prot=1
IP: s=172.16.12.2 (Ethernet1) d=224.1.1.1 len 114, mroute olist null
```

用来解码多播包，这是一条轻易不要用的命令，特别是当有较大多播流量通过路由器时，输出信息会搞 得你眼花缭乱，更会使路由器晕头转向。

debug ip mroute

```
R6# debug ip mrouting 224.1.1.1
MRT: Create (*, 224.1.1.1), RPF Null, PC 0x6032D254
MRT: Create (172.16.12.2/32, 224.1.1.1), RPF Ethernet1/0.0.0.0, PC x6032D378
```

用来动态监视多播路由表的变化。

debug ip pim

```
R4#  debug ip pim 224.1.1.1
PIM: Send Router-Query on Ethernet0
PIM: Send Router-Query on Ethernet1
PIM: Received Router-Query on Ethernet0 from 172.16.6.1
```

用来查看 PIM 邻居之间的消息交换，图中显示该路由器通过 E0和E1口向外发出 PIIM 路由器查询，但只在E0口上收到了来自 172.16.6.1 的回应。

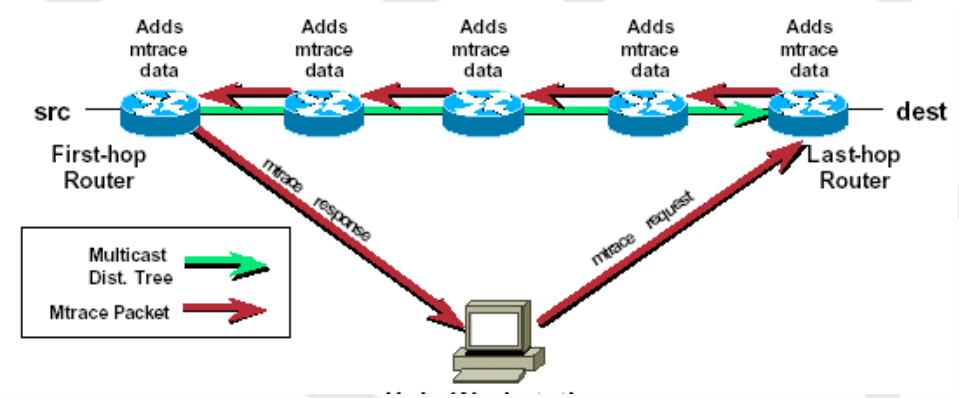
```
R4#
PIM: Building Join/Prune message for 224.1.1.1
PIM: For RP, Join-list: 172.16.8.1/32, RP-bit, WC-bit
PIM: Send periodic Join/Prune to RP via 172.16.6.1 (Ethernet0)
PIM: Received RP-Reachable on Ethernet0 from 172.16.8.1
     for group 224.1.1.1
PIM: Update RP expiration timer (270 sec) for 224.1.1.1
```

现在看到的还是 debug ip pim 的输出，记录的是此路由器向 RP（172.16.8.1）发送注册消息并收到 RP 发回的 RP-Reachable 消息。

Other useful commands

mtrace/mstat

源于 UNIX 系统中的 mtrace 命令，它的工作方式很特殊，让我们通过下面的图来进行解释：



上图中，绿色箭头表示从源到接收者的多播流传递路径，红色箭头表示mtrace包传递的路径。mtrace 采用专用的 IGMP 包类型，0x1F 表示查询/请求，0x1E 表示响应。查询/请求包发往与接收者相邻的末跳路由器，末跳路由器将它转为单播的 traceroute 请求，按单播路由上溯到与源相邻的首跳路由器，在此过程中，每个经由的路由器都将查询到达时间、流入接口、流出接口、上一跳路由器地址、输入输出包计数、源/组包计数、路由协议、TTL 门限 值以及转发/错误码等信息记入包中。

首跳路由器加入自己的响应数据后将单播 traceroute 信息包转换为 mtrace 类型送回查询主机。类似于单播的 traceroute 命令，mtrace 也显示多播信息的转发路径及各段延时。我们可以用它来查找多播信息传递过程中的中断点，也可以用它来发现次佳路由。

```
dallas-gw>mtrace bloom-iptv-svr bwilliam-ss5 224.2.156.43
Type escape sequence to abort.
Mtrace from 172.17.67.43 to 171.68.37.121 via group 224.2.156.43
From source (?) to destination (bwilliam-ss5.cisco.com)
Querying full reverse path...
 0  bwilliam-ss5 (171.68.37.121)
-1  dallas-gw (171.68.37.1) PIM  thresh^ 0  3 ms
-2  wan-gw4 (171.68.86.193) PIM  thresh^ 0  32 ms
-3  bloomington-mn-gw (171.68.27.2) PIM  thresh^ 0  717 ms
-4  bloom-mnlab (171.68.39.28) PIM  thresh^ 0  730 ms
-5  bloom-iptv-svr (172.17.67.43)
dallas-gw>
```

上图显示的是 mtrace 命令的输出示例，其中显示了多播数据包传递时经由的路径及每一跳的时延。

mstat 命令以图形的方式输出多播转发路径，可用于分析网络中任意两点间的多播信息转发情况，报告显示每个节点丢弃/重复包的数量以及 TTL 值和延时。主要用于在网络中查找存在大量包丢弃或包复制的节点。

```
dallas-gw>mstat 172.17.67.43 bwilliam-ss5 224.2.156.43
  Source         Response Dest     Packet Statistics For      Only For Traffic
172.17.67.43     171.68.86.194     All Multicast Traffic      From 172.17.67.43
     |        __/  rtt 547  ms    Lost/Sent = Pct  Rate      To 224.2.156.43
     v       /     hop 547  ms    ---------------------      ---------------------
172.17.67.33
171.68.39.28     bloom-mnlab
     |      ^      ttl   0
     v      |      hop -409 ms     -11/168 = --%  16 pps      0/67 = 0%  6 pps
171.68.39.1
171.68.27.2      bloomington-mn-gw
     |      ^      ttl   1
     v      |      hop 379  ms     -9/170 = --%   17 pps     -3/67 = --%  6 pps
171.68.27.1
171.68.86.193    wan-gw4
     |      ^      ttl   2
     v      |      hop 28   ms     -3/195 = --%   19 pps      0/70 = 0%   7 pps
171.68.86.194
171.68.37.1      dallas-gw
     |      \__    ttl   3
     v         \   hop 0    ms         196        19 pps           70    7 pps
171.68.37.121    171.68.86.194
  Receiver       Query Source
```

上图是 mstat 的输出示例，注意：重复包的数量在 Lost 列中以负值表示。

PIM Sparse Mode

PIM Sparse Mode Overview

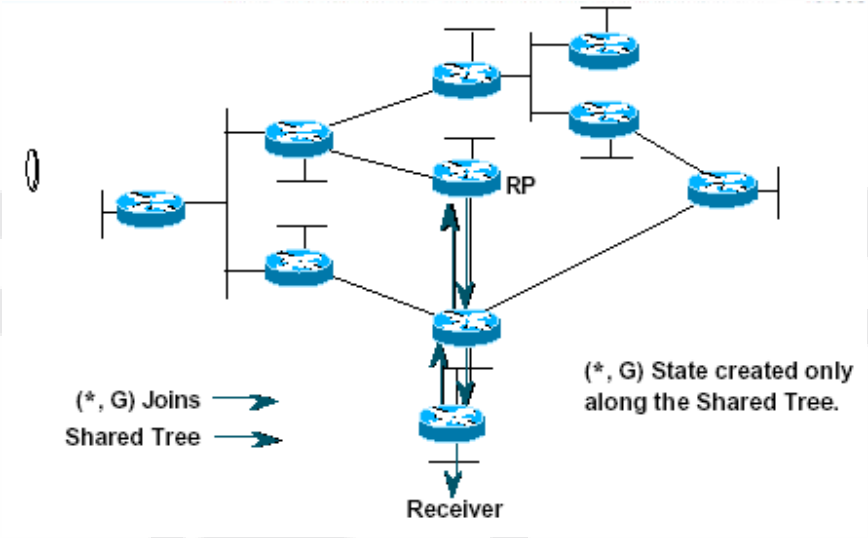
PIM SM是显示加入模式，接收者向RP（会聚点）发送PIM加入消息通告其加入组播的行为，发送 者也要发送注册消息到RP进行注册。RP是共享树的根，由源发出的数据先送至RP，再由RP沿共享树传递到每个接收者所在的网络。

经过RP的转发可能不是最佳路径，这时最后一跳路由器可以修正经RP沿共享树的转发为沿基于源的SPT（Shortest Path Tree）的转发。在共享树中，RPF检验使用RP的地址；在SPT中，RPF检验使用多播源地址。

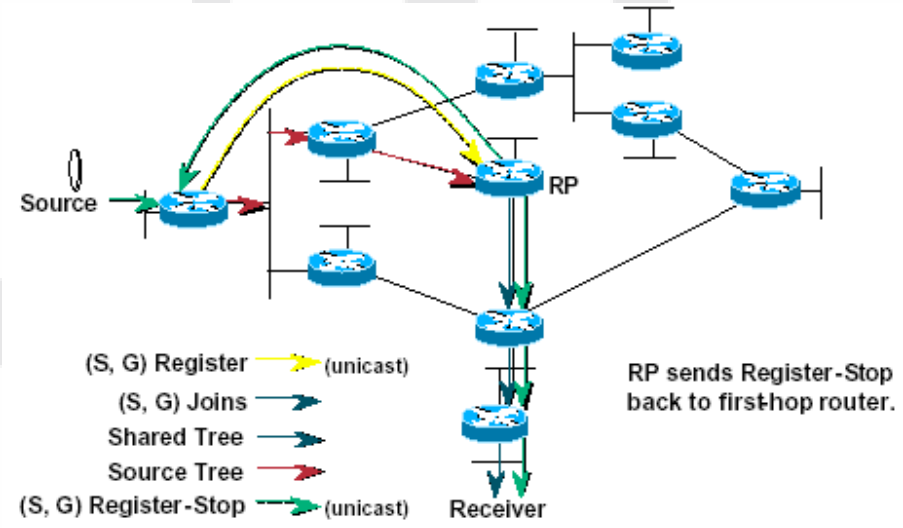
通常一个多播网络中只使用一个RP,但是我们可以借助访问列表基于组地址范围将多播信息转 发路径的管理分派给多个RP，它们可以位于网络中的不同位置，这有利于优化数据流的转发， 一个多播组只需要一个RP。RP可以静态配置，也可以动态学习（通过Auto-RP或PIM v2 BSR）。 数据包在转发时，省先在多播路由表里

查找匹配的（S，G）条目，如果找不到，则按照（*，G）条目指定路径转发。

源树及共享树的形成



上图中显示的是Receiver向RP注册以及共享树形成的过程，所有途经路由器中产生相应的（*，G） 条目。

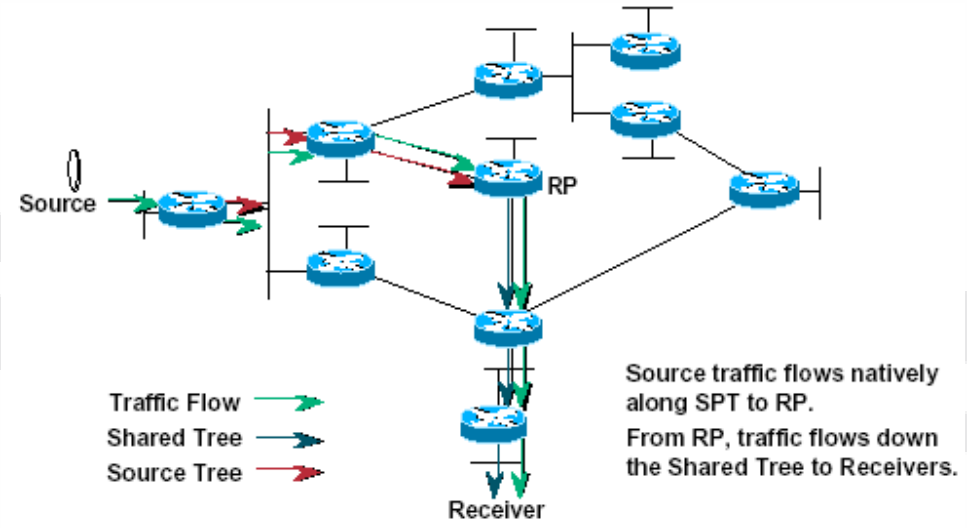


而多播源向RP的注册过程要稍稍复杂一点，这一点从图中不同类型的消息流的数量上就能看得出。 下面我来解释一下整个过程：

1. 当关于组G的活动源向首跳路由器发送第一个多播包时，首跳路由器就知道它应该向RP注册 这个源并且请求RP构建一条由RP返回首跳路由器的源树。
2. 首跳路由器把该多播消息封装成注册消息，以单播形式发往RP。
3. RP收到这一注册消息后，解除注册消息的封装后将它沿共享树向下传递。
4. 同时发送一个（S，G）加入消息到源S以创建一条（S，G）SPT，也就是说，在SPT途经的所 有路由器（包括RP）中创建（S，G）转发条目。
5. 当SPT形成后，发自源S的多播数据流将沿SPT从源S直接传到RP。
6. RP一旦接到从源S发出的沿SPT传来的多播包，则向首跳路由器发送注册停止消息，然后首跳 路由

器停止向RP发送注册消息。

7. 最后，多播数据包可沿从源到RP的SPT传至RP，再沿共享树从RP传至接收者。



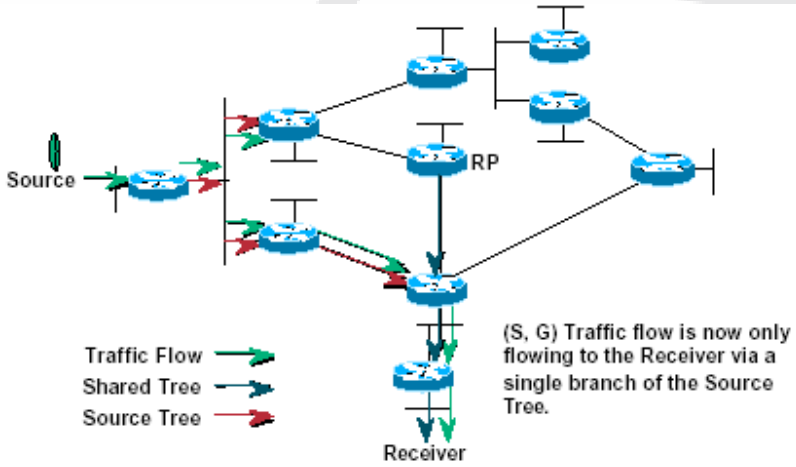
上图是最后形成的转发树，但是就本图而言，这个转发路径并不理想，事实上有一条不经RP的更短的路径可以用来传递这一多播数据流。对这一次佳路径的修正是通过SPT Switchover的过程来实现的。

SPT Switchover

上图中显示的转发树是由从源到RP的源树和从RP到接收者的共享树构成，尽管这两条路径本身都是最短的，但是合在一起后的结果并不理想，最靠近接收者的末跳路由器可以对这条次佳路径加以修正：

1. 末跳路由器根据SPT-Threshold的值决定何时对转发路径加以修正，缺省的情况下这个阈值等于0，表示第一个包沿共享树到达后马上对转发路径加以修正。
2. 末跳路由器向源发送（S，G）加入消息来重新构建一条旁路RP的SPT。
3. 最后，末跳路由器沿共享树向RP发送(S,G)修剪消息，阻止RP经共享树向末跳路由器转发属于（S，G）源组的多播数据。（这个修剪在RP上生效之前，接收者可能会收到重复数据）
4. 如果RP在它的所有属于共享树的接口上都收到了（S，G）修剪消息，那么它也会向源发送（S，G）修剪消息。
5. 通过这种方式，PIM SM以更经济的方式建立了从源到接收者的（S，G）SPT。

下图是最后形成的从源直接到接收者的SPT。



PIM Neighbor Discovery PIM邻居发现

PIM通过周期性的Hello消息发现并维持邻居关系，多路访问网络中，路由器向224.0.0.13

（All-PIM-Router）发送Hello消息。并进行DR(Designated Router)的选举（有最高IP地址的胜出）。

DR在PIM SM中用于为多路访问网络中的接收者向RP发送Join消息，以及为多路访问网络中的多 播源发送注册消息，在密模式中DR无意义，当然了，如果用的是IGMP版本1，则DR做为IGMP Querier代表所有路由器向多路访问网络内的接收者发送定期的查询。

```
wan-gw8>show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface          Uptime    Expires   Mode
171.68.0.70       FastEthernet0      2w1d      00:01:24  Sparse
171.68.0.91       FastEthernet0      2w6d      00:01:01  Sparse (DR)
171.68.0.82       FastEthernet0      7w0d      00:01:14  Sparse
171.68.0.86       FastEthernet0      7w0d      00:01:13  Sparse
171.68.0.80       FastEthernet0      7w0d      00:01:02  Sparse
171.68.28.70      Serial2.31         22:47:11  00:01:16  Sparse
171.68.28.50      Serial2.33         22:47:22  00:01:08  Sparse
171.68.27.74      Serial2.36         22:47:07  00:01:21  Sparse
171.68.28.170     Serial0.70         1d04h     00:01:06  Sparse
171.68.27.2       Serial1.51         1w4d      00:01:25  Sparse
171.68.28.110     Serial3.56         1d04h     00:01:20  Sparse
171.68.28.58      Serial3.102        12:53:25  00:01:03  Sparse
```

PIM State PIM

每台路由器对当前多播信息的转发都有自己的理解，但这里指的PIM状态指整个网络内的信息转发情况。

PIM状态主要通过多播路由表体现，其中包含（*，G）和（S，G）条目，（S，G）条目较（*，G） 有更高的优先权，所有条目中都包含Incoming interface、Upstream neighbor和Outgoing interface list。Incoming interface是通过RPF检验的接口，且只有从Incoming interface 收到的多播数据才会被转发到OIL。

```
sj-mbone> show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:13:28/00:02:59, RP 10.1.5.1, flags: SCJ
  Incoming interface: Ethernet0, RPF nbr 10.1.2.1,
  Outgoing interface list:
    Ethernet1, Forward/Sparse, 00:13:28/00:02:32
    Serial0, Forward/Sparse, 00:4:52/00:02:08

(171.68.37.121/32, 224.1.1.1), 00:01:43/00:02:59, flags: CJT
  Incoming interface: Serial0, RPF nbr 192.10.2.1
  Outgoing interface list:
    Ethernet1, Forward/Sparse, 00:01:43/00:02:11
    Ethernet0, forward/Sparse, 00:01:43/00:02:11
```

上图是一个多播路由表的示例，下面我们来解释这张表的维护过程：

PIM SM Route Maintenance

（*，G）是基于共享树的转发条目，只有当没有（S，G）匹配时才采用。当一台路由器收到（*， G）加入消息或来自属于组播组G的成员的IGMP报告时，创建（*，G）条目，如果有（S，G）条 目产生，则相应的（*，G）条目也会出现。（*，G）中的Incoming interface为该路由器指向RP的接口（根据RPF），OIL为收到(*,G)加入消息的接口、直接连接有接收者的接口或手工设置的静态转发接口。(命令：**ip igmp static-group <group>**）当OIL=NULL或没有子（S，G）条目时（*，G）条目将被删除。（S，G）是指定源的具体条目，相对于（*，G）条目有更高的优先权。 当收到(S，G)加入消息或PIM SM中首跳路由器发向RP的注册消息时，（S，G）条目被创建。如果相应的（*，G）条目不存在，则它必须被创建。（S，G）条目中的Incoming interface和Upstream neighbor由RPF决定。OIL复制于（*，G）条 目，但是不包含Incoming interface.

来看看OIL的管理： 收到（S，G）或（*，G）加入消息时将收到该消息的接口加入相应条目的OIL，（*，G）的OIL。如果变化，则（S，G）的OIL无条件跟随一起变化。收到（S，G）或（*，G）修剪消息时将收到该

消息的接口从相应条目的OIL中删除，（*，G）的 OIL如果变化，则（S，G）的OIL无条件跟随一起变化。此外，OIL中的接口过期时限为3分钟， 如果3分钟没有收到下游路由器的加入消息或来自直接相连成员的IGMP报告的话，也将会被删除。

当（*，G）条目的OIL由空转为非空时发送（*，G）加入消息，当（*，G）条目的OIL由非空转为空时发送（*，G）修剪消息。

当（S，G）条目的OIL由空转为非空时发送（*，G）加入消息，当（S，G）条目的OIL由非空转为空且从Incoming interface收到相关数据包时发送（*，G）修剪消息。

当（S，G）的RPF信息与（*，G）的RPF信息不一致时（表明SPT和共享树在此处分离）发送（S， G）RP-bit修剪消息。

PIM State Flags

S 表示Sparse Mode，仅见于（*，G）条目。

C 表示有组成员与路由器直接相连。

L 表示路由器自身是该多播组成员。（当使用Auto-RP时，所有路由器自动加入224.0.1.40）

P 表示OIL为空，该路由器正向上游路由器发送修剪消息。

T 表示至少有一个数据包已经通过SPT送达。

J 在（*，G）中表示收到下一个包后转发路径将由共享树切换到SPT，在（S，G）中表示该条目是SPT Threshold超过后被创建的。

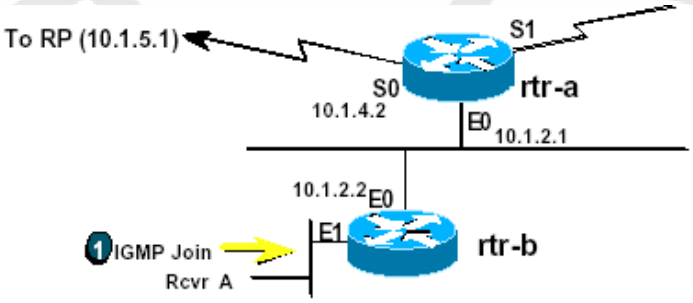
F 在（S，G）中表示该路由器与源直接相连（首跳路由器），它也出现在多播边界路由器上，（*，G）条目会随（S，G）条目上的F标志的出现而出现。

R 表示RP-bit，在修正经RP沿共享树的转发为SPT的过程中用于修剪（S，G）经共享树的转发， 关于这点我们后面还会详细讲述。

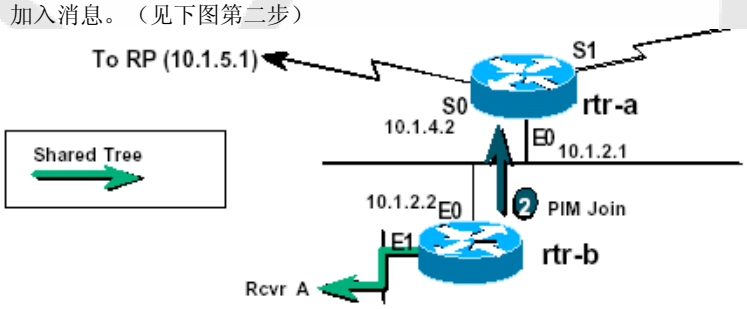
X 只见于（S，G）条目，当代理加入计数器运行时使用，一般用于SPT与共享树合并时，后面还 有讲述。

PIM Sparse Mode Join

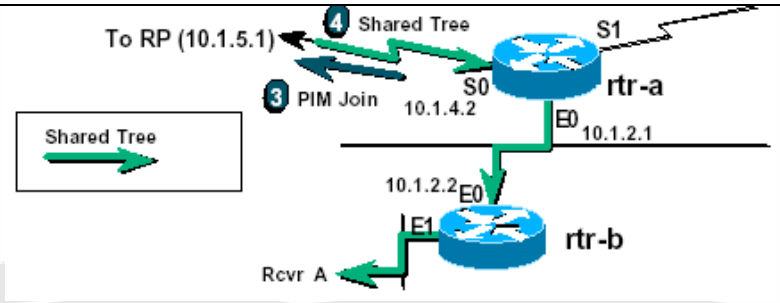
第一步，接收者A向224.0.0.2（All-Routers）发出IGMP Join消息请求接收多播组G的数据。（见下图第一步）



第二步，末跳路由器收到该请求消息后，首先查看(*，G)条目是否已经存在，如果已经存在， 则只需将收到该请求的接口加入 OIL，如果不存在，则先创建（*，G）转发条目，并向指向 RP 路径上的上游路由器发送（*，G）加入消息。（见下图第二步）



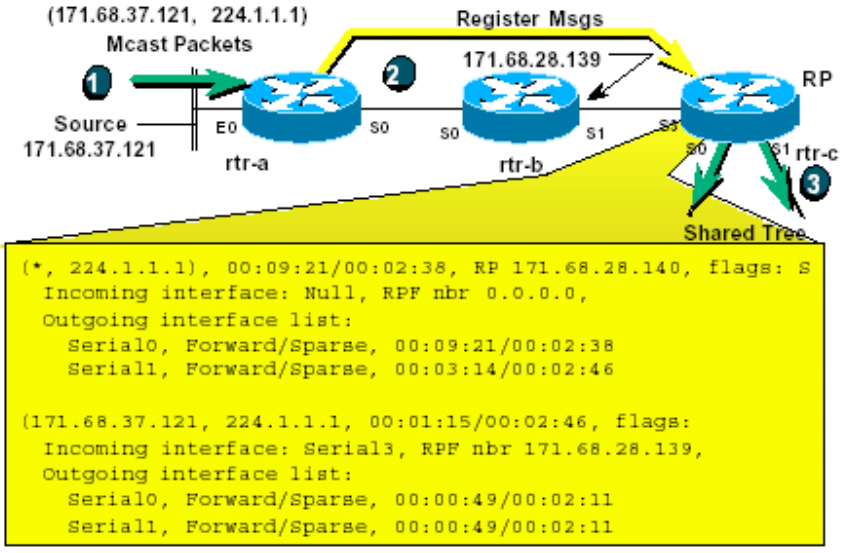
类似的过程在到达RP途经的路由器上依次重复，直到RP或一台已经属于共享树的路由器为止， 最后生成一条从RP到末跳路由器的共享树。（见下图三四步）



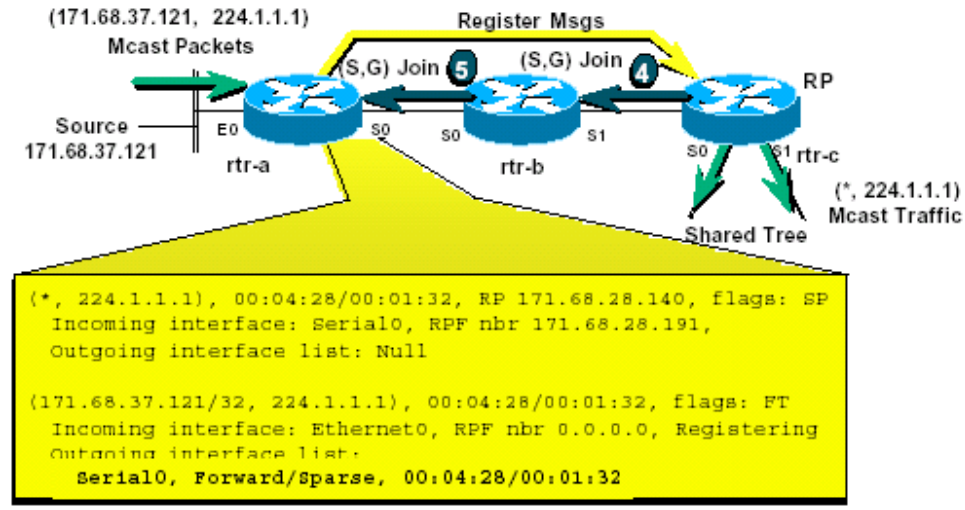
PIM Sparse Mode Registering

PIM稀疏模式注册过程

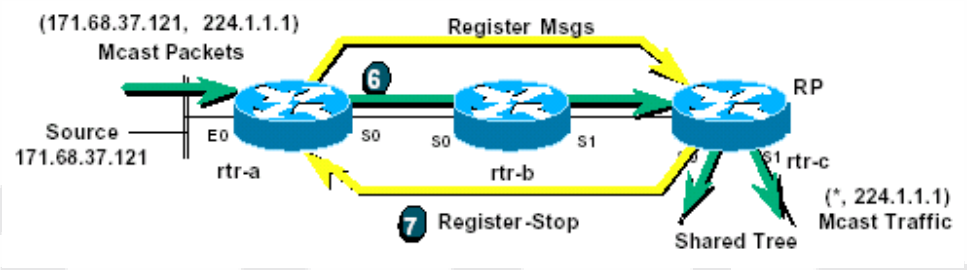
首先，多播源向组G发送多播消息，首跳路由器收到该组播数据后将其封装在单播形式的注册消 息包内发向RP，并在多播路由表内创建（S，G）和（*，G）条目。RP收到该注册消息后，解出 内部的多播数据包沿共享树传递给接收者。（见下图中一、二、三步）



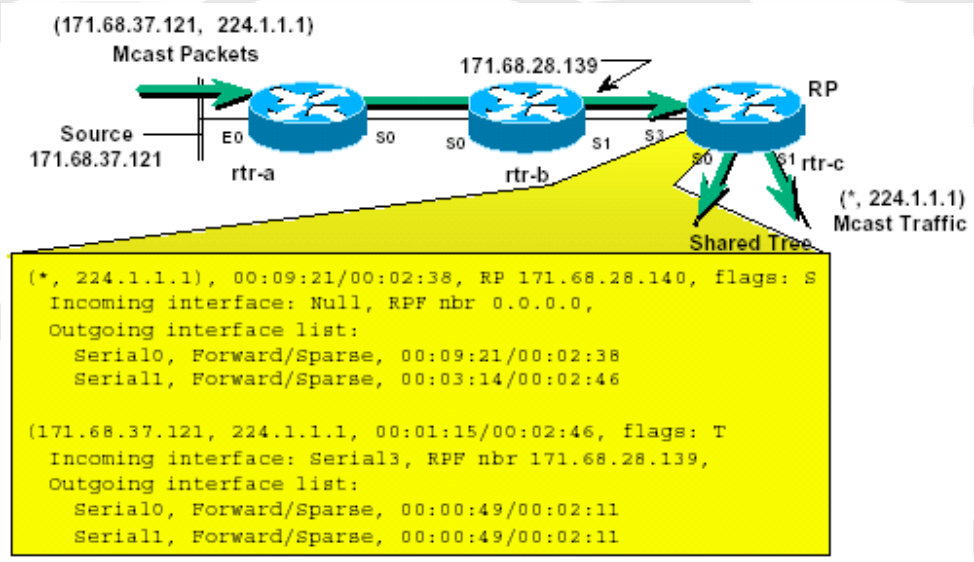
接下来RP沿指向源的方向向上游路由器发送（S，G）加入条目以构建源树，沿途路由器都做类 似操作，直到首跳路由器，源树形成，注意首跳路由器已将S0加入（S，G）条目的OIL中。（见下图中四、五步）



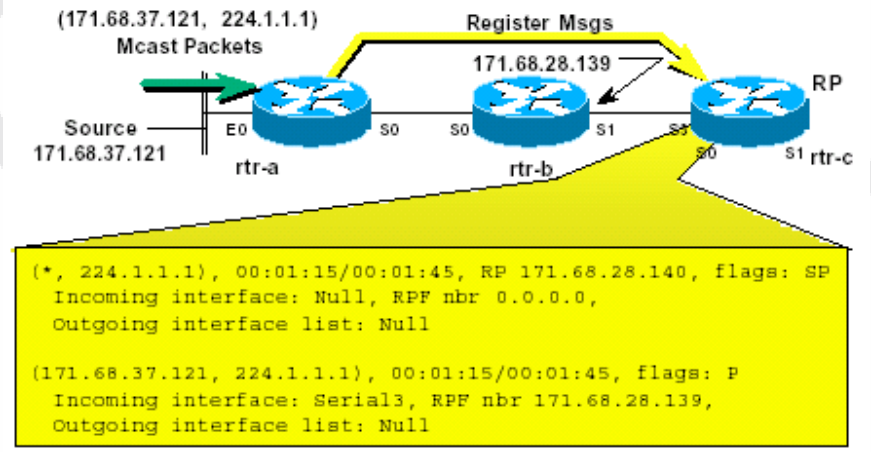
源树构建完毕后，RP将收到经源树送达的多播数据包，此时同样的多播数据RP将收到两份（另一份封装在注册消息中）收到经源树送达的多播数据包后，RP向首跳路由器发送注册停止消息。（见下图中六、七步）



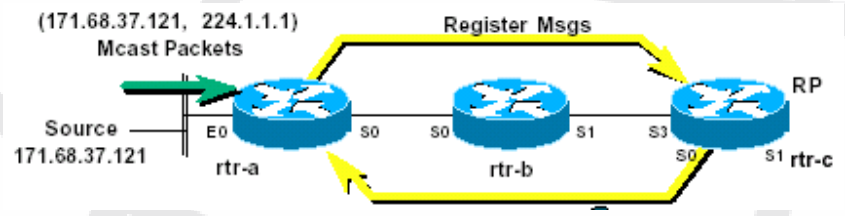
下图是最后形成的源树，SPT途经的路由器中均存在（S，G）转发条目。



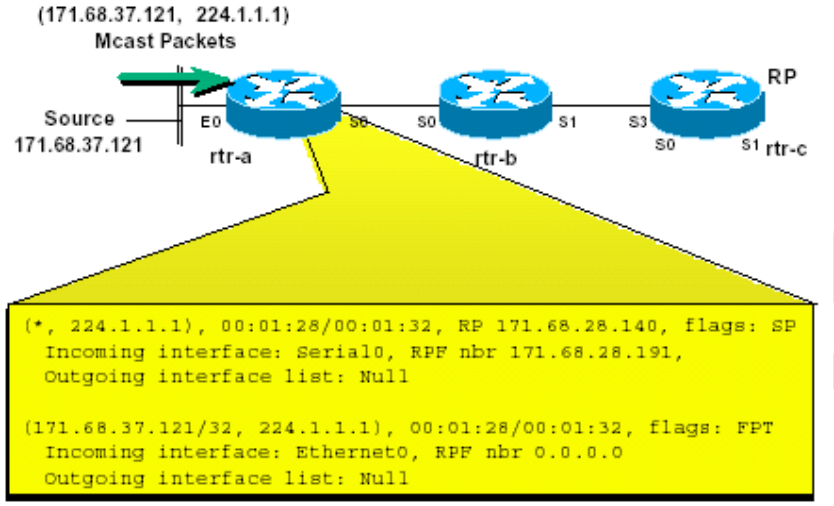
请注意：刚才的例子当中我们假定在源发送多播数据之前已经存在属于该组的接收者且共享树 已经建立，如果共享树尚不存在，那么刚才的过程会有很大的不同。下图中，首跳路由器向RP 发送注册消息时RP（*，G）条目的OIL为空，也就是说共享树不存在，没有接收者。



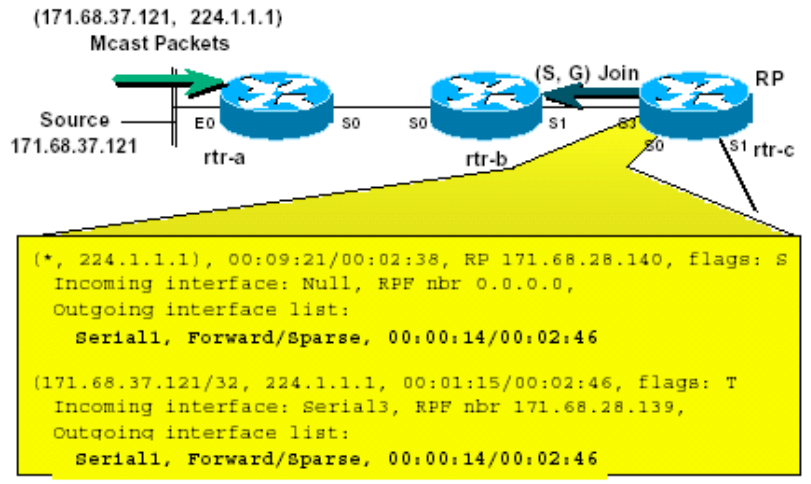
这种情况下，RP不向上游路由器发送（S，G）加入请求构建源树，而是直接向首跳路由器发送 注册停止消息，见下图。



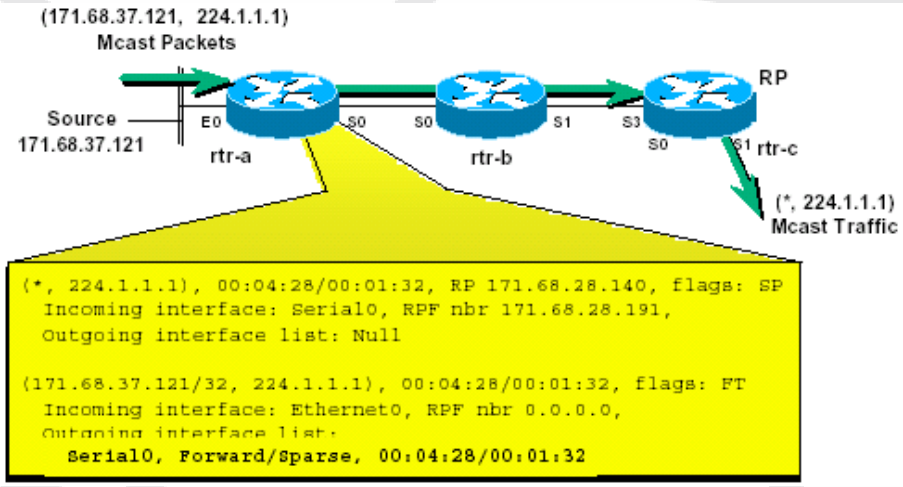
首跳路由器的注册消息将被抑制，多播转发表中（S，G）和（*，G）的OIL均为空，三分钟之后，该条目超时消失，一个重新注册的过程再次开始。见下图。



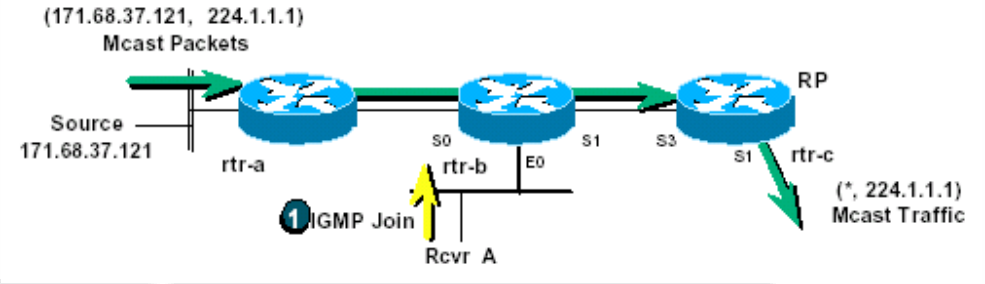
只要多播源不停止发送数据，这个注册和抑制的过程就会继续下去，并且导致RP中的（S，G） 和（*，G）条目始终存在，尽管它们的OIL为空，如果有接收者出现，RP将主动向SPT所经上游 路由器主动发送（S，G）加入消息。见下图。



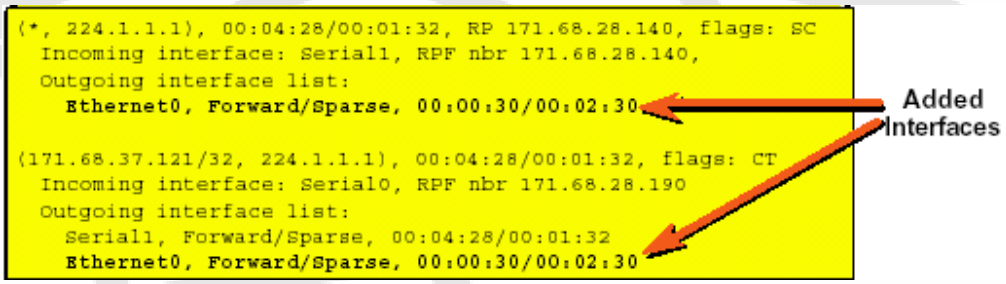
于是源树将在接收者出现之后很快建立起来，如下图。



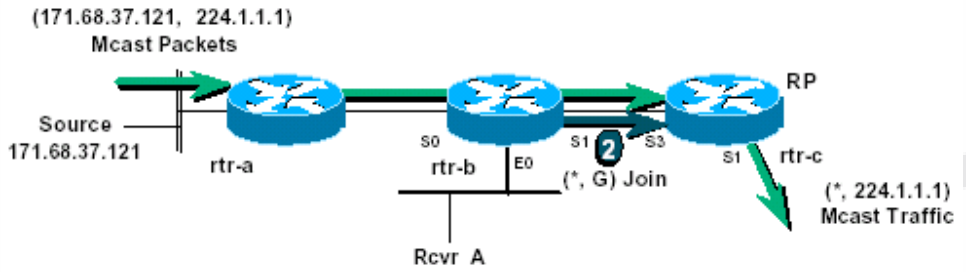
什么情况都可能存在，如果一个接收者出现在源树上会怎么样呢？如下图，源树和共享树都已经建立并正常转发多播数据，此时，一个接收者向属于源树的一台路由器发出了加入请求。



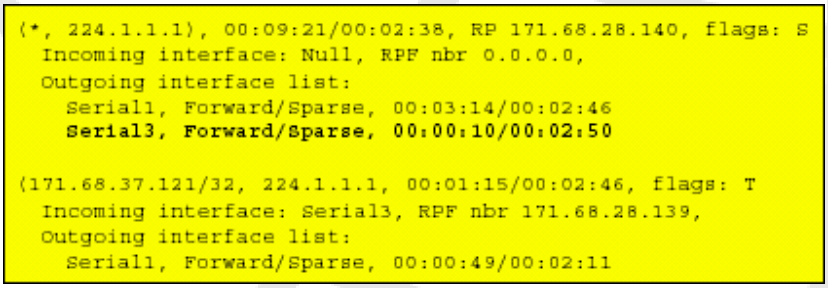
收到该加入请求的路由器会在（S，G）和（*，G）的OIL中将相应接口加入。如下图。



问题得到解决了，该接收者已经可以接收到数据了，但是因为（*，G）条目OIL非空了，该路由器会向RP发送加入请求以加入共享树来接收数据，如下图。



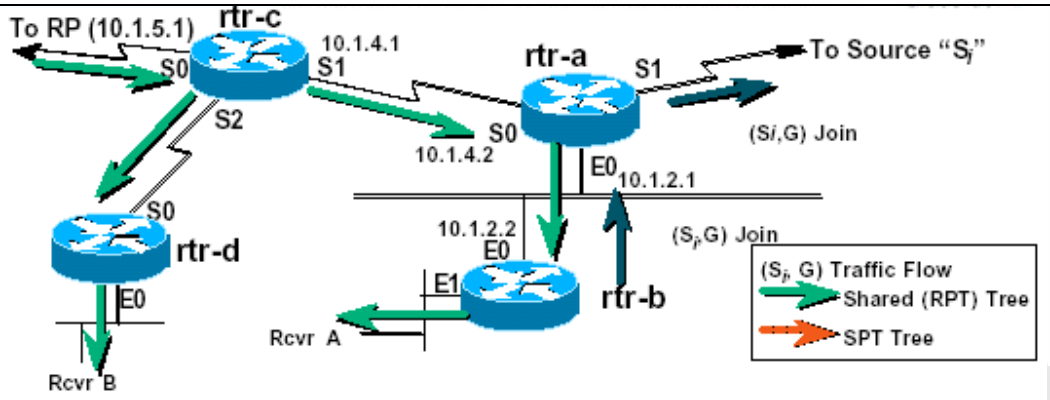
于是RP的转发表变成了下面的样子，由源树发至RP的多播消息又经共享树送回来了，你可能觉得这很荒唐，但这是必要的，因为组G的源可能不只一个。根据（*，G）与（S，G）的同步原则，S3应该被加入（S，G）的OIL，但是S3是（S，G）的Incoming Interface，所以不加入，最后的结果是（S，G）的数据流并不会由RP再转发回rtr-b。



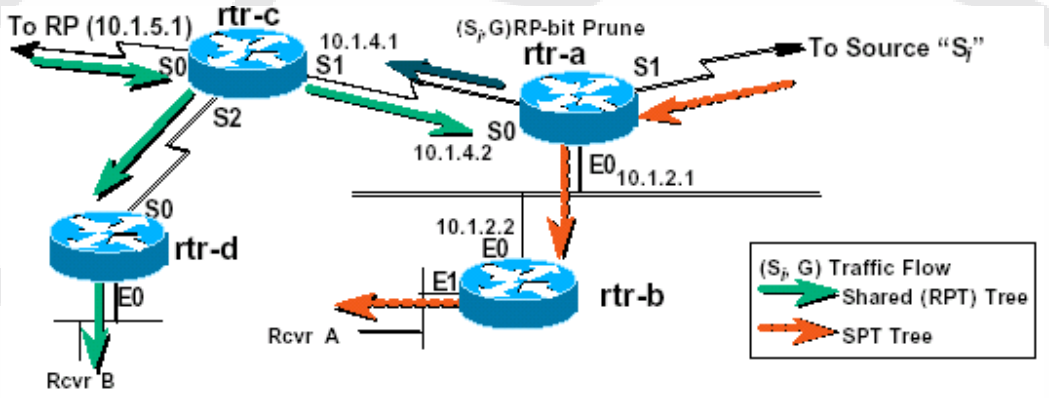
PIM Sparse Mode SPT-Switchover PIM稀疏模式SPT翻转

经由RP的转发路径通常不是从源到接收者的最佳路径，SPT翻转可以将经由源树和共享树的路径修正为从源直接到接收者的SPT，这样做的优点是可以减少多播数据转发延时，缺点是在多播路由器中产生较多的（S，G）转发条目。下面我们来介绍整个过程。

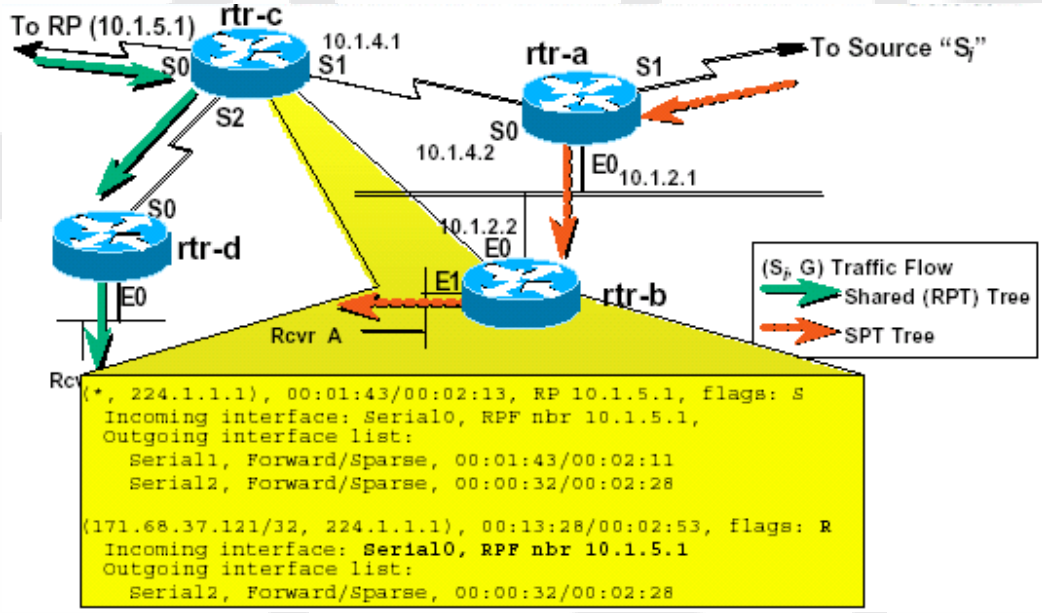
经共享树到达的多播流量超过SPT-Threshold（阈值）时，末跳路由器将会沿到达源的路径向上游路由器发送（*，G）加入消息以构建从源直接到接收者的SPT，这个消息一路传递一直到源。



从源到接收者的SPT形成后，末跳路由器向RP发送（S，G）RP-bit剪裁消息，通知RP停止通过共 享树向其转发（S，G）数据流。



我们看看最终的结果，rtr-c的转发表中，（S，G）的OIL为（*，G）的OIL减去Incoming Interface和RP-bit Prune消息中指定的接口，于是只剩S2，不再向rtr-a转发（S，G）多播流。

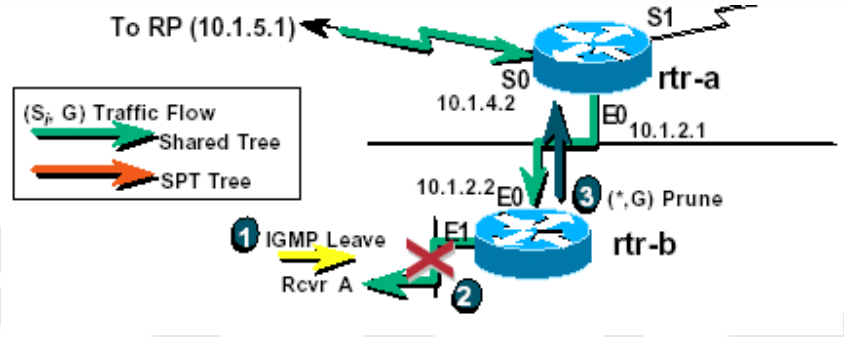


［注意］ 此后每隔一分钟，末跳路由器将检查多播数据流量与SPT-Threshold的大小，如果当前流量小于阈值，路由器执行Switchback（回转）的过程，重新切换到经由共享树的接收，以发现 网络结构和多播数据传递路径的新变化。

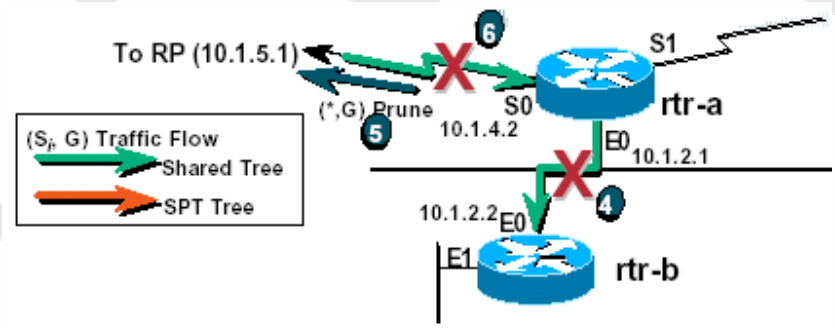
PIM Sparse Mode Pruning PIM稀疏模式修剪

下面我们将介绍哪些情况下路由器将向上游路由器发送修剪消息。 第一种情况，当叶子路由器接收到来自最后一个接收者的IGMP离开消息时，它会将相应接口从（*，G）和（S，G）中删除[注意：下图

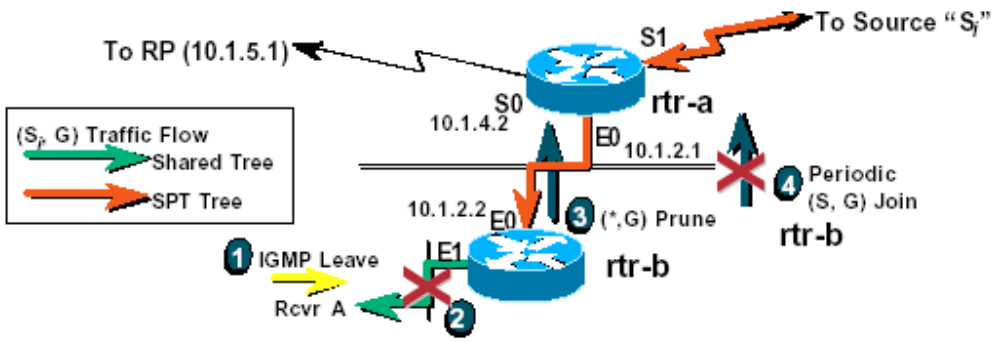
中只有（*，G）条目]，如果删除后OIL为空，它会向上游路由器发送修剪消息。



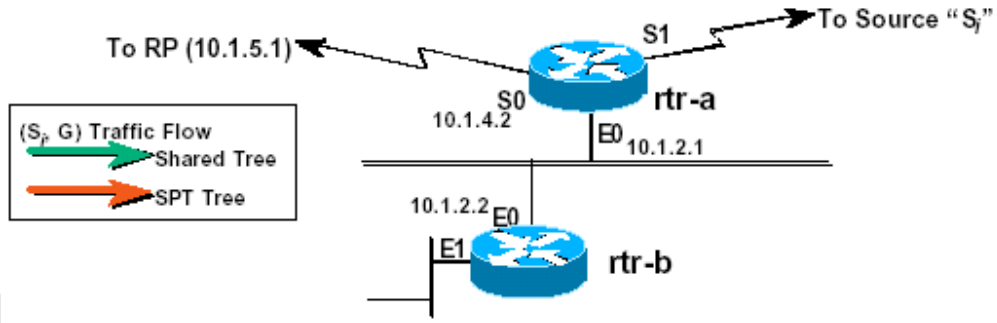
最后，如果沿途路由器中相关条目的OIL也为空，则修剪消息会一种传送下去，不需要的流量转发被剪裁掉。



如果存在SPT，则针对（*，G）发送修剪消息的共享树上游路由器的修剪消息，并停止周期性 的发向SPT的上游路由器的（S，G）加入消息。



最后的修剪结果一定是所有不必要的流量转发全部被修剪掉。



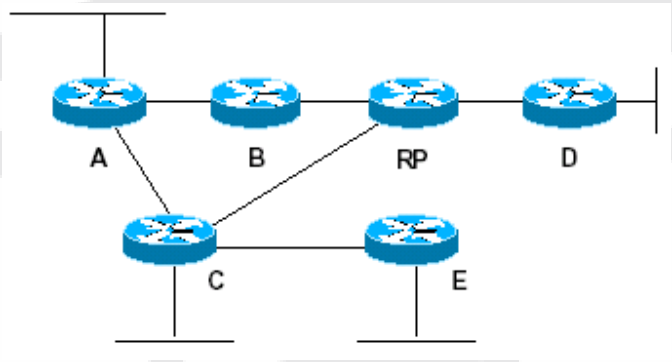
缺省情况下，加入/修剪状态信息三分钟后会过期，所以路由器必须周期性的向所有邻居路由器发送加入/修剪消息。周期性的加入消息将会刷新多播转发条目的OIL中的相应接口的超时计数器，如果该计数器为

0， 则接口从OIL中删除，如果OIL为NULL，将发送修剪消息到上游路由器。周期性的修剪消息将会导致路由器从OIL中删除掉收到消息的接口，或沿共享树向RP发送（S，G）RP-bit修剪消息。多播数据包的接收将复位（S，G）超时计数器，（S，G）条目超时值为0则该条目被删除（例如 源停止发送数据时）。

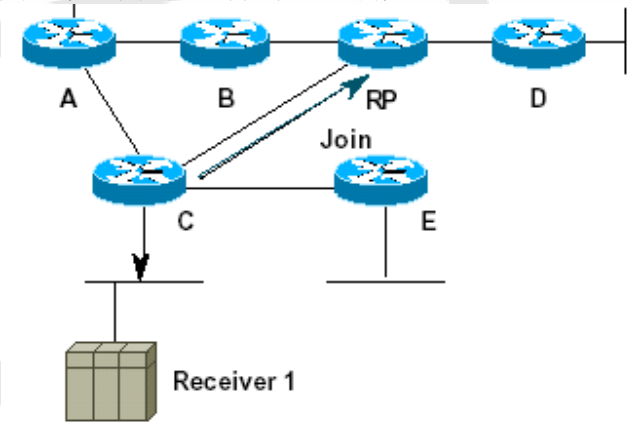
PIM Sparse Mode Review

类似前面讲过的密模式，在本单元的最后我们通地一个例子来总结一下稀疏模式的工作原理。

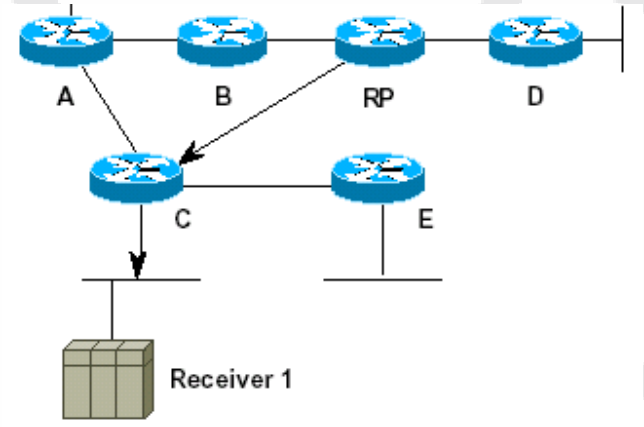
下图是本例的网络结构。



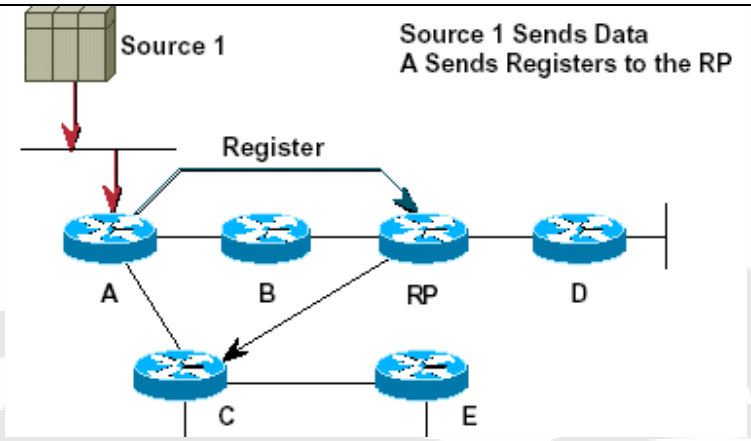
有接收者出现，末跳路由器向RP发送注册消息。



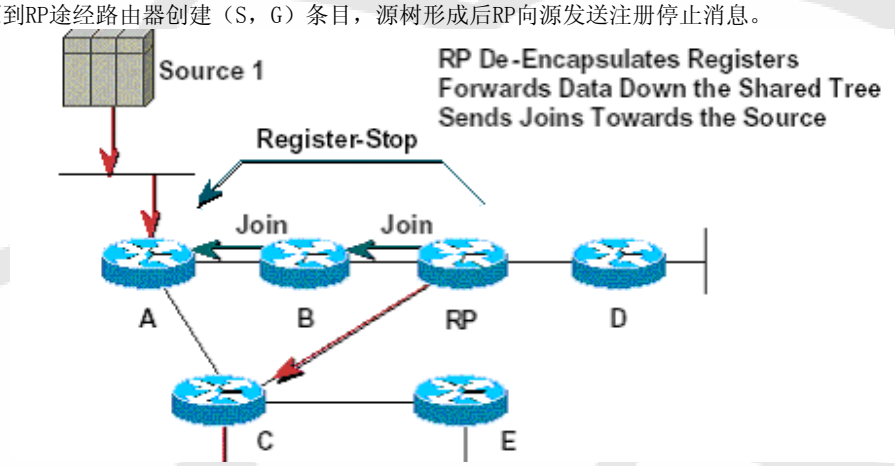
末跳路由器与RP之间的路由器均创建（*，G）条目，共享树形成。



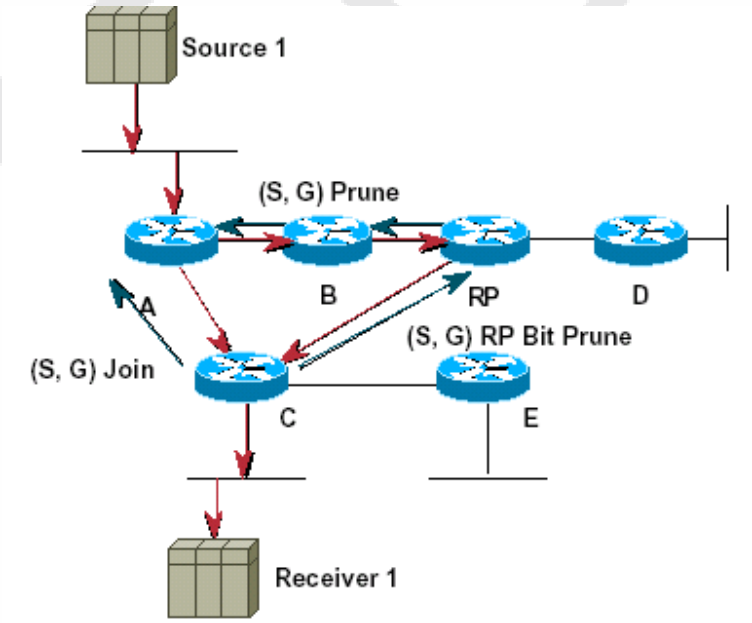
有源产生，首跳路由器向RP发送注册消息。



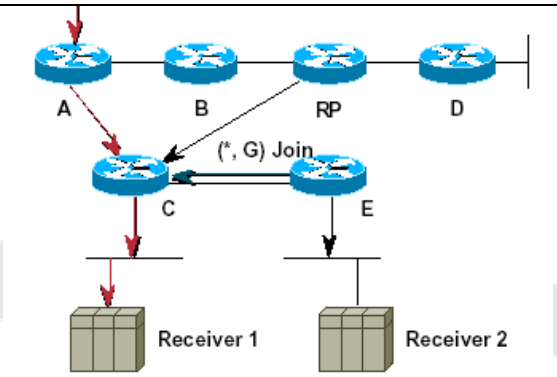
从源到RP途经路由器创建（S，G）条目，源树形成后RP向源发送注册停止消息。



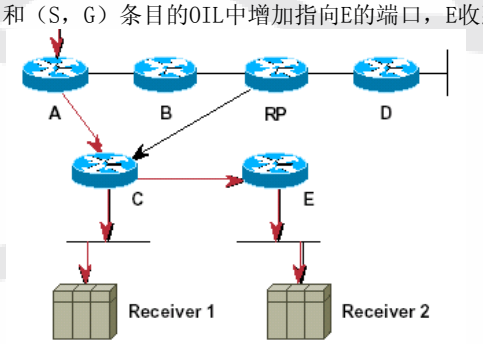
接下来是Switchover过程，末跳路由器修正经RP沿共享树传送的多播数据转发路径为直接的SPT。并向RP发送RP-bit修剪消息消除不必要的流量转发。



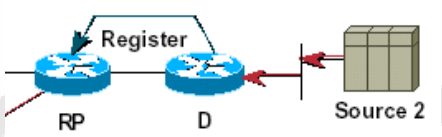
下面是一个新的接收者加入。



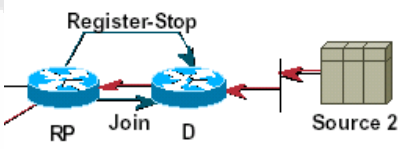
路由器C的转发表中（*，G）和（S，G）条目的OIL中增加指向E的端口，E收到来自C的多播数据。



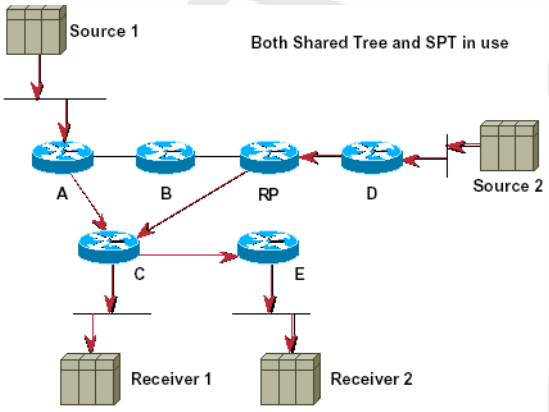
接下来又有新的源。



完成了到RP的注册。



最后的转发路径是这样的。



PIM Sparse Mode Configuration Steps

第一步，在路由器启动多播路由支持。

全局命令：**ip multicast-routing** 在所有路由器上配置，否则可能会形成组播路由黑洞。

第二步，以相关接口下启用对多播的支持。

接口命令：**ip pim { dense-mode | sparse-mode | sparse-dense-mode }** 第三种模式根据多播组它在

处理该多播路由信息时所采用的工作模式。 要防止组模式与接口模式不匹配的情况发生，疏密模式永远会匹配组模式，适用于采用Auto-RP 方案的网络或与单纯采用稀疏模式和单纯采用密集模式的网络混合存在。 千万不要以为接口模式决定组模式，接口模式只决定路由器在该接口下如何工作，组模式决定 于RP信息。对于一组多播组，如果RP被设置且有效，那么组模式为稀疏模式，否则为密集模式， 路由表内的（*，G）条目会分别标
以S和D。

[结论]永远使用sparse-dense-mode模式，并尽量保证所有路由器对RP的访问的有效性。（除非你喜欢用Dense模式）

为了保证RP的有效性，防止网络因为失效导致网络切换至密集模式，我们可以指定静态RP，但是为了防止静态RP阻碍Auto-RP协议的运行，必须与访问列表相结合使用。如下例：

```
ip pim rp-address <RP-of-last-resort> 10
access-list 10 deny 224.0.1.39 access-list 10 deny 224.0.1.40 access-list 10 permit any
On every router:        ip multicast-routing
On every interface:     ip pim sparse-dense-mode
On routers B and C:     ip pim send-rp-announce loopback0 scope 16
                        ip pim send-rp-discovery loopback0 scope 16
```

Auto-RP Overview

所有路由器自动学习与特定组相关的 RP 地址，除了 Candidate RPs（RP 候选者）和 MappingAgents（映射代理）外的路由器不需要为 RP 做任何设置。通过专用的两个多播地址 224.0.1.39（Cisco-Announce）和 224.0.1.40（Cisco-Discovery）以PIM DM（否则存在 Chicken and Egg 问题）方式传递 RP 相关信息。可以存在多个 RP 以作备份，可以通过管理范围对消息的传递加以限制，BSR 不个备这一功能，这一功能对减少多播信息对广域网带宽的占用非常有效。

Candidate RPs

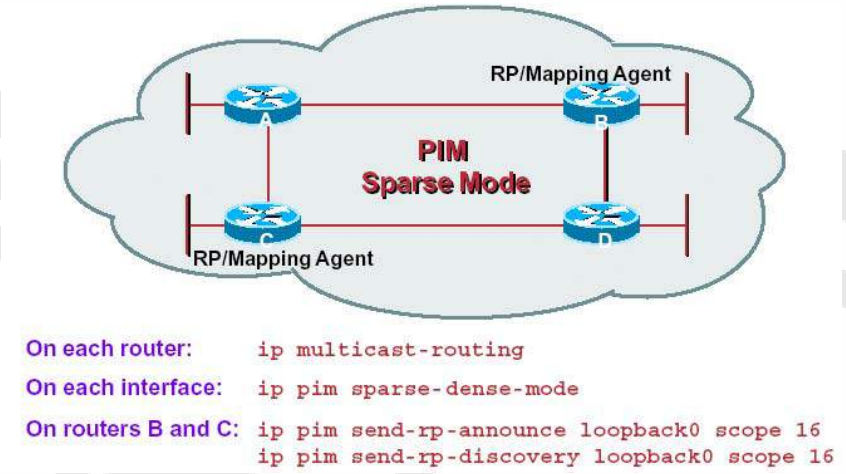
RP 候选者以固定周期向 224.0.1.39 组播地址发送 RP-Announcement 消息，这个消息用来说明该路由器是一个 RP 候选者，rp-announce-interval 的缺省值为 60s。RP 声明中包括：组范围（缺省为 224.0.0.0/4）、候选 RP 的地址，保持时间缺省为三分钟， 即三倍的 rp-announce-interval。
在全局模式下以下述命令设置：**ip pim send-rp-announce <intfc> scope <ttl> [group-list acl]**。Intfc 用于指定 RP 声明中的源地址取自哪个接口。

Group-list 中的 Deny 在不同的 IOS 版本中意义不同，12.0（1.1）以前的版本中表示当前路 由器不是相应组范围的候选 RP，12.0（1.1）以后版本中表示该组范围永远采用密模式。

Mapping Agents

用于接收发自 Candidate RPs 的声明，自动加入 224.0.0.39 这个多播组。所有声明存储在缓 存中，为每个特定组范围选举具有最高 IP 地址的候选者作为 RP。我们可以通过 **show ip pim rp mapping** 命令来查看 MA 的缓存。[注意：所有 MA 的缓存内容必须一致。]向 224.0.1.40 地址发送 Cisco-Discovery 消息，每 60 秒或检测到变化时发送。消息中包含从 多个候选者中选出的 RP。

可以通过如下命令设置：**ip pim send-rp-discovery [<interface>] scope <ttl>**。Interface 用于指定消息包源地址取自哪个接口，如果不设置，源地址为送出接口，这样将会导致一个 MA以多个地址出现。所有其它路由器自动加入 224.0.1.40 以接收 Cisco-Discovery 消息。通过接收 Cisco-Discovery消息以确定负责特定组的 RP。



通常一个网络中应该至少设置两个 C-RP 和 MA，一台路由器可以同时担当这两种角色。Intfc 最好使用回环接口来定义。需要支持多播的每台路由器的每个接口下都应该配置成疏密模式，因为 Auto-RP 采用密模式 PIM 工作，其它多播数据采用稀疏模式工作。

PIMv2 BSR Overview

至少需要一个 BSR（BootStrap Router），当然可以有多个以作备份（BSR 从这些 BSR 候选者中选出）。RP 候选者以单播形式向 BSR 发送声明（RP 通过接收 BSR 发出的声明知道 BSR 的地址），所有的声明存储于 RP-set。BSR 周期性的向 224.0.0.13（All-PIM Routers）发送声明，声明中包含 RP-set 和 BSR 的 IP 地址，RP-set 从 BSR 开始一跳一跳的传遍整个网络，所有路由器按照相同的算法从 RP-set 中选出相同的 RP。BSR 发送的声明不能设置管理范围。

Candidate RPs

RP 候选者以固定周期向 BSR 发送单播 PIMv2 RP-Announcement 消息，这个消息用来说明该路由器是一个 RP 候选者，rp-announce-interval 的缺省值为 60s。RP 声明中包括：组范围（缺省为 224.0.0.0/4）、候选 RP 的地址，保持时间缺省为三分钟， 即三倍的 rp-announce-interval。
在全局模式下以下述命令设置：**ip pim rp-candidate <intfc> [group-list acl]**。Intfc 用于指定RP 声明中的源地址取自哪个接口。这条命令可以在一个路由器内配置多次。

BSR（BootStrap Router）

用于接收和传递 C-RP 信息给所有路由器，接收到来自 C-RP 信息后存储于 Group-to-RP 映射缓存。每隔 60 秒或映射缓存有变化时 BSR 通过所有接口向 224.0.0.13(All-PIM-Router)发送 TTL为 1 的组播消息，消息中包含所有 Group-to-RP 信息和 BSR 的地址。这个消息会被所有的 路由器向前传递（通过 RPF 检验的消息发向除接收端口外的其它端口，TTL=1）。

多个 C-BSR（引导路由器候选者）通过选举优先级和 IP 地址最高者为活动 BSR，这个选举 过程是

抢先式的。活动 BSR 根据 C-BSR 的优先级随时变化，作为接收者的 All Other Routers在 Accept Any 和 Accept Preferred 两种状态中改变。

BSR通过以下命令设置 ： ip pim bsr-candidate <intfc> <hash-length> [priority <pri>]<hash-length>用于指定 hash mask 的长度，这一长度决定了 RP 服务于多播组的范围有多大。Priority 定义了当前BSR的优先级，优先级缺省等于 0。

Static RP

如果要配置，则应该在所有路由器上配置，所有路由器必须使用相同的 RP 地址，当静态配置的RP失效时，路由器不可能切换到其它的备用 RP（除非使用 Anycast-RP，在 RP 之间运行 MSDP）。

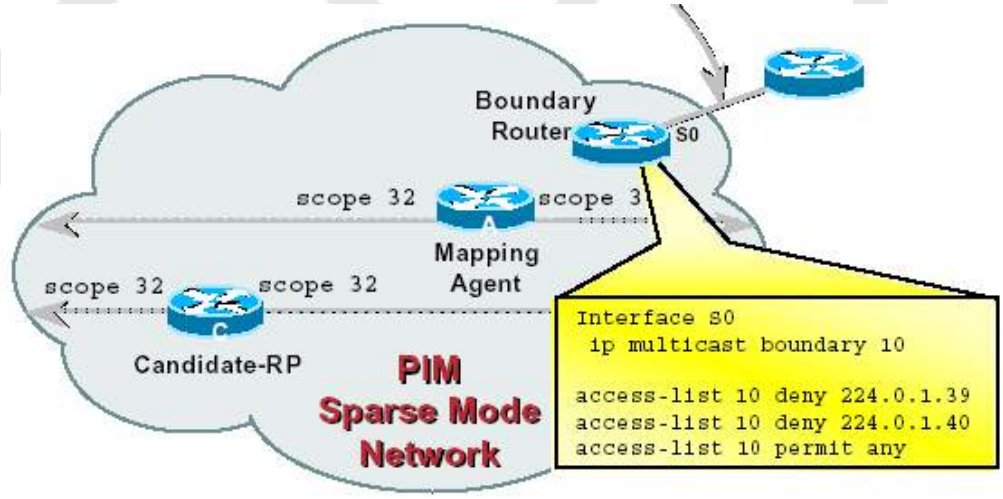
以下命令实现静态 RP 的指定：ip pim rp-address <address> [group-list <acl>][override]，group-list 指定组范围，缺省为 224.0.0.0/4，这是很危险，因为它把 Auto-RP 多播组（224.0.1.39 和 224.0.1.40）也包括进来了，注意这两个多播组是使用密模式进行维护的。所以我们至少 应该使用访问列表将这两个组排除。override 参数指示静态配置优先于 Auto-RP 学得的内容。

静态 RP 的配置比较容易理解，但是管理工作量很大，由于没有冗余能力，可靠性也不强， 不适于在大的网络中使用。

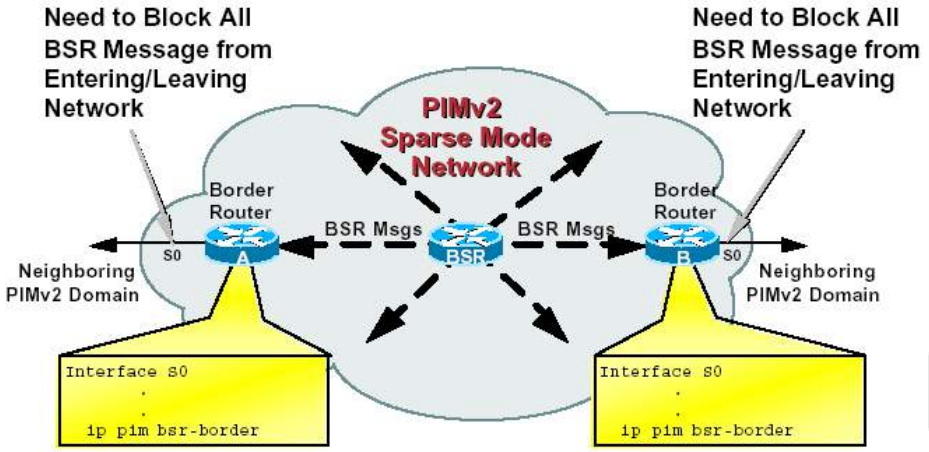
Tuning RP Operations

RP 应该放在哪里呢？这并不重要，因为 SPT 会对经共享树的转发加以修正，RP 只是源和目的的初始多播路由信 息的汇合点，大量的多播流量并不经过 RP 传递，RP 不会成为瓶颈，除非你把 SPT-Threshold 设成很大的一个值。如果要保证 RP 的性能，需要多快的 CPU和多大的内存呢？原文中给出了对 RP 的 CPU 有较大开销的任务和多播转发表的内存开销计算方法，这里不准备介绍给大家，因为性能的问题不太容易量化的评测。如果 RP 性能下降，我们可以选择更换 CPU，增加内存，减少 SPT 阈值和多 RP 负载均衡等 办法来解决。

Auto-RP 声明和发现消息中的 TTL 的值设成多大合适？ 这个跟你的网络结构有关，只要保证所有的 MA/C-RPs 都可以收到来自 C-RPs/MA 的消息就 可以了。不妨把 TTL 值设得大一些，我们可以在网络边界使用 ip multicast boundary 的命令 来限定 Auto-RP 多播信息的传递，如下图。



对于 PIMv2 BSR，我们使用命令 ip pim bsr-border 来限定消息传递，如下图。



如何才能防止不正确的配置导致 RP 信息的不一致呢？

这是一个比较令人头痛的问题，事实上没有绝对的保证，我们只能通过 **ip pim accept-rp** 命令对可接受 RP 进行限制。

命令用法如下图：

Ip pim accept-rp <rp-addr> [acl]

Ip pim accept-rp auto-rp [acl]

Ip pim accept-rp 0.0.0.0 [acl]

Auto-rp 和 0.0.0.0 只能设置一条，acl 如果省略掉则表示 224.0.0.0/4。

对 RP 的限定可以用来控制组模式，只有通过 RP 验证的组才可以使用稀疏模式。也可以用来限定哪些（*，G）的加入消息是可接受的，还可以用来限定哪些（S，G）注册消息是可接受的。此外，我们可以在 MA 中使用 RP 过滤来验证 RP 候选者的有效性，进而控制 RP。命令如下：

ip pim rp-announce-filter rp-list <acl> [group-list <acl>]rp-list 和 group-list 分别用来指定哪些 C-RP 和多播信息是可以接受的。

[注意]group-list 如果不指定表示不接受一切组。

Debugging RP Operation

先来看看调试 Auto-RP 时需要注意的问题：

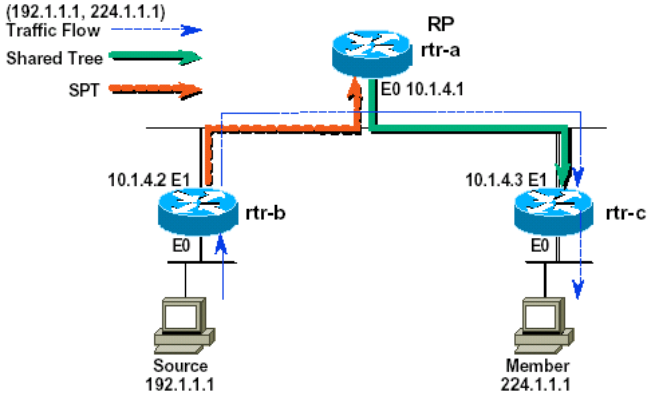
清楚Auto-RP的工作原理是基本的，要特别注意Group-to-RP信息的一致性，先检查所有MA的Group-to-RP 表，再检查其它路由器。因为 Auto-RP 的信息交换基于密模式，要特别注意静态 RP 的配置对 Auto-RP 的影响，还要 注意 NBMA 网络中运行 DM 时出现的问题。

再来看看调试 BSR 时要注意的问题：

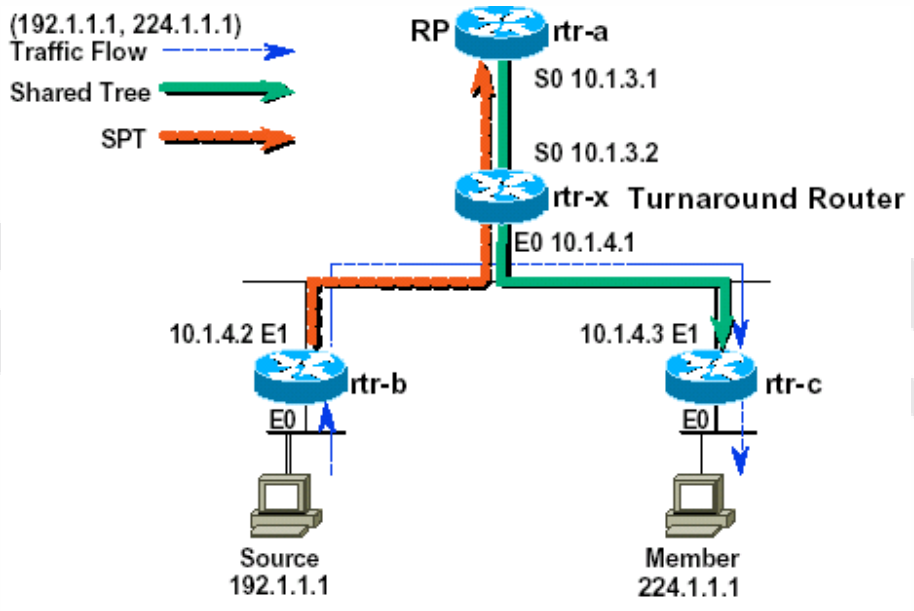
清楚BSR的工作原理是基本的，要特别注意Group-to-RP信息的一致性，先检查BSR的Group-to-RP 表，再检查其它路由器。

Special Cases

RP on a Stick，下图是一个特例，具体的原理比较复杂，为了解决这种网络结构下出现的问题采取了很多办法，具体内容不做介绍了，注意蓝色线表示实际的多播数据转发路径。



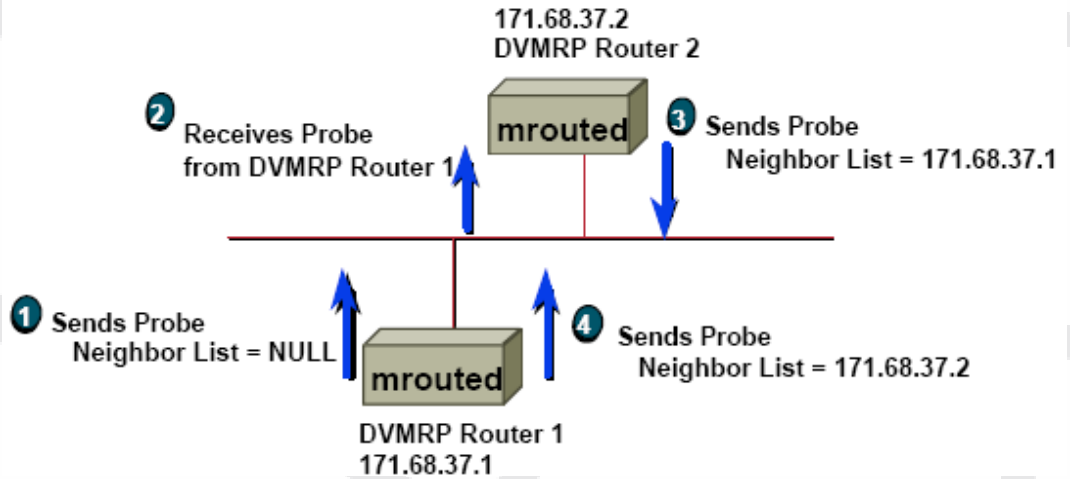
TurnAround Router 同样是一个特例，不解释原理了，注意蓝色线表示实际转发路径。



DVMRP

DVMRP Neighbor Discovery

DVMRP探针(Probe)消息定期向所有DVMRP路由器组地址(224.0.0.4)组播



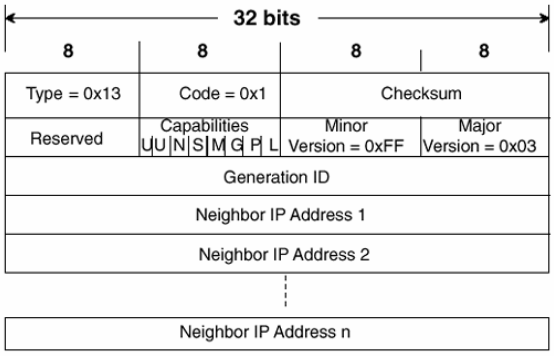
1. 路由器发送探针信息包，此时R1没有受到任何来自其他路由器的探针，邻居表为NULL
2. R2收到R1发送的探针，并向接口上的DVMRP邻居内部列表中加入R1的地址
3. R2在邻居列表上发送带有R1 ip的探针
4. R1收到探针，并将R2加入邻居列表，同时发送带有R2 ip的探针

此时双向邻接的工作已经完成,同时每隔10s发送一个，维持邻居关系，35s后未收到Probe包，则宣告邻居死亡

DVMRPv3通过IGMPv2查询来决定指定路由器，指定路由器是唯一能发送多播会话响应子网中IGMP查询的路由器

Message Type

1. Probe消息

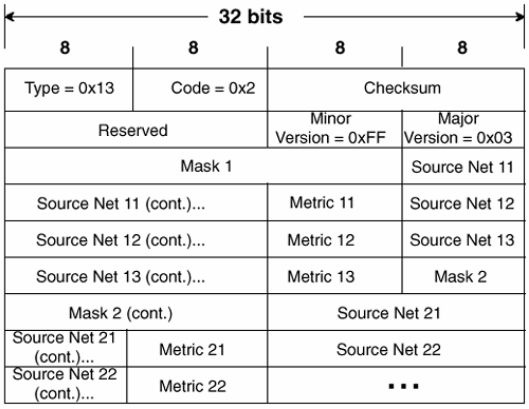


它具有4种功能

a. 他们让路由器通过发起路由器列出所有检测到的DVMRP路由器，互相知道彼此位置
b. 他们可以让DVMRP路由器互相通告彼此的能力
c. 在有多条通路都可以到达下游组员时，它可以选出指定转发
d. 每10s发送一次，35s没收到宣告邻居死亡

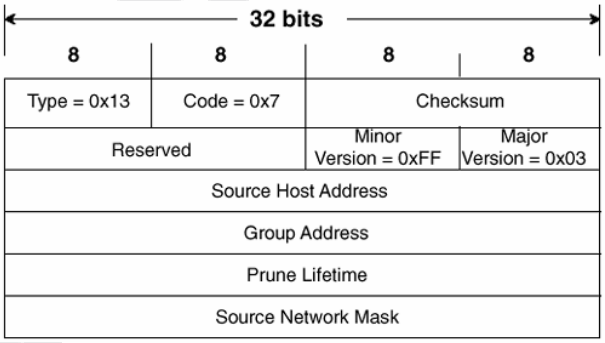
Bit	Flag	Capability
0	L	This router is a leaf router.
1	P	This router understands pruning.
2	G	This router sends Generation IDs.
3	M	This router handles Mtrace requests.
4	S	This router supports the DVMRP MIB.
5	N	This router understands netmasks appended to Prune, Graft, and Graft Ack messages.
6, 7	U	Unused.

2. DVMRP Route Report

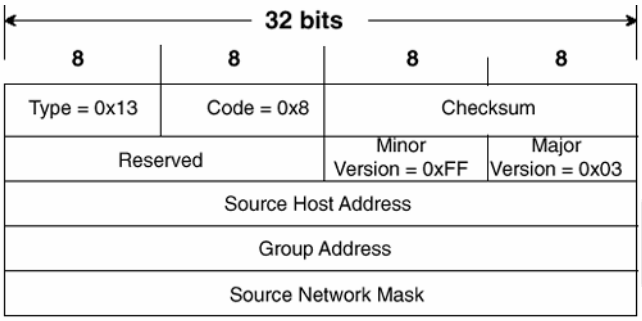


每60s发送一次，含有自己的直连网络和其他路由器获得的路由条目

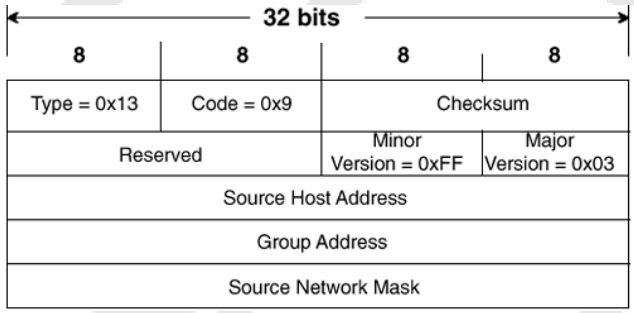
3. DVMRP Prune



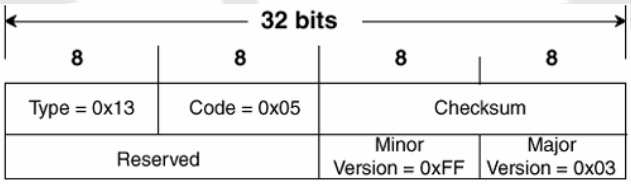
4. DVMRP Graft



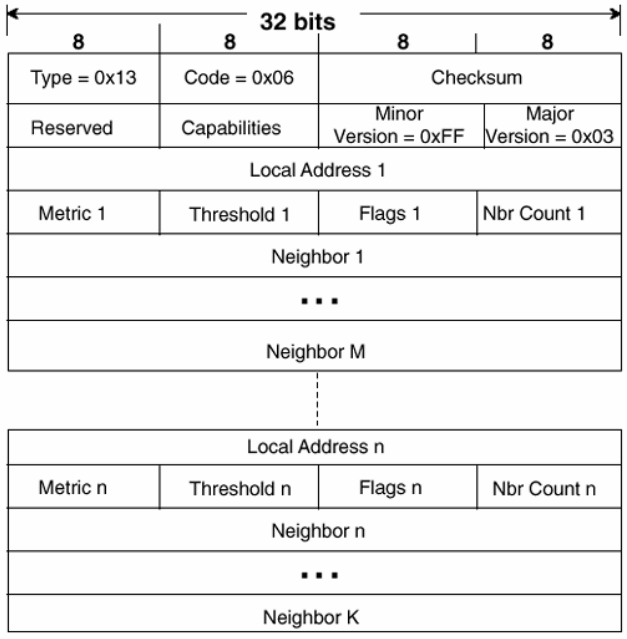
5. DVMRP Graft Ack



6. DVMRP Ask Neighbors2



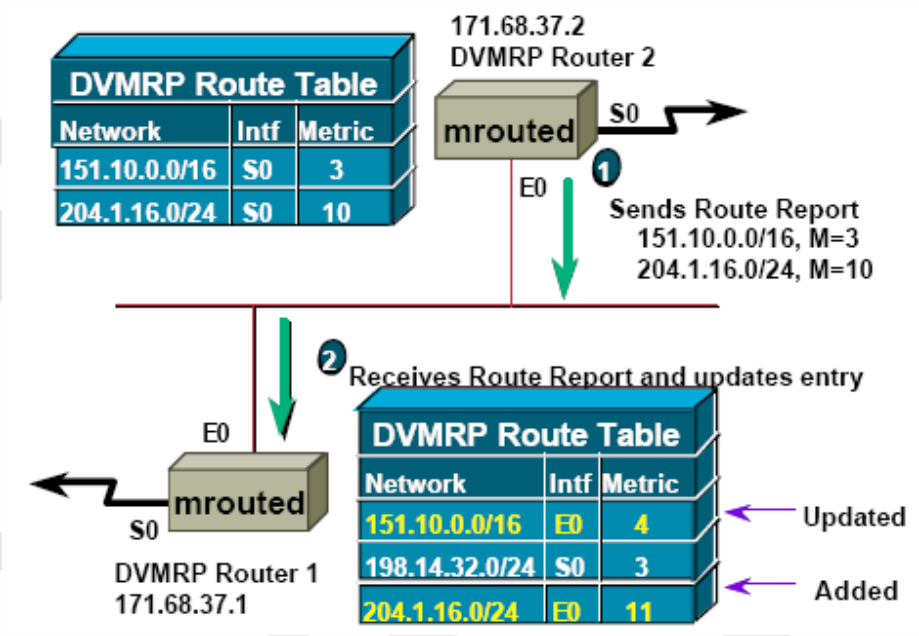
7. DVMRP Neighbors2



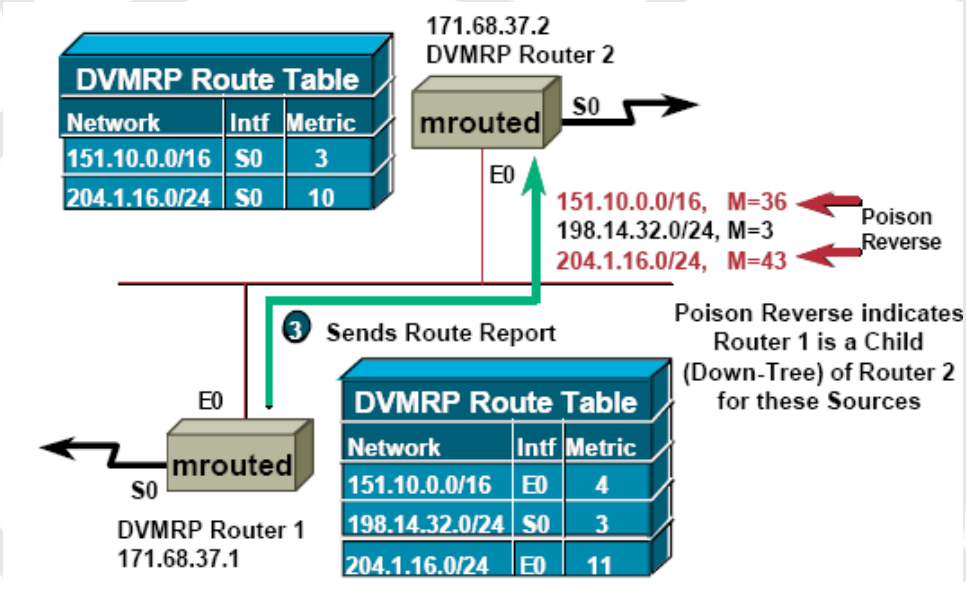
DVMRP Route-Table

DVMRP采用了RIP的很多变量来对外告知路由表和直连子网，路由更新每60s发送一次，如果140s未更新，则删除路由。Metric被设置为1~63，其中1~31为可达源，33~63为依赖源

Metric定义类似于RIP，每经过一个路由器+1



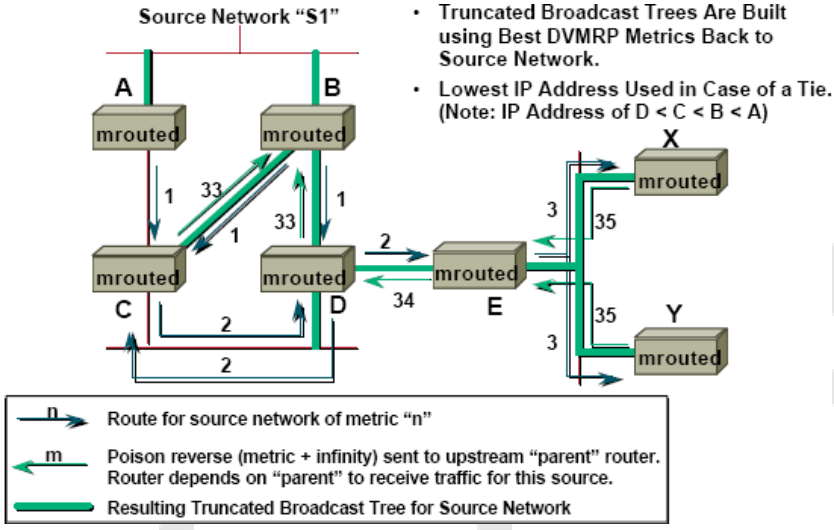
为了能够正确地剪除，DVMRP路由器必须知道靠它转发组播包的下游邻居，由下游路由器向上游路由器发送一条毒性逆转的路由条目，即在Metric上加上无限(32)，则上游路由将其认为依赖路由



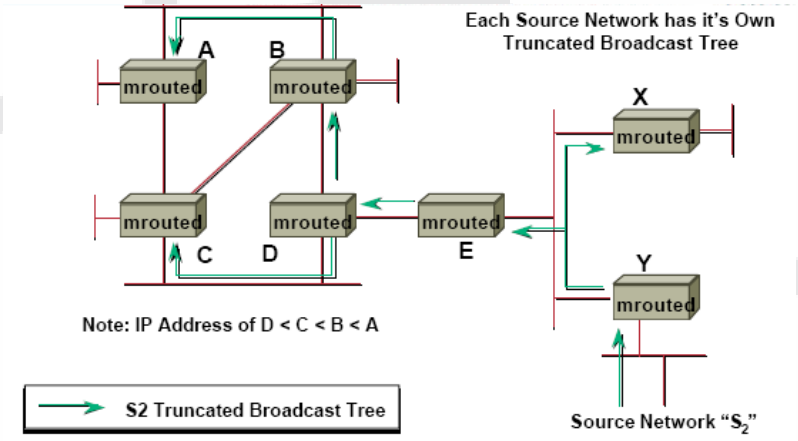
DVMRP Source-Tree

路由器A，B都向路由器路由器C，D向网络S通告一条Metric=1的指向S的路由。由于D对于S来说位于路由器B的下游，

因此为了向网络S逆向抑制通告路由，并向B返回通告，将其路由变为依赖路由



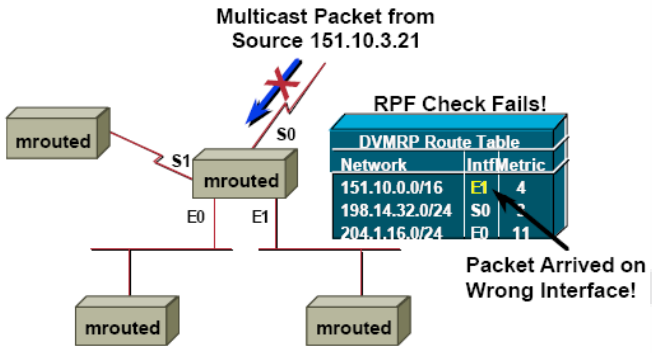
如果源在路由器Y上，其结果如下



Multicast-Forwarding

由于组播路由是一种颠倒的路由，因此在DVMRP路由表中的信息被用于确定是否在正确的接口收到一个输入的组播信息包。否则，为了防止组播循环将放弃该信息包，基于输入接口的转发再次被当作逆向路径转发(RPF)，并且把为了确定信息包到达正确的接口所进行的测试称为RPF检查。

一个组播信息包到达错误的接口的RPF检查如下



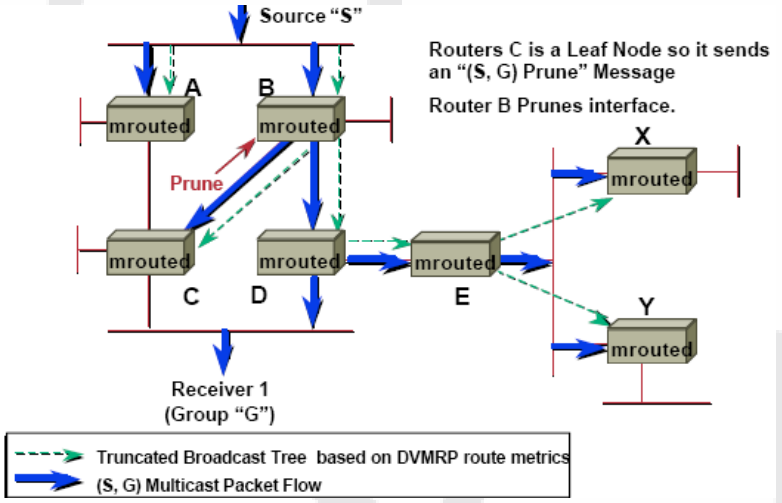
本例中，一个来自源151.10.3.21的组播信息包通过接口s0抵达。但是，依据DVMRP路由表中的151.10.0.0/16，来自源的组播信息包必须通过接口E1(不是S0)到达。于是，RPF检查失败，而且组播信息包被悄悄丢弃。

注意：通常情况下，DVMRP操作稳定时．信息包不会抵达错误的接口，因为上游路由器不会转发信息包，除非下游路由器逆向抑制第一个位置的路由.然而若网络拓扑刚刚改变，且DVMRP路由选择尚未在所有路由器上收敛(注意，这是基于路由选择的距离向量)，并有突如其来的DVMRP操作发生时，上述差错就会出现

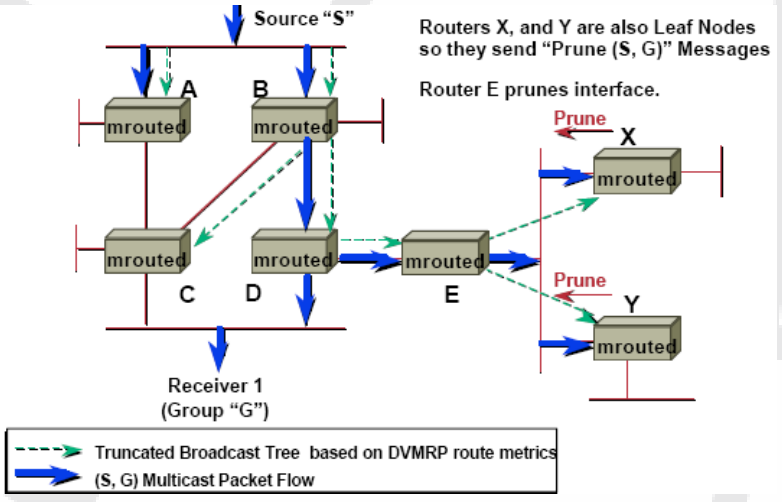
Pruning

即使在分布树上没有正在使用的接收站点，本过程也会发生。和大多数密集模式一样．DVMRP使用扩散-剪枝机制向网络中的所有路由器开始发送组播信息。就DVMRP来说，信息沿着截断的广播树向下被扩散到任一存在的接收站点。

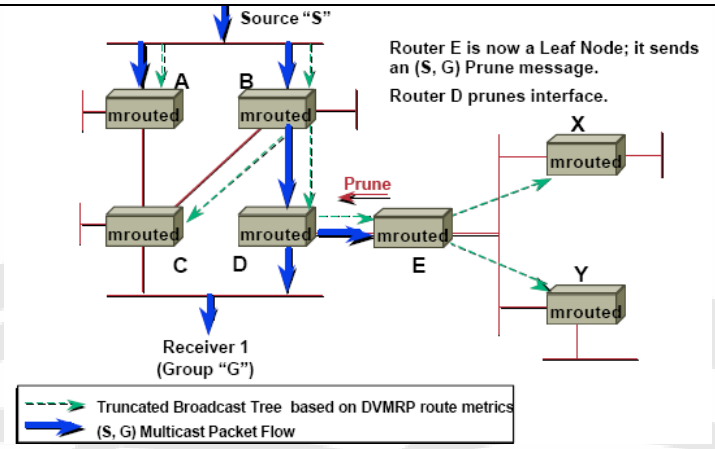
但是．为了保存宝贵的网络资源．你必须沿着截断广播树分支向下切断(或修剪)，通问没有接收站点的信息流；因此，没有任何直接相连的接收站点的叶路由器为了停止不需要的组播信息流及剪枝截断广播树的不需要的分支，而向战新广播树上发送DVMRP信息。在DVMRP剪枝后仍然留下的是对特定源的”十分布树或5PT”。不幸的是．由于DVMRP足扩散—剪枝协议，因此．只要剪枝超时。由DVMRP产生的源分布树就恢复截断广播树(一般地，DVMRP剪枝只有两分钟的有效时间．2分钟之后它们终止，信息再次扩散)和PIM-DM一样，这种扩散-剪枝方式也能引起在网络的DVMRP路由器里建立(S，G)状态。即使只有很少的接收站点.



注意：由于被网络中所有路由器的DVMRP路由表项的内容所描述．因此截断广播树本身作为DVMRP剪枝的结果不会被剪枝；这些DVMRP路由表项不会被剪枝信息修改-相反地，在DVMRP剪枝消息中(通常作为某个具体的(S，G)信息流的剪枝)收到的信息被存储在路由器中的某个独立数据结构中；此信息用于修改在截断广播树下面的(S，G)信息流；



当E的所有下游节点发出Prune消息后，E将往上游节点发出Prune消息

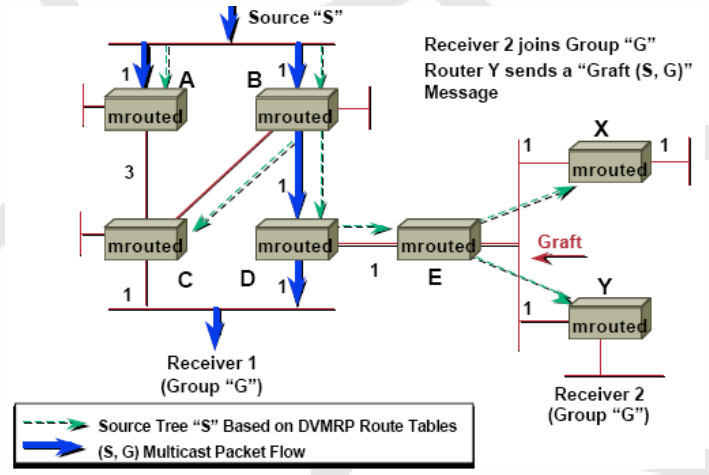


Grafting

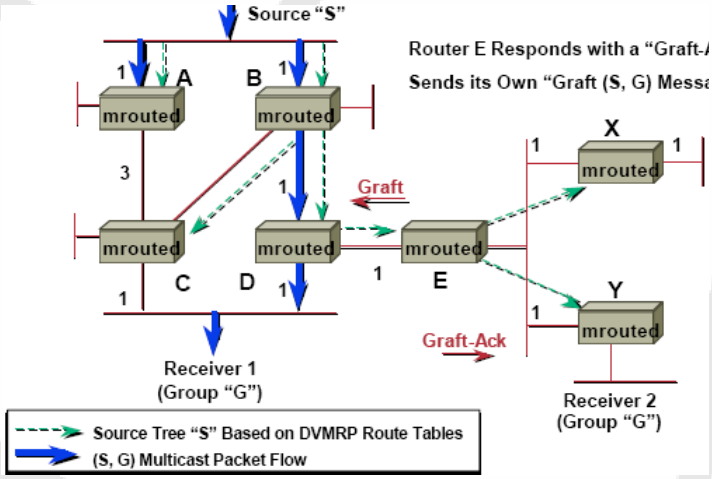
DVMRP提供一种可靠的嫁接机制，此机制把先前面已经剪枝掉的分支嫁接回来。如果没有这一机制，组中新主机的加入等待时间可能受到严重影响。因为在组播信息开始流动之前．上游路由器的剪枝状态必须结束．由于基于沿着被剪枝的分支的路由器数和使用的超时性．所以，在主机开始收到组播之前许多时间已经逝去。通过使用嫁接机制．DVMRP把加入延迟减少列几毫秒。

与剪枝机制的不可靠下一样，嫁接机制由于使用嫁接确认消息而变得可靠-这些消息由上游路由器返回以作为收到的嫁接消息。这一步骤防止由十阻塞而引起的嫁接消息丢失，阻塞会导致嫁接过程失败：

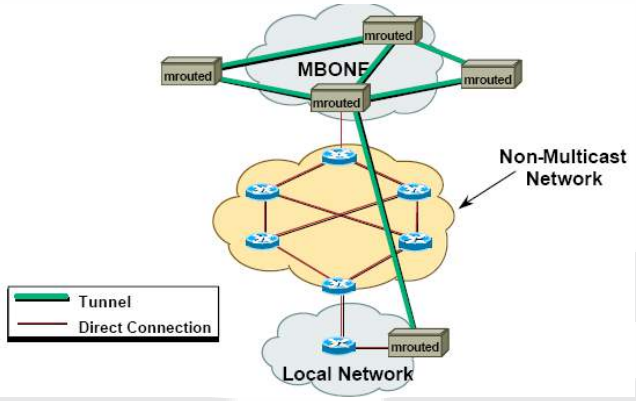
在路由器Y上的接收站点2加入组播组之后紧接着由于路由Y仍然存在(Si，G)状态，其表示路由器被剪枝，因此它知道为了重建信息流，必须向上游路由器E发出嫁接消息



E收到消息后，会向上级路由器发送嫁接请求，待确认后给Y发送Graft-Ack



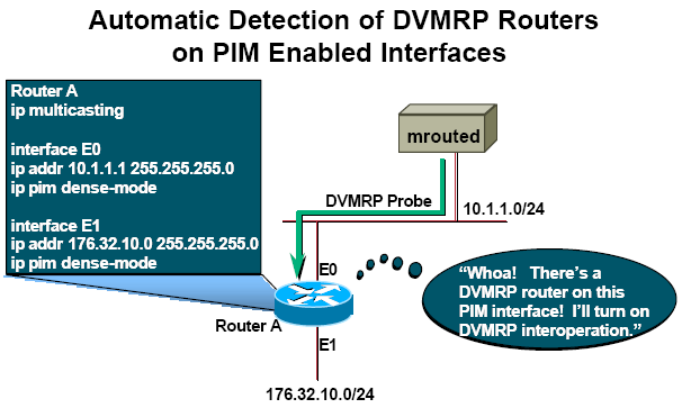
Tunnel



由于DVMRP仅支持31跳，所以为了扩展DVMRP，可以使用一些其他措施，例如使用tunnel

PIM-DVMRP

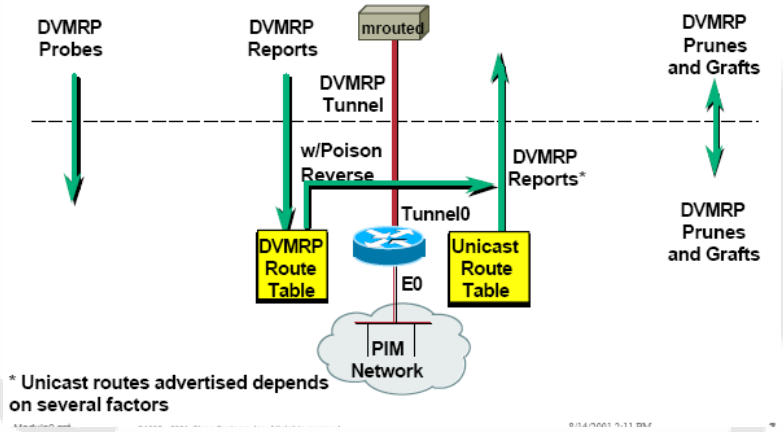
PIM-DVMRP Interoperability



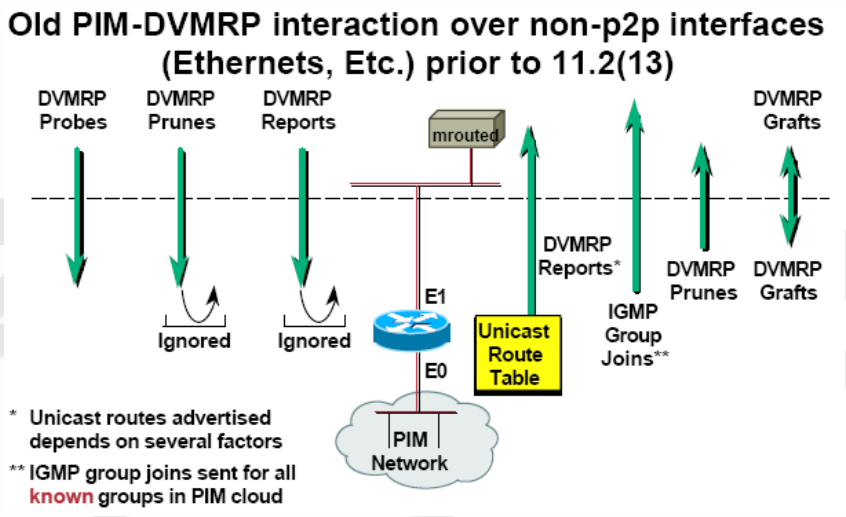
1. 采用DVMRP-Tunnel进行P2P互联

```
Interface  Tunnel 0
Ip unnumbered Ethernet 0
Ip pim spare-dense-mode
Tunnel source Ethernet 0
Tunnul destination 192.168.1.10
Tunnul mode DVMRP
```

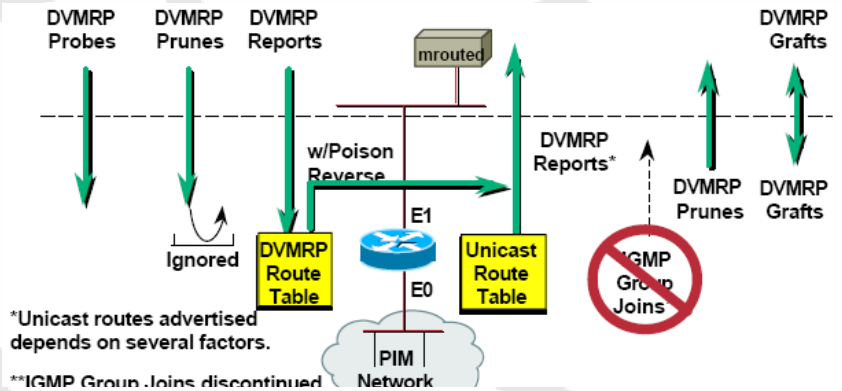
PIM对DVMRP消息的处理



2. 非P2P环境下对DVMRP的处理

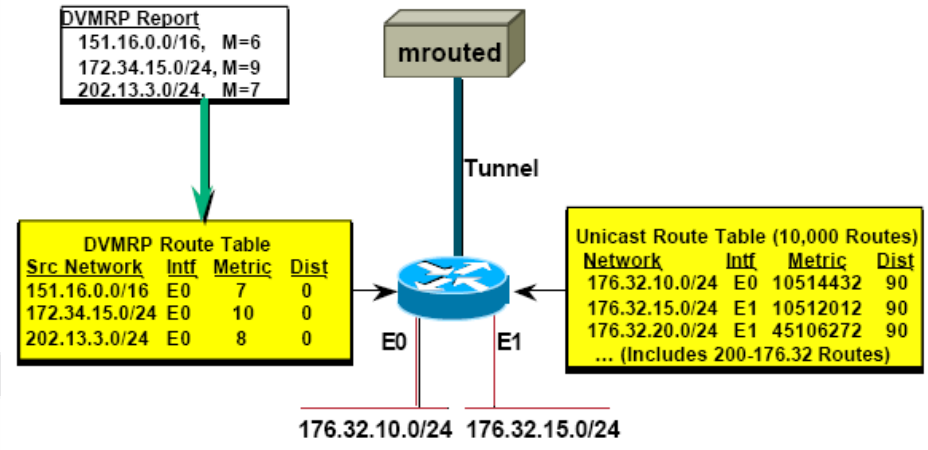


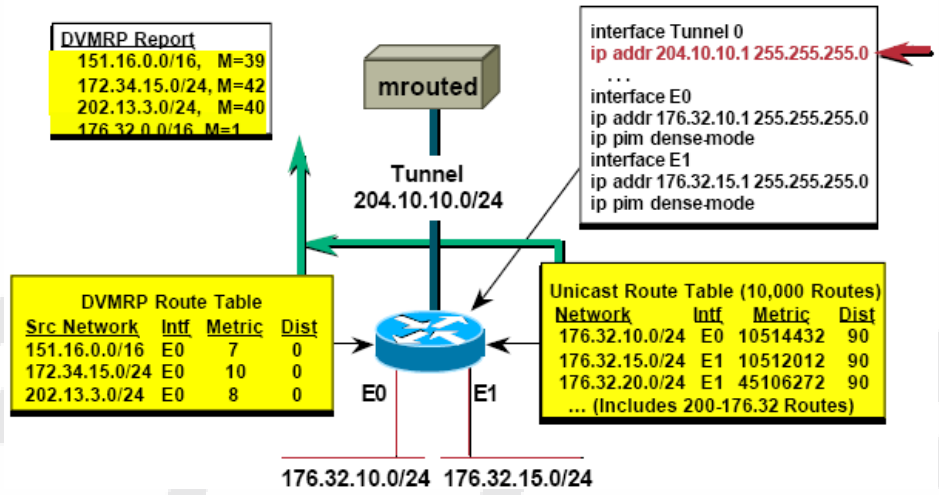
Ip dvmrp unicast-routing



可以通过ip dvmrp accept-filter过滤不必要的DVMRP路由器

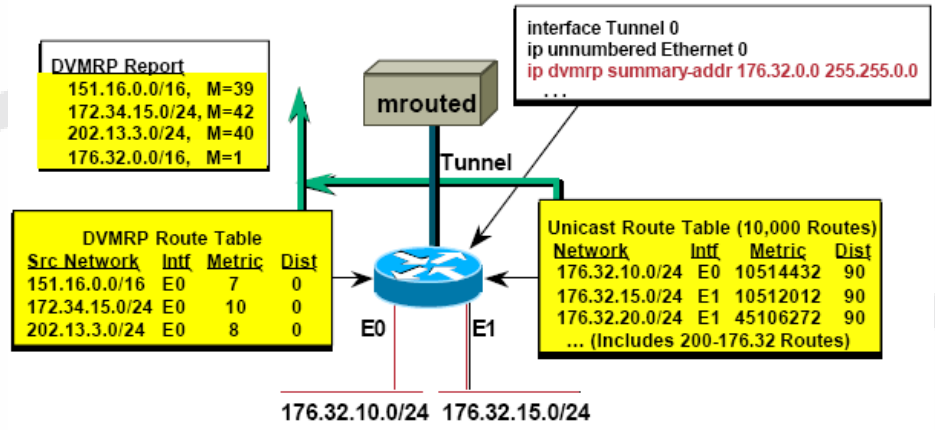
PIM-DVMRP Route-Exchange



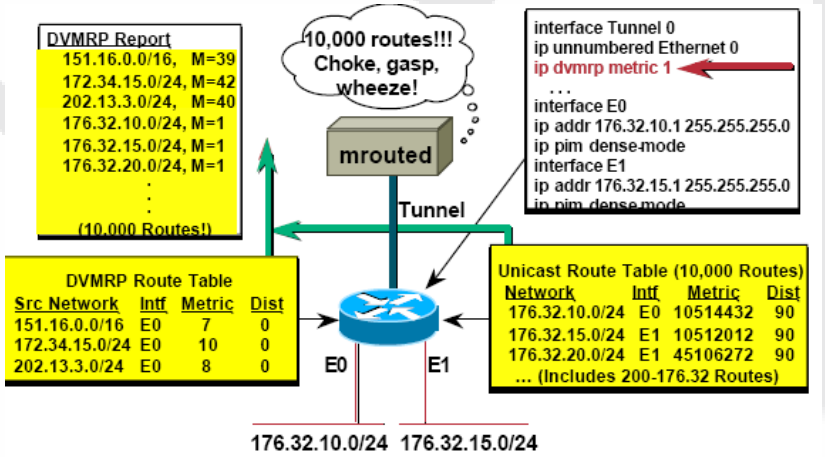


Ip dvmrp summary-address address mask

进行地址会聚，减小路由器负担



Ip dvmrp metric 修改metric值



No ip dvmrp auto-summary 防止地址自动会聚

Ip dvmrp default-information | originate only 控制默认路由

Ip dvmrp distance 调DVMRP路由的距离

Ip dvmrp metric-offset in|out

Ip dvmrp route-limit 限制路由

ip dvmrp output-report-delay <delay> [<burst>]

Defaults: delay = 100ms; burst = 2

Generating "route-hog" Warnings

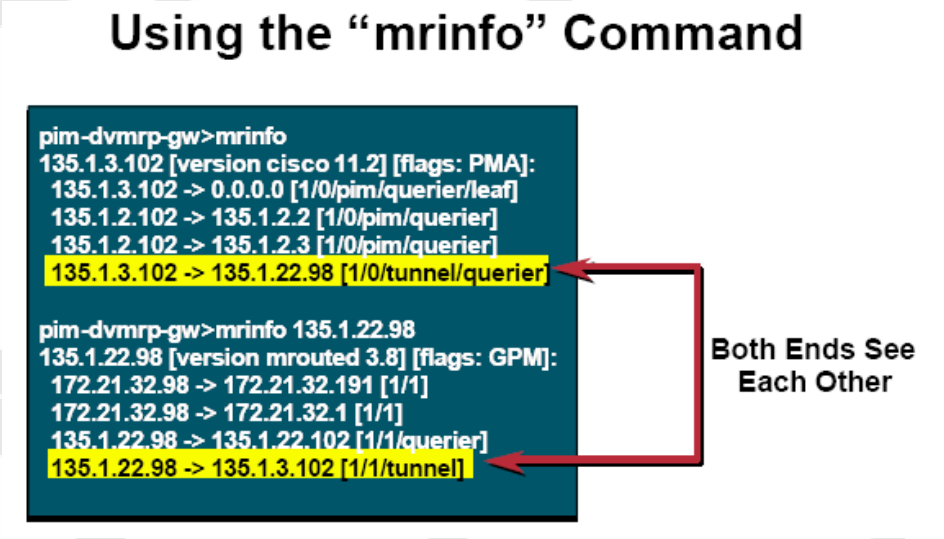
ip dvmrp routehog-notification <route-count>

Ignore DVMRP neighbors that don't support Pruning.

ip dvmrp reject-non-pruners

Debug

```
pim-dvmrp-gw> show int tunnel 0
Tunnel0 is up, line protocol is up
Hardware is Tunnel
Interface is unnumbered. Using address of Ethernet0 (135.1.3.102)
MTU 1500 bytes, BW 9 Kbit, DLY 500000 usec, rely 255/255, load 1/255
Encapsulation TUNNEL, loopback not set, keepalive set (10 sec)
Tunnel source 135.1.3.102 (Ethernet0), destination 135.1.22.98
Tunnel protocol/transport IP/IP (DVMRP), key disabled, sequencing
disabled
Checksumming of packets disabled, fast tunneling enabled
Last input 00:00:05, output 00:00:08, output hang never
Last clearing of "show interface" counters never
Input queue: 0/75/0 (size/max/drops); Total output drops: 0
```




```
pim-dvmrp-gw# show ip dvmrp route
DVMRP Routing Table - 8 entries
130.1.0.0/16 [0/3] uptime 00:19:03, expires 00:02:13
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
135.1.0.0/16 [0/3] uptime 00:19:03, expires 00:02:13
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
135.1.22.0/24 [0/2] uptime 00:19:03, expires 00:02:13
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
171.69.0.0/16 [0/3] uptime 00:19:03, expires 00:02:13
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
172.21.27.0/24 [0/3] uptime 00:19:04, expires 00:02:12
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
172.21.32.0/24 [0/2] uptime 00:19:04, expires 00:02:12
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
172.21.33.0/24 [0/3] uptime 00:19:04, expires 00:02:12
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
172.21.120.0/24 [0/3] uptime 00:19:04, expires 00:02:12
        via 135.1.22.98, Tunnel0, [version mrouted 3.8] [flags: GPM]
```

```
DVMRP debugging is on
pim-dvmrp-gw#
Mar 20 11:39:36.335: DVMRP: Aging routes, 0 entries expired
Mar 20 11:39:41.271: DVMRP: Received Probe on Tunnel0 from 135.1.22.98
Mar 20 11:39:45.335: DVMRP: Building Report for Tunnel0 224.0.0.4
Mar 20 11:39:45.335: DVMRP: Send Report on Tunnel0 to 135.1.22.98
Mar 20 11:39:45.335: DVMRP: 2 unicast, 8 DVMRP routes advertised
Mar 20 11:39:47.335: DVMRP: Aging routes, 0 entries expired
Mar 20 11:39:51.371: DVMRP: Received Probe on Tunnel0 from 135.1.22.98
Mar 20 11:39:52.379: DVMRP: Received Report on Tunnel0 from 135.1.22.98
```

```
pim-dvmrp-gw# debug ip dvmrp detail
DVMRP debugging is on
Mar 20 11:42:45.337: DVMRP: Building Report for Tunnel0 224.0.0.4
Mar 20 11:42:45.337: DVMRP: Report 130.1.0.0/16, metric 35, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 135.1.0.0/16, metric 35, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 135.1.22.0/24, metric 34, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 171.69.0.0/16, metric 35, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 172.21.27.0/24, metric 35, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 172.21.32.0/24, metric 34, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 172.21.33.0/24, metric 35, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 172.21.120.0/24, metric 35, from DVMRP table
Mar 20 11:42:45.337: DVMRP: Report 135.1.2.0/24, metric 1
Mar 20 11:42:45.337: DVMRP: Report 135.1.3.0/24, metric 1
Mar 20 11:42:45.337: DVMRP: Send Report on Tunnel0 to 135.1.22.98
Mar 20 11:42:45.337: DVMRP: 2 unicast, 8 DVMRP routes advertised
```

Advanced IP Multicast

Bandwidth Control of Multicast

组播流量控制一般采用丢弃数据报文的方式，被设计用来控制非法源或者和单播通信共享带宽

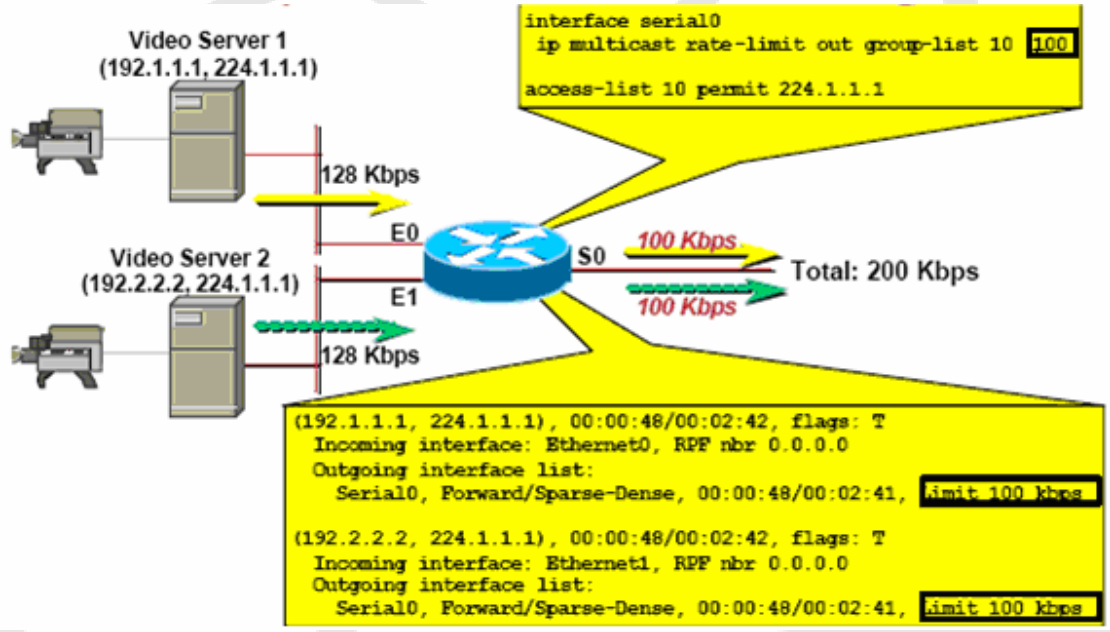
组播带宽限制分为2种

1．基于接口的速率限制

2．基于留的速率限制---可以对任何一个(S,G)或者(*,G)进行限制

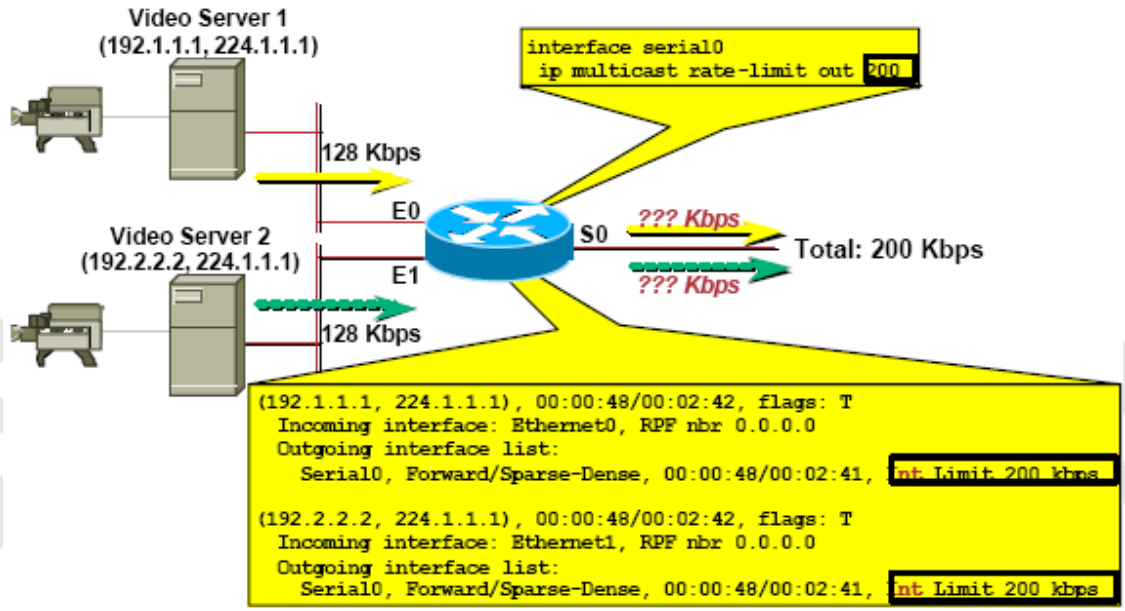
ip multicast rate-limit in | out { [video] | [whiteboard] }[group-list <acl>] [source-list <acl>] [<kbps>]

Flow-based Rate Limiting



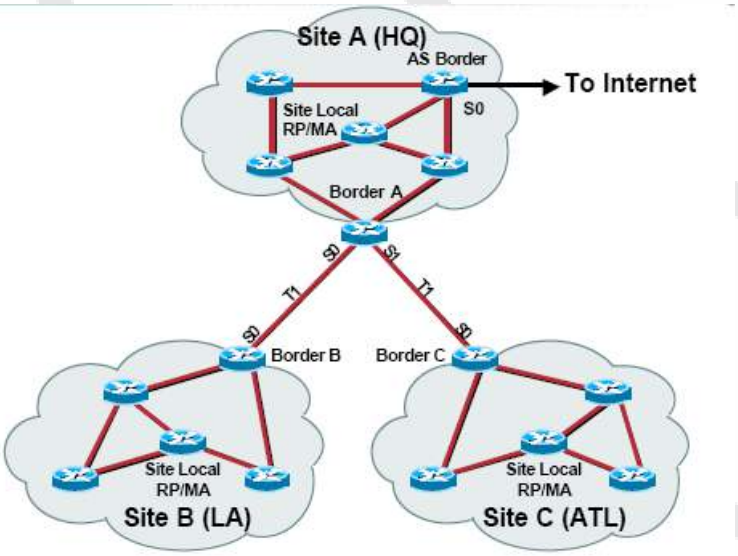
当限制值大于原有流量时，外出流量依旧未128Kbps

Interface-based Rate Limiting



基于接口的流量限制，将无法识别流量的大小，只能设置多少，使用多少带宽

BW Control via Admin-Scoping

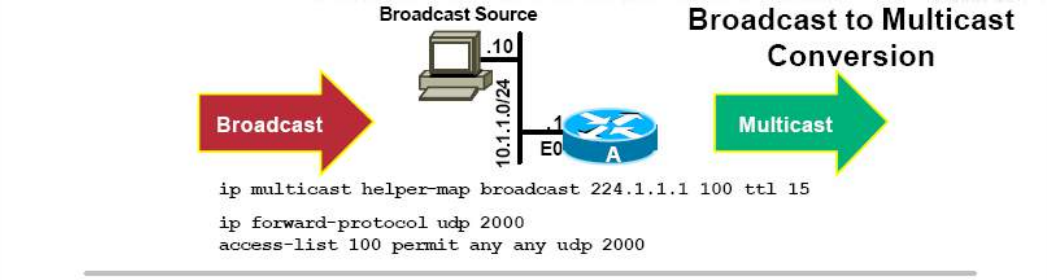


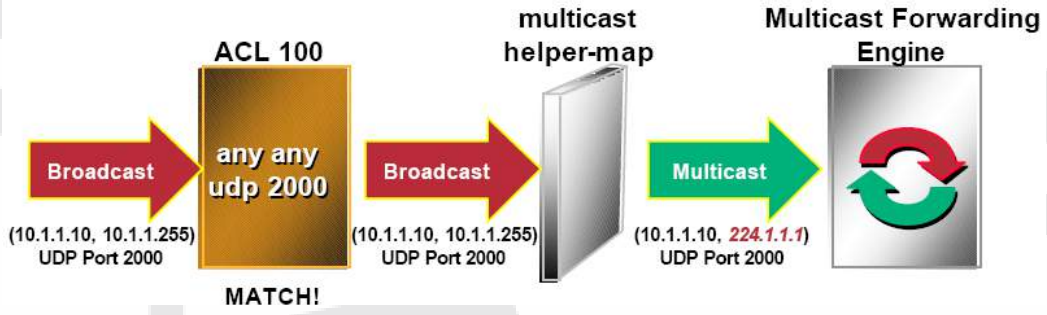
这些配置中，**ip multicast boundary**命令阻汇所有本地组播 信息通过边界串行接口进入或离开本场所。该组播边界也避免了由于可能的不应该离开本地场所的高速率组播信息因离开本地场所而使该接口发生阻塞。

```
ip multicast boundary 10
ip multicast ttl-threshold 16
```

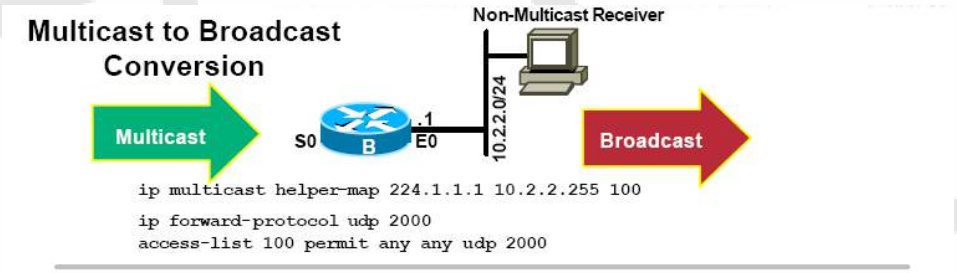
除了组播边界外，ip multicast ttl-threshold命令也定义了边界接口，以避免本地场所候选RP(c—RP)的通知泄痛出去；

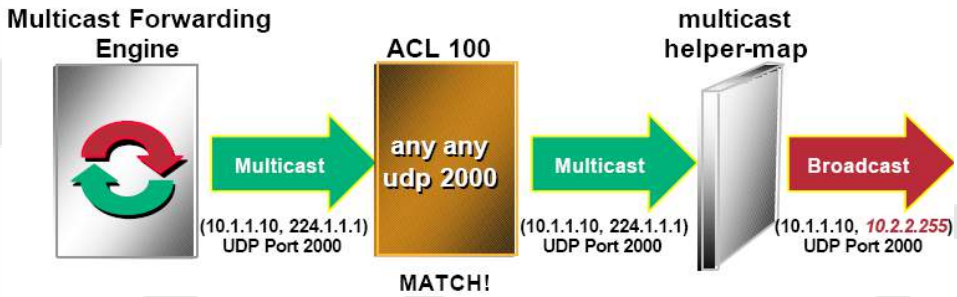
```
ip multicast helper-map broadcast 224.1.1.1 100 ttl 15
ip forward-protocol udp 2000
access-list 100 permit any any udp 2000
```





ip multicast helper-map 224.1.1.1 10.2.2.255 100
ip forward-protocol udp 2000
access-list 100 permit any any udp 2000





MBGP

Introduction

MBGP：多协议边界网关协议（MBGP：Multiprotocol BGP）

多协议边界网关协议（MBGP）的特征是对原来的 BGP 增加使整个网络能够组播路由，并能够在 BGP 自治系统间连接组播拓朴的能力。换句话说，MBGP 是增强版的携带 IP 组播路径的 BGP。BGP 携带了两组路径，一组是提供单播路由，另一组是提供组播路由。协议独立组播（PIM）使用连接组播路由的路由器建立数据分配树。当需要链接组播通信量，或限制通信量的资源使用时；也可能当网络访问点需要交换所有的组播通信量时，MBGP 都是非常有用的。MBGP 允许单播路由拓朴不同于组播路由拓朴。

BGP－4 携带三个唯一的 IPv4 信息：

NEXT-HOP 属性（即 IPv4 地址）；

AGGREGATOR（包含一个 IPv4 地址）；

NLRI（即 IPv4 地址前缀）。

任何 BGP 说话者，包括一个 MBGP 说话者，都需要有一个 IPv4 地址用于 AGGREGATOR 属性。为了使 BGP4 能够为多路网络层协议支持路由 BGP－4，需要附加两点能力：

连接特定网络层协议和下一跳信息的能力；

连接特定网络层和 NLRI 的能力。

关于 NLRI，MBGP中定义了两种属性：

MP_PEACH_NLRI，用来告知对等可行性路径，允许路径告知网络层用于下一跳的路径地址，同意特定地路径报告部分或所有子网的连接点（SNPAs）；

MP__UNREACH__NLRI，用来撤消服务器上的多路不可行性路由。

为了提供后台兼容性，同时也能简化进入 BGP-4 多路协议的能力的介绍，两种新的属性，多路协议可获得 NLRI（MP_UNPEACH_NLRI）并且 MBGP 可以用于不可获得的多路协议。MP_PEACH_NLRI 用来携带可获得目的文件组，同时利用下一跳信息转发这些目的文件。MP_UNPEACH_NLRI 主要用于携带不可获得目的文件。这两种属性都是可选的且不传递的。按照这种方式，不支持多路协议能力的说话者将忽略这些属性携带的信息，并不再将它传送给其他 BGP 说话者。

协议结构

可获得的多协议 NLRI — MP_REACH_NLRI （Type Code 14）：各属性列表如下：

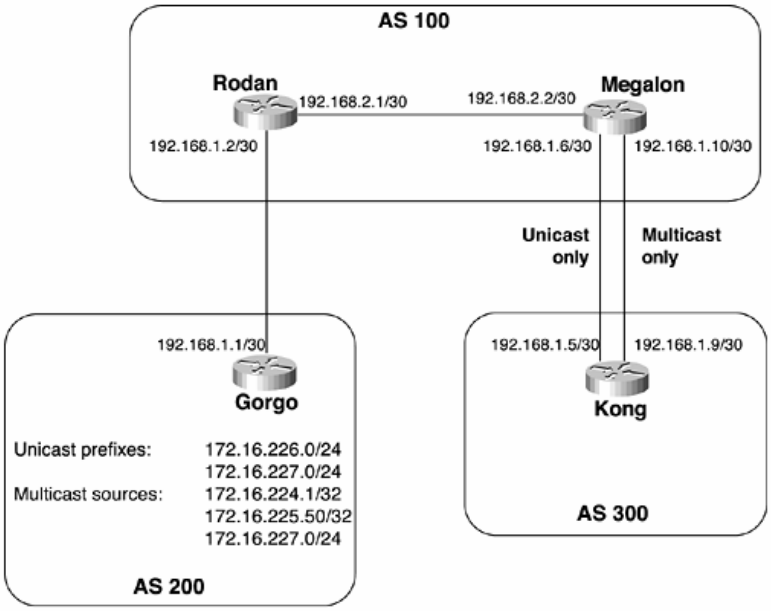
2 Bytes		1Byte	1Byte
Address Family Identifier		Subsequent Address Family Identifier	Length of Next Hop Network Address
Network Address of Next Hop （variable）			
Number of SNPAs	Length of first SNPA	First SNPA （variable）	Length of second SNPA （1 Byte）
Second SNPA （variable）	Length of Last SNPA （1 Byte）	Last SNPA （variable）	Network Layer Reachability Information （variable）

- Address Family Identifier — 传送与网络地址相关的网络层协议 ID。
- Subsequent Address Family Identifier — 提供另外的网络层可抵达信息。
- Length of Next Hop Network Address — 表示“Network Address of Next Hop”（下一跳网络地址）字段的长度（八位）。
- Network Address of Next Hop — 可变长字段，包含到达目的地的路径上的下一跳路由器的网络地址。
- Number of SNPAs — 表示下面字段中列出的 SNPAs 数目。值为 0 时表示该属性中没有 SNPA。
- Length of Nth SNPA — 表示“Nth SNPA of Next Hop”字段的长度（Semi-Octets）
- Nth SNPA of Next Hop — 包含一个路由器的 SNPA，其网络地址包含在“Network Address of Next Hop”字段。
- Network Layer Reachability Information — 列出可行线路的 NLRI。
- Address Family Identifier — 传送与 NLRI 相关的网络层协议。
- Subsequent Address Family Identifier — 提供另外的网络层可抵达信息（Network Layer Reachability Information）。
- Withdrawn Routes — 列出正在被撤出服务的线路 NLRI。

不可获得的多协议 NLRI - MP_UNREACH_NLRI：各属性列表如下：

Address Family Identifier （2 Bytes）	Subsequent Address Family Identifier （1 Byte）	Withdrawn Routes （variable）

Configurations




```
Gorgo
router bgp 200
 no synchronization
 network 172.16.226.0 mask 255.255.255.0
 network 172.16.227.0 mask 255.255.255.0
 neighbor 192.168.1.2 remote-as 100
 no auto-summary
 !
 address-family ipv4 multicast
 neighbor 192.168.1.2 activate
 network 172.16.224.1 mask 255.255.255.255
 network 172.16.225.50 mask 255.255.255.255
 network 172.16.227.0 mask 255.255.255.0
 exit-address-family
Rodan
router bgp 100
 no synchronization
 neighbor 192.168.1.1 remote-as 200
 neighbor 192.168.254.2 remote-as 100
 neighbor 192.168.254.2 update-source Loopback0
 neighbor 192.168.254.2 next-hop-self
 !
 address-family ipv4 multicast
 neighbor 192.168.1.1 activate
 neighbor 192.168.254.2 activate
 neighbor 192.168.254.2 next-hop-self
 exit-address-family
Megalon
router bgp 100
 no synchronization
 no bgp default ipv4-unicast
 neighbor 192.168.1.5 remote-as 300
 neighbor 192.168.1.5 activate
 neighbor 192.168.1.9 remote-as 300
 neighbor 192.168.254.1 remote-as 100
 neighbor 192.168.254.1 update-source Loopback0
 neighbor 192.168.254.1 activate
 neighbor 192.168.254.1 next-hop-self
 no auto-summary
 !
 address-family ipv4 multicast
 neighbor 192.168.1.9 activate
```

```
 neighbor 192.168.254.1 activate
 exit-address-family
Kong
router bgp 300
 no synchronization
 no bgp default ipv4-unicast
 neighbor 192.168.1.6 remote-as 100
 neighbor 192.168.1.6 activate
 neighbor 192.168.1.10 remote-as 100
 no auto-summary
 !
 address-family ipv4 multicast
 neighbor 192.168.1.10 activate
 exit-address-family
```

```
Kong#show ip bgp ipv4 unicast
BGP table version is 7, local router ID is 10.254.254.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.226.0/24  192.168.1.6                           0 100 200 i
*> 172.16.227.0/24  192.168.1.6                           0 100 200 i

Kong#show ip bgp ipv4 multicast
BGP table version is 10, local router ID is 10.254.254.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight Path
*> 172.16.224.1/32  192.168.1.10                          0 100 200 i
*> 172.16.225.50/32 192.168.1.10                          0 100 200 i
*> 172.16.227.0/24  192.168.1.10                          0 100 200 i
Kong#
```

SSM

Introduction

源特定组播（SSM：Source Specific Multicast）是一种区别于传统组播的新的业务模型，它使用组播组地址和组播源地址同时来标识一个组播会话，而不是向传统的组播服务那样只使用组播组地址来标识一个组播会话。SSM保留了传统PIM-SM模式中的主机显示加入组播组的高效性，但是跳过了PIM-SM模式中的共享树和RP（Rendezvous Point，集合点）规程。在传统PIM-SM模式中，共享树和RP规程使用(*，G)组对来表示一个组播会话，其中（G）表示一个特定的IP组播组，而（*）表示发向组播组G的任何一个源。SSM直接建立由(S,G)标识的一个组播最短路径树（SPT：Shortest Path Tree），其中（G）表示一个特定的IP组播组地址，而（S）表示发向组播组G的特定源的IP地址。

SSM 的一个（S,G）对也被称为一个频道(Channel)，以区分传统PIM-SM组播中的任意源组播组（ASM：Any Source Multicast）。由于ASM支持点到多点和多点到多点两种组播业务模式，因此源的发现过程是ASM复杂性的原因。例如在PIM-SM模式中，用户点击浏览器中的组播内容，接收端设备只被通知到组播组的内容，而没有被通知到组播源的信息。而在SSM模式中，用户端将同时接收到组播源和组播组信息。 v
因此,SSM特别适合于点到多点的组播服务，例如网络娱乐频道、网络新闻频道、网络体育频道等业务，但如果要求多点到多点组播服务则需要ASM模式。

PIM-SSM是对传统PIM协议的扩展，使用SSM，用户能直接从组播源接收组播业务量，PIM－SSM利用PIM-SM

的功能，在组播源和客户端之间，产生一个SPT树。但PIM-SSM在产生SPT树时，不需要汇聚点(RP)的帮助。Vtdd6[
一个具有SSM功能的网络相对于传统的PIM-SM网路来说，具有非常突出的优越性。网络中不再需要汇聚点，也不再需要共享树或RP的映射，同时网络中也不再需要MSDP协议，以完成RP与RP之间的源发现。

MSDP

Introduction

MSDP：组播源发现协议

（MSDP：Multicast Source Discovery Protocol）

组播源发现协议（MSDP）描述了一种连接多 PIM-SM（PIM-SM ： PIM Sparse Mode） 域的机制。每种 PIM-SM 域都使用自己独立的 RP ，它并不依赖于其它域内的 RP 。该优点在于：

不存在第三方（ Third-party ）资源依赖域内 RP 。

PIM-SM域只依靠本身的 RP 。

接收端域：只带接受端的域可以获取数据而不用全局通告组成员。MSDP可以和其它非PIM-SM协议一起使用。

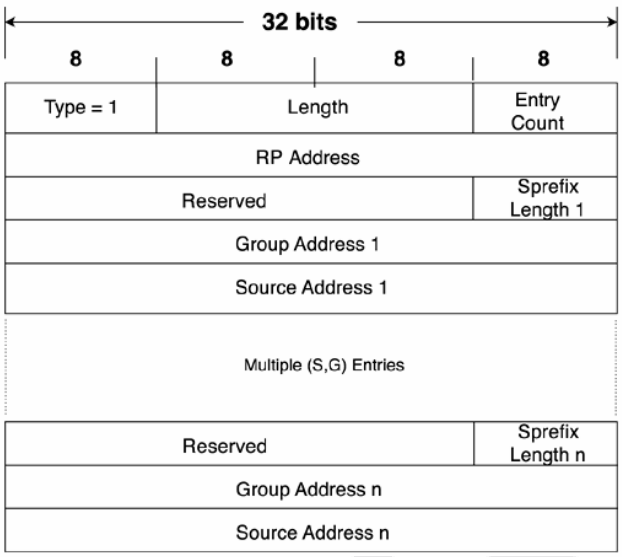
PIM-SM 域内的 MSDP 发话路由器与其它域内的 MSDP 对等设备之间存在一种 MSDP 对等关系，这种关系通过 TCP 连接形成，在其中控制信息进行交换。每个域都有一个或多个连接到这个虚拟拓扑结构。

这种拓朴结构使得域能从其它域发现组播源。如果组播源想知道含有接收端的域，那么 PIM-SM 中的标准源树建立机制就会被用于在域内分配树上传送组播数据。

MSDP使用TCP 639 端口建立对等连接(高ip侦听，低ip连接)，和BGP一样，对等间连接必须明确配置，当PIM DR在RP注册源时，RP向所有的MSDP对等体发送源激活消息，然后其他MSDP路由器将SA泛洪，为防止环回，现检查MBGP，再检查BGP

Message-Type

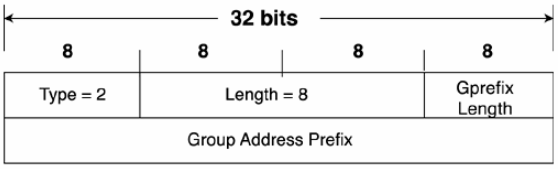
Source Active TLV



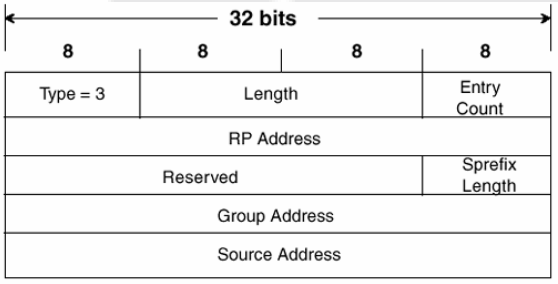
Entry Count	ecifies the number of (S, G) entries being advertised by the specified RP address.
RP Address	the IP address of the originating RP.
Reserved	set to all zeroes.
Sprefix Length	specifies the prefix length of the associated source address. This length is always32
Group Address	the multicast IP address to which the associated source is sending multicast packets.

Source Address the IP address of the active source.

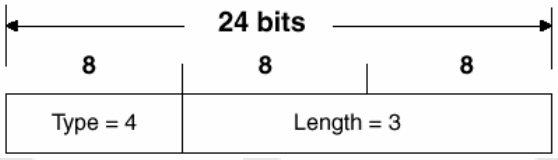
Source Active Request TLV



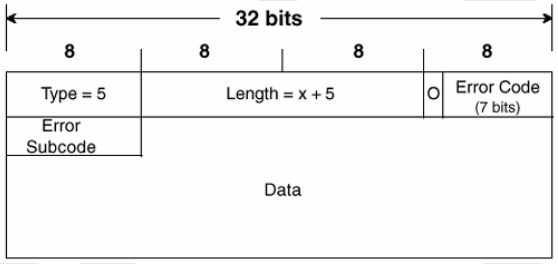
Source Active Response TLV



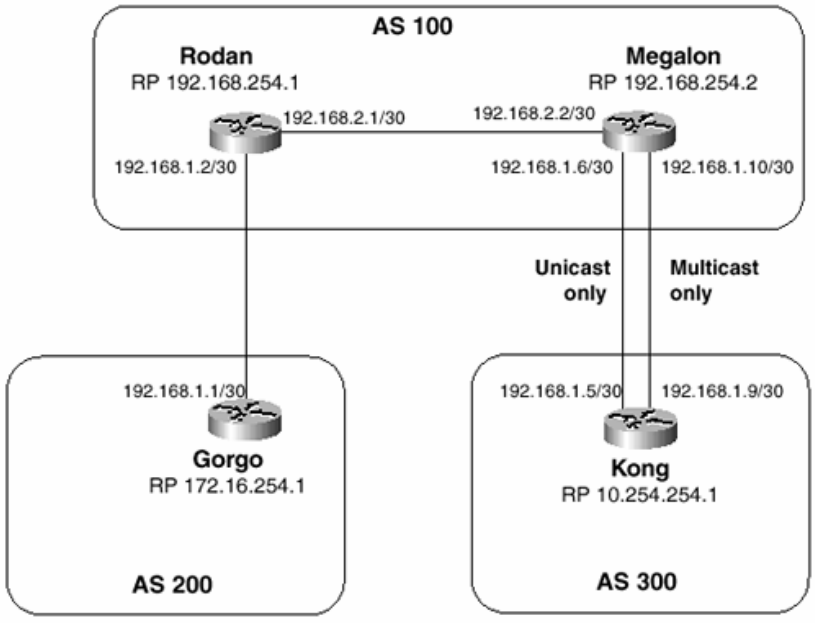
Keepalive TLV



Notification TLV



Configuration



Gorgo
ip msdp peer 192.168.1.2

Kong
ip msdp peer 192.168.1.10

Rodan
ip msdp peer 192.168.1.1
ip msdp peer 192.168.254.2 connect-source Loopback0

Megalon
ip msdp peer 192.168.254.1 connect-source Loopback0
ip msdp peer 192.168.1.9

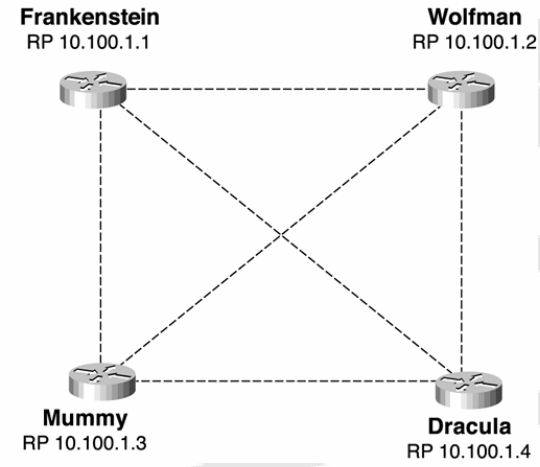
```
Megalon#show ip msdp peer
MSDP Peer 192.168.254.1 (?), AS 100
Description:
  Connection status:
    State: Up, Resets: 0, Connection source: Loopback0 (192.168.254.2)
    Uptime(Downtime): 3d22h, Messages sent/received: 5683/5677
    Output messages discarded: 0
    Connection and counters cleared 3d22h    ago
  SA Filtering:
    Input filter: none, route-map: none
    Output filter: none, route-map: none
  SA-Requests:
    Input filter: none
    Sending SA-Requests to peer: disabled
  Peer ttl threshold: 0
  Input queue size: 0, Output queue size: 0
MSDP Peer 192.168.1.9 (?), AS 300
Description:
  Connection status:
    State: Up, Resets: 0, Connection source: none configured
    Uptime(Downtime): 3d22h, Messages sent/received: 5674/5694
    Output messages discarded: 0
    Connection and counters cleared 3d22h    ago
  SA Filtering:
    Input filter: none, route-map: none
    Output filter: none, route-map: none
  SA-Requests:
    Input filter: none
    Sending SA-Requests to peer: disabled
  Peer ttl threshold: 0
  Input queue size: 0, Output queue size: 0
Megalon#
```

一份更详细的MSDP配置

```
ip pim rp-address 192.168.254.2
ip msdp peer 192.168.254.1 connect-source Loopback0
ip msdp description 192.168.254.1 Rodan in AS 100
ip msdp sa-filter out 192.168.254.1 list 101
ip msdp filter-sa-request 192.168.254.1 list 1
ip msdp sa-request 192.168.254.1
ip msdp ttl-threshold 192.168.254.1 5
ip msdp peer 192.168.1.9
ip msdp description 192.168.1.9 Kong in AS 300
ip msdp sa-filter in 192.168.1.9 list 101
ip msdp sa-filter out 192.168.1.9 list 103
ip msdp sa-request 192.168.1.9
ip msdp ttl-threshold 192.168.1.9 2
ip msdp cache-sa-state list 101
ip msdp redistribute list 102
!
access-list 1 permit 229.50.0.0 0.0.255.255
```

```
access-list 101 permit ip 10.254.0.0 0.0.255.255 224.0.0.0 31.255.255.255
access-list 102 permit ip 192.168.224.0 0.0.0.255 224.0.0.0 31.255.255.255
access-list 103 permit ip 172.16.0.0 0.0.255.255 230.0.0.0 0.255.255.255
access-list 103 permit ip 192.168.224.0 0.0.0.255 224.0.0.0 31.255.255.25
```

MSDP Mesh Groups




```
Frankenstein
ip pim rp-address 10.100.1.1
ip msdp peer 10.100.1.3 connect-source Loopback0
ip msdp description 10.100.1.3 to Mummy
ip msdp peer 10.100.1.2 connect-source Loopback0
ip msdp description 10.100.1.2 to Wolfman
ip msdp peer 10.100.1.4 connect-source Loopback0
ip msdp description 10.100.1.4 to Dracula
ip msdp mesh-group Boogeymen 10.100.1.3
ip msdp mesh-group Boogeymen 10.100.1.2
ip msdp mesh-group Boogeymen 10.100.1.4
```

Wide Area Networking

Introduction to Wide Area Networks

WAN是覆盖地理范围相对较为广阔的数据通信网络,它一般是利用公共载体，提供的设备进行传输.

WAN技术运行在OSI 的最下3 层

Defining WAN Terms

1 customer premises equipment,CPE: .客户前端设备，位于用户(subscriber)前端,用户所拥有的设备

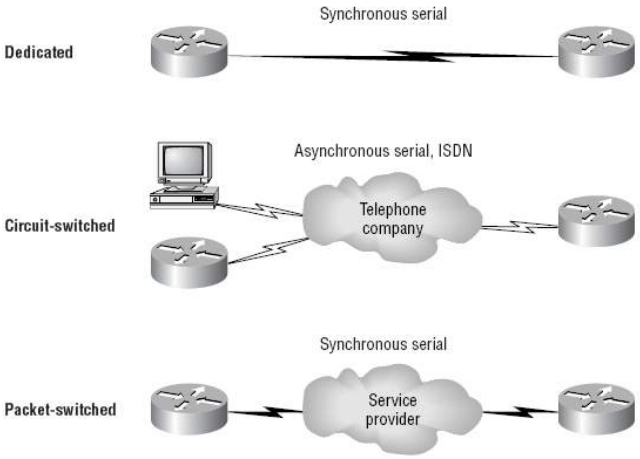
2.demarcation point: 分界点，服务提供商(service provider,SP)和CPE 的分隔点,一般位于电信机房由电信公司所拥有.用户这边连接到CSU/DSU 或者ISDN 接口来扩展延伸分界点

3.local loop: 本地回路，把分界点连接到central office(CO)的最近交换局

4.CO: 中心局，连接用户到服务商交换环境网络的点,有时候CO也叫做point of presence(POP)

5.toll network: Internet service provider(ISP)拥有,各种网络设备资源集合的网络

WAN Connection Types



Leased Lines(租用线路**)**,有时候也叫专线或点对点连接.预先布置好的通信路径,该路径从客户端通过电信公司的网络连接到远程网络.因为这样的通信线路通常是通过从电信公司租用而来,所以就叫做租用线路.这样线路方式一般由带宽和距离来定价,价格相对其他技术比如帧中继(Frame Relay)更为昂贵.速度可以达到45Mbps,一般使用HDLC 和PPP 的封装格式

Circuit Switching (电路交换**)**,这样的方式是连接只在有数据需要传输的时候才进行连接,通信完成后终止连接.这个和日常中打电话的过程很相似.一般用于对带宽要求过低的数据传输.例子有综合业务数字网络(Integrated Service Digital Network,ISDN).router 向远程站点发送数据时,交换线路用远程网络的线路号进行启动.对于ISDN,实际情况为拨远程ISDN 线路的电话号码.当2个网络连接并验证以后,就开始传输数据,数据传输完成,连接终止.

Packet switching包交换**(**或者翻译为分组交换**)**,用户共享电信公司资源,成本较低.在这样的网络中,网络连接电信公司网络,许多客户共享电信公司网络.然后电信公司在客户站点之间建立虚拟线路,数据包通过网络进行传输.这类例子有帧中继,ATM,X.25 等.速度可以从56Kpbs 达到T3 的45Mbps.采用多路复用的方式。

WAN Support

1.帧中继(Frame Relay):

一种包交换的技术,高性能,运行在OSI 的最下2 层即物理层和数据链路层.它其实是X.25 技术的简化版本,省略了X.25 技术的一些功能比如窗口技术和数据重发功能,这是因为帧中继工作在性能更好的WAN设备上;而且它比X.25 有更好的传输效率,速度可以从64Kbps 达到T3 的45Mbps.它还提供带宽的动态分配和拥塞控制

功能

2. ISDN:

ISDN 是1 种在已有的电话线路上传输语音和数据等数字服务.如果你对那种传统的拨号(dial-up)上网的速度感到不满的时候，你可以使用ISDN 的方式.ISDN 也可作为比如帧中继或者T1 连接的备份连接

3. 平衡链路访问过程(Link Access Procedure,Balanced,LAPB):

工作在OSI参考模型的数据链路层,是1 种面向连接的协议,一般和X.25 技术一起进行数据传输.因为它有严格的窗口和超时功能,所以使得代价很高

4. 高级数据链路控制(High-Level Data-Link Control,HDLC):

这个是由IBM 创建的同步数据链路控制(Synchronous Data Link Control,SDLC)衍生而来的.工作在OSI 参考模型的数据链路层.相比LAPB,HDLC 成本较低.HDLC 不会把多种网络层的协议封装在同1 个连接上.各个厂商的HDLC 都有他自己鉴定网络层协议的方式,所以各个厂商的HDLC 是不同的,私有化的

5. 点对点协议(Point-to-Point Protocol,PPP):

1 种工业标准(industry-standard)协议.因为各个厂商的HDLC 私有,所以PPP 可以用在不同厂商的设备之间的连接.PPP 使用网络控制协议(Network Control Protocol,NCP)来验证上层的OSI 参考模型的网络层协议

6. 异步传输模式(Asynchronous Transfer Mode,ATM):

国际电信联盟电信标准委员会(ITU-T)制定的信元(cell)中继续标准.ATM使用固定长度的53 字节长的信元方式进行传输,ATM网络的面向连接的

CISCO对WAN协议的支持

```
Kaka#config t
Enter configuration commands, one per line. End with CNTL/Z.
Kaka(config)#int s0/0
Kaka(config-if)#encapsulation ?
atm-dxi          ATM-DXI encapsulation
bstun            Block Serial tunneling (BSTUN)
frame-relay      Frame Relay networks
hdlc             Serial HDLC synchronous
lapb             LAPB (X.25 Level 2)
ppp              Point-to-Point protocol
sdlc             SDLC
sdlc-primary     SDLC (primary)
sdlc-secondary   SDLC (secondary)
smds             Switched Megabit Data Service (SMDS)
stun             Serial tunneling (STUN)
x25              X.25
```

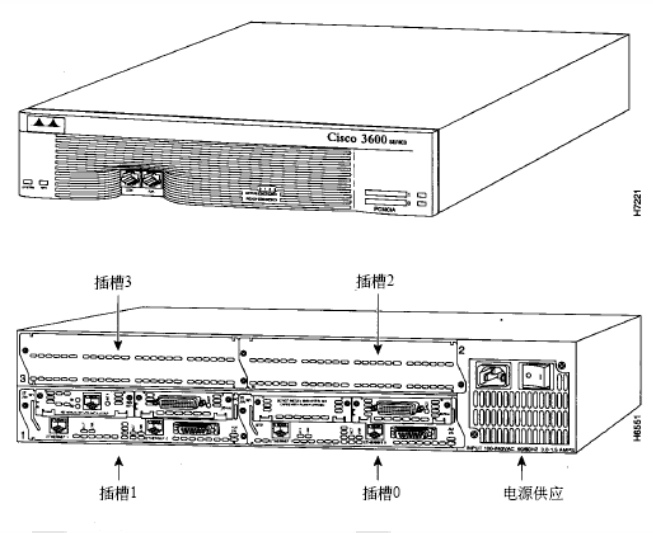
Cabling the Wide Area Network

Cisco 的串行连接支持几乎所有类型的WAN 服务.HDLC,PPP 和帧中继使用相同的物理层定义的接口,但是和ISDN的不一样.我们先来回顾下router 的接口类型:

LAN Interface

常见的以太网接口主要有AUI,BNC 和RJ-45 接口,还有FDDI,ATM,千兆以太网等都有相应的网络接口,下面分别介

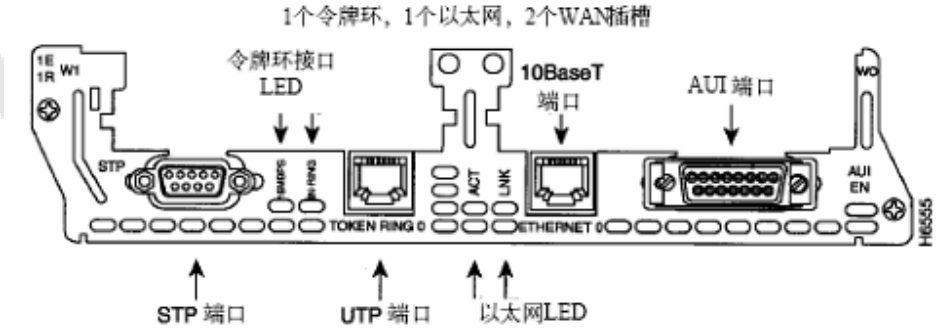
绍主要的几种局域网接口



以Cisco 3600为例，介绍各种路由器常见的接口

(1).AUI 接口

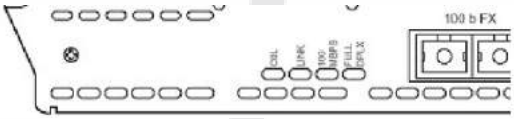
AUI 接口它就是用来与粗同轴电缆连接的接口,它是一种D 型15 针接口,这在令牌环网或总线型网络中是一种比较常见的接口之一.router 可通过粗同轴电缆收发器实现与10Base-5 网络的连接.但更多的则是借助于外接的收发转发器(AUI-to-RJ-45),实现与10Base-T 以太网络的连接.当然,也可借助于其他类型的收发转发器实现与细同轴电缆(10Base-2)或光缆(10Base-F)的连接



(2).RJ-45 接口

RJ-45 接口是我们最常见的接口了,它是我们常见的双绞线以太网接口.因为在快速以太网中也主要采用双绞线作为传输介质,所以根据接口的通信速率不同RJ-45 接口又可分为10Base-T 网RJ-45 接口和100Base-TX 网RJ-45 接口两类.其中,10Base-T 网的RJ-45 接口在router 中通常是标识为ETH,而100Base-TX 网的RJ-45 接口则通常标识为10/100bTX.

(3).SC 接口



SC 接口也就是我们常说的光纤接口,它是用于与光纤的连接.光纤接口通常是不直接用光纤连接至工作站,而是通过光纤连接到快速以太网或千兆以太网等具有光纤接口的switch.这种接口一般在高档router 才具有,如图

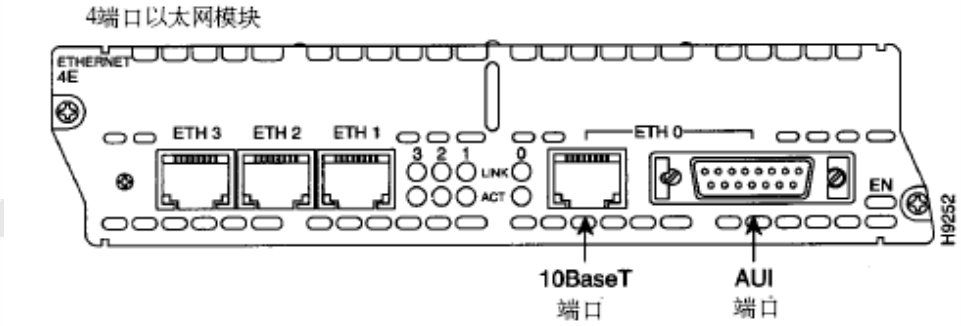
所示:

WAN Interface

在上面就讲过,router 不仅能实现局域网之间连接,更重要的应用还是在于局域网与广域网、广域网与广域网之间的连接.但是因为广域网规模大,网络环境复杂,所以也就决定了router 用于连接广域网的接口的速率要求非常高,在以太网中一般都要求在100Mbps 快速以太网以上.下面介绍几种常见的广域网接口

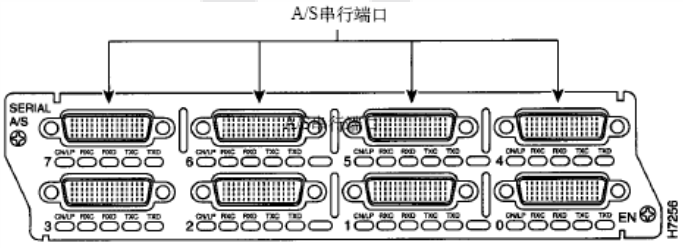
(1).RJ-45 接口

利用RJ-45 接口也可以建立广域网与局域网VLAN 之间,以及与远程网络或Internet 的连接.如果使用router 为不同VLAN 提供路由时,可以直接利用双绞线连接至不同的VLAN 接口.但要注意这里的RJ-45 接口所连接的网络一般就不太可有是10Base-T 这种了,一般都是100Mbps 快速以太网以上.如果必须通过光纤连接至远程网络,或连接的是其他类型的接口时,则需要借助于收发转发器才能实现彼此之间的连接



(2).AUI 接口

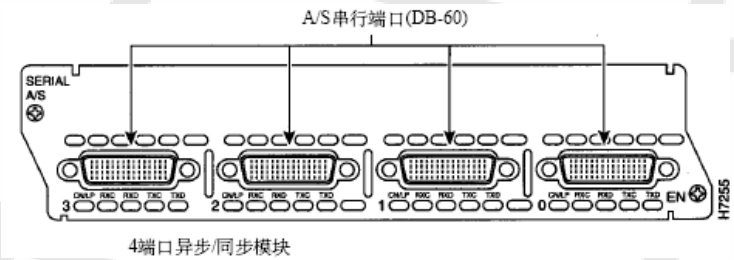
AUI 接口我们在局域网中也讲过,它是用于与粗同轴电缆连接的网络接口,其实AUI 接口也被常用于与广域网的连接,但是这种接口类型在广域网应用得比较少.在Cisco 2600 系列router上,提供了AUI 与RJ-45 两个广域网连接接口,



8端口异步/同步模块

(3).高速同步串口

在router 的广域网连接中,应用最多的接口还要算高速同步串口(Serial)了.这种接口主要是用于连接目前应用非常广泛的DDN,帧中继,X.25,PSTN 等网络连接模式.在企业网之间有时也通过DDN 或X.25 等广域网连接技术进行专线连接.这种同步接口一般要求速率非常高,因为一般来说通过这种接口所连接的网络的两端都要求实时同步



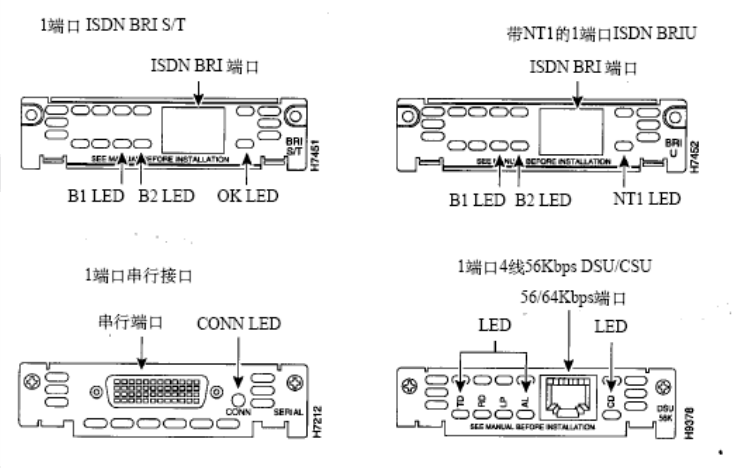
4端口异步/同步模块

(4).异步串口

异步串口主要是应用于Modem 或Modem 池的连接,如图8 所示.它主要用于实现远程计算机通过公用电话网拨入网络.这种异步接口相对于上面介绍的同步接口来说在速率上要求就松许多,因为它并不要求网络的两端保持实时同步,只要求能连续即可,主要是因为这种接口所连接的通信方式速率较低

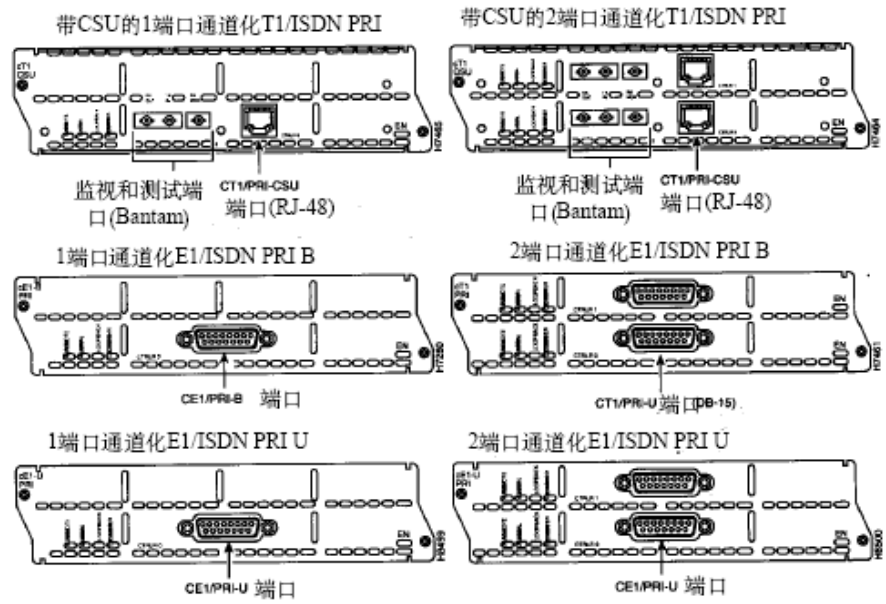
(5).ISDN BRI 接口

因ISDN 这种互联网接入方式连接速度上有它独特的一面,所以在当时ISDN刚兴起时在互联网的连接方式上还得到了充分的应用.ISDN BRI 接口用于ISDN 线路通过router 实现与Internet 或其他远程网络的连接,可实现128Kbps 的通信速率.ISDN 有两种速率连接接口,一种是ISDN BRI(基本速率接口);另一种是ISDN PRI(基群速率接口).ISDN BRI 接口是采用RJ-45标准,与ISDN NT1 的连接使用RJ-45-to-RJ-45 直通线.如图所示的BRI 为ISDN BRI 接口:



(6).其它特殊接口

3600系列路由器的多种通道化T1/ISDN-PRI和E1/ISDN-PRI网络模块



Serial Transmission and Parallel Transmission

串行传输(serial transmission):1 次1 位,WAN 普遍使用这种方式传输

并行传输(parallel transmission):1 次8 位

Cisco 使用私有的60 针脚的串行连接器.连接器的另外1 端的类型可以有以下几种:

1.EIA/TIA-232

2.EIA/TIA-449

3.V.35(与CSU/DSU 连接)

4.X.21(X.25 中使用)

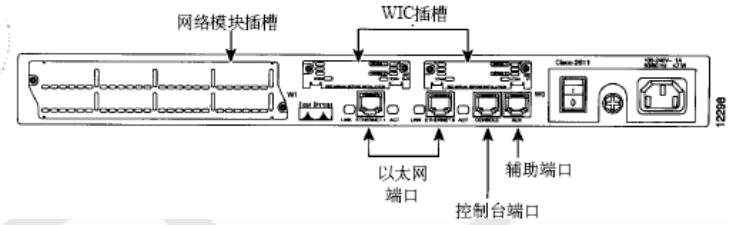
5.EIA-530

Data Terminal Equipment(DTE) and Data Communication Equipment(DCE)

Router 的接口默认是DTE,它们和DCE 比如CSU/DSU 相连,DCE 的主要作用就是提供时钟频率

Fixed and Modular Interfaces

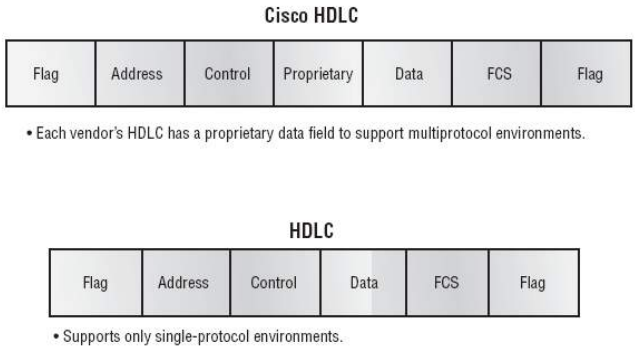
Cisco全系列路由器采用一些固定接口，和一些具有模块化的接口，正如如上接口的图片，都是一个个的模块，需要相应的功能的时候，购置相应的模块，后期还可以对某些功能进行升级，例如2600系列还可以加载语音模块。



High-Level Data-Link Control(HDLC) Protocol

HDLC 是1 种ISO 标准,面向比特(bit-oriented)的数据链路层协议.它定义了在同步串行连接的封装方法.HDLC 是种在租用线路上使用的点对点协议.HDLC 不使用验证(authentication)在面向字节(byte-oriented)的协议中,控制信息使用整个字节进行编码;但是在面向比特的协议中,使用单独的1 个比特(bit)来代表控制信息.面向比特的协议包括SDLC,LLC,HDLC,TCP,IP等

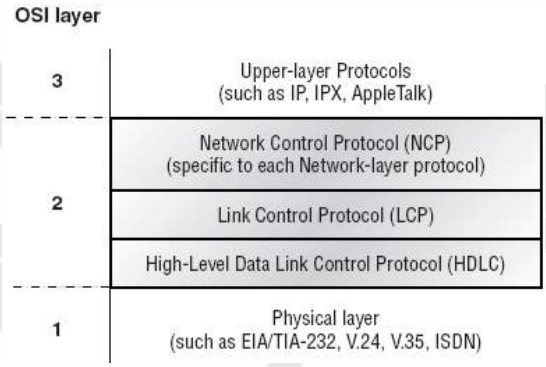
HDLC 是Cisco 同步串行连接中默认的封装格式.当然,Cisco 的HDLC 是私有的,即不能和其他厂商的HDLC 相互通信.而且各个厂商的HDLC 均是私有的.来看看Cisco的HDLC和HDLC的帧的格式,如图:



假如你有2 个不同厂商的设备,就不能使用HDLC,就要使用PPP

Point-to-Point Protocol(PPP)

PPP 是OSI 参考模型层2 协议,可以使用在异步串行连接比如拨号(dial-up)或者同步串行连接比如ISDN上.它使用链路控制协议(Link Control Protocol,LCP)来建立和保持连接.PPP 的主要目的是通过数据链路层点对点的传输OSI 参考模型层3 数据包.来看下PPP 的协议栈,如图:



PPP 的4 个组件如上图.注意PPP 的协议栈只定义在OSI 参考模型的层1 和层2.NCP 用于建立和配置多种网络层协议.PPP 允许采用多种网络层协议.PPP 可以工作在任何DCE/DTE 接口比如EIA/TIA-323-C(以前为RS-232-C),ITU-T(原CCITT)V.35 等.唯一要求是必须提供全双工线路

Dial

MODEM

DTE-DCE

DTE（数字终端设备）---DCE（数字传输设备）

EIA/ETA-232-C interface

使用DB-25接口，其中8根用于连接DTE和DCE

数据传送

2 TxD DTE将数据传输给DCE

3 RxD DTE接受来自DCE的数据

7 GrD 提供电压度量的参考地

硬件流控

4 RTS DTE有空闲缓存

5 CTS CTE有空闲缓存

20 DTR 告诉DCE，DTE已经作好接受准备

8 CD 载波检测

6 DSR 数据采集准备就绪

错误控制和数据的压缩标准：

错误控制的几个标准：一般包括MNP和LAPM。是两个不同厂商的协议。

压缩：

MNP5有2：1的压缩比

V.42BIS有4：1的压缩比

v.44有6：1的压缩比

调制解调的标准：

开放协议：一般来讲注重v.90和v.92他们的下行速度都是56k。而上行速度v.90为33k。v.92有48k

私有协议比如：K56flex和x2 他们都是56k的标准

用v.90协议要注意，如果用两个MODOM互相连接的话。那么速度就是33K。而不是56K。因为上行速度的原因。

其中，压缩标准和MODON的传输速度是有很大关系的。

比如，使用v.90的标准。用v.42bis的压缩标准的话（4：1）那么理论上速度可以达到4*56K的下行速度。

Configuration

DTE-DCE

配置一部调制解调器需要分几个部分。例如，用户需要知道在哪里物理地连接调制解调器，以及如何在Cisco IOS中对其进行编址。用户还需要知道物理连接的特性，如通信的速度、采用的流控制方法等等。

在Cisco 接入服务器(Cisco Access Server)中,有4条不同类型的线路可以使用：Console(CON或CTY)、Auxiliary（AUX）、Asynchronous（TTY）和Virtual Terminal（VTY）。并不是所有的Cisco 设备都有AUX和T T Y线路，但所有的设备都有CON和VTY线路。CON或CTY线路都要参考控制台端口。读者已经了解到，控制台端口用以配置路由器。在管理Cisco接入服务器时，用户不需要配置任何其他接口。控制台端口的默认通信设置如下：

- 9600bps。
- 8数据位。
- 无奇偶校验。
- 1个停止位。
- 无流控制。

只要用户有了到控制台端口和终端访问程序（例如HyperTerminal或Tera Term Pro等）的物理访问能力，通过这些设置就可以访问IOS。控制台端口通常编址为线路0。

下一类线路就是异步端口。诸如Cisco 2509等设备有8个物理异步端口，每个端口上都可以连接外部的调制解调器。

Cisco 接入服务器产品（AS5x00）可以处理ISDN PRI。然而为了支持从POTS线路来的连接，这些产品仍然需要有调制解调器。这是因为Cisco 接入服务器需要将进来的模拟信号转换为数字信号。因此， Cisco 创建了可以容纳许多调制解调器的网络模块（在有些模块中可容纳多达120个调制解调器）。这些调制解调器即众所周知的MICA调制解调器，被认为是像2509中物理异步连接那样的异步连接。二者之间的唯一区别是这里的异步线路上连接着调制解调器。在必要的情况下，用户可以独立于其他调制解调器之外来配置每个调制解调器（例如，当用户的每条线路上连接着几个不同的调制解调器时）。但是，在多数时候，异步线路是集中在一起、作为一个单独的单元进行配置的。

Cisco IOS中可用的TTY线路数量要取决于在Cisco 接入服务器中有多少内部MICA调制解调器与/或异步端口。如果辅助端口可用，则将立即出现在异步线路之后。辅助端口并不作为第2个控制台端口使用，而是担任着备份的异步端口的角色。

在Cisco 接入服务器中最后一类线路是V T Y线路，telnet线路这个名字对人们大概更为熟悉。VTY线路不与任何物理接口关联，而是通过配置文件在路由器中动态地创建。有多少VTY线路，就可以建立多少到Cisco 接入服务器的telnet对话。要确定哪些线路可用，可以在全局配置模式下使用IOS命令show lines。其中有一个控制台端口（线路0）、1 6个异步接口（线路1 ~ 16）、一个辅助端口（线路17）和5个虚拟终端线路（线路18 ~ 22）。

```
Router>show line
Tty Typ Tx/Rx A Modem Roty AccO Accl Uses Noise Overruns
*   0 CTY - - - - - 0 0 0/0
```

```
A   1 TTY 9600/9600 - - - - - 0 0 0/0   //   *表示当前被占用
2 TTY 9600/9600 - - - - - 0 0 0/0      //   A 表示活动
I   3 TTY 9600/9600 - - - - - 0 0 0/0      //   I   处于非活动状态
4 TTY 9600/9600 - - - - - 0 0 0/0
5 TTY 9600/9600 - - - - - 0 0 0/0
6 TTY 9600/9600 - - - - - 0 0 0/0
7 TTY 9600/9600 - - - - - 0 0 0/0
8 TTY 9600/9600 - - - - - 0 0 0/0
9 TTY 9600/9600 - - - - - 0 0 0/0
10 TTY 9600/9600 - - - - - 0 0 0/0
11 TTY 9600/9600 - - - - - 0 0 0/0
12 TTY 9600/9600 - - - - - 0 0 0/0
13 TTY 9600/9600 - - - - - 0 0 0/0
14 TTY 9600/9600 - - - - - 0 0 0/0
15 TTY 9600/9600 - - - - - 0 0 0/0
16 TTY 9600/9600 - - - - - 0 0 0/0
17 AUX 9600/9600 - - - - - 0 0 0/0
18 VTY - - - - - 0 0 0/0
19 VTY - - - - - 0 0 0/0
20 VTY - - - - - 0 0 0/0
21 VTY - - - - - 0 0 0/0
22 VTY - - - - - 0 0 0/0
```

在Cisco IOS中对这4种线路（CON、TTY、AUX和VTY）进行编址时，可以使用相对线路编号方式和绝对线路编号方式两种方法来识别线路。用show line命令查找TTY值，就可以识别出任何线路的绝对线路编号。

绝对线路编号	相对线路编号	绝对线路编号	相对线路编号
线路0	CON 1	线路17	AUX 0
线路18	VTY 0	线路19	VTY 1
线路20	VTY 2	线路21	VTY 3
线路22	VTY 4		

要用相对编址配置Cisco 路由器的线路，要用下面的命令：

```
Router(config)#line line-type line-number
line-type   con      0        Configure relative console line 0
            aux      0        Configure relative auxiliary line 0
            tty      x-y      Configure the relative tty line, x is the starting number
                              of the tty lines and y is the last tty line available
            vty      0-z      Configure relative telnet session line where 0 is the first
                              line and z is the number of vty lines available
```

要用绝对编址配置Cisco 路由器的线路：

```
Router(config)#line line –number
      line-number   x   The absolute line to be configured
```

稍变化一下前面命令，可以同时配置多条线路：

```
Router(config)#line first-line last-line
first - line   x     The absolute line to be configured
last - line   x     The absolute line to be configured
```

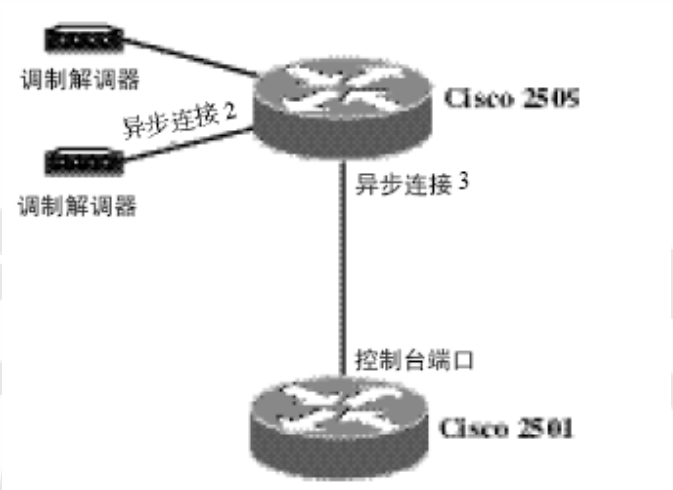
当使用固定配置的路由器时，要提前确定某特定接口的相对线路编号方式时，可使用表中所列的简单规则。

接口	关键字	线路数目	线路号
控制台	con	1	Line 0
异步设备	tty	x	Lines 1 through x
辅助	aux	1	Line x + 1
虚拟终端	vty	y	Line x+1 through y

连接到异步设备

Cisco 接入服务器（Cisco Access Servers）可支持传入和传出的异步连接。传入连接的典型例子就是用户

拨入Cisco 接入服务器。传入连接是最容易让人理解的一种连接类型。



在上图中，表示两部调制解调器和一个Cisco 2509连接在一个Cisco 2501的控制台端口和辅助端口上。传出异步连接可以使用户能够修改调制解调器的配置。或者，用户可以连接到Cisco 2501的控制台端口，这样，即使是在2501的所有物理接口都不支持远程登录会话时也可以对调制解调器进行配置。这是路由器中的一个小秘密。这种连接称为反向远程登录（reverse telnet）。

要执行反向远程登录，必须在需要反向远程登录的线路上执行下面这两个命令： **transport input all**和 **modem inout**。

```
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line 1 48
Router(config-line)#transport input all
Router(config-line)#modem inout
```

既然Cisco 接入服务器支持反向远程登录，那么用户只需要用telnet [ipaddress port]命令远程登录到感兴趣的地址便可

```
Router#telnet 192.168.12.1 2001
Trying 192.168.12.1, 2001 ... Open
**********************************
*Welcome to kaka's   router *
**********************************

User Access Verification
Password :
```

注意如果在异步端口之外建立反向远程登录有困难，那么除要用到**transport input all** **modem inout**命令外，还可以使用命令**no-exec**

端口20xx定义了用户想远程登录到异步线路；端口xx01指定了登录到哪条异步线路（在此例中是指Cisco 2501的线路1或AUX 0）。我们建议用户在回送接口中设置一个未注册的IP地址，并用这个IP地址来执行反向远程登录。为什么不简单地使用真正接口中的IP地址呢？当然也可以这样做，但是如果此接口崩溃（物理上断开连接或者网络卡失效），则IP地址就不再有效了，那么用户就不能反向远程登录了—或者，更严重的是，在配置调制解调器的时候，可能会失去连接。

服务	基准TCP端口
Telnet	2000
Raw TCP协议	4000
Telnet 协议，二进制模式	6000

Xremote 协议 9000

终止一个反向远程登录会话分两步：首先需要挂起反向远程登录会话，然后再在全局配置模式中终止会话。要挂起反向远程登录会话，按下Ctrl+Shift+6，再输入x即可。要终止反向远程登录会话，发布命令**disconnect [session number]**。用命令**show sessions**便可以查找出**session number**。

用命令**show sessions**和**disconnect**来终止反向远程登录会话。

用命令**ip host**可以简化连续的将远程登录连接反向的过程，如下所示：

```
Router(config)# ip host hostname port ip-address-of-router
hostname WORD Name used to represent the shortcut
port 20XX Port address of asynchronous port,
XX is the absolute line number
ip-address-of-router a.b.c.d IP address of the router


Router#show sessions
Conn Host Address Byte Idle Conn Name
* 1 192.168.12.1 192.168.12.1 0 0
192.168.12.1
Router#disconnect 1
Closing connection to 192.168.12.1 [confirm]
```

用**ip host**命令简化反向远程登录会话

```
RouterE(config)#ip host RouterA 2001 192.168.12.1
RouterE(config)#end
RouterE#
RouterE#Router A
Trying modem (192.168.12.1, 2001)... Open
* * * * * * * * * * * * * * * * * *
*Welcome to Router A *
* * * * * * * * * * * * * * * * * *
User Access Verification
Password :
```

线路配置

用户要建立到Cisco 接入服务器的异步连接，需要确定正确配置了调制解调器和线路。我们可以单独地配置每条线路，也可以将所有的线路作为一个整体来同时配置。

配置异步端口线路的命令有许多。要有效地应用这些命令，需要知道它们各自的目的。Speed命令确定了线路的速度。线路速度并不是拨号连接的速度，而是DTE到DCE的通信速度。记住要实现有效的压缩，线路速度需要大于拨号连接的速度，这是非常重要的；否则，调制解调器就会缺乏数据，从而浪费了带宽。

用于异步端口的典型的flowcontrol方法是通过硬件实现的。它等价于控制数据流的RTS和CTS。调制解调器在其缓冲器满时会通知路由器，并通知路由器停止发送数据。

由于在默认情况下， Cisco路由器的异步端口并不接收传入的网络连接，所以用户需要识别在异步连接上允许使用哪种协议。Transport命令允许用户定义所用的协议（ clat、mop、pad、rlogin、telnet或v120）。**Transport input all**命令则允许所有的流量类型在线路上传递。

```
Router(config-line)#transport direction-preference protocol
direction-input protocols that can preference connect on the Cisco Access Server
output protocols that can be used on for outgoing connections
Router#config t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#line 65 76
Router(config-line)#speed 115200
Router(config-line)#flowcontrol hardware
Router(config-line)#transport input all
Router(config-line)#modem inout
```

```
Router(config-line)#stopbits 1
Router(config-line)#end
Router#
                    preference      preferred protocol
protocol            all             all protocols
                    pad             X.3 PAD
                    rlogin          Unix rlogin protocol
                    telnet          TCP/IP Telnet protocol
                    v120            Async over ISDN
```

Modem命令为Cisco接入服务器提供了极大的灵活性。Modem命令可以确定我们所支持的异步连接的方向。要指定通信的方向，可使用下面的命令：

```
Router(config-line)#modem direction-preference
direction-dialin protocols that can connect to preference
the Cisco Access Server
inout preferred protocol
```

Cisco 接入服务器支持许多异步协议，如AppleTalk远程访问协议（Apple Talk Remote Access Protocol，简写为ARAP）、串行线路Internet 协议（ Serial Line Internet Protocol，简写为SLIP）和点到点协议（Point to Point Protocol，简写为PPP）。Cisco IOS中autoselect命令可以检测正在

使用的协议。要配置异步线路以支持不同的协议，可使用下面的命令：

```
Router(config-line)#autoselect protocol
Protocol        arap            Support ARAP connections
                during-log in   Support character-based connections
                ppp             Support PPP connections
                slip Support    SLIP connections
```

其他的许多命令中，提出的选项不是这个协议就是那个协议，与此不同，多次使用**autoselect**命令来支持感兴趣的协议是很司空见惯的事情。

配置一条异步线路以支持传入的连接

```
Router(config)#line 65 76
Router(config-line)#modem dialin
```

配置PPP、ARAP和during login的命令

```
Router(config)#line 65 76
Router(config-line)#autoselect ppp
Router(config-line)#autoselect arap
Router(config-line)#autoselect during-login
```

在从前，当两个调制解调器建立连接后使用电子公告牌系统时，用户大概需要回车多次才可出现用户名/口令提示符。这种终端连接在Hyper Termianl或Kermit等程序下是很正常的。通过**autoselect during-login**命令，只要用户建立起与接收调制解调器的连接， Cisco 接入服务器就可以立即显示提示符。如果用户用的是终端程序如Hyper Termianl等来建立连接，那么就会看到一个登录屏幕。如果用户用PPP连接，那么，一建立连接就可以开始发送包。

AT命令与初始化字符串

用于配置调制解调器和解决调制解调器故障的最基本的命令来自AT命令集。AT命令集最初是由Hayes开发的，是为了与其Smartmodem 300协同工作的。

AT命令集可以分解为两种不同的指令：配置命令和操作命令（通常称为S-Registers）。为了能给大多数调制解调器发送命令，用户必须用AT命令启动。从中，通过使用一系列彼此互补的命令，用户可以配置调制解调器的各种属性。

配置一个Cisco MICA调制解调器。

这些命令的结果是：

AT&FS0=1E1S29=6

需要配置的各种属性

配置命令	操作命令
定义流控制	拨入电话号码
设置调制	挂起电话
设置数据压缩	测试调制解调器
定义RS-232信令特征	询问调制解调器

各种MICA AT命令

要求的效果	命令
设置为出厂默认值	&F
应答电话	S0=1
启用到工作站的命令回显	E1
使用V. 9 0调制	S29=6

这套命令集非常有限，可能不会按照用户希望的方式初始化调制解调器。创建调制解调器脚本时要做一件非常重要的工作，就是将调制解调器设置回为出厂默认值。由于在调制解调器中可以设置的变量很多，所以需要确定我们的工作是从一种已知的状态开始的。如前面的例子所示，几个AT命令是搁在一起的，这称为初始化字符串（initialization string）。

虽然用来配置初始化字符串的许多命令都非常相似，但在不同的供应商、甚至是不同的型号之间，命令都是各异的。要了解用于配置某特定的调制解调器的命令，请参考随调制解调器的文档。因为每次启动Cisco 接入服务器或用户中断会话时，都要在调制解调器上用到初始化命令，所以需要一种方法来使初始化字符串自动执行。

MODEM-Auto Configuration

调制解调器自动配置用于使连在Cisco 接入服务器上的调制解调器的配置自动进行。针对一些最常用的调制解调器， Cisco 接入服务器带有许多预配置的初始化字符串。预配置的初始化字符串存储在modemcap数据库中。

以下是Cisco IOS 11.3所带的位于modemcap数据库中的一些入口：

a) codex_3260
b) usr_courier
c) usr_sportster
d) hayes_optima
e) global_village
f) viva
g) telebit_t3000
h) microcom_hdms
i) microcom_server
j) nec_v34
k) nec_v110
l) nec_piafs
m) cisco_v110
n) mica

除前面的入口外，用户还可以创建自己的初始化字符串。例如因为不使用其中列出的任何调制解调器，或者因为需要修改预配置的脚本时，我们大概就需要创建自己的初始化字符串。每种modemcap数据库入口使用的是特定供应商自己独特的一套命令集。要查看某种特定的modemcap入口，可以使用show modemcap命令。显示出一列usr_courier modemcap入口知道的配置参数。

```
Router#show modemcap usr_courier
Modemcap values for usr_courier
Factory Defaults (FD): &F
Autoanswer (AA): S0=1
Carrier Detect (CD): &C1
Drop with DTR (DTR): &D2
Hardware Flowcontrol (HFL): &H1&R2
Lock DTE speed (SPD): &B1
DTE locking speed (DTE): [not set]
Best Error Control (BER): &M4
Best Compression (BCP): &K1
```

```
No Error Control (NER): &M0
No Compression (NCP): &K0
No Echo (NEC): E0
No Result Codes (NRS): Q1
Software Flowcontrol (SFL): [not set]
Caller ID (CID): [not set]
On-hook (ONH): H0
Off-hook (OFH): H1
Miscellaneous (MSC): [not set]
Template entry (TPL): default
Modem entry is built-in.
```

创建新的**modemcap**数据库入口

如果modemcap数据库的入口不能满足要求，那么您可以创建新的人口。在创建新的入口之前，我们必须知道需要修改的属性及其相应的命令、以及新的modemcap入口。我们可以修改以下属性：

- 自动应答。
- 最佳错误控制。
- 载波检测。
- 出厂默认值。
- 其他命令。
- 无回显。
- 无结果。
- 锁定调制解调器速度。
- 最佳压缩。
- 呼叫者ID。
- DTR。
- 硬件流控制。
- 无压缩。
- 无错误控制。
- 软件流控制。

要创建并编辑一个新的modemcap入口，可使用modemcap edit命令。Modemcap edit命令的格式如下：

```
Router(config)#modemcap edit entry-name attribute-name setting
```

entry-name	WORD	Name for the new modemcap entry
attribute-name	autoanswer	Edit entry for autoanswer
	best-compression	Edit entry for best copression
	best-error-control	Edit entry for best error control
	caller-id	Edit entry for Caller ID
	carrier-detect	Edit entry for carrier-detect
	dtr	Edit entry for DTR
	factory-default	Edit entry for factory default
	hardware-flowcontrol	Edit entry for hardware flowcontrol
	miscellaneous	Edit entry for miscellaneous commands
	no-compression	Edit entry for no compression
	no-echo	Edit entry for no echo
	no-error-control	Edit entry for no error control
	no-results	Edit entry for no results (quiet mode)
	software-flowcontrol	Edit entry for software flowcontrol
	speed	Edit entry for locking modem speed

创建一个新的modemcap入口

```
Router#configure terminal
```

```
Router(config)#modemcap edit TestModem autoanswer AA
```

修改的假想的Test Modem调制解调器的自动应答属性。然而，在实际情况中，您大概需要修改许多属性，此时，您需要给每个属性都输入**modemcap edit**命令。

注意modemcap数据库是区分大小写的。如果用户不注意，就可能意外地创建名称相同而大小写不同的多个入口。

有时，我们也许会想扩展modemcap数据库中一个入口的功能。但很遗憾，我们不能修改任何预定义的modemcap入口。因此，这时我们必须创建一个与已经定义的那个modemcap入口非常相似的modemcap入口。因为这里的命令的含义都很隐晦（如&f），所以创建modemcap数据库入口比较费事，而且很容易出错。

Cisco提供了一种方法，可使用户利用预定义的modemcap入口作为模板来创建新的modemcap入口。因为是从已知的良好状态下创建入口，并从此处进行修改，所以就减少了可能出现的错误。复制预定义的modemcap入口的命令如下：

```
Router(config)#modemcap edit new-modemcap-entry template
predefined-modemcap-entry
        new-modemcap-entry WORD Name for the new modemcap entry
        predefined-modemcap-entry WORD Name of the modemcap entry to copy
```

用户需要注意，该命令需要从全局配置模式下执行。从这里，用户可以用modemcap edit命令修改新的modemcap入口，该入口存储在位于NVRAM的配置文件中。

配置连在Cisco 接入服务器上的调制解调器

我们已经掌握了许多信息，但我们仍然需要将其应用到Cisco接入服务器的线路上。要从MICA modemcap入口将命令应用到Cisco 接入服务器的线路上，可按如下配置Cisco 接入服务器以便使用MICA modemcap入口

```
Router(config)#line 65 76
Router(config-line)#modem autoconfigure type mica
```

调制解调器的自动发现

Cisco 提供了一种自动发现哪一部调制解调器连接在Cisco 接入服务器上的方法。Cisco IOS从modemcap数据库的第1个入口启动，试图将连着的调制解调器初始化。如果调制解调器初始化成功， IOS就无须进行进一步的工作。如果没能初始化，则IOS转向modemcap的下一个入口。IOS将重复这一过程，直到调制解调器成功地进行了初始化，否则， IOS将用完所有的数据库入口。

如果IOS遍历了所有的modemcap入口，还是没能初始化调制解调器，那么用户就必须创建一个modemcap入口来支持这个调制解调器。当我们知道入口之一将配置调制解调器，但不知道哪一个入口来做这项工作时， **modem auto-discovery**命令就很有用了。然而在实际工作中，用**modem auto-discovery**命令初始化调制解调器承担着一种惩罚—时间。在对拨入线路有巨大要求的环境中， IOS遍及所有modemcap入口所需的额外时间就是用户不能使用调制解调器的时间。在小规模环境下，这也许不是什么问题。但是在具有几百部调制解调器的环境下，就可能会用10分钟或更长的时间来等待调制解调器初始化。此外，路由器内核有几个线程可用于调制解调器的初始化。如果用户有许多调制解调器，则路由器就要用很长的时间来对其进行配置。

对话脚本

我们可以用对话脚本（chat script）来指导调制解调器初始化、拨出、登录到远程系统。对话脚本是通过接收一些期望的输入并将一些预定义的数据作为结果发送出去而进行的。以下是创建一个对话脚本的命令：

```
Router(config)# chat-script chat-script-name expect-string send string
        chat-script-name WORD Name for the new chat script
```

expect-string WORD Expected string

send-string WORD Response string

要定义两台设备间发生的信号交换，对话脚本就要对预期的输入进行处理并发送对。例如当执行AT命令时，将会得到来自调制解调器的“ OK”信号。这就告诉我们调制解调器正在活动，可以发送另一个命令，如发送AT D T拨叫电话号码。Start-chat命令可使用户在任何一个未使用的异步线路上启动对话脚本。命令的格式如下：

Router(config)# start-chat chat-script-name line-typecon line-number

chat-script-name	WORD	Name of the chat script
	line-typecon	Apply to the console line
	aux	Apply to the auxiliary line
	tty	Apply to the asynchronous line
	vty	Apply to the telnet line
line-number	x	Apply to this absolute line number

debug modem

有许多命令可以帮助我们检验调制解调器的配置并解决其故障。如show line命令可显示Cisco 接入服务器的基本线路。用户可以用debug confmodem命令来调试调制解调器的配置过程。这对于确定初始化时调制解调器的故障类型、或者它究竟是否进行了初始化是非常有用的。

在断开连接时debug confmodem命令的输出

```
06:18:40: %LINK-5-CHANGED: Interface Async65, changed state to
reset
06:18:45: %LINK-3-UPDOWN: Interface Async65, changed state to down
06:18:46: TTY65: detection speed (115200) response —OK—
06:18:46: TTY65: Modem command: -AT&F-06:
18:46: TTY65: Modem configuration succeeded
06:18:46: TTY65: Detected modem speed 115200
06:18:46: TTY65: Done with modem configuration
```

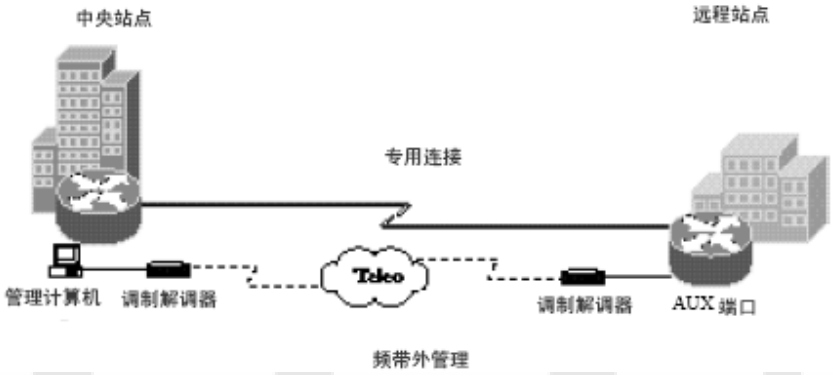
解决故障的另外的命令之一就是debug modem命令该命令可让用户看到物理层建立过程中的发展进程。

当呼叫Cisco 接入服务器时debug modem命令的输出

```
Router#debug modem

Modem control/process activation debugging is on
06:23:31: Modem 2/4 Mica: configured for Answer mode, with Null
signaling, 0x0 tone detection.
06:23:31: Modem 2/4 Mica: in modem state CALL_SETUP
06:23:32: Modem 2/4 Mica: in modem state CONNECT
06:23:36: Modem 2/4 Mica: in modem state LINK
06:23:45: Modem 2/4 Mica: in modem state TRAINUP
06:23:47: Modem 2/4 Mica: in modem state EC_NEGOTIATING
06:23:48: Modem 2/4 Mica: in modem state STEADY
06:23:48: Modem 2/4 Mica: CONNECT at 33600/31200 (Tx/Rx), V34+,
LAPM, V42bis
06:23:48: TTY69: DSR came up
06:23:48: tty69: Modem: IDLE->READY
06:23:48: TTY69: EXEC creation
06:23:48: TTY69: set timer type 10, 30 seconds
06:23:50: TTY69: Autoselect(2) sample 7E
06:23:50: TTY69: Autoselect(2) sample 7EFF
06:23:50: TTY69: Autoselect(2) sample 7EFF7D
06:23:50: TTY69: Autoselect(2) sample 7EFF7D23
06:23:50: TTY69 Autoselect cmd: ppp negotiate
06:23:50: TTY69: EXEC creation
06:23:50: TTY69: create timer type 1, 600 seconds
06:23:50: TTY69: destroy timer type 1 (OK)
06:23:50: TTY69: destroy timer type 0
06:23:52: %LINK-3-UPDOWN: Interface Async69, changed state to up
```

用户可以看到建立呼叫时MICA调制解调器2/4通过的各个阶段。它采用LAPM进行压缩，用V. 42bis进行错误校正，协议以33600bps的速度下载并以31200bps的速度上载。

Case



配置路由器A

1. 连接设备

1）应用连到2 5引脚公头电缆的R J 4 5头，将一个US Robotics调制解调器连接到路由器A的AUX端口。

2）将标准的POTS线路插到连在路由器A上的调制解调器上。

3）用连到25引脚公头电缆的9引脚母头，将调制解调器与管理计算机连接。

4）US Robotics调制解调器连在了管理计算机上，将剩余的POTS线路插入US Robotics调制解调器。

2. 配置AUX端口

5）用相对编址配置AUX端口。

```
RouterA (config) # line aux 0
```

6）在AUX端口上启用口令校验。

```
RouterA (config - line) # login
RouterA (config - line) #password cisco
```

7）允许入站和出站会话。

```
RouterA (config-ce) #modem inout
```

8）允许任何传输协议连接到该端口。

```
RouterA (config - line) # transport input all
```

9）设置AUX端口的线路速度。

```
RouterA ( config- line) #speed 38400
```

10）定义使用停止位的个数。

```
RouterA (config-line) # stopbits 1
```

11）定义流控制方法。

```
RouterA (config-line) #flowcontrol hardware
```

3. 配置回送接口

12）创建回送接口。

```
RouterA (config) #interface loopback 1
```

13）分配用于反向telnet的IP地址。

```
RouterA (config-if) #ip address 192.168.1.1 255.255.255.0
```

4. 配置调制解调器

14）为反向远程登录到调制解调器创建别名。

```
RouterA (config) #ip host auxmodem 2001 192.168.1.1
```

15）反向登录到调制解调器。

```
RouterA # telnet auxmodem
```

16）输入第9步中的A U X调制解调器口令。

Password: cisco

17）将调制解调器设置回为出厂设定默认值。

AT&F

18）将调制解调器设置为正常的C D操作。

AT&C1

19）将调制解调器设置为正常的D T R操作。

AT& D2

20）将调制解调器设置为使用硬件流控制。

AT & H1

21）将调制解调器设置为只在RT S上接收数据。

AT & R2

22）确定串行端口数据速率。

AT & B1

23） 禁用数据压缩。

AT & K0

24） 将调制解调器设置为在第1次振铃时应答。

ATS00 = 001

25）断开反向远程登录会话。

RouterA # disconnect 1

任务2—拨入路由器A上的AUX端口

1）打开管理计算机上的HyperTerminal。

2）命名您准备创建的连接（如“调制解调器”）。

3）选择调制解调器连接的COM端口。

4）将调制解调器设置为出厂默认值（如果不这样做，会导致以外的情况发生）。

AT & F

5）拨打连接在路由器A上的调制解调器的电话号码。

ATDT 54742334

6）输入任务1的第9步中的AUX调制解调器口令。

Password: cisco

路由器A配置文件

```
!
hostname RouterA
!
enable secret cisco
!
ip subnet-zero
ip host auxmodem 2001 192.168.1.1
!
interface Loopback1
ip address 192.168.1.1 255.255.255.0
no ip directed-broadcast
!
interface Ethernet0
no ip address
no ip directed-broadcast
shutdown
!
interface Serial0
no ip address
no ip directed-broadcast
shutdown
!
no ip address
no ip directed-broadcast
shutdown
!
ip classless
!
line con 0
transport input none
```

```
line aux 0
password cisco
login
modem Inout
transport input all
stopbits 1
flowcontrol hardware
line vty 0 4
login
!
end
```

PPP and CHAP

Basic Introduction

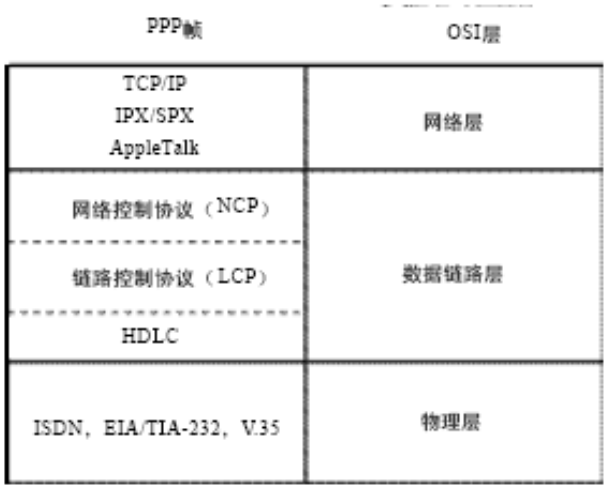
点到点协议（ Point to Point Protocol，简写为PPP）是在点到点链路上传输数据报的另一种方法。PPP在很多方面都优于SLIP，其中最重要的一点是它的可扩展性。PPP由以下3个组件组成：

- 封装方法（HDLC）。
- 链路控制协议（Link Control Protocol，LCP）。
- 网络控制协议（Network Control Protocol，NCP）。

PPP是一种面向位的协议，可在同步或异步链路上运行。PPP使用高级数据链路控制(High-level Data Link Control，HDLC）的变体作为封装的基础。这种封装在同一条链路上同时为多个网络层协议提供多路复用技术（multiplexing）。

链路控制协议（Link Control Protocol，LCP）赋予PPP以多功能性，考虑到包格式、包大小和认证的协商。它还使PPP具有确定何时线路为失败、何时正常运行的功能。

网络控制协议（Network Control Protocol，NCP）实际上是一套协议。每个子协议都是为处理各自网络层协议所需的错综复杂的配置而设计的。



HDLC

HDLC规程主要由HDLC帧结构、规程要素、规程类别三个部分组成[1]。 为了能够实现HDLC的基本功能并能按照各项标准的规定灵活采用不同的CRC校验算法，我们必须了解HDLC基本的帧结构形式。

HDLC是面向比特的链路控制规程，其链路监控功能通过一定的比特组合所表示的命令和响应来实现，这些监控比特和信息比特一起以帧的形式传送。

以下是ISO/IEC 3309标准规定的HDLC的基本帧结构。

起始标志	地址数据	控制数据	信息数据	帧校验序列	结束标志
01111110	8bits	8bits	8bits	16或32bits	01111110

其它的HDLC标准也有类似的HDLC帧结构。每帧的起始和结束以“7E”（01111110）做标志，两个“7E”之间为数据段（含地址数据、控制数据、信息数据）和帧校验序列。帧校验采用CRC算法，对除了插入的“零”以外的所有数据进行校验。为了避免将数据中的“7E”误为标志，在发送端和接收端要相应地对数据流和帧校验序列进行“插零”及“删零”

操作。

各种HDLC协议间的区别之一是帧校验序列的CRC算法不同，这种不同表现在几个方面：

a、 HDLC帧校验序列的位数不同，如16位和32位等；

b、 CRC生成多项式不同，如对于16位的CRC，CCITT V .41标准的多项式是x16+x12+x5+1，ANSI CRC-16标准的多项式是x16 +x15+x2+1等；

c、 CRC序列的起始化条件不同，如可以初始化为全“0”、全“1”等；

d、 CRC计算结果的处理方式不同，如可以直接把CRC结果发送，或对CRC结果取反后再发送等；

e、 对接收到的数据做CRC校验时，合格判据不同，因为有了上述的不同处理自然会得到不同的结果，由此造成合格判据不同。

Configuration

可以在同步接口，异步接口，ISDN BRI和PRI上用encapsulation ppp封装PPP

```
RouterA (config) # interface serial 0
RouterA (config) # encapsulation ppp
```

LCP

在TCP/IP和IPX/SP X等网络层协议可以被路由之前， LCP必须打开一个连接并协商配置。

LCP有4个主要的组件，如下：

- 认证。
- 回拨。
- 压缩。
- 多链路PPP。

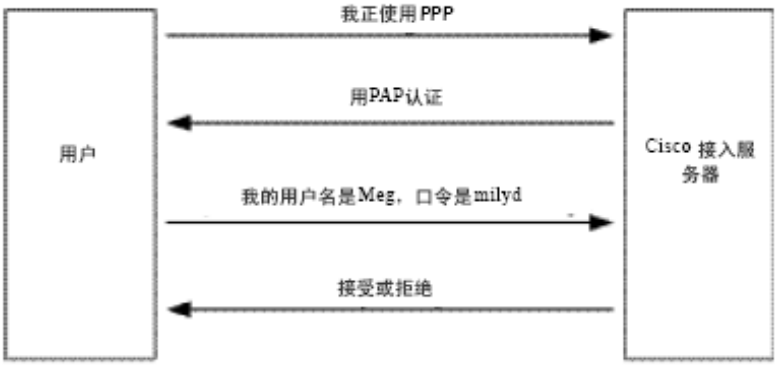
1. PPP认证

PPP认证是客户端连接好并成功与一调制方法进行协商后、在LCP协议组中被协商的第1个协议。如果认证失败，Cisco 接入服务器就会立即中断连接。

现有两种常用的认证协议：口令认证协议（ Password Authentication Protocol，简写为PAP）和质询握手认证协议（ Challenge Handshake Authentication，简写为CHAP）。注意虽然认证不是使用PPP时所必需的，但它的确是构成任何安全策略都需要的部分。

（1）口令认证协议

口令认证协议（Password Authentication Protocol，PAP）是最简单的认证协议。一旦客户端连接到Cisco 接入服务器， PPP就可以进行认证。在这段时间内，用户名/口令反复地被发送到Cisco 接入服务器，直到认证被接受或拒绝。认证拒绝通常是以连接中断的形式出现的。当用户名/口令用PAP发送时，没有任何加密措施来隐藏用户名或口令



由于发送的数据只是用户名和口令，所以很容易遭受“重演攻击”。所谓的重演攻击（playback attack）就是入

侵者在认证过程发生时“收听”线路，截取并记录下认证过程。随后，入侵者就能够使用记录下的认证过程来访问用户系统。虽然这也不甚容易，但的确可以做到。

与PPP封装相似，Cisco 接入服务器的配置是在接口上进行的。命令的格式如下：

Router(config-if)#ppp authentication pap

（2）质询握手认证协议

要提高安全性，用户可以使用一种更为安全的认证方法：质询握手认证协议（ Challenge Handshake Authentication，简写为CHAP）。在CHAP认证过程中，Cisco 接入服务器向连接主机发送一条质询，质询是以随机数的形式出现的。这个数字称为加密密钥（ encryption key），由Cisco 接入服务器产生，且必须是唯一的、不可预测的。

然后主机用单向散列函数—通常用M D 5给用户名和口令加密。主机发送的这一新加密的消息称为响应。

Cisco 接入服务器接收到响应后，将用这一连接的密钥对用户名和口令进行解密，并将其与数据库中的有效用户名和口令进行比较。这些安全特点使得CHAP比PAP更安全。因为在质询中发送的随机数是唯一的且不可预测的，CHAP防止了重演攻击。由于主机用加密密钥来对用户名和口令进行加密，所以口令和用户名都不是以明确的文本形式在线路上发送的。

CHAP还通过重复地每两分钟质询主机一次来提供额外的保护。这种重复的质询限制了信息暴露在任何单独的攻击之下。

Cisco 接入服务器的配置基本上与PAP的配置相同。命令格式如下：

Router(config-if)#ppp authentication chap

注意当主机试图进行认证时，决定认证协议的是接收端的路由器。管理员指定是用PAP认证协议还是CHAP认证协议，或者二者均采用，而且入局主机（incoming host）必须支持这个认证协议。

配置PAP：

在点到点两端接口分别封装PPP，然后用**ppp authentication pap callin**启用PAP身份验证，

使远程设备进入时进行验证参数**callin**指当我主动拨出去的时候不验证对方，当对方拨进来的时候验证对方（双向验证中有效）在全局用**username kaka password nichole**指定一个本地用户名/密码数据库，该用户名和密码要和远端路由器接口上指定的**ppp pap sent-username**中用户名和密码一致（用户名区分大小写）；或者指定一个用户名/密码服务器.如果是拨号接口，还要用**dialer-map ip 1.1.1.1 name kaka 1985223**指定被拨的远程主机名和号码双向验证的话，两端都要配，有一边验证不成功，就连接不成功

配置CHAP ：

在点到点两端接口分别封装PPP，然后用**ppp authentication chap**启用CHAP身份验证

在全局用**username kaka password nichole**指定一个本地用户名/密码数据库，该用户名和密码要和远端路由器接口上指定的**ppp chap hostname kaka**和**ppp chap password nichole**中用户名和密码一致，或和默认用户名相同（用户名区分大小写）；或者指定一个用户名/密码服务器，如果是拨号接口，还要用**dialer-map ip 1.1.1.1 name**

kaka 1985223指定被拨的远程主机名和号码。可以用ppp authentication pap chap或ppp authentication chap pap同时启用PAP和CHAP，谁在前面谁优先，用于对方的验证方式不确定

2. 创建用户账号

既然读者已经基本理解了PAP和CHAP认证，现在就需要学习如何创建用户账号以与认证请求相比较。

Cisco 路由器具有创建本地用户账号数据库的能力。在本地Cisco 接入服务器上创建用户名是一个非常直接的过程。在全局配置模式用username命令，语法如下：

Router(config)#username user-account-name encryption-type password

. 第1种加密类型为0，它指出，准备输入的口令是没有被加密的，这就允许在创建用户账号时输入纯文本的口令。第2种加密类型为7，指出Cisco路由器已经对准备输入的口令进行了加密。如果用户使用加密类型0创建用户账号，大概不希望能访问配置文件的人知道每个人的口令。要在配置文件中对用户口令进行加密，就必须使用service password-encryption命令：

Router(config)#service password-encryption

一旦用户启用了口令加密，那么配置文件中的所有口令就都加密了。一旦用户的口令被加密，那么即使使口令加密无效也不能对用户口令进行解密，口令仍然保持为加密状态。

3. PPP 回拨

PPP回拨是在LCP内的扩展，客户端可以请求Cisco 接入服务器回拨客户端。为了使PPP回拨可以正确地进行协商，就必须要对客户端和Cisco 接入服务器进行配置，让二者均支持回拨技术。用户可以用回拨来控制访问成本，或将回拨作为安全策略的一部分内容。通常，回拨并不用作安全特征。但是，用户可以在用户名数据库中将用户与预先确定的电话号码相关联，强迫Cisco 接入服务器只承认回拨请求，抛弃所有不请求回拨的呼叫，从而加强安全。这就要求客户端在试图访问Cisco 接入服务器时，必须位于一预先确定的位置，从而加强了安全性。

只要客户端成功地进行了认证，就可以请求回拨。假定PPP回拨被启用， Cisco 接入服务器会立即中断呼叫，并回拨原来的客户端。让Cisco 接入服务器结束连接，就是要向客户端说明Cisco 接入服务器准备回拨客户端。在这种方法的作用下，即使是Cisco 接入服务器中没有启用回拨，也允许客户端保持连接状态。

Router(config-if)#ppp callback callback-option

callback-option	accept	Accept a callback request
	initiate	Initiate callback without ppp callback negotiation
	request	Request a callback

accept选项允许Cisco 接入服务器在特定的接口中从客户端接受回拨请求。这就规定，无论何时客户端拨入并认证，只要请求回拨， Cisco 接入服务器就要启动回拨进程。用户可能回遇到有的PPP客户端不完全符合RFC1570，但是仍然想应用回拨选项。在这种情况下，客户端必须要将自身置于应答模式（answer mode），但不能在LCP协商过程中指明要使用回拨。initiate命令并不遵从RFC1570，但允许用户回拨客户端。

Router(config-if)#ppp callback accept

注意只有在那些进行了配置以便响应回拨的协议（PPP、ARAP和SLIP）下，回拨才能正常工作。要启用回拨，用户必须要在异步线路上应用autoselect ppp命令以支持PPP回拨。通常，当需要呼叫远程路由器的路由器指定回拨选项时，要用到reques t选项.如前所述，用户可以将回拨用作一种安全方法。安全性是通过在用户拨入时指定呼叫的电话号码而获得的。如果用户使用本地用户名数据库（即对接入服务器也是本地的），可以用下面的命令配置回拨：

Router(config)#username username callback-attributes

callback-attributes	call-back-dialstring	string	Dial string to use forthis user
	call-back-line	line	Line to callback on (aux, con, tty, vty)
	callback-rotary	number	ogical rotary groupnumber

4. PPP压缩技术

对于速度慢的链路而言，压缩是一种非常有用的工具。它可以加速传输速率，使得在外表上看来似乎是有了更多的带宽。因为压缩在LCP内是协商的选项，所以客户端和Cisco 接入服务器都必须支持它，否则就被禁用。

在Cisco 接入服务器上，可以配置下面的4种压缩：

- 预测压缩。
- 堆栈压缩。
- Microsoft 点到点压缩。
- TCP头压缩。

要配置MPPC、Predicator和Stacker压缩，请使用下面的命令：

Router(config-if)#compress compression-type compression-option

compression-type	mppc	Use MPPC compression
	predictor	Use Predictor compression
	stac	Use Stacker compression
compression-option	ignore-pfc	Ignore negotiated compression and use MPPC

要配置TCP头压缩，请使用下面的命令：

Router(config-if)#ip tcp header-compression compression-option

compression-option	passive	Passive only compress headers for clients that send compressed headers (optional)

虽然以上各种压缩方法都是为了相似的任务，但却对Cisco 接入服务器有不同的影响。例如，Predicator倾向于加强内存，而Stacker和MPPC倾向于增强CPU。

5. 多链路PPP

多链路PPP（Multilink PPP）允许路由器在多个接口上将包分割或重新合并。多链路PPP是一种在呼叫连接过程中进行协商的LCP选项。在LCP协商过程中，客户端指示它可以将多个物理链路合并为一个“束”，之后这个束的运作就象是一个单独的WAN链路在运作.

多链路PPP提出了下面这些与多个WA N链路上负载的平衡相关的问题：

- 多供应商之间协同工作的能力。
- 包划分。
- 包顺序和负载平衡。

要配置多链路PPP，请使用下面的命令：

Router (config-if) #ppp multilink

NCP

网络控制协议（Network Control Protocol，简写为NCP）是一些单独定义了的协议的集合，负责在呼叫建立时协商网络层属性，然后在PPP包中将网络层协议封装以便于传输。要启动NCP协议，必须先要成功地完成LCP协商。NCP了建立和断开单条数据链路上的多个三层协议会话 - 多路复用

Configuration PPP

Configurating TCP/IP connections

1. 配置接入地址池

Router(config)#ip local pool address-pool-name first-ip-address last-ip-address

address-pool-name	default	Make this IP address pool the default
	WORD	Name for this IP address pool block
first-ip-address	a.b.c.d	First IP address in the block of IP addresses

last-ip-address	a.b.c.d	Last IP address in the block of IP addresses

只有TCP/IP地址而没有其他属性使得应用TCP/IP非常困难，这还是最好的情况。因此，用户需要给出拨号客户端的TCP/IP信息。用下面的命令，可以分发大量的信息：

Router(config)#async-bootp ip-attribute a.b.c.d

ip-attribute	dns-server	a.b.c.d	Set DNS server for remote client to use
	lpr-server	a.b.c.d	Set LPR server for remote client to use
	nbns-server	a.b.c.d	Set WINS server for remote client to use
	time-server	a.b.c.d	Set time server for remote client to use

使用ip unnumbered interface将一个LAN接口的Ip地址作为WAN信息包流的IP地址，节省地址空间

PPP Debug

Cisco 在IOS中提供了许多调试PPP连接的工具，使得PPP连接的排错工作非常容易。debugPPP命令中有很多解决PPP连接故障时常要用到的选项。

解决PPP连接故障的命令如下：

Router#debug ppp ppp-option

ppp-option	authentication	Debug the PPP authentication process
	multilink	Debug Multilink activity
	negotiation	Debug the PPP negotiation process

当用户试图读取排错的输出时，需要解释6条不同的消息，这6条消息又可以分解为以下两个组：

- 什么设备发出了什么信息。
- 有何响应。

有两个不同的值来识别设备发出了什么信息。字母O识别从Cisco 接入服务器发出的命令，而字母I识别发送到Cisco接入服务器的通信。若试图识别从Cisco 接入服务器发出的响应，用户可以利用下表来解释4种可能的命令。

CONFREQ	配置请求
CONFREJ	配置拒绝
CONFACK	配置承认
CONFNACK	配置未承认

综合利用这些命令，可以帮助用户查出任何PPP连接内发生的故障。

debug ppp authentication命令的输出

```
RouterA#debug ppp authentication
PPP authentication debugging is on
%LINK-3-UPDOWN: Interface Async1, changed state to up
As1 PPP: Treating connection as a dedicated line
As1 PPP: Phase is ESTABLISHING, Active Open
As1 LCP: O CONFREQ [Closed] id 30 len 25
As1 LCP: ACCM 0x000A0000 (0x0206000A0000)
As1 LCP: AuthProto CHAP (0x0305C22305)
As1 LCP: MagicNumber 0xE093FFC9 (0x0506E093FFC9)
As1 LCP: PFC (0x0702)
As1 LCP: ACFC (0x0802)
As1 LCP: I CONFACK [REQsent] id 30 len 25
As1 LCP: ACCM 0x000A0000 (0x0206000A0000)
As1 LCP: AuthProto CHAP (0x0305C22305)
As1 LCP: MagicNumber 0xE093FFC9 (0x0506E093FFC9)
As1 LCP: PFC (0x0702)
As1 LCP: ACFC (0x0802)
As1 LCP: I CONFREQ [ACKrcvd] id 2 len 36
As1 LCP: ACCM 0x00000000 (0x020600000000)
```

```
As1 LCP: MagicNumber 0x046D5ADC (0x0506046D5ADC)
As1 LCP: PFC (0x0702)
As1 LCP: ACFC (0x0802)
As1 LCP: Callback 6 (0x0D0306)
As1 LCP: MRRU 1614 (0x1104064E)
As1 LCP: EndpointDisc 3 26ff.2052.4153 (0x13090326FF20524153)
As1 LCP: O CONFREJ [ACKrcvd] id 2 len 20
As1 LCP: Callback 6 (0x0D0306)
As1 LCP: MRRU 1614 (0x1104064E)
As1 LCP: EndpointDisc 3 26ff.2052.4153 (0x13090326FF20524153)
As1 LCP: I CONFREQ [ACKrcvd] id 3 len 20
As1 LCP: ACCM 0x00000000 (0x020600000000)
As1 LCP: MagicNumber 0x046D5ADC (0x0506046D5ADC)
As1 LCP: PFC (0x0702)
As1 LCP: ACFC (0x0802)
As1 LCP: O CONFACK [ACKrcvd] id 3 len 20
As1 LCP: ACCM 0x00000000 (0x020600000000)
As1 LCP: MagicNumber 0x046D5ADC (0x0506046D5ADC)
As1 LCP: PFC (0x0702)
As1 LCP: ACFC (0x0802)
As1 LCP: State is Open
As1 PPP: Phase is AUTHENTICATING, by this end
As1 CHAP: O CHALLENGE id 9 len 27 from "RouterA"
As1 LCP: I IDENTIFY [Open] id 5 len 24 magic 0x046D5ADC MSRAS-1-AQUIGGLE
As1 CHAP: I RESPONSE id 9 len 24 from "meg"
As1 CHAP: Unable to validate Response. Username meg: Authentication failure
As1 CHAP: O FAILURE id 9 len 19 msg is "% Access denied"
As1 PPP: Phase is TERMINATING
As1 LCP: O TERMREQ [Open] id 31 len 4
As1 LCP: I TERMACK [TERMsent] id 31 len 4
As1 LCP: State is Closed
As1 PPP: Phase is DOWN
As1 PPP: Phase is ESTABLISHING, Passive Open
As1 LCP: State is Listen
LINK-5-CHANGED: Interface Async1, changed state to reset
As1 LCP: State is Closed
As1 PPP: Phase is DOWN
```

客户端和服务器都在协商使用CHAP认证。User Meg的出现说明认证过程失败了，大概是因为口令错误引起的。

调试PPP连接的最有用的工具之一是debug ppp negotiations命令。

debug ppp negotiations命令的输出（只限于IP协商）

```
As1 IPCP: I CONFREQ [REQsent] id 9 len 40
As1 IPCP: CompressType VJ 15 slots CompressSlotID (0x0206002D0F01)
As1 IPCP: Address 0.0.0.0 (0x030600000000)
As1 IPCP: PrimaryDNS 0.0.0.0 (0x810600000000)
As1 IPCP: PrimaryWINS 0.0.0.0 (0x820600000000)
As1 IPCP: SecondaryDNS 0.0.0.0 (0x830600000000)
As1 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
As1 IPCP: O CONFREJ [REQsent] id 9 len 22
As1 IPCP: CompressType VJ 15 slots CompressSlotID (0x0206002D0F01)
As1 IPCP: PrimaryWINS 0.0.0.0 (0x820600000000)
As1 IPCP: SecondaryWINS 0.0.0.0 (0x840600000000)
As1 IPCP: I CONFACK [REQsent] id 2 len 10
As1 IPCP: Address 192.168.101.1 (0x0306C0A86501)
As1 IPCP: I CONFREQ [ACKrcvd] id 12 len 22
As1 IPCP: Address 0.0.0.0 (0x030600000000)
As1 IPCP: PrimaryDNS 0.0.0.0 (0x810600000000)
As1 IPCP: SecondaryDNS 0.0.0.0 (0x830600000000)
As1 IPCP: O CONFNAK [ACKrcvd] id 12 len 22
```

```
As1 IPCP: Address 192.168.100.26 (0x0306C0A8641A)
As1 IPCP: PrimaryDNS 192.168.1.11 (0x8106C0A86F6F)
As1 IPCP: SecondaryDNS 192.168.1.12 (0x8306C0A86F70)
As1 IPCP: I CONFREQ [ACKrcvd] id 13 len 22
As1 IPCP: Address 192.168.100.26 (0x0306C0A8641A)
As1 IPCP: PrimaryDNS 192.168.1.11 (0x8106C0A86F6F)
As1 IPCP: SecondaryDNS 192.168.1.12 (0x8306C0A86F70)
As1 IPCP: O CONFACK [ACKrcvd] id 13 len 22
As1 IPCP: Address 192.168.100.26 (0x0306C0A8641A)
As1 IPCP: PrimaryDNS 192.168.1.11 (0x8106C0A86F6F)
As1 IPCP: SecondaryDNS 192.168.1.12 (0x8306C0A86F70)
As1 IPCP: State is Open
As1 IPCP: Install route to 192.168.100.26
```

ISDN and DDR

Basic Introduction

ISDN与其他WAN服务相比，有以下优越性：

- 与模拟呼叫相比，建立呼叫的速度特别快。
- 比租用线路便宜。
- 比模拟线路的传输速度快。

ISDN有几种不同的变体：基本速率接口（ Basic Rate Interface，BRI）、美国的主速率接口（U.S. Primary Rate Interface，T1 PRI）和欧洲的主速率接口（ European Primary Rate Interface，E1 PRI）。虽然它们基于同样的技术，但其不同之处在于其信道的数目。

ISDN形式	可以使用的B信道	B信道的传输速率（Kbps）	D信道的数目	D信道的传输速率（Kbps）
BRI	2	64	1	16
T1 PRI	23	64	1	64
E1 PRI	30	64	1	64

信道化T1/E1

信道化T1多路复用一条线路来逻辑地在同一条物理介质上创建多条信道。虽然这看似有多条物理线路，实际上只有一条。创建逻辑信道的方法有很多，时分多路复用（ Time Division Multiplexing，TDM）和频分多路复用（ Frequency Division Multiplexing，FDM）是其中两个例子。

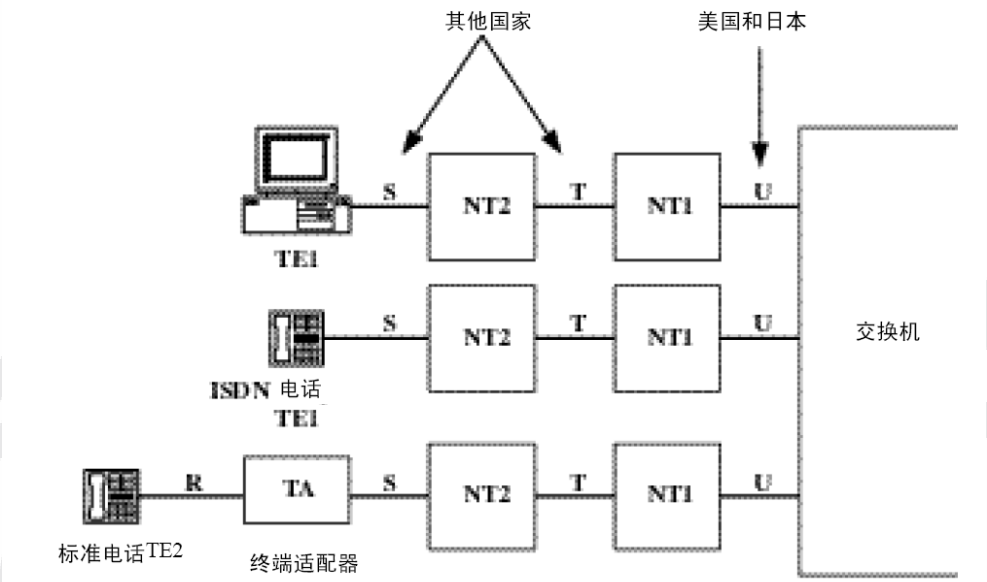
TDM是通过将信道划分为时间片来工作的。这种多路复用技术对于数字信号而言工作得很好。ISDN使用了一种称为脉冲码调制（ Pulse Code Modulation，PCM）的方法来代表一个线路上的多个信道的。PCM由ITU-T I.431定义，在物理线路上创建时隙，每个时隙用来为其信道代表数据。

D信道

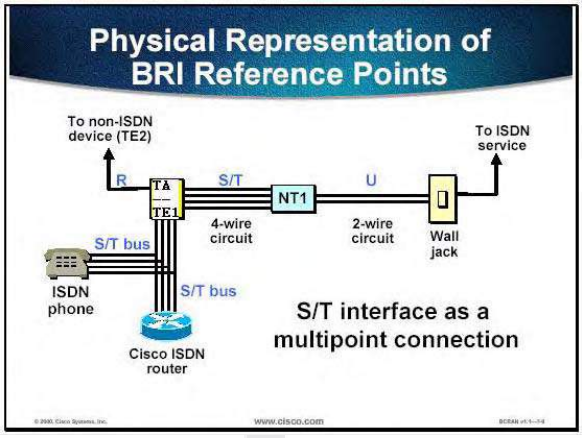
D信道的主要目的在于为ISDN线路提供信令与控制。有几个协议是用来建立或中断ISDN线路上的呼叫的。

参考点

参考点用来定义不同设备之间的信令标准。下图给出了4种参考点并说明了它们是如何彼此关联的。



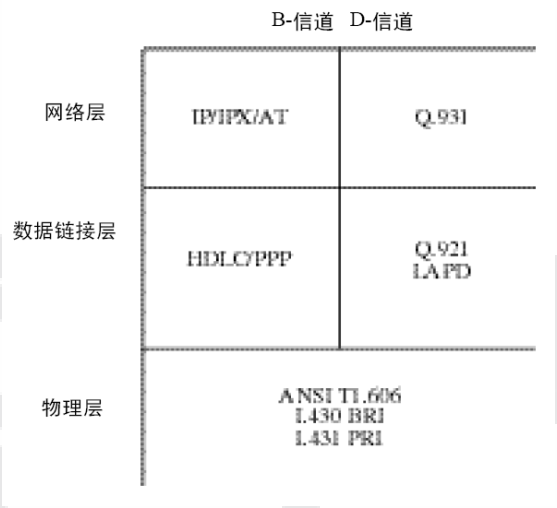
日本和美国的电话公司提供了到U参考点的接口，而欧洲的电话公司提供了到S/T参考点的连接。S/T参考点是具有4条线路的无源总线网络，这种结构可使得多台设备连接到此参考点上，这与将多部电话连接到标准的POTS线路是相同的。而U参考点只允许有一台设备与其相连接。美国生产的大多数Cisco ISDN路由器都有一个NT1接口，并且该接口可以连接到U参考点。这样用户就无需NT1设备了，但是S/T接口不能连接额外的与ISDN兼容的设备。美国的业界已经定义了协议ANSI T1.601来规定U接口上的信令特点。



ISDN protocols

ITU-T将ISDN协议组织为3类：Q系列、E系列和I系列。

1. Q系列协议　规定了设备间交换和信令协议，示例包括：
   - Q.921 ISDN 用户-网络接口-数据链路层规范。
   - Q.931 用于基本呼叫控制的ISDN用户-网络接口网络层规范。
2. I系列协议　用于定义与ISDN有关的概念和接口。示例包括：
   - I. 430 基本用户-网络接口-物理层规范。
   - I. 431 主速率用户-网络接口-物理层规范。
3. E系列协议　用于定义电话网络标准。示例包括：
   - E.164 对ISDN时代的规划进行编号。
   - E.172 ISDN时代的呼叫路由。



数据链路层

ISDN D信道的数据链路层是由Q.921定义的，也就是众所周知的D信道链路访问协议（Link Access Protocol，D-channel，LAPD）。LAPD为点到点或点到多点的链路提供了串行、同步和全双工的通信。

以下是LAPD帧的主要组成部分

- Flag 所有的帧都以相同的二进制序列开始和结束： 0 111111 0（7Eh）。
- Address 该字段可识别发送或接收帧的ISDN设备和协议
- Control 该字段用来识别帧的类型。它可以携带序列和确认信息
- Data 该字段包含着消息数据，且长度可变，但必须是8位排列（octet aligned，指的是编码规则，简单的数

  据流适合于在通信信道上进行传输中，此编码规则就是要在简单的数据流中代表复杂的数据结构）。
- Frame Check Sequence（帧校验序列） 这一两字节的字段包含着帧校验和的计算。

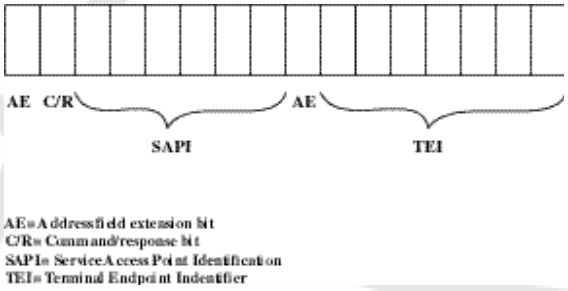
LAPD编址

LAPD编址具有在同一物理信道上多路复用几个逻辑链接的能力。地址的长度在9位到13位之间（3位用于开销），并由2个字段组成：终端终点标识符（ Terminal Endpoint Identifier，TEI）和服务访问点标识符（ Service Access Point Identifier，SAPI），这两个字段允许多个终端设备（terminal equipment，TE）可以彼此区分。

TE设备具有内置的本地ISDN接口。

SAPI值	第3层实体
0	呼叫控制过程
1	使用I.451的包通信
16	使用X.25的包通信
63	第2层管理功能

TEI是一个7位的字段，在Address字段中位于第2个8位的位置上。因此，用户从理论上可以在一个单独的接口上拥有多达127个TEI—虽然ITU-T对TE2的数量进行了限制，规定每BRI有8个TE2。注意TE2不具备本地的ISDN



支持。

网络层消息

ISDN网络层消息是在LAPD I帧的Information字段中携带的。从理论上说，任何第3层协议都可以用在D信道上，但我们只推荐使用其中的两个： Q.931，用于用户-网络信令；X.25 分组层协议（Packet Layer Protocol，简写为PLP），用于X.25包的交换。

ISDN网络层消息的格式包括以下组成部分：

- Protocol Discriminator（协议鉴别器） 8位
- Call Reference Value（呼叫参考值） 1 5位
- Message Type（消息类型） 7位
- Other Information Elements（其他信息成分） 可变长度Protocol Discriminator （协议鉴别器）字段用来指定对消息进行编码的协议（ X.25或Q.931）。

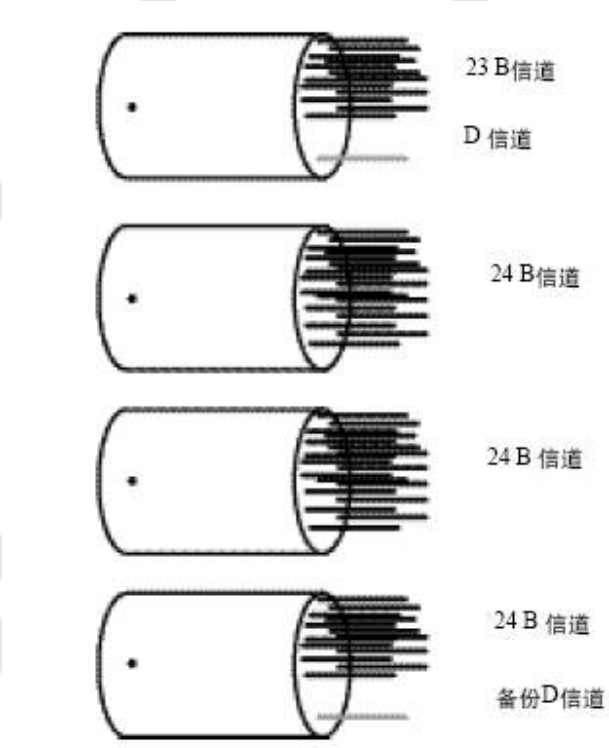
Call Reference Value（呼叫参考值）字段用于用户和网络之间，用来识别激活的呼叫。在呼叫开始时此字段就被分配，并在呼叫终止之前一直保持在原位置上。Call Reference Value在本地的具有显著的意义。

Message Type（消息类型）字段用来识别网络层信令消息。在Q.931（25）和Q.932（8）之间定义了33条消息。

消息名称	功 能
	呼叫建立
Alerting	振铃线
Call Proceeding	所有接收到的必要信息，继续进行呼叫
Connect	呼叫建立
Connect Acknowledge	确认呼叫建立
Progress	呼叫建立过程中的中间信号
Setup	带有服务要求的呼叫请求
Setup Acknowledge	接收到的Setup，需要更多信息
	呼叫信息的各个阶段
Hold	请求保持呼叫
Hold Acknowledge	同意Hold请求
Hole Reject	拒绝Hold请求
Resume	请求恢复原来挂起的呼叫
Resume Acknowledge	同意Resume呼叫
Resume Reject	拒绝Resume呼叫
Retrieve	请求检索正在保持着的呼叫
Retrieve Acknowledge	同意Retrieve请求
Retrieve Reject	拒绝Retrieve请求
Suspend	请求挂起呼叫
Suspend Acknowledge	同意Suspend请求
Suspend Reject	拒绝Suspend请求
User Information	在信令路径上的用户-用户信息
	呼叫清除
Disconnect	挂断呼叫
Release	放弃呼叫
Release Complete	同意放弃呼叫
Restart	重新启动第3层协议
Restart Acknowledge	同意重新启动

消息名称	功能
	其他
Congestion Control	用户信息的流控制
Facility	请求任选用户服务
Information	在呼叫建立过程中提供其他信息
Notify	分配呼叫参考值
Register	注册呼叫参考值
Status	说明信道状态
Status Enquiry	在congestion control后请求信道状态

使用ISDN PRI线路上的D信道可以为另一条ISDN PRI线路提供信令。



一单独的D信道控制多条ISDN PRI线路的示例，其中还有一条备份的D信道

这样，用户就得到一额外的时隙，可以转换为B信道—这种技术称为NFAS（Non-FacilityAssocited Signaling）。只有当具有3条以上的PRI时，才能真正使用NFAS。如果没有备份D信道，并且带有D信道的PRI失效的话，那么整套线路就崩溃了。

Config ISDN PRI

虽然ISDN非常复杂，但是只要理解了概念，配置ISDN PRI并不困难。要配置ISDN PRI线路来接受传入的模拟呼叫和ISDN呼叫，用户需要注意以下问题：

- 控制器类型（T 1 / E 1）。
- 组帧和线路编码。
- 时钟控制。

- 交换机类型。
- 时隙识别。
- RBS 。
- D信道串行端口的配置。
- 接受模拟呼叫。

控制器类型

有两种ISDN PRI控制器类型可以供用户使用：T1和E1。为您提供ISDN PRI服务的本地电话公司将规定您购买何种控制器。首要之处在于，T1控制器应用于北美和日本，而欧洲采用的是E1。使用下面的命令，以选择控制器的接口进行配置：

```
Router(config)#controller controller-type slot-number
Controller-type     T1   Configure T1 controller
                    E1   Configure E1 controller
slot-number         x    Controller unit number
```

组帧与线路编码

组帧和线路编码是由ISP提供的属性。如果用户采用了错误的组帧和线路编码，PRI线路就不能工作。在一正常的T1帧中，有193个位，其中1 9 2个位用于数据， 1个位用于组帧。遗憾的是，在T1中，这个单独的位不能简单自如地来传送保持T1线路同步的信令信息。因此，人们创建了超帧格式（superframe format，SF）来处理这个问题。SF，也就是D4，由12个T1帧组成， 这12个组帧位要经历下面的12位模式： 100011011100。也就是说，第1个帧中的组帧位为1，下面3个帧的组帧位为0，再下两个帧的组帧位为1，直到12位模式结束。一旦到达了最后，就再重新开始。接收器通过在1 2个帧的每1 9 3个位中寻找这种特定的模式，就可以建立帧的同步。

另一种组帧方法，称为T1扩展超帧格式（ extended superframe format，ESF）。在ESF中，2 4个帧组成一组，产生了2 4个组帧位。这些组帧位可分解为以下3种功能：

- 帧排列序列（Frame alighment sequence，FAS） 6位
- 帧校验序列（Frame check sequence，FCS） 6位
- 维护信道（ maintenance channel） 12位

FAS在其6位中使用了一种重复的模式（001011），以确保帧进行了正确的同步。如果接收器失去了同步（滑动，slip），它将在下面的5个帧中查找到相应的位的模式（24个T1帧每ESF帧*5个ESF帧=120个帧）。

FCS用了一个6位的循环冗余码校验（cyclic redundancy check，CRC）来确定在前一个帧中是否存在位错误。F C S只用于错误检测，而不能进行错误校正。

维护信道，也称为设备数据链接（Facilities Data Link，FDL），是一个4Kbps的附带信道，用于网络维护和操作。然而， PRI并不对该信道有何具体操作。

最后一种组帧方法是CRC4和NO-CRC4。这些方法只在E1时可用。CRC4和NO-CRC 4根据E1是欧洲的还是澳大利亚的，而具有不同的算法。

T1电路的要求之一就是要维持一定的密度水准，这样转发器就可以维持计时了。如果发送的长字符串都是二进制0，那么转发器的计时很容易出现延误。因此，必须考虑一种方法以确保每隔一段时间就出现一个二进制1或者标志脉冲。目前存在3种方法，通常，应用何种方法取决于采用的线路编码的类型。

组帧和线路编码配置是在控制器上执行的，不同的控制器上的组帧和线路编码是不同的。

```
Router(config-controller)#framing   frame-type
frame-type          sf      Configure T1 controller for superframe format
                    esf     Configure T1 controller for extended superframe format
```

crc4 Configure E1 controller for cyclic redundancy check 4

组帧	线路编码
超帧（SF）	信号交替反转（Alternate Mark Inversion，简写为AMI）
扩展超帧（ESF）	B8ZS（Binary 8-Zero Substitution）
循环冗余码校验4（CRC4）	高密度双极型（ High-Density Bipolar 3，简写为HDB 3）

要配置控制器上的线路编码，请应用以下命令：

```
Router(config-controller)#linecode linecoding-type
linecoding-type     ami     Configure T1 controller for alternate mark
                    b8zs    Configure T1 controller for extended superframe format
                    hdb3    Configure E1 controller for cyclic redundancy check 4
Router#config t
Router(config)#controller t1 0
Router(config-controller)#framing esf
Router(config-controller)#linecode b8zs
```

时钟控制

要向ISDN交换机发送数据，用户必须要使用时钟以便可以正确地对传输进行计时。时钟是在T1/E1控制器上定义的。

```
Router(config-controller)#clock source line clock - source
clock-source    primary     Configure T1 interface as the primary
                            TDM clock source
                secondary   Configure T1 interface as the secondary
                            TDM clock source
                internal    Will use its own internal clock
```

交换机类型

选择何种交换机乃取决于提供ISDN PRI的电话公司。安装了ISDN PRI线路后，电话公司将位用户指示使用的交换机类型。

以下是Cisco IOS可支持的7种不同的交换机类型：

- AT&T 4ESS 交换机。
- AT&T 5ESS交换机。
- Northern Telecom 交换机。
- NET5 European 交换机。
- National ISDN 交换机。
- 日本交换机。
- 澳大利亚交换机。

在IOS 11 . 3之前，用户只能在全局配置中选择ISDN交换机的类型。而从IOS 11.3起，用户可以用两种方法来选择交换机类型：在全局配置中选择，或在特定接口上选择。如果用户是在全局配置中选择交换机类型，那么只能一次设定所有的接口。而在接口上选择交换机类型，就可以让不同的接口支持不同的交换机类型。

要全局地配置交换机类型，请应用下面的命令：

```
Router(config)#isdn switch-type telco-switch-type
telco-switch-type   primary- 4 ess      AT&T 4ESS switch type for the U.S.
                    primary-5ess       AT&T 5ESS switch type for the U.S.
```

primary-dms100	Northern Telecom switch type for the U.S.
primary-net5	European switch type for NET5
primary-ni	National ISDN switch type
primary-ntt	Japan switch type
primary-ts014	Australia switch type

在接口上配置交换机类型的命令和语法与全局配置是相同的。用户不要配置T1控制器接口，而要配置ISDN线路上的D信道这样命令提示就改为：

```
Router(config-if)#isdn switch-type telco-switch-type
```

注意要使配置改变生效，必须重新加载路由器。

时隙识别

在配置ISDN PRI线路时，用户必须指定供应商的ISDN交换机分配的固定时隙的数量。下面的命令将应用于控制器的接口：

```
Router(config-controller)#pri-group timeslots-range
timeslots-range     1-24        For use with a T1
                    1-31        For use with an E1
```

D信道串行接口配置

既然用户已经定义了时隙的数量，就需要识别哪一个信道将为D信道。一旦识别出D信道的时隙，用户就需要为其配置其他的一些属性。因为所有的T 1线路都使用最后一条信道作为D信道，而E1线路使用第15条信道为D信道，所以配置D信道并不困难。

```
Router(config)#interface serial timeslot
timeslots       x:23        x is the slot of the T1 network module
                x:15        x is the slot of the E1 network module
```

在串行接口上用户可以配置的一项可选属性是： TEI协商何时进行。

```
Router(config-if)#isdn tei-negotiation when-to-negotiate
when-to-negotiate       first-call      Negotiate when first ISDN call
                                        is placed or received
                        powerup         Negotiate when powered up
```

接受模拟呼叫

由于ISDN呼叫采用D信道建立和拆卸呼叫，因此用户就必须为标准的模拟呼叫预先采取措施。标准模拟呼叫使用带内信令发誓来建立呼叫，而ISDN采用的是带外信令方式。当传入一个呼叫时， Cisco IOS就会检查D信道上的数据，看呼叫是否为语音呼叫（记住，模拟调制解调器呼叫看起来就像是标准的电话呼叫）。如果是语音呼叫（与ISDN设备进行的呼叫对立），CiscoIOS就会将该呼叫路由到内部的调制解调器中进行处理。进行此过程的命令如下：

```
Router(config-if)#isdn incoming-voice source-type
source-type         modem       Voice calls will be handled as modems
                    data 56     Voice calls will be handled as data at 56 Kbps
                    data 64     Voice calls will be handled as data at 64 Kbps
```

记住要在ISDN PRI线路的D信道上应用该命令，如果有多条D信道（因为有多条ISDN PRI线路），您就需要在所有的D信道上都应用该命令。

Config ISDN BRI

虽然ISDN BRI既可以支持拨入连接，也可以支持拨出连接，但本节重点讲述对拨入连接的支持（BRI上的调制解调器连接）。

从IOS 12.0开始，带ISDN BRI的Cisco 路由器就可以支持传入的模拟连接以及传入的ISDN BRI拨入连接了。

注意目前，只有IOS 12.03（T）和12.04（T）可以在3600系列路由器中支持ISDN BRI的56Kbps连接。

配置ISDN BRI并不像配置ISDN PRI那样困难。要使ISDN BRI线路支持传入呼叫，用户需要关注以下3个问题：

- 交换机类型。
- 服务配置文件标识符。
- 接受模拟呼叫。

交换机类型

与ISDN PRI相似，用户必须在ISDN BRI接口上配置交换机类型。交换机类型的选择取决于运行的IOS的版本。在IOS 11.3之前，只能在全局配置下选择交换机类型。从IOS 11.3开始，用户可以用两种方法来配置交换机类型：在全局配置或在接口上进行。通过在接口上配置交换机类型，用户可以支持多种交换机类型（如果有多个BRI接口）。

配置ISDN BRI接口的交换机类型的命令格式与配置ISDN PRI接口上的交换机类型的命令格式相同，然而配置ISDN BRI线路的选项却稍有差异。要全局地配置交换机类型，请使用以下命令：

```
Router(config)#isdn switch-type telco-switch-type
telco-switch-type    basic-1tr6        1TR6 switch type for Germany
                     basic -5ess       AT&T 5ESS switch type for the U.S.
                     basic -dms100     Northern DMS-100 switch type
                     basic -net3       NET3 switch type for the UK and Europe
                     basic-ni          National ISDN switch type
                     basic-qsig        QSIG switch type
                     basic-ts013       TS013 switch type for Australia
                     ntt               NTT switch type for Japan
                     vn3               VN3 and VN4 switch types for France
```

服务配置文件标识符

通常，服务配置文件标识符（ Service Profile Identifier，SPID）是由13个数字组成的数，服务供应商可以通过SPID使终端与一终端服务配置文件关联。这样，服务供应商就可以为用户设备分配服务特征（如呼叫转发、呼叫者I D、呼叫等待等等）。

并不是所有的交换机类型都需要SPID。目前，只有DMS-100和NI1要求有SPID。与帧中继的DLCI相似，SPID只在本地意义重大。

SPID的配置是在ISDN BRI接口上进行的。要配置ISDN BRI SPID，请使用以下命令：

```
Router(config- if ) # isdn B-channel-number spid-number phone-number
B-channel-number     spid 1          Specify SPID for first B channel
                     spid 2          Specify SPID for second B channel
spid-number          xxxxxxxxxx      Thirteen digit SPID assigned by service provider
phone-number         xxxxxxx         Phone number associated with the SPID
```

接受模拟呼叫

BRI上的调制解调器使得小型办公室可以支持56Kbps的拨入连接，而无须昂贵的PRI或额外的拨号线路，因此降低了每月重复的开销。要支持BRI上的调制解调器，用户需要一种方法来解释带内信令，并解调模拟信号。支持模拟呼叫的命令如下：

```
Router(config-if)#isdn incoming-voice source-type
source-type          modem           Voice calls will be handled as modems
                     data 56         Voice calls will be handled as data at 56 Kbps
                     data 64         Voice calls will be handled as data at 64 Kbps
```

用户在发出这些命令来配置封装和认证时要仔细。虽然在所有的接口上都支持封装和认证，要获得理想的效果，用户还是需要理解在何处应用这些命令。例如，当支持BRI上的调制解调器时，用户要配置BRI接口以支持物理层属性，但是还将使用一组异步接口来配置数据链路层和网络层特征。然而当支持ISDN BRI到ISDN BRI的连接时，若非用户使用拨号装置接口，就必须要配置BRI接口以支持物理层、数据链路层和网络层要素。

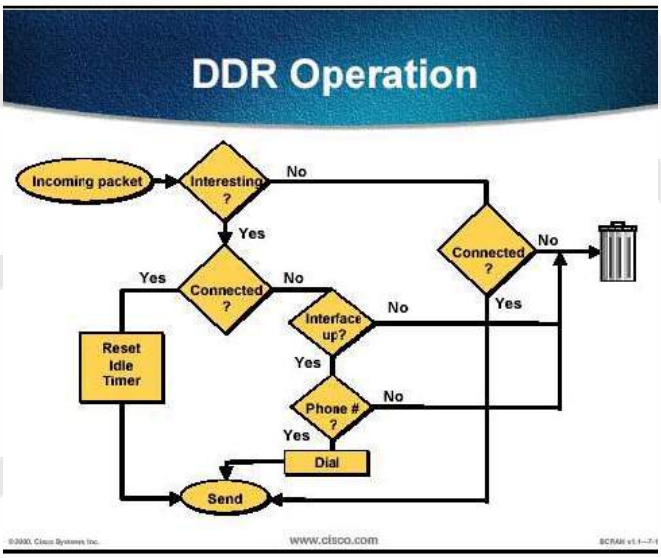
Debug ISDN

Show isdn status

Debug isdn q921

DDR

DDR的运行过程：



首先判断一个要出去的包是否为感兴趣的，如果是NO的话，则看现在链路是否已经CONNECTED。如果已经连接则发送数据，如果没有连接，则丢弃。

再看，如果这个包为感兴趣的流量，那么现在来看连接是否已经被建立，如已经被建立，则发送数据，这里要注意一点，这里有一个闲置记时器（idle-time），如果这个包为感兴趣的那么这个时候会重置这个记时器，如果不是敢兴趣的那么记时器继续记时。

再看，如果当时这个连路没有连接起来，那么这个时候接口如果是UP的那么会用一个电话号码进行拨号，然后SEND出去。若拨号失败，则丢弃数据。

这里还要注意一点就是。这个线路只要是NO SHUT了就是UP的。当然这个时候是假的UP。如果他DOWN了会造成路由表的一个错误。如果使用外置TA必须支持V.25bis拨号（串型线路上也须支持）。

在ISDN上配置DDR：

首先要定义一个感兴趣流量（dialer-list)在接口上**dialer-group** 。定义一条静态路由。知道下一跳地址用**dialer map**。否则用Dialer string 号码。

也可以通过ACL来定义一个感兴趣流量（更细致），定义好ACL之后在**dialer-list**里调用ACL

指定dialer map 语句。这里是用来拨号,可在ISDN接口、同步、异步后配置）。一般加上Boardcast。

测试DDR命令：**ISDN TEST CALL INT number 、dialer string number、isdn call int bri 0**

断开**isdn discon int type number B1|B2|all**

配置ISDN和DDR。

了解**dialer idle-timeout 30**和**dialer fast-idle 10**的区别

前者：在设置的时间内没有前往目的地的感兴趣流量，连接将被断开。默认120秒

后者，如果有其它呼叫在等待使用线路时，当前呼叫空闲多少时间后将被断开，默认20秒。

（无感兴趣数据后，断开连接之前的等待时间。）

PPP的配置

多链路PPP。接口模式**ppp multilink**

dialer load-threshold load either 定义什么时候使用第2条Link。

这里LOAD为1-255 （在D通道上配置，如BRI0）

CALLER ID：允许那些号码进行拨入。保证了一些安全性 **if# isdn caller number** 配置主叫方ID过滤。

验证：**called-party number**应答哪些电话号码的子号，相当于我们的分机号。CLID=Call line id，提高ISDN连接的安全性，过滤到来的ISDN呼叫来实现，根据许可号码表对主叫号码进行验证

速率的协商 **if# isdn answer1 number**

if# isdn answer2 number

检验

show isdn status（能显示出TEI－Terminal Endpoint Identifier的取值，范围是0-127，CISCO将0用于主ISDN，0-63为手动，64-126为动态，127用于TEI分配广播请求。）可显示出1-3层信息包括交换类型、状态、拨号接口、感兴趣流量的定义、PPP协商、身份验证失败。显示BRI接口上1-3层的情况。

debug isdn q921可以查看D通道的2层消息！（Router与ISDN Switch）

debug isdn q931查看呼叫的建立和放弃的过程。display call setup and teardown info，与debug isdn event为3层排错的命令。接入Router和ISDN SW之间。

show interface bri x y （Y=1或2） 用于显示ISDN BRI接口的信息，未指定数字则显示D通道信息。

(spoofing＝欺骗)。低版本IOS只支持FIFO，不支持WFQ。查看BRI的PPP命令：**sh int bri**

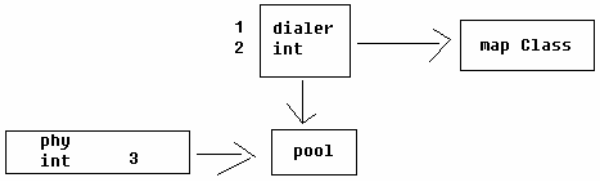
Show ppp multilink 、**debug dialer**(检查拨号DDR，是否过载)。

PPP MLP排错：

1.CHAP/PAP/CallID、

2.在Router上Dial Load threshold的问题。

3.WFQ启用？



优化DDR

拨号原形：增加灵活性。

dialer inter. dialer pool . physical inter 必须的。

dialer map-class，这个是可选的。

每一个dialer interface（逻辑的端口）在使用的时候要调用dialer pool里面的一个成员。

physical interface要划分到dialer pool里成为他的成员以供dialer inter来调用

dialer map-class是为了优化配置而加入的可选项。

循环组：

可以将一种接口配置应用于一组物理接口，在有多个呼叫和多个呼叫目的地情况下很有用。

拨号循环组：

命令：全局下

创建**Interface Dialer Groupnumber**

接口下，将物理接入加入到组：**dialer rotary-group groupnumber**。将一系列物理接口加入到同一个循环组中

不能将CON（Line0）加入到循环组中。

异步接口组：命令：

Interface Group-async 1

Group-range 1-8。

配置 **dialer profiles**：

它是将逻辑配置和物理接口分开。可定义封装、ACL、呼叫数和特性的启用禁用。它是每个呼叫动态地将逻辑配置和物理配置关联起来，它是一种新的DDR模式，决定了用户的特征，动态地绑定到用于接收/发起DDR呼叫的物理接口。支持PPP、HDLC、FR、X.25封装。使用Dialer Profile前BRI/PRI的B信道继承相同的物理接口配置，用作备用接口（standby mode），直到不再处于备用状态后才可用。Dialer Profile解决了这种问题。

首先熟悉3个命令：

(1) 在逻辑端口 dialer端口上:**dialer string number class map class-name**

这个意思是说让逻辑端口拨号，这里给给出了电话号码 number。后面的class 参数可以指定一个map-class.

(2) 在逻辑端口dialer端口上:**dialer pool number**

这里的number是说这个逻辑端口在使用的时候调用哪个池里面的成员，这里number应该为一个池的号码

(3) 在物理端口上：**dialer pool-member 101 (pri 1000)**这里也可以指定他的优先级，级别越高，越先被调用这里说说明了这个物理端口属于pool 101这个池。

要注意几点:

一个物理端口可以是多个池的成员，一个池可以有多个dialer interface来调用。一个dialer interface 只能调用一个池的成员。

排除断开故障：检查拨号闲置时间值、检查ACL。

检验dialer profiles

show dialer inter bri0显示有关（如DDR拨出）拨入/出呼叫的统计信息（号码、成功与否、时间、状态、通道、线路状况、IP、接口信息等）。

show interface dia1

debug dialer 调试Dial（实时）或Dial Profile的问题。

排除拨入故障：

1. 检查Dialer int的Pool、
2. 物理接口上的验证、
3. 检查Dialer Int远程拨号名。

无法拨号原因：

1. Dialer Group未定义（Dialer-int）
2. Dialer-List、
3. 无空闲Dialer（物理接口）
4. 不能Place Call（无号码，未在Dialer Int配置号码）。

Debug Dialer Packet 若呼叫意外断开或总也断不开，检查拨号空闲定时器值和感兴趣流量定义。

Dialer in-band 将接口的空闲超时设置为默认120s。但这个值不会出现在sh run中

ISDN for backup connection

配置备份的拨号

当主链路失败的时候起用备份线路。

备份线路也可以为主链路进行一定的负载分担

具体配置：接口模式：**backup interface dialer 0** 配置在主接口上。指定哪个接口用做备份

backup delay 10 10

第一个10为设置当主链路失效后10秒开启备份线路，第2个10为，当主链路恢复的时候等待10秒中DOWN掉备份线路。

如果一个链路被设置为另外一个连路的备份线路的时候，这个时候当主链路没失效的时候应该为STANDY BY状态。

一个端口只要一被设置为一个端口的备份端口的话他就不能做其他的任何事情了。这里就突出了DIALER INTERFACE的特点。

做负载分担时候的配置:**if# backup load 50 20**这个意思是说当流量占到主链路的50%的时候开启备份线路。

这个时候当两个线路上的总流量的和为主链路的20%的时候DOWN掉备份线路。

物理断口做备份的限制，当他做了一个链路的备份的时候他就不能做别的任何事情了。

```
inter dia 0
ip unnumber lo0
encap ppp
dialer remote-name kaka
dialer pool 1
dialer string 11111111
dialer-group 9
inter bri 0/0
encap ppp
ppp authentica chap
dialer pool-member 1
inter s 0
ip add 1.1.1.1 255.0.0.0
backup inter dia 0
backup delay 10 5
```

线路备份方式：

1. 拨号备份、
2. 浮动静态路由、
3. 按需电路、
4. Snapshot快照路由、
5. Dialer Watch监视路由表。

负载分担（路由）

注意一些路由选择协议。

比如。如果是OSPF。他只能做等代价的负载均衡。所以这个时候要指定OSPF的COST。那么，如果是EIGRP的话，他可以实现不等代价的负载均衡。当然要指定一个倍数关系参数是：**variance 3**意思是说相差3倍可以负载均衡。Variance用于配置非等价负载均衡，指定最佳度量值和可接受的最差度量值之间的比值（几倍范围是可接受的）。 最小的Metric和最大相差的倍数。

traffic-share {balanced|min}：指定如何参与EIGRP负载均衡的路由之间分配流量，默认4条，最多6条。Balanced指定流量与度量值成反比，如果Varance为3则，最佳路由的流量是最差路由的3倍。Min表示使用度量值最小的路由来传输数据。

验证配置

```
show inter s0
show int dia 0
```

通过dialer watch来实现备份

```
inter dia 1
dialer watch-group 1
dialer watch-list 1 ip 10.1.2.0 255.255.255.0
```

这里在DIALER接口实现了一个**dialer watch 1**

那么定义一个**dialer watch-list 1**。这里监视的是路由10.1.2.0。如果这个路由失效，那么dialer 1会触发一个拨号，

这里只是一个思想，具体的配置没有描述。

NAT

Basic NAT Concepts

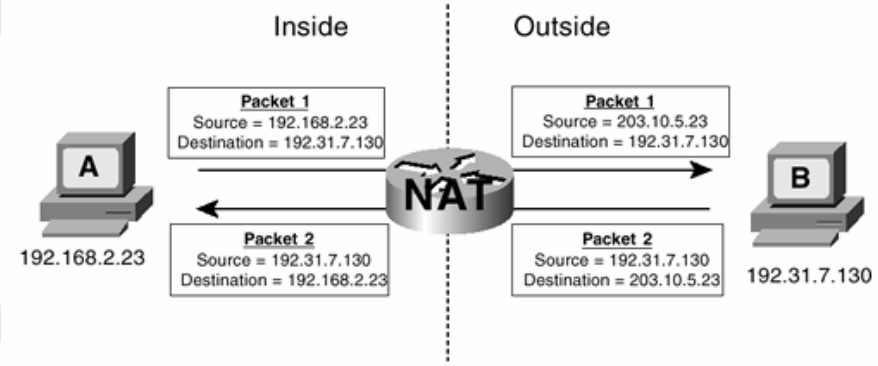
NAT的优点和缺点：

优点：节省IP地址，能够处理地址重复的情况，增加了灵活性，消除了地址重新编号，隐藏了内部的IP地址。

缺点：增加了延迟，丢失了端到端IP的跟踪过程，不能够支持一些特定的应用程序（如MSN5.0以下版本），需要更多的内存来储存一个NAT表，需要更多的CPU来进行处理NAT的过程。

NAT的概念：

INSTDE（内部看）和OUTSIDE（外部看）。本地和全局。



NAT的原理

转换内部的源地址，转换外部的源地址，PAT，解决地址重叠的问题。

地址分类

1. IL 内部本地，地址不对外公布
2. IG 内部全局，外部可以知道内部设备
3. OG 外部全局，分布给外部设备的地址，不会向内部公布
4. OL 外部本地，通过这个地址，内部设备可以知道外部设备

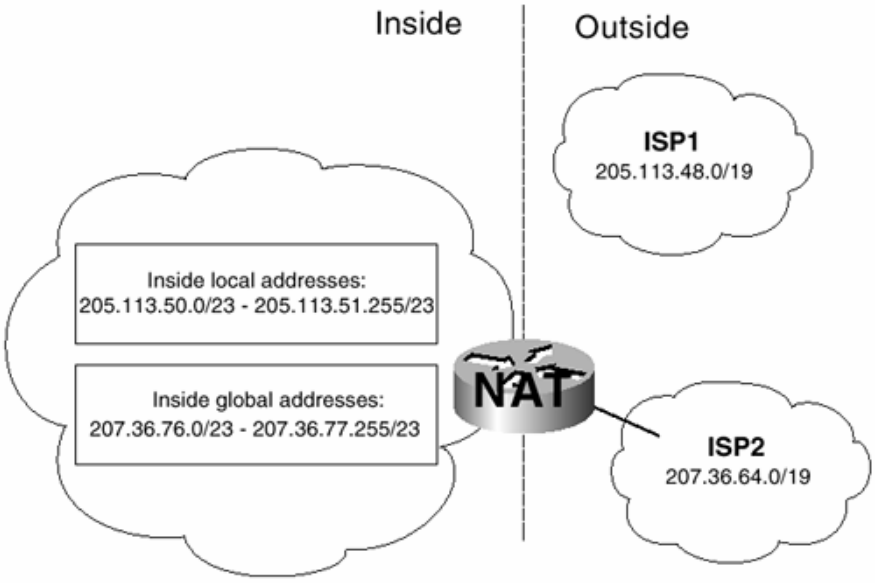
NAT在cisco路由器中采用NAT地址翻译表进行地址映射

```
NATrouter#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside global
---  203.10.5.23       192.168.2.23       172.16.80.91       192.31.7.130
---  203.10.5.23       192.168.2.23       ---                ---
---  ---               ---                172.16.80.91       192.31.7.130
NATrouter#
```

地址翻译表在一个条目刚加入时，启动一个定时器，如果超时，则去掉该条目，默认时间为86400s（24h）

通过ip nat translation timeout 可以修改

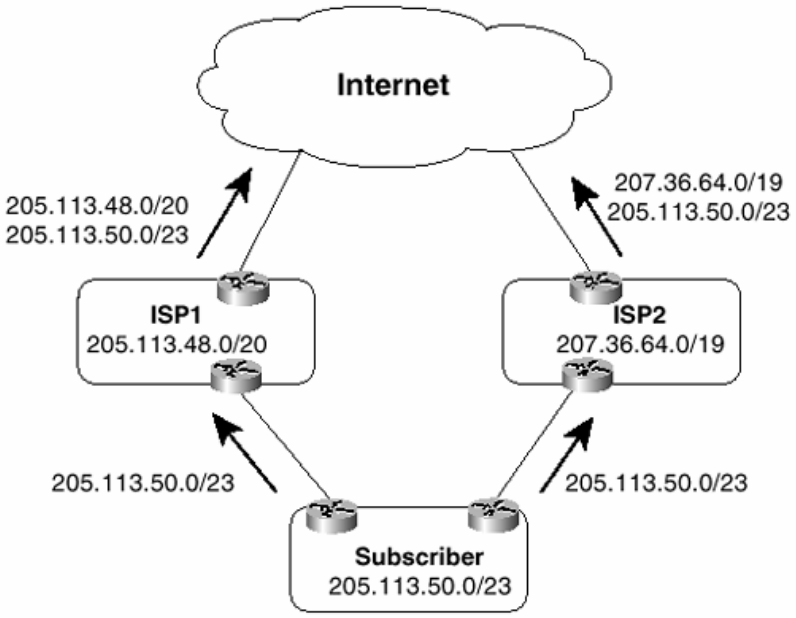
NAT and ISP Migration



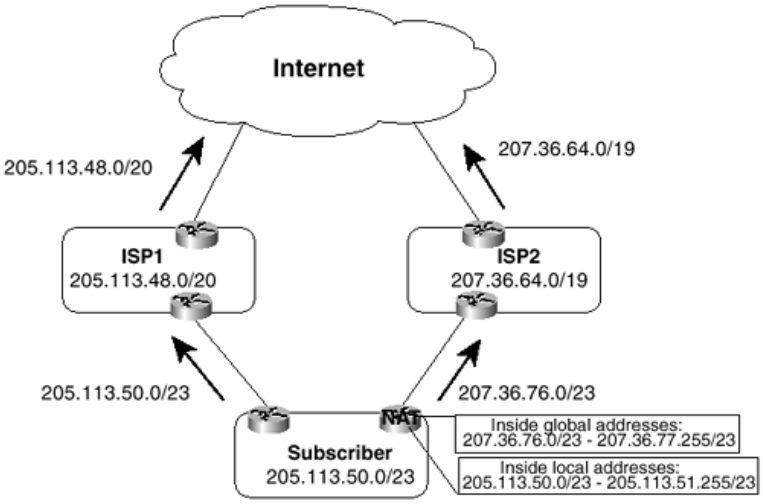
ISP变迁时通过NAT可以极少的变动内部网络的情况下，进行ISP的迁移。

NAT and Multi-AS

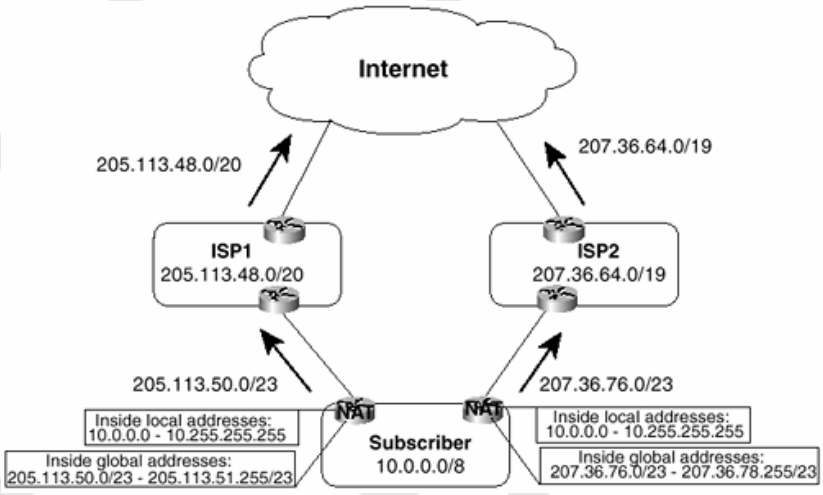
在没有NAT以前，需要在ISP通告的BGP路由条目中打洞



使用NAT后，可以采用将一个ISP的地址段NAT到另一个ISP的地址段



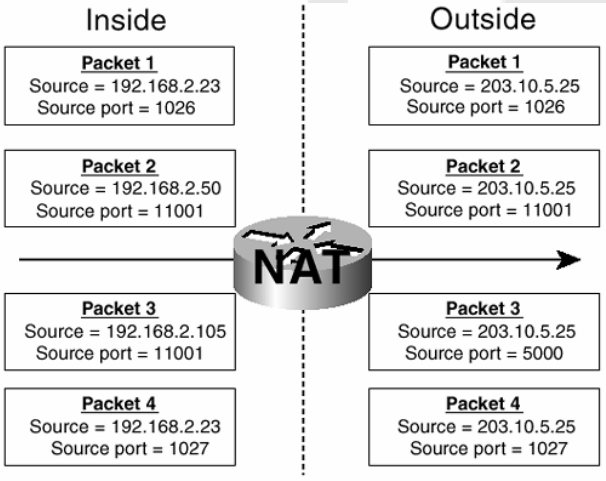
或者使用RFC1918保留地址，然后在2个ISP的边界路由器上，同时做NAT



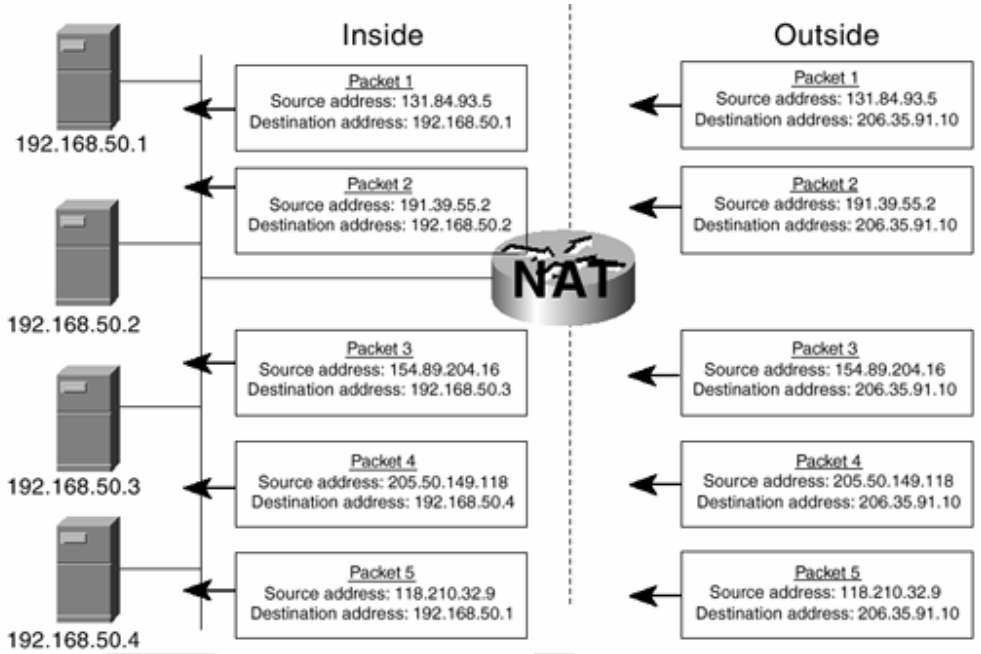
但这种方式只限于小规模网络，很多路由器不支持NAT的线速发包，所以在较多主机的时候，会产生很大的delay

PAT

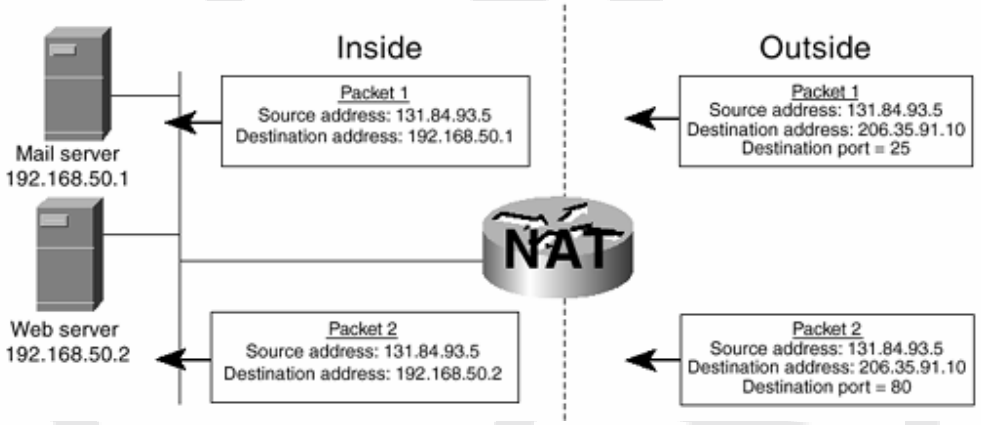
Cisco将多个ip映射到一个ip上可以通过PAT 将其映射到不同的端口



NAT and TCP Load Distribution



NAT and Virtual Server

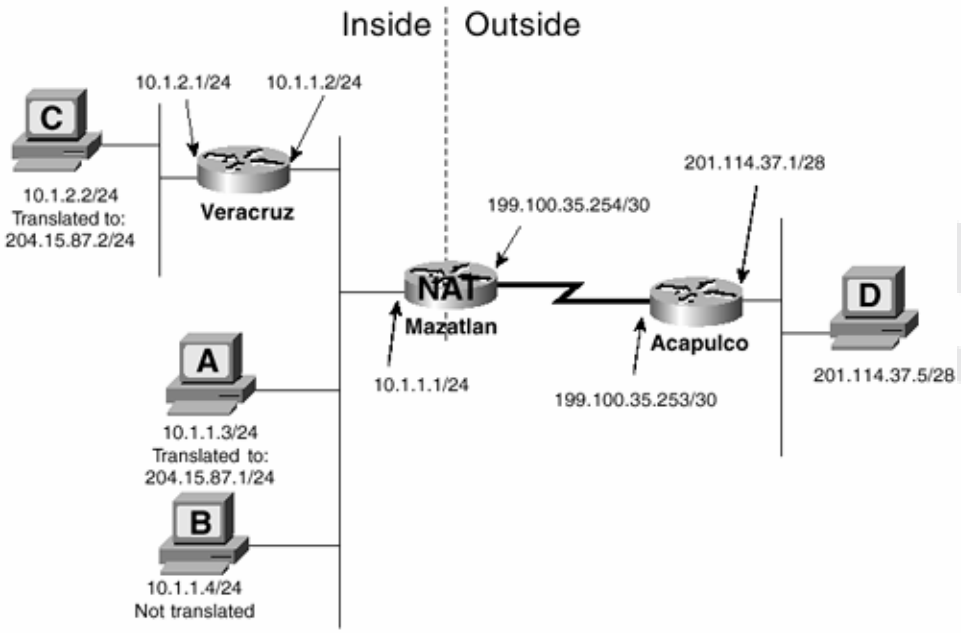


NAT Issues

由于NAT对IP和TCP信头的综合处理，使得一些服务在使用上会出现异常

1. 信头的checksum需要重新计算
2. IPSec等加密服务由于不能让NAT修改Ip报头，所以会工作异常
3. ICMP报文被修改
4. DNS解析时需要静态的NAT
5. FTP需要服务器打开被动模式，并采用PASV传送文件

NAT Configuration



Case1 Static-NAT

Mazatian Config

```
interface Ethernet0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
!
interface Serial1
 no ip address
 encapsulation frame-relay
!
interface Serial1.705 point-to-point
 ip address 199.100.35.254 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 705
!
router ospf 100
 network 10.1.1.1 0.0.0.0 area 0
 default-information originate
!
ip nat inside source static 10.1.2.2 204.15.87.2
ip nat inside source static 10.1.1.3 204.15.87.1
!
ip route 0.0.0.0 0.0.0.0 199.100.35.253
```

在E0口上启用**ip nat inside** 将其定义为内部端口
在S1.705上启用**ip nat outside** 将其定义为外部端口
通过**ip nat source static**定义静态的地址映射

```
Mazatlan#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 204.15.87.2        10.1.2.2           ---                ---
--- 204.15.87.1        10.1.1.3           ---                ---
```

ip nat outside source static 201.114.37.5 10.1.3.1将主机D配置外部**NAT**，使其看上去像内部网的一部分

```
Mazatlan#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 204.15.87.2        10.1.2.2           ---                ---
--- 204.15.87.1        10.1.1.3           ---                ---
--- ---                ---                10.1.3.1           201.114.37.5
```

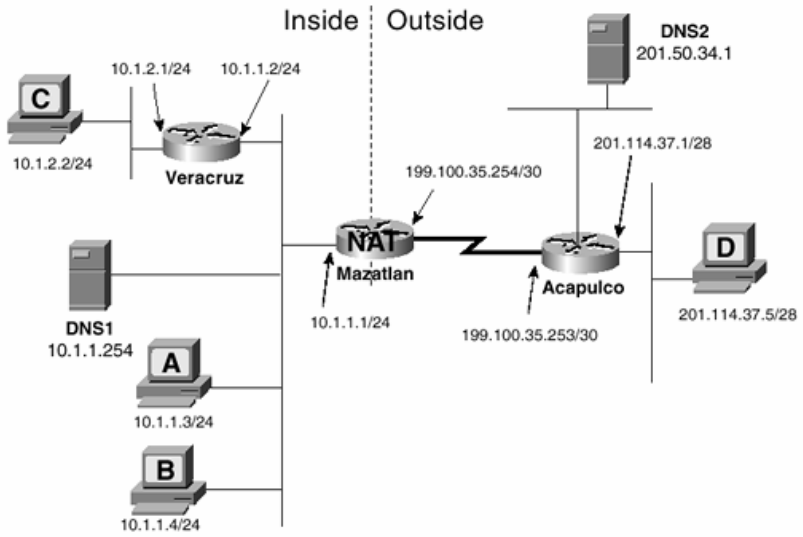
此后内部地址将自动映射到外部地址

```
Mazatlan#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
--- 204.15.87.2        10.1.2.2           ---                ---
--- 204.15.87.1        10.1.1.3           ---                ---
--- ---                ---                10.1.3.1           201.114.37.5
--- 204.15.87.1        10.1.1.3           10.1.3.1           201.114.37.5
--- 204.15.87.2        10.1.2.2           10.1.3.1           201.114.37.5
```

为了让内部主机只能通过OL地址访问，可以做如下的访问控制过滤

```
interface Ethernet0
 ip address 10.1.1.1 255.255.255.0
 ip access-group 101 in
 ip nat inside
!
interface Serial1
 no ip address
 encapsulation frame-relay
!
interface Serial1.705 point-to-point
 ip address 199.100.35.254
 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 705
!
```

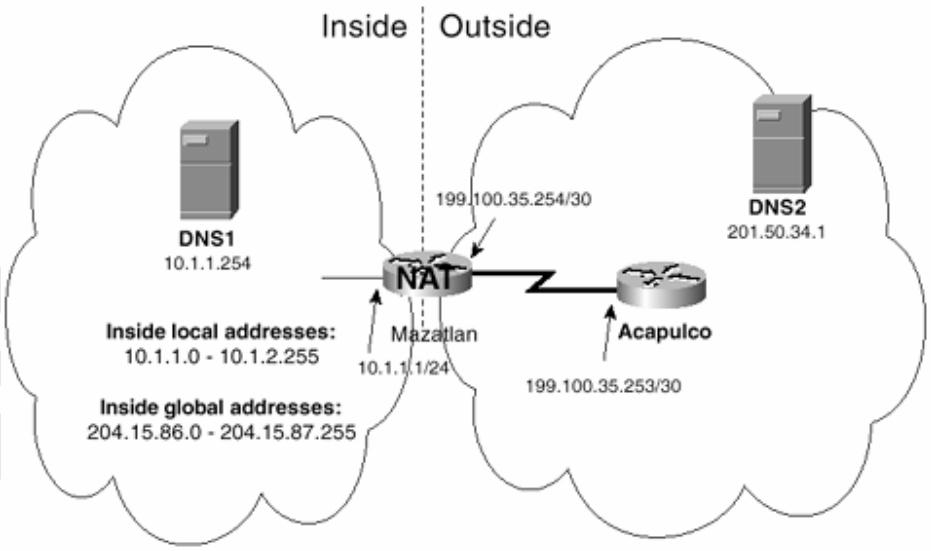
```
router ospf 100
 network 10.1.1.1 0.0.0.0 area 0
 default-information originate
!
ip nat inside source static 10.1.1.3
 204.15.87.1
ip nat inside source static 10.1.2.2
 204.15.87.2
ip nat outside source static 201.114.37.5
 10.1.3.1
!
ip route 0.0.0.0 0.0.0.0 199.100.35.253
!
access-list 101 permit ip any host 10.1.3.1
```



DNS1为内部授权DNS，DNS2为外部授权DNS

```
ip nat inside source static 10.1.1.3 204.15.87.1
ip nat inside source static 10.1.2.2 204.15.87.2
ip nat inside source static 10.1.1.4 204.15.87.3
ip nat inside source static 10.1.1.254 204.15.87.254
ip nat outside source static 201.114.37.5 10.1.3.1
ip nat outside source static 201.50.34.1 10.1.3.2
```

Case2 Dynamic-NAT




```
interface Ethernet0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
!
interface Serial1
 no ip address
 encapsulation frame-relay
!
interface Serial1.705 point-to-point
 ip address 199.100.35.254 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 705
!
router ospf 100
 network 10.1.1.1 0.0.0.0 area 0
 default-information originate
!
ip nat pool PoolOne 204.15.86.1 204.15.86.254 netmask 255.255.255.0
ip nat pool PoolTwo 204.15.87.1 204.15.87.253 prefix-length 24
ip nat inside source list 1 pool PoolOne
ip nat inside source list 2 pool PoolTwo
ip nat inside source static 10.1.1.254 204.15.87.254
!
ip route 0.0.0.0 0.0.0.0 199.100.35.253
!
access-list 1 permit 10.1.1.0 0.0.0.255
access-list 2 permit 10.1.2.0 0.0.0.255
```

可以在地址池中加入**type mtch-host**参数，以匹配主机

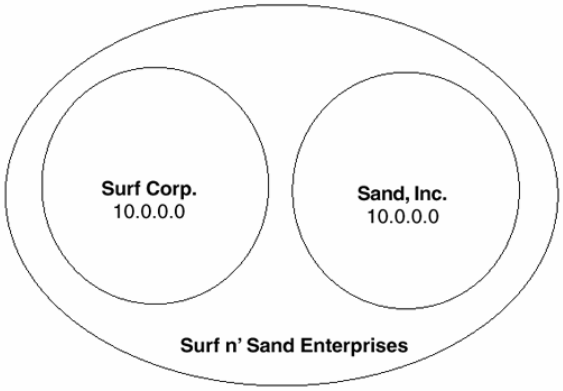
如果存在较多的**IL**地址共享**IG**地址，则可以减小**translation**计时器，防止地址不够分

Ip nat translation timeout 0

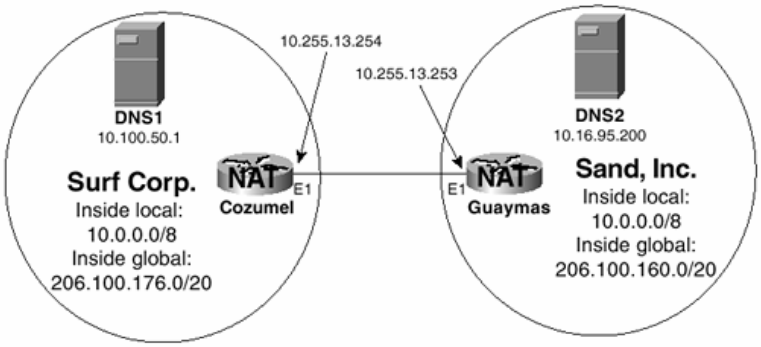
Show ip nat translation verbose 可以查看因超时被清除的路由信息

Case3 A Network Merger

由于公司合并等原因，2个网络将合并在一起，此时可能产生地址冲突，可以采用NAT解决地址冲突问题



在两个网络的边界路由器上采用NAT

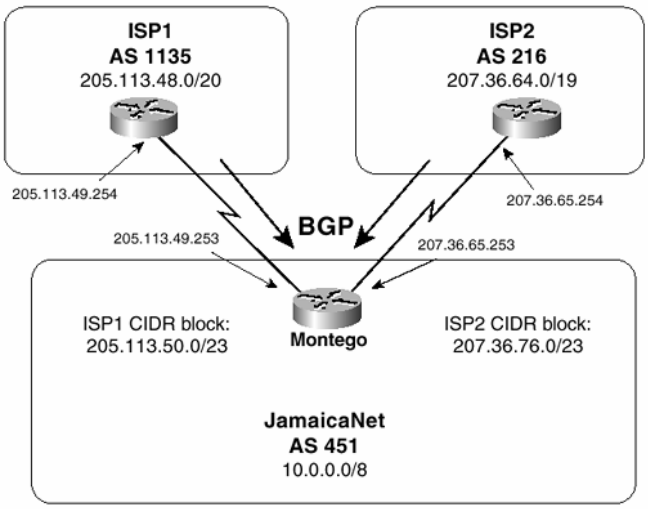



```
Cozumel
interface Ethernet0
 ip address 10.100.85.1 255.255.255.0
 ip nat inside
!
interface Ethernet1

 ip address 10.255.13.254 255.255.255.248
 ip nat outside
!
router ospf 1
 redistribute static
 network 10.100.85.1 0.0.0.0 area 18
!
ip nat pool Surf 206.100.176.2 206.100.191.254 prefix-length 20
ip nat inside source list 1 pool Surf
ip nat inside source static 10.100.50.1 206.100.176.1
!
ip route 206.100.160.0 255.255.240.0 10.255.13.253
!
access-list 1 deny     10.255.13.254
access-list 1 permit any
Guaymas
interface Ethernet0
 ip address 10.16.95.1 255.255.255.0
 ip nat inside
!
interface Ethernet1
 ip address 10.255.13.253 255.255.255.248
 ip nat outside
!
interface Serial1
 no ip address
 encapsulation frame-relay
!
interface Serial1.508 point-to-point
```

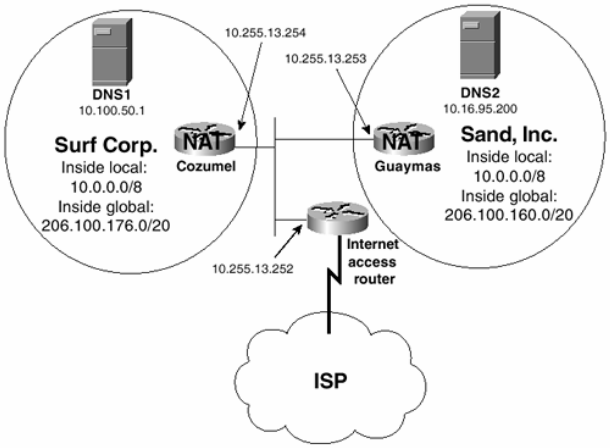
```
 ip address 10.18.3.253 255.255.255.0
 ip nat inside
 frame-relay interface-dlci 508
!
router eigrp 100
 redistribute static metric 1000 100 255 1 1500
 passive-interface Ethernet1
 network 10.0.0.0
 no auto-summary
!
ip nat pool Sand 206.100.160.2 206.100.175.254 prefix-length 20
ip nat inside source list 1 pool Sand
ip nat inside source static 10.16.95.200 206.100.160.1
!
ip route 206.100.176.0 255.255.240.0 10.255.13.254
!
access-list 1 deny    10.255.13.253
access-list 1 permit 10.0.0.0
```

Case4: ISP Multihoming with NAT

在未实现NAT以前，常采用BGP进行配置



现在可以采用如下方式配置NAT



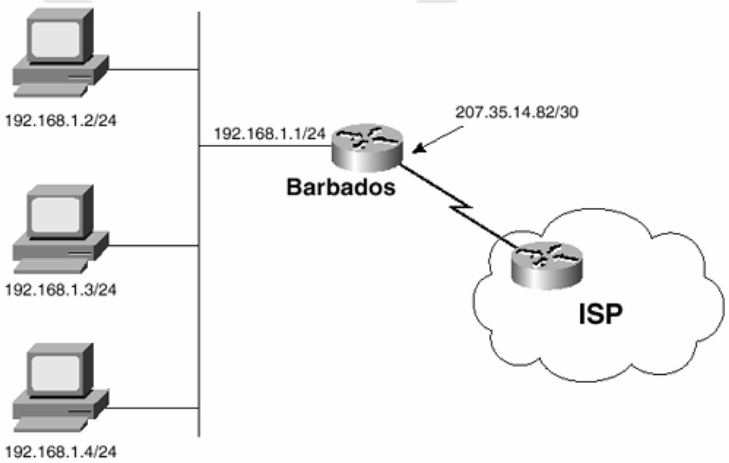

```
interface Ethernet0
 ip address 10.1.1.1 255.255.255.0
 ip nat inside
!
interface Ethernet1
```

```
 ip address 10.5.1.1 255.255.255.0
 ip nat inside
!
interface Serial1
 no ip address
 encapsulation frame-relay
!
interface Serial1.708 point-to-point
 description PVC to ISP1
 ip address 205.113.49.253 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 708
!
interface Serial1.709 point-to-point
 description PVC to ISP2
 ip address 207.36.65.253 255.255.255.252
 ip nat outside
 frame-relay interface-dlci 709
!
router ospf 10
 network 10.0.0.0 0.255.255.255 area 10
 default-information originate always
!
router bgp 451
 neighbor 205.113.49.254 remote-as 1135
 neighbor 207.36.65.254 remote-as 216
!
ip nat pool ISP1 205.113.50.1 205.113.51.254 prefix-length 23
ip nat pool ISP2 207.36.76.1 207.36.77.254 prefix-length 23
ip nat inside source route-map ISP1_MAP pool ISP1
ip nat inside source route-map ISP2_MAP pool ISP2
!
access-list 1 permit 10.0.0.0 0.255.255.255
access-list 2 permit 207.36.65.254
!
route-map ISP1_MAP permit 10
 match ip address 1
 match interface Serial1.708
!
route-map ISP2_MAP permit 10
 match ip address 1
 match ip next-hop 2
```

将一个IL地址映射到多个IG地址，可以在声明中加入extendable参数

```
ip nat inside source static 10.5.1.2 207.36.76.100 extendable
```

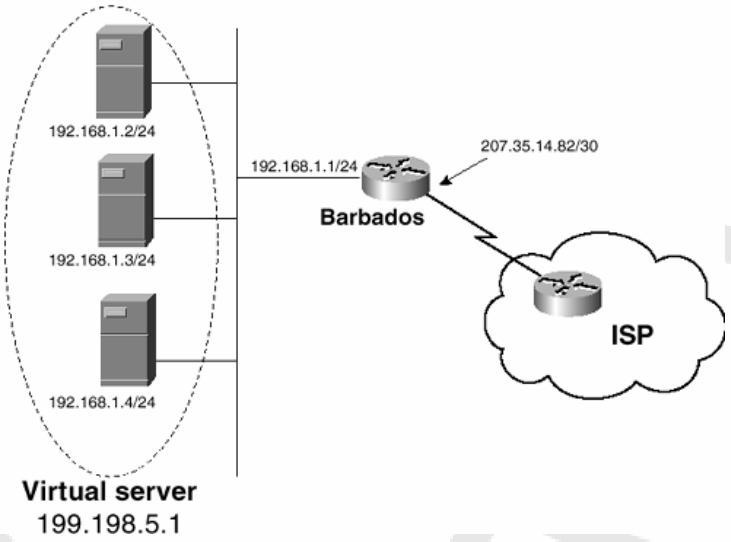
Case5: PAT




```
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
```

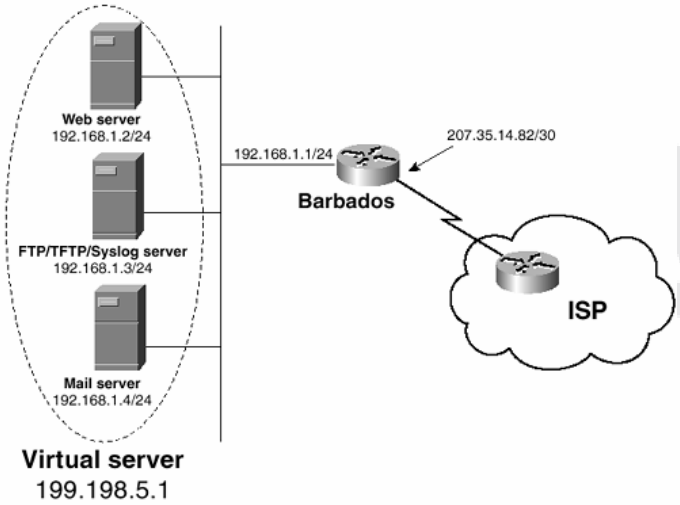
```
!
interface Serial0
 ip address 207.35.14.82 255.255.255.252
 ip nat outside
!
ip nat inside source list 1 interface Serial0 overload!
ip route 0.0.0.0 0.0.0.0 Serial0
!
access-list 1 permit 192.168.1.0 0.0.0.255
Barbados#show ip nat translations
Pro Inside global      Inside local       Outside local      Outside global
tcp 207.35.14.82:11011 192.168.1.3:11011  191.115.37.2:23    191.115.37.2:23
tcp 207.35.14.82:5000  192.168.1.2:11000  191.115.37.2:23    191.115.37.2:23
udp 207.35.14.82:3749  192.168.1.2:3749   135.88.131.55:514  135.88.131.55:514
tcp 207.35.14.82:11000 192.168.1.4:11000  191.115.37.2:23    191.115.37.2:23
tcp 207.35.14.82:11002 192.168.1.2:11002  118.50.47.210:23   118.50.47.210:23
udp 207.35.14.82:9371  192.168.1.2:9371   135.88.131.55:514  135.88.131.55:514
icmp 207.35.14.82:7428 192.168.1.3:7428   135.88.131.55:7428 135.88.131.55:7428
tcp 207.35.14.82:5001  192.168.1.2:11001  135.88.131.55:23   135.88.131.55:23
tcp 207.35.14.82:11001 192.168.1.4:11001  135.88.131.55:23   135.88.131.55:23
```

Case6: TCP Loading Balance




```
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
!
interface Serial0
 ip address 207.35.14.82 255.255.255.252
 ip nat outside
!
ip nat pool V-Server 192.168.1.2 192.168.1.4 prefix-length 24 type rotary
ip nat inside destination list 1 pool V-Server
!
ip route 0.0.0.0 0.0.0.0 Serial0
!
access-list 1 permit 199.198.5.1
```

Case7: Service Distribution




```
interface Ethernet0
 ip address 192.168.1.1 255.255.255.0
 ip nat inside
!
interface Serial0
 ip address 207.35.14.82 255.255.255.252
 ip nat outside
!
ip nat inside source static tcp 192.168.1.4 25 199.198.5.1 25 extendable
ip nat inside source static udp 192.168.1.3 514 199.198.5.1 514 extendable
ip nat inside source static udp 192.168.1.3 69 199.198.5.1 69 extendable
ip nat inside source static tcp 192.168.1.3 21 199.198.5.1 21 extendable
ip nat inside source static tcp 192.168.1.3 20 199.198.5.1 20 extendable
ip nat inside source static tcp 192.168.1.2 80 199.198.5.1 80 extendable
!
ip route 0.0.0.0 0.0.0.0 Serial0
```

show ip nat statistics可以看到一些翻译地址的统计信息

debug ip nat、debug ip nat detailed可以查看NAT翻译的过程。显示协议、源/目的Port、方向。

Clear ip nat translation * 删除动态转换条目。

xDSL and cable Modem

宽带的概述：

有高速的介入，能够提供语音和视频的服务，而且是ALWAYS ON的。普通的几种宽带介入的方法：DSL.CABLE MODEM.卫星，无线。

Cable Modem: CATV（公用天线电视网）

1) 初始化和注册过程：Modem通电后，扫描并锁定下行路径中的RF数据通道；

2) MODEM读取下行路径中特定的维护信息，这些信息指出了如何、在什么地方、何时通过上行路径进行通信。

3) MODEM与CMTS通信，以建立第1层和第2层连接。

4) MODE向DHCP服务器请求IP地址和配置信息，要给MODEM提供IP地址，DHCP服务器必须支持RFC2131，然后MODEM请求一个内部IP地址（通常是有线电视运营商使用的私有地址，用于测试和诊断），供自己使用。

5) MODEM向TFTP服务器请求DOCSIS配置文件，这是由专门的DOCSIS编辑人员创建的ASCII文件。为处理MODEM的请求，TFTP服务器必须支持RFC1350。

6) MODEM向CMTS注册，以协商并确保QoS。

7) 初始化完成，其下游的PC可以向DHCP服务器请求自己的IP地址，这是全局IP地址，被分配给与MODEM相连的PC或CABLE Modem工作在指定的带宽，基于DOCSIS

DSL技术

分为：ADSL非对称数字用户线，SDSL对称数字用户线。上行和下行速度一样。上行慢，下行快。

G.SHDSL是一个新的标准，有ITU来指定的，

IDSL不需要拨号。类似与专线。（ISDN DSL）

VDSL非常高的速度，但是距离比较断，

HDSL高速度的数字用户线。替代E1、T1。

ADSL速度为1-8M下行。上行640K-1。5M。。最大的距离为5.5KM（18000英尺）。

一般来说速度越快要求的距离就越短。

ADSL可以和POTS共存。不象其他 DSL技术。

提供了一个慢的上行和快的下行速度。

ADSL的三个标准：CAP。DMT。G.LITE。ADSL的标准必须和ISP初向匹配。

早期ADSL和POTS的共存

用户端，有一个分离器使得数字和语音信号隔离。一个接电话，一个接ADSL的设备。

经过一个所谓的分离器，分出后一根走电话公司（电话SW），一根接到ADSL的DSLAM。

两个端之间走的其实是POTS+ADSL

电话线内：频率0到4KHZ是为了POTS。

从20-250KHZ是为了上行。250-1.1MHZ是为了下行。这个是CAP的技术。

DMT是这样的：

0-4khz是为了语音的。

25-163KHZ是为了上行的，

163-1100KHZ是为了下行的。

这里上行和下行的频率是把通道划分为了多个子通道，每个通道为4KHZ。

桥接技术：网关地址为BVI地址，被分配和网关同一网段。

CPE是一个客户前端设备（ADSL MODEM）。

PPPOE的技术：主要是进行一个出内网的验证，依靠客户端软件来连接，验证可用本地或RADIUS AAA服务器。

PPPOA的技术：CPE到ISP出走的是ATM线路。ATM（AAL5）同一广播域。ADSL在Router上为ATM，例：**interface atm 0/0**

ADSL三种封装：PPPOE（RFC2516）、PPPOA、RFC1483/2684桥接。

配置PPPOE

1) 配置PPPOE的虚拟拨号网络VPDN组，必须。

2) 指定ATM INTERFACE，配置PVC和封装。

3) 创建、配置拨号接口，用于协商IP和MTU 1492

4) 配置PAT，用于共享拨号接口的动态公共IP地址。

5) 配置DHCP SERVER

6) 配置一个静态默认路由。

(1) vpdn enable 开启VPDN

vpdn-group kaka 创建一个VPDN组为kaka

在相应的局部配置模式：**request-dialin**向对方发出拨入的请求

protocol pppoe说明协议为PPPOE

(2) 配置ATM的接口：必须

inter atm 0

pvc vpi/vci

进入响应的局部配置模式：**pppoe-client dial-pool-number number** 指定是哪个拨号池的成员。

(3) 然后配置一个拨号接口：必须

inter dia0

ip address negotiated 这里是说明IP地址由对方协商给我。

encap ppp封装

dialer pool 1 指定POOL

ip mut 1492

ppp chap hostanme kaka

ppp chap password nichole 这里说明是让对方验证我方。而不用验证对方。

Dialer-group number指定拨号组，其对应的拨号列表定义了感兴趣的数据流。

(4) PAT的配置

指定出入口。然后

ip nat inside source list 1 inter dia0 overload

access-list 1 permit ip 10.0.0.0 0.255.255.255 any

下面举例子：

interface e0

ip add *.*.*.*

ip nat inside

inter dia0

ip add negotiated

ip nat outside

encap ppp

dialer pool 1

no cdp enable

ppp chap hostname cisco

ppp chap pass nichole

ip nat inside source list 101 interface dia0 overload

access-list 101 permit ip 10.0.0.0 0.255.255.255 any

(5) 下面配置DHCP服务器。可选的。。

全局：**ip dhcp pool 123**

进入相应的局部配置模式：**import all** 这里是说DNS和WINS服务器分配给客户，这里的两个地址是由拨号后ISP给的

network 10.10.0.0 255.255.0.0 指定分配IP地址范围。

default-router 10.0.0.1指定要分配出去的网关。缺省地址。

(6) 要指定一个静态默认路由

ip ruote 0.0.0.0 0.0.0.0 dialer0

实际当中的物理接口比如ATM接口的IP应该不配置，有DIA接口的IP ADD NEGOTIATED来决定。

配置PPPOA：不用配置VPDN。

1) 给ATM接口配置ATM PVC和封装，指定服务提供商提供的VCI/VPI，将ATM接口加入到拨号池中，并设置其DSL运行模式。

2) 配置拨号接口，使用IPCP IP地址协商和PPP身份验证CHAP或PPP

3) 配置PAT

4) 配置DHCP。

5) 配置默认静态路由。

实例：

```
int atm 0
dsl operating-mode auto
pvc 1/32
enc aa15mux ppp dialer
dialer pool-member 1
ip add neg
enc ppp
dialer pool number
no cdp enable
ppp chap host
ppp chap pass
ip nat Inside source list xx int dial 0
access permit
int e 0
ip nat inside
int dialer 0
in nat outside
```

5：检验配置

show dsl int atm0 能看到DSL模式、厂商代码、线路情况

```
827-1#show dsl int atm 0

                ATU-R (DS)                      ATU-C (US)
Modem Status: Showtime (DMTDSL_SHOWTIME)
DSL Mode:       ITU G.992.1 (G.DMT)
ITU STD NUM:    0x01                            0x1
Vendor ID:      'ALCB'                          'GSPN'
Vendor Specifi0x0000                            0x0002
Vendor Country0x00                              0x00
Capacity Used:97%                               100%
Noise Margin:  5.0 dB                           6.0 dB
Output Power: 9.5dBm                            12.0dBm
<output omitted>

                        Interleave    Fast      Interleave    Fast
Speed (kbps):           7616           0        896           0
<output omitted>
```

- Showtime will appear once the DSL modem has trained

检查第1层问题：Modem Status显示为showtime(DMTDSL_SHOWTIME)则说明第1层连接正常，问题出现在第2层。如果不正常，检查ATM接口是否为Down，如果为Down则可能是DSL Active Pin（活动针脚）不正确或服务商未开启DSL服务。如是在CISCO827上出现问题，还应检查电源是否正常（需供应正负12V），DSL运行模式：建议使用dsl operating-mode auto。

检查第2层问题：检查VPI/VCI设置正确与否，用Debug atm events，接口应处于UP/UP，同时Ping，通OK，不通联系ISP。查看VPI/VCI值，并做必要的修改，如60s内没有调试输出联系ISP。如果包计数器不断增大说明Router正在发送接收数据。与PPP协商相关的故障：1.远程设备（ISP）无响应；2.LCP没有打开；3.PAP/CHAP身份验证失败；4.IPCP失败。可使用Debug ppp negotiation。

show interface atm 0检查物理层故障。

线路质量的好坏和CPE到DSLAM的距离是影响DSL链路质量的关键。

CD（载波检测）

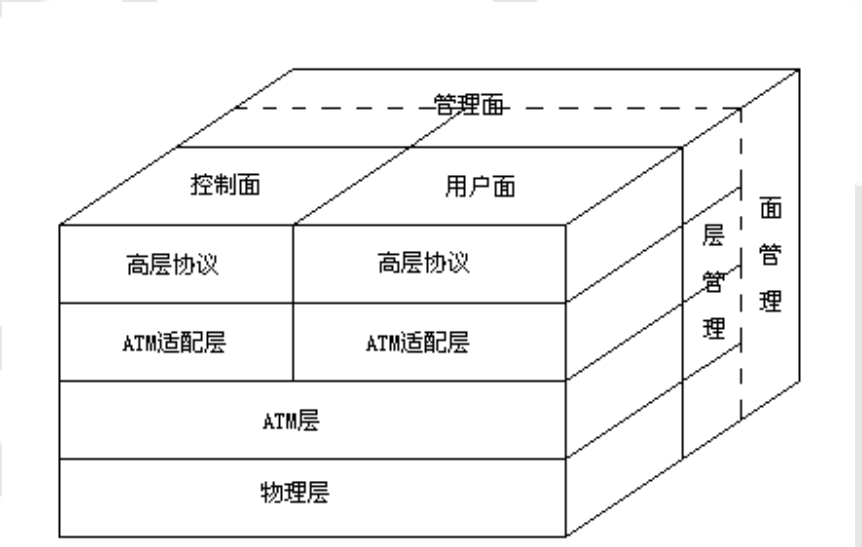
1.CD灯ON说明2层没问题，off为1层问题。2.问ISP支持DSL芯片。3.检查ATM端口和线4.电源问题。5.DSL操作模式

ATM

Basic Introduction

在ITU-T的I.321建议中定义了B-ISDN协议参考模型，如下图。它包括三个面：用户面、控制面和管理面，而在每个面中又是分层的，分为物理层、ATM层、AAL层和高层。

协议参考模型中的三个面分别完成不同的功能：



用户平面：采用分层结构，提供用户信息流的传送，同时也具有一定的控制功能，如流量控制、差错控制等；

控制平面：采用分层结构，完成呼叫控制和连接控制功能，利用信令进行呼叫和连接的建立、监视和释放；

管理平面：包括层管理和面管理。其中层管理采用分层结构，完成与各协议层实体的资源和参数相关的管理功能，如元信令。同时层管理还处理与各层相关的OAM信息流；面管理不分层，它完成与整个系统相关的管理功能，并对所有平面起协调作用。

PhysicalLayer

ATM意即异步传输模式（asynchronous transfer mode)。这种模式可以与同步T1线路做一对比。在T1线路中每125us都有一个T1帧生成，该速率由主时钟控制，每帧的第k时隙中有从相同源来的1字节数据。T1是同步的。而ATM不严格要求信元交替地从不同的源到来，每一列从各个源来的信元，没有特别的模式，信元可以从任意不同的源到

来，而且，不要求从一台计算机来的信元流是连续的，数据信元可以有间隔，这些间隔由特殊的空闲信元（idle cell）填充。

ATM并不标准化传输的信元格式。实际上，它指出仅发送单个信元是可以的，并且指出信元可以被装入到T1，T3，SONET或FDDI（光纤LAN）线路上发送。对于以上的这些例子，有标准规定信元如何封装到这些系统提供的帧里。

在最初的ATM标准中，主速率为155.52Mb/s，另外还有一个4倍于它的速率（622.08Mb/s）。选择此速率是为了和SONET兼容，SONET是电话系统中用于光纤线路的分帧标准。基于T3（44.736Mb/s）和FDDI（100Mb/s）上的ATM也已出现。

ATM的传输介质常常是光纤，但是100m以内的同轴电缆或5类双绞线也是可以的。光纤可达数千米远。每个链路处于计算机和一个ATM交换机之间或两个ATM交换机之间。换句话说，ATM链路是点到点的（和LAN不一样，它在一条电缆上有许多发送方和接收方）。通过让信元从一条线路进入交换机并且从多条线路输出，可以获得广播效果。每条点到点链路是单向的。对于全双工操作需要两条链路，每个方向的流量占用一条。

ATM的物理层包括两个子层，即物理介质子层（PM）和传输会聚（TC）子层。其中物理介质子层提供比特传输能力，对比特定时和线路编码等方面作出了规定，并针对所采用的物理介质（如光纤、同轴电缆、双绞线等）定义其相应的特性；传输会聚子层的主要功能是实现比特流和信元流之间的转换。

对于输出，ATM层提供信元序列，PDM子层进行必要的编码，并且以比特流的方式发送它们。对于输入，PDM子层从网络中获得输入的比特，并且向TC子层提交一个比特流。信元的边界并没有标记出来，TC子层负责找出信元在何处开始和结束。但这不仅困难，而且在理论上行不通，因此TC层去掉了这一功能。因为TC层管理分帧，所以它属于数据链路功能，因此我们在第3章讨论它。

Interface

ITU-T和ATM论坛将物理接口分为三类，即基于SDH、基于信元和基于PDH。下面从不同的角度介绍：

传统的数字信令。

DS0	64kbit/s
DS1(T1)	1.544Mbit/s
DS2(T2)	6.312Mbit/s (4 DS1,96 DS0)
DS3(T3)	44.736Mbit/s (28 DS1,672 DS0)
DS4	274.176Mbit/s (4032 DS0)

SONET的同步传送信令(STS)

STS-1(OC-1)	51.84Mbit/s
STS-3(OC-3)	155.52Mbit/s (3STS-1)
STS-12(OC-12)	622.08Mbit/s (12STS-1)
STS-24(OC-24)	1244.16Mbit/s (24STS-1)
STS-48(OC-48)	2488.32Mbit/s (48STS-1)

ANSI标准

STS-1	51.84Mbit/s
STS-3c	155.52Mbit/s
STS-12c	622.08Mbit/s
DS3	44.736Mbit/s

CCITT标准

DS1	1.544Mbit/s
E1	2.048Mbit/s

DS2	6.312Mbit/s
E2	8.448Mbit/s
E3	34.368Mbit/s
DS3	44.736Mbit/s
E4	139.264Mbit/s
STM-1	155.52Mbit/s（与STS-3相同）
STM-4	622.08Mbit/s（与STS-12相同）

ATM Switch

1、ATM基本排队原理

ATM交换有两条根本点：信元交换和各虚连接间的统计复用。信元交换即将ATM信元通过各种形式的交换媒体，从一个VP/VC交换到另一个VP/VC上。统计复用表现在各虚连接的信元竞争传送信元的交换介质等交换资源，为解决信元对这些资源的竞争，必须对信元进行排队，在时间上将各信元分开，借用电路交换的思想，可以认为统计复用在交换中体现为时分交换，并通过排队机制实现。

排队机制是ATM交换中一个极为重要的内容，队列的溢出会引起信元丢失，信元排队有是交换时延和时延抖动的主要原因，因此排队机制对ATM交换机性能有着决定性的影响。基本排队机制有三种：输入排队、输出排队和中央排队。这三种方式各有缺点，如输入排队有信头阻塞，交换机的负荷达不到60%；输出排队存储器利用率低，平均队长要求长，而中央排队存储器速率要求高、存储器管理复杂。同时，三种方式有各有优点，输入队列对存储器速率要求低，中央排队效率高，输出队列则处于两者之间，所以在实际应用中并没有直接利用这三种方式，而是加以综合，采取了一些改进的措施。改进的方法主要有：

减少输入排队的队头阻塞。

采用带反压控制的输入输出排队方式。

带环回机制的排队方式。

共享输出排队方式。

在一条输出线上设置多个输出子队列，这些输出子队列在逻辑上作为一个单一的输出队列来操作。

2、ATM交换机构

为实现大容量的交换，也为了增加ATM交换机的可扩展性，往往构造小容量的基本交换单元，再将这些交换单元按一定的结构构造成ATM交换机构(Fabric)，对于ATM交换机构来说，研究的主要问题是各交换单元之间的传送介质结构及选路方法，以及如何降低竞争，减少阻塞。

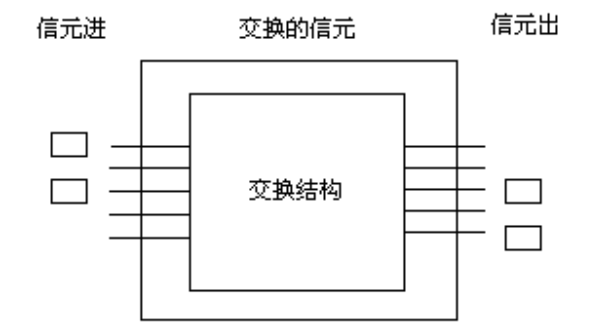
ATM交换机构分类方法不一，有一种分法为：时分交换和空分交换，其中时分交换包括共享总线、共享环和共享存储器结构，空分交换包括全互连网和多级互连网。

3、ATM交换机

ATM信元交换机的通用模型如下图所示。它有一些输入线路和一些输出线路，通常在数量上相等（因为线路是双向的）。在每一周期从每一输入线路取得一个信元（如果有的话）。通过内部的交换结构（switching fabric），并且逐步在适当的输出线路上传送。从这一角度上来看，ATM交换机是同步的。

一个通用的ATM交换机

交换机可以是流水线的，即进入的信元可能过几个周期后才出现在输出线路上。信元实际上是异步到达输入线路的，因此有一个主时钟指明周期的开始。当时钟滴答时完全到达的任何信元都可以在该周期内交换。未完全到达的信元必须等到下一个周期。



信元通常以ATM速率到达，一般在150Mb/s左右，即大约超过360,000信元/s，这意味着交换机的周期大约为2.7um。一台商用交换机可能有16条~1,024条输入线路，即它必须能在每2.7um内接收和交换16个~1,024个信元。在622Mb/s的速率上，每700ns就有一批信元进入交换结构。由于信元是固定长度并且较小（53字节），这就可能制造出这样的交换机。若使用更长的可变长分组，高速交换会更复杂，这就是ATM使用短的、固定长度信元的原因。

4、ATM交换机的分类

各种ATM交换设备由于应用场合的不同，完成的功能也略有差异，主要区别有接口种类、交换容量、处理的信令这几方面。

在公用网中，有接入交换机、节点交换机和交叉连接设备。接入交换机在网络中的位置相当于电话网中的用户交换机，它位于ATM网络的边缘，将各种业务终端连入ATM网中。节点交换机的地位类似于现有电话网中的局用交换机，它完成VP/VC交换，要求交换容量较大，但接口类型比接入交换机简单，只有标准的ATM接口，主要是NNI接口，还有UNI接口或B-ICI接口，信令方面，只要求处理ATM信令。交叉连接设备与现有电话网中的交叉连接设备作用相似，它在主干网中完成VP交换，不需要进行信令处理，从而实现极高速率的交换。

在ATM专用网中，有专用网交换机、ATM局域网交换机。专用网交换机作用相当于公用网中的节点交换机，具有专用网的UNI和NNI接口，完成P-UNI和P-NNI的信令处理，有较强的管理和维护功能。ATM局域网交换机完成局域网业务的接入，ATM局域网交换机应具有局域网接口和ATM P-UNI接口，处理局域网的各层协议以及ATM信令。

ATM中的数据链路层

ATM物理层大体包括了OSI物理层和数据链路层，包括功能像OSI物理层的物理介质决定了子层和与数据链路功能一样的传输汇集（TC）子层。对于ATM，没有特殊的物理层特性。相反，是由SONET，FDDI及其他传输系统运送ATM信元的。因此，我们这里将集中于TC子层的数据链路功能。

当一个应用程序产生了一条要发送的消息后，此消息要进入传输线路上，向下传到ATM协议栈，加上头部和尾部，并把分段放入ATM信元中。最后，这些信元到达TC子层进行传输。让我们看一下出了门后，在路上所发生的事情。

信元传输

第一步是进行头部的校验和。每个信元都有一个5字节的头部，头部中包括4字节的虚拟电路及控制信息和1字节的校验和。校验和只包括了前4个头部字节，而不占用有效载荷字节。它是由32个头部位除以多项式x^8+x^2+x+1后，所得的余数构成的。校验和加上常数01010101。

做出只校验头部的决定，是为了减少由于头部错误，而造成不正确传递信元的可能，也为了避免其校验开始要大得多的有效载荷字段的校验。如果确需校验有效载荷字段，就要上到较高的层上完成这一功能。由于校验和字段只位于头部，因此这8位校验和字段被称为头部错误控制HEC（header error control）。

一旦产生出HEC，并插入信元头部，那么此信元就作好了发送准备。传输手段分成两组：异步的和同步的。当使用异步方式时，只要准备好了发送它，就可以发送，没有时间限制。

使用同步方式，信元就必须按照事先确定的时间节拍发送。如果在需要时无数据信元可用，TC子层就必须发明一个，这种信元称为空闲信元（idle cell）。

无数据信元的另一种类型是操作和维护OAM（operation and maitenance）信元。ATM机制也使用OAM信元来交换控制及其他必需的信息，以保证系统的运行。把ATM输出速率与从事传输系统的速率相匹配是TC子层的重要任务。

在接收方，空闲信元在TC子层中进行处理，但OAM信元交给了ATM层。

TC子层的另一项重要任务是：如果有的话，针对从事传输的系统，产生成帧信息。比如，一个ATM摄象机在线路上只产生一系列信元，但它也可能用ATM信元产生SONET帧，嵌入SONET有效载荷中。在后一种情况下，TC子层将产生SONET或帧，并把ATM信元打包，这并不完全是一个不必要的步骤，因为SONET有效载荷不能支持53字节信元的整数倍。

尽管电话公司明确地使用SONET作为ATM的传输系统，但是也可以定义成把ATM对应到其他系统的有效载荷字段，并且这种新帧已在工作。尤其是，映射成T1，T3或FDDI帧也是可以的。

信元接收

在输出处，TC子层的工作是取得一系列信元，在每个信元上增加一个HEC，把此结果转变成比特流，并通过加入OAM信元，将比特流匹配为进行物理传输系统的速率。在输入处，TC层准确地进行逆变换。它取来到达的比特流，设定信元边界，确定信元头（丢弃拥有不合法头部的信元），处理OAM信元，并把数据信元上传给ATM层。

最困难的部分是在到来的比特流中设定信元边界。在某些情况下，进行传输的物理层提供了帮助。然而，有时物理层对成帧并不能提供帮助。这时应该怎么办？

技巧是使用HEC。随着比特流到达TC子层，保留一个40位移位寄存器，比特流从左边进入，右边出来。TC子层然后审查这40位，看是否可能存在一个合法的信元头部。如果有，最右边的8位将是合法的HEC，而最左边的32位则不是。如果不存在这种情况，则缓冲区没有存在一个合法信元，在这种情况下，缓冲区中所有的位都向右移动一位，使得后端空出一位，于是一个新的输入位就加到最左端。不断重复此过程，直到定位一个合法的HEC。此时，明确了信元边界，因为移位寄存器包括了一个有效的头部。

ATM中的网络层

ATM层处理从源端到目的端移动着的信元，在ATM交换机中的确包含了路由选择算法和协议，它也处理全局寻址问题。因此从功能上说，ATM层发挥着和网络层相同的功能。ATM层并不能保证百分之百的可靠性，不过一个网络层的协议也不需要如此。

因为ATM层具有网络层的功能，而不具有数据链路层所具备的功能，并且，ATM层同现有的网络层类似，因此我们仍在本章中讨论ATM层协议。

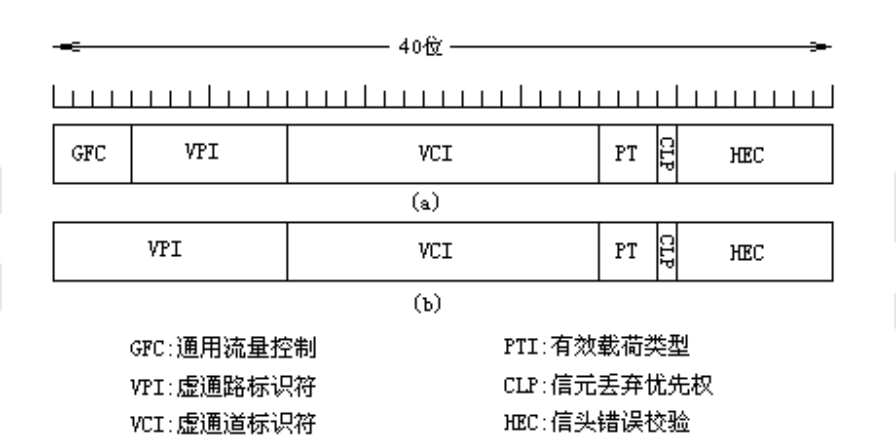
唯一的问题是ATM层不具有数据链路层协议所具有的特性：一个用于导线两端的机器之间的单站段协议，就像第3章中的协议1到协议6。ATM层却具有网络层协议的功能：端到端虚电路连接、交换、路由选择。

对于面向连接的协议来说，ATM层是不同寻常的，因为它不提供任何确认。但ATM层仍然提供了强有力的保障：沿着一条虚电路发送的信元将永远不会失去顺序。如果阻塞发生了，允许ATM子网丢弃信元，但是在任何情况下，它都不能对在一条单独的虚电路中传递的信元重新排序。然而，对于在不同的虚电路中传递的信元并没有提供顺序上的保障。

信元格式

在ATM层，有两个接口是非常重要的，即用户-网络接口UNI（user-network interface）和网络-网络接

口NNI（network-network interface）。前者定义了主机和ATM网络之间的边界（在很多情况下是在客户和载体之间），后者应用于两台ATM交换机（ATM意义上的路由器）之间。两种格式的ATM信元头部如下图。信元传输是最左边的字节优先，在一个字节内部是最左边的比特优先。



连接建立

从技术上讲，连接建立并不是ATM层的一部分，而是由控制平台使用的一个高度复杂的叫做Q.2931(stiller,1995)的ITU协议来处理的。然而，逻辑上处理建立网络层连接的地点是网络层，并且类似的网络层协议都是在这里进行连接建立的，因此我们在这里讨论它。

用于连接建立和连接释放的消息

消息	由主机发送时的含义	由网络发送时的含义
SETUP	请建立一条虚电路	进入呼叫
CALL PROCEEDING	我看见了进入呼叫	将尝试你的呼叫请求
CONNECT	我接受进入呼叫	接受你的呼叫请求
CONNECT ACK	谢谢接受	谢谢发出呼叫
RELEASE	请终止呼叫	另一端已足够坏
RELEASE COMPLETE	对 RELEASE 的确认	对 RELEASE 的确认

ATM网络允许建立多点播送通道。一个多点播送通道有一个发送者和多于一个的接收者。它们是通过如下方法建立起来的：用通常的方法在源端和目的端之间建立一条连接，接着发送ADD PARTY消息把第二个目的端连接到前一个呼叫返回的虚电路上去，接下来就可以发送其余的ADD PARTY来增加目的端的个数。

ATM有3种地址格式。第1字节指明该地址是3种地址格式中的哪一种。第1种有20字节长，是基于OSI地址格式的。第2和第3字节指明国家，第4字节给出了基于地址部分的格式，其他包括3字节指明权限，2字节指明域（domain），1字节指明区域，还有6字节的地址，以及其他一些信息项。在第2种地址格式中，第2和第3字节指定一个国际组织，而不是国家；地址的其余部分和格式与第1种相同。另一种是旧的使用15位十进制数的ISDN电话号码（CCITT E.164）作为地址的格式。

路由选择和交换

当建立虚电路时，SETUP消息沿着网络从源端走向目的端。路由选择算法决定了消息要走的路径，从而也就决定了虚电路的路径。ATM标准中没有指定任何特定的路由选择算法，所以人们就可以从我们在本章前面几节中讨论的路由选择算法中选择一种，或者选用另外不同的算法。

交换机的大部分工作量是花费在如何从一个信元里的虚电路信息里得到输出线路的选择上。除了在每一个方向上的最后一个站段外，路由都是在VPI字段上进行的，而不是在VCI字段；在最后一个站段，信元在交换机和主机之间传送。在两台交换机之间只使用虚通路。

在局域网中，事情简单得多，一条简单的虚通路就可以为所有的虚电路所使用。

服务类型

恒定比特率CBR（constant bit rate）主要用来模仿铜线或者光导纤维。没有差错校验，没有流量控制，也没有其余的处理。这个类别在当前的电话系统和将来的B-ISDN系统中作了一个比较圆滑的过渡，因为话音级的PCM通道，T1电路以及其余的电话系统都使用恒定速率的同步数据传输。

可变比特率VBR（variable bit rate）被划分为两个子组别，分别是为实时传输和非实时传输而设立的。RT-VBR主要用来描述具有可变数据流并且要求严格实时的服务，比如交互式的压缩视频（例如电视会议）。NRT-VBR用于主要是定时发送的通信场合,在这种场合下,一定数量的延迟及其变化是可以被应用程序所忍受的,如电子邮件。

可用比特率ABR（available bit rate）术语是为带宽范围已大体知道的突发性信息传输而设计的。ABR是唯一一种网络会向发送者提供速度反馈的服务类型。当网络中拥塞发生时会要求发送者减小发送速率。假设发送者遵守这些请求，采用ABR通信的信元丢失就会很低。运行着的ABR有点象等待机会的机动旅客：如果有空余的座位（空间），机动的旅客就会无延迟地被送到空余座位处；如果没有足够的容量，他们就必须等待（除非有些最低带宽是可用的）。

未指定比特率UBR（unspecified bit rate）不做任何承诺，对拥塞也没有反馈，这种类型很适合于发送IP数据报。如果发生拥塞，UBR信元也会被丢弃，但是并不给发送者发送反馈，也不给发送者希望放慢速度的期望。

各种ATM服务类型的特性

服务特性	CBR	RT-VBR	NRT-VBR	ABR	UBR
带宽保证	是	是	是	可选	不
适用于实时通信	是	是	不	不	不
适用于突发通信	不	不	是	是	是
有关于拥塞的反馈	不	不	不	是	不

服务质量

服务质量在ATM网络中是一个重要的话题，这部分因为ATM网络都是用作实时传输的，比如音频和视频。当一条虚电路建立时，传输层（典型地为主机中的一个进程，“客户”）和ATM网络层（例如：一个网络操作者，也即“运载提供者”）都要遵守一个定义服务的协定。

协定的第一部分是通信量描述符（traffic descriptor）。它描述要提供的载荷。协定的第二个部分指定客户所要求的和通信提供者同意的服务质量。无论是载荷还是服务，都是要以可度量的数量来描述的，这样约定就可以被客观的决定。

为了使具体的通信量协定成为可能，ATM标准定义了一系列的服务质量Qos(quality of service)，客户和通信提供者可以协商这些参数的值。对于每一个服务质量参数，其最差情况下的值被指定了，要求通信提供者必须要达到或者超过该值。在某些情况下，参数是一个最小值，而在另外一些情况下它是一个最大值。也是在这里，服务质量在每个方向上都是单独指定的。其中一些比较重要的列在了下表中，但它们并不是对所有的服务类型都适用。

一些服务质量参数

参数	缩写词	含义
峰值信元速率	PCR	信元发送的最大速率
持续信元速率	SCR	长时间的平均信元传输速率
最小信元速率	MCR	最小的可接受的信元传输速率
信元延迟变化极值	CDVT	最大的可接受的信元抖动
信元丢失比率	CLR	信元丢失或提交得太迟的比例
信元传送延迟	CTD	信元提交时拖延的时间（中间值和最大值）
信元延迟变化	CDV	信元提交时间的变化幅度
信元错误比率	CER	提交无错信元的比例
严重错误信元块比率	SECBR	出错信元的比例
信元错误目的地比率	CMR	信元提交至错误目的地的比例

通信量整形和控制

使用和增强服务质量参数的机制是基于（部分地）一种特定的算法，也即通用信元速率算法GCRA(generic cell rate algorithm)。它的工作原理是检查每一个信元，看是否遵从了虚电路的参数。

GCRA有两个参数，它们指定了最大的允许到达率（PCR）和其中可以忍受的到达时间变化量（CDVT）。PCR的倒数，T=1/PCR是最小的信元到达间隔值。

GCRA算法被称为虚拟调度算法(virtual scheduling algorithm)，然而从另一种角度来看，它等同于一个漏桶算法。可把一个合乎协定的信元想象成是倒入一个漏桶的T单位的流体。这个桶以1单位/us的速度漏液体，因此Tus之后它就空了。如果信元正好是以1信元/Tus的速度到达，那么每一个到达的信元都会发现桶刚刚空出来，该信元会把桶内重新装上T单位的液体。因此当一个信元到达时，液体水位升至T，以后就线性递减直到为零。当一个信元提前Lus到达时，桶就应该溢出。对于一给定的T，如果我们把L设置得很小，桶的容量将会很难超过T，因此所有的信元必须以一种非常规范的间隔顺序发送。然而，如果我们现在增加L的值，使它远远大于T，桶将会容纳很多的信元，因为T+L>>T。这就意味着发送者可以以峰值速率一个接一个地发送一些突发性数据，而它们仍然能够被正确地接收。

GCRA正常情况下是通过给定参数T和L来指定的。T正好是PCR的倒数；L就是CDVT。GCRA也用来保证在任何一段较长时间内平均信元传输速率不会超过SCR。

除了提供了一条规则来看哪一个信元是合乎协定的，哪一个是不合乎协定的之外，GCRA也用于通信整形，以消除某些突发性传输。CDVT越小就意味着越好的平滑效果，但也增大了因为不合乎协定而丢弃信元的机率。在一些实现中把GCRA漏桶和一个令牌桶结合起来，以提供进一步的平滑。

拥塞控制

ATM网络必须既要处理由于大于系统处理能力的通信量而引起的长期拥塞，又要处理由于通信中的突发性传输而引起的短期拥塞。结果人们使用了几种不同的策略。它们当中最重要的可分为3类：

1、许可证控制

很多ATM网络中有以固定速率产生数据的实时通信源。告诉这一类的通信源减慢发送速率是行不通的（想象一种有一个红灯的新型数字电话。当通知拥塞发生时，红灯就会亮，讲话者将被要求速率减慢25%）。

因此，ATM网络把防止拥塞发生放在第一的位置。然而，对于CBR、VBR、UBR类通信量，根本就没有动态拥塞控制，因此在这里预防拥塞发生将远远比拥塞发生后再去恢复强得多。预防拥塞的一个主要工具是许可证控制。当一台主机需要一条新的虚电路时，它必须描述出希望被提供的通信和服务，网络便作出检查来看是否有可能，在不对已存在连接造成有害的影响的前提下处理该连接。可能需要检查多条可能的线路，从而发现哪

一条将可以做此项工作。

2、资源预订

同许可证控制密切相关的是事先预定资源的技巧，这通常是在呼叫建立时进行。因为通信量描述符给出了信元发送峰值速率，网络就有可能沿通路预留足够的带宽来处理该峰值速率。

3、基于速率的拥塞控制

在CBR和VBR通信中，因为信息源固有的实时和半实时的特性，所以即使在发生拥塞的情况下，一般也不可能让发送者减慢发送速率。在VBR服务中，没有人会担心。如果有太多的信元，把多出来的丢弃掉就是。

在ABR通信中，网络去通知一个或多个发送者并且请求它们暂时减慢发送速率直到网络恢复，这是可能的也是合理的。

怎样检测、通知和控制ABR通信中的拥塞是ATM标准发展过程中的一个热门话题，问题集中在以下两个方面：一是基于信用的解决方案，一种是基于速度的解决方案。

交换机厂商们反对基于信用的解决方案。他们不想进行所有计算，以记住这些信用，同时，也不想预先提供很多缓冲区，并认为所需要的开销总量太大。因此，采用了基于速度的拥塞控制系统。其基本模型是每个发送端在k信元数据之后传送一个特殊的资源管理RM(resource management)信元。这个信元的传输通路与k信元相同，但是它由交换机进行特殊处理。当RM信元到达接收端时，对它进行检测、修改并且再将它发送回发送端。另外，还提供了其他两种拥塞控制装置。第一种是超载荷交换机能够自发地产生RM信元，并将它们发送回发送端。第二种是超载荷交换机能够对从发送端传送到接收端的信元数据设置其中间PTI位的值。当然这两种方法没有一个是完全可靠的。

ATM中的传输层

很难说清ATM是否有传输层。一方面，ATM层具有网络层的功能，并且其上还有一层（AAL），从分层角度看AAL便是传输层。一些专家同意这一观点。此处所使用的协议之一（AAL5）功能上类似于UDP，而UDP无疑是传输层协议。另一方面，没有任何一个AAL协议像TCP那样提供可靠的端到端的连接（尽管这些协议只需做很小的变化即可）。另外，在多数应用中，在AAL之上还使用了另一个传输层。不再细究了，就在这一章中讨论AAL层及其协议，而不管它是不是真正的传输层。

ATM网络的AAL层与TCP具有本质区别，其主要原因是设计者对传输音频和视频数据流更有兴趣，为此迅速传送比精确地传送更重要。ATM层连续输出53字节的信元。信元中没有差错控制、没有流量控制以及其他种类的控制。所以，它不能很好地满足多数应用的要求。为了弥补这一不足，在建议I.363中，ITU在ATM层之上定义了一个端到端的层。这一层称为ATM适配层AAL(ATM adaptation layer)，它经历了一段曲折的历史：充满了错误、反复修订以及未完成的工作。

AAL的目标是向应用提供有用的服务，并将它们与在发送端（方）将数据分割为信元、在接收端（方）将信元重新组织为数据的机制隔离开来。它按照3个坐标轴来组织服务空间：

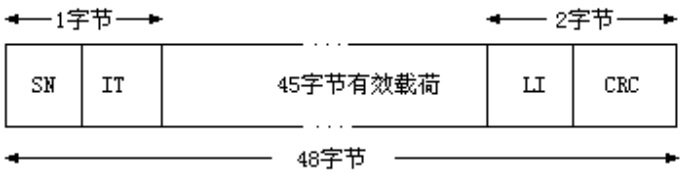
1、实时服务和非实时服务。

2、恒定比特率服务和变化的比特率服务。

3、面向连接的服务和非连接的服务。

原则上，用3个坐标轴和每个坐标轴上的2个值可以定义8种不同的服务，如下图。ITU觉得只有其中的4个有使用价值，并分别命名为类A、B、C、D。其他几种则未得到支持。从ATM4.0开始，该图有些过时，所以在这里提出它来主要是作为背景信息，以帮助读者了解为什么AAL协议设计为目前这个样子。目前主要的不同是传输类（ABR、CBR、NRT-VBR、RT-VBR和UBR）之间，而不是这些AAL支持的服务类之间。

AAL支持的基本服务类（现已过时）



为了处理这4类服务，ITU定义了4个协议而后来发现对于类C和类D的技术要求十分相似，从而将AAL3和AAL4合为AAL3/4。计算机工业当时昏然不觉，后来才发现它们都不令人满意。后来暂且定义了另一种协议--AAL5来解决这个问题。

ATM适配层的结构

ATM适配层的上面部分称为会聚子层(cenvergence sublayer)。其作用是向应用程序提供一个接口。它又是由两个子部分组成：一个是对所有应用程序都通用的公共部分（相对于给定的AAL协议），另一个是与应用程序相关的子部分。其中每个部分的作用都是与协议相关的，但是可以包括报文分帧和错误检测。

在发送端，会聚子层负责接收来自于应用程序的比特流（数据）或随机长度的报文，并将它们分为44~48字节的单元以便传输。确切的大小有赖于所用的协议，因为一些协议要占用48字节ATM载荷中的一部分作为自己的头。在接收端，该子层将信元重组为原始的报文。通常情况下，报文分界（如果存在）是要保留的。

AAL下面的部分称为分割和重组SAR(segmentation and reassembly)子层。它将会聚子层交给它的数据单元加上头和尾从而构成信元有效载荷。接着，这些载荷被交给ATM层进行传送。在接收端，SAR子层将信元重组为报文。SAR子层基本上只涉及信元，而会聚子层则与报文打交道。

SAR子层对于某些（但不是所有的）服务类来说，还有另外一些功能。特别是，它有时候可以进行错误检测和多路复用。SAR子层对于所有服务类都是存在的但功能的强弱则依赖于其特定的协议。

AAL1

AAL1是用于A类传输的协议。A类传输是指实时的、恒定比特率的、面向连接的传输，例如非压缩的音频和视频数据。输入的是比特流，不存在报文分界。对于这种传输，并没有使用像停--等这样的错误检测协议，因为由超时和重发机制引入的延迟是不能接受的。但是，丢失信元时会通知应用程序，由它采取措施（如果可能的话）来进行弥补。AAL1使用了一个会聚子层和SAR子层。会聚子层检测丢失和误入的信元，平缓输入的数据速率从而以恒定的速度发送信元。最后，会聚子层将输入的报文或比特流分解为46字节或47字节的单元，然后交给SAR子层处理。在另一端（接收方）它取出这些数据单元，重组为原始的输入。AAL1的会聚子层没有自己的协议头信息。

相反，AAL1的SAR子层有自己的协议。其信元格式如下图。两种格式都是以1字节的头开始：其中包含3字节的信元序号SN（用于检测是否丢失或误入了信元）；该字段之后是3位的序号保护字段SNP（即校验和），可以改正信元序号字段中的单个错误并检测出现两个错误的情况。

当必须保留报文分界时使用P信元。指针(Pointer)字段用于给出下一段报文起始位置的偏移量。

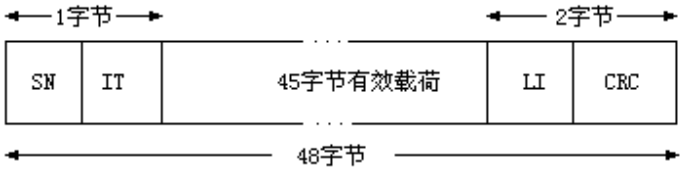
AAL2

AAL1是针对简单的、面向连接的、实时数据流而设计的，除了具有对丢失和误入信元的检测机制外，它没有错误检测功能。对于单纯的未经压缩的音频或视频数据，或者其中偶尔有一些较重要的位的其他任何数据流都没有什么问题，AAL 1就已经足够了。

对于压缩的音频或视频数据，数据传输速率随时间会有很大的变化。例如，很多压缩方案在传送视频数据时，先周期性地发送完整的视频数据，然后只发送相邻顺序帧之间的差别，最后再发送完整的一帧。当镜头

静止不动并且没有东西发生移动时，则差别帧很小。其次，必须要保留报文分界，以便能区分出下一个满帧的开始位置，甚至在出现丢失信元或坏数据时也是如此。由于这些原因，需要一种更完善的协议。AAL2就是针对这一目的而设计的。

像在AAL1中一样，AAL2的会聚子层没有本身协议而SAR子层有本身协议。SAR信元的格式如下图：



序号SN(Sequence Number)字段用于记录信元的编号以便检测信元丢失或误入。信息类型IT(Information Type)字段用于指明该信元是报文的开始、中间或末尾。长度指示LI(Length Indicator)字段指明有效载荷是多大，单位为字节（有效载荷可能小于45字节）。最后，CRC字段是整个信元的校验和，可以检测出错误。

标准中并没有注明各字段的大小。据说在标准化进程的最后关头，委员会成员觉得AAL2有许多问题，以致不能投入使用，但为时已晚，没有办法组织标准化的进程。最后委员们去掉了所有的字段大小的设定以使正式标准能够按时颁布，但这样便没有人能够实际使用它。

AAL3/4

开始时，ITU为服务类C和D制订了不同的协议（服务类C和D分别是对数据丢失或出错敏感，但不具有实时性的面向连接和非连接的数据传输服务类）。后来ITU发现没有必要指定两套协议，于是便将它们合二为一，形成了一个单独的协议，即AAL3/4。

AAL3/4可以按两种模式进行操作，即流和报文。在流模式中不保留报文分界信息。以下将集中讨论流模式。在每种模式中都可能出现可靠的传输和不可靠的（即不保证可靠性）的传输。

AAL3/4具有一个其他协议中没有的性能--支持多路复用。AAL3/4的这一功能允许来自于一台主机的多个会话（如远程登录）沿着同一条虚电路传输并在目的端分离出来。使用一条虚电路的所有会话得到相同质量的服务，因为这是由虚电路本身性质所决定的。

和AAL1、AAL2不一样，AAL3/4具有会聚子层协议和SAR子层协议。从应用程序到达会聚子层的报文最大可达65535字节。首先将其填充为4的整数倍字节。接着加上头和尾信息。在会聚子层对保温进行了重构，并加上了头和尾信息后，便将报文传送给SAR子层，由SAR子层将报文分为最大44字节的数据片。

AL3/4具有两层协议开销：每个报文需要增加8字节，每个信元增加4字节。总之，它是一种开销极大的机制，尤其是对短的报文。

AAL5

从AAL1到AAL3/4协议主要是由电信工业设计的并被ITU标准化，它没有太多地考虑计算机工业的要求。由于两个协议层所导致的复杂性及低效性，再加上校验和字段十分短（仅10位），使一些研究人员萌生了一个制订新的适配层协议的念头。该协议被称为简单有效的适配层SEAL(simple efficient adaptation layer)，经过论证，ATM论坛接受了SEAL，并为它起名叫AAL5。

AAL5向其应用程序提供了几种服务。一种选择是可靠服务（即采用流控机制来保证传输，以防过载）；另一种选择是不可靠服务（即不提供数据传输保证措施），通过选项使校验错的信元或者丢失或者传送给应用程序（但被标识为坏信元）。AAL5支持点到点方式和多点播送方式的传输，但多点播送方式未提供数据传输的保证措施。

像AAL3/4一样，AAL5支持报文模式和流模式。在报文模式中，应用程序可以将长度从1字节~65535字节的数据报传送到AAL层。当到达会聚子层时，将报文填充至有效载荷字段并加上尾部信息，选择填充数据（0字节~47字节），以使整个报文（包括填补的数据和尾部信息）为48字一节的整数倍。AAL5没有会聚子层头，只有一个8字节的尾。

用户到用户UU（User to User）字段不用于AAL层本身，而是为了自己的目的供更高一层（可能是会聚子层的特定服务子部分）使用，例如，排序或者多路复用。长度（Length）字段指出真正的有效载荷是多少，以字节为单位，不包括填充的字节数。0值用于终止未传送完毕的报文。CRC字段是基于整个报文的标准32位校验和，包括填充数据和尾部信息（CRC字段设置为0）。尾部的一个8位的字段留作将来使用。

报文交给SAR子层，然后发送出去。在SAR子层不增加任何头、尾信息，而是将报文分成48字节的单元，并将每个单元送到ATM层进行传输。它还通知ATM层将最后信元的PTI字段置为1，以便保留报文分界。（这时出现了一个问题：这是一种不正确的协议层混合体，因为AAL层不该使用ATM层的头部信息。）

AAL5较AAL3/4的主要优点是更加高效。虽然AAL3/4对每个报文只增加4字节的头信息，但它还要为每个信元增加4字节的头信息，因而使有效载荷的容量减少到44字节，对于长的报文，无效数据占8%。AAL5的每个报文有一个稍大的尾部（8字节），但每个信元无额外开销。信元中没有顺序号，可以通过长的校验和来弥补，从而可以检测丢失的、误插的或错误的信元，而不需要使用顺序号。

在因特网中，与ATM网接口的一般方法是使用AAL5的有效载荷字段来传输IP分组。与这种方法相关的各种问题在RFC 1483和RFC 1577中进行了讨论。

AAL协议的比较

各种AAL协议似乎不必要地相似，并且考虑得很不周到，把会聚子层和SAR子层区分开也是有疑问的，尤其是因为AAL5的SAR子层并无任何自己的特点。用稍微增强一些的ATM层头部信息来提供像排序、多路复用和数据分帧的功能便足够了。

AAL给人的整体印象是变体很多，变体之间存在很多细微的差别，而且尚未完工。原来的4个服务类A、B、C、D实际上已被废除。AAL1可能确实没有必要存在；AAL2不完整；AAL3和AAL4永无出头之日；AAL3/4效率低而且校验和字段位数太少。

将来的一切都依赖于AAL5，但到目前为止，AAL5尚有很多改进的余地。AAL5报文应该有一个顺序号和一位用于区分数据还是控制报文的标志位，从而可以成为一种可靠的传输协议。可以用尾部的未用空间来实现上述功能。

SSCOP--特定服务的面向连接协议

尽管有这么多不相同的AAL协议，但没有一种支持简单可靠的点到点的传输连接。需要这种服务的应用程序可以使用另一种协议--特定服务的面向连接协议SSCOP(service specific connection oriented protocol)。但是，SSCOP只是用于控制，不能用于数据传输。

SSCOP用户发送报文，每个报文都被赋予一个24位的顺序号。报文最大可达64KB，而且不能分开。它们必须按顺序传送。不像某些可靠的传输协议，它丢失报文时总是有选择性地进行重传而不是回到序号n，重传n以后的所有的报文。

SSCOP从根本上说是一种动态滑动窗口协议。对于每个连接，接收方保留准备接收报文序号的窗口，及标明该报文是否已经存在的位图(bitmap)。这个窗口在协议操作期间可以改变大小。

SSCOP的不寻常之处是对确认的处理方法：它没有捎带机制。取而代之的是发送方定期地查询接收方，要求它发送回表明窗口状态的位图。据此，发送方丢弃已被对方接收的报文并更新其窗口。

ATM网可分为三大部分：公用ATM网、专用ATM网和ATM接入网。

公用ATM网是由电信管理部门经营和管理的ATM网，它通过公用用户网络接口连接各专用ATM网和ATM终端。作为骨干网，公用ATM网应能保证与现有各种网络的互通，应能支持包括普通电话在内的各种现有业务，另外还必须有一整套维护、管理和记费等功能。目前还没有一个商用的公用ATM网，有关公用ATM网的协议也正在不断地完善之中。

专用ATM网是指一个单位或部门范围内的ATM网，由于它的网络规模比公用网要小，而且不需要记费等管理规程，因此专用ATM网是首先进入实用的ATM网络，新的ATM设备和技术也往往先在ATM专用网中使用。目前专用网主要用于局域网互连或直接构成ATM LAN，以在局域网上提供高质量的多媒体业务和高速数据传送。

接入ATM网主要指在各种接入网中使用ATM技术，传送ATM信元，如基于ATM的无源光纤网络(APON)、混合光纤同轴(HFC)、非对称数字环路(ADSL)以及利用ATM的无线接入技术等。

ATM主要接口

1、UNI(User-Network Interface)

UNI为ATM网中的用户网络接口，它是用户设备与网络之间的接口，直接面向用户。UNI接口定义了物理传输线路的接口标准，即用户可以通过怎样的物理线路和接口与ATM网相连，还定义了ATM层标准、UNI信令、OAM功能和管理功能等。按UNI接口所在的位置不同，又可分为公用网的UNI和专用网的UNI(PUNI)，这两种UNI接口的定义基本上是相同的，只是PUNI由于不必象公网的接口那样过多地考虑严格的一致性，所以PUNI的接口形式更多、更灵活、发展也更快一些。

2、NNI(Network to Network/Network Node Interface)

NNI可理解为网络节点接口或网络/网络之间的接口，一般为两个交换机之间的接口，与UNI一样，NNI接口也定义了物理层、ATM层等各层的规范，以及信令等功能，但由于NNI接口关系到连接在网络中的路由选择问题，所以特别对路由选择方法做了说明。同样，NNI接口也分为公网NNI和专用网中的NNI(PNNI)，公网NNI和PNNI的差别还是相当大的，如公网NNI的信令为3、7号信令体系的宽带ISDN用户部分B-ISUP，而PNNI则完全基于UNI接口，仍采用UNI的信令结构。

3、B-ICI(BISDN Inter-Carrier Interface)

B-ICI定义为两个公用ATM网之间的接口，为分别属于两个运营者的UNI接口提供了连接，它的定义基于NNI接口，其特点是支持不同网络间的多种业务传送，包括基于信元的PVC方式业务、PVC方式的帧中继业务、电路仿真业务、SMDS以及SVC业务等。

4、DXI(Data Exchange Interface)

DXI定义在数字终端设备DTE和数字连接设备DCE之间，DTE通过DXI与DCE相连，再通过ATM UNI接口接入ATM网中，DCE完成了不符合ATM标准的数据终端到ATM的适配过程，相当于终端适配器。

5、FUNI(Frame Based UNI Interface)

FUNI的意义与DXI相似，FUNI将ATM适配功能完全移入了交换机内部，终端和ATM交换机之间传送FUNI帧，所以与基于信元的DXI接口相比，FUNI在接入线上有更高的效率。

网络，但组播服务器可能最终成为瓶颈。

第二种方法称为组播网，该组中每个节点与其它节点建立点到多点连接。这样，所有的节点都可以向其它节点发送和从它们接收数据。对于一个含N个节点的组来说，将需要N个点到多点连接，不适于含节点数目很多的组。

这两种方法都用于Armitage建议的组播地址解析服务器(MARS)。MARS服务于一簇节点，一簇中所有的端系统配置以MARS的ATM地址。当一个端系统想向特定的组播群发信息时，它建立与MARS的连接，发出MARS_REQUEST

信息，MARS返回MARS_MULTI信息，此信息含有该组的组播服务器的地址或组成员的地址，如果该组支持组播服务器，请求节点就建立与该服务器的连接，将数据发送给该服务器，由该服务器将数据转发给组中的节点；在组播网方案中，请求节点与组中的节点建立点到多点连接并通过该连接发送数据。

MPOA

1、MPOA的原则

MPOA的目的是在LANE环境中有效地传输子网间的unicast数据。MPOA集成了LANE和NHRP以保留LANE，同时通过旁路路由器提高子网间通信的效率。MPOA允许网络层路由记算和数据传送物理地分离，这称为虚拟路由。路由计算由位于路由器中的服务器--即MPS--执行，数据传送由边缘设备中的客户--即MPC--执行。

在入口点，MPC检测通过ELAN传送给含有MPS的路由器的数据流，当它发现能够旁路当前路由路径的捷径时，它使用基于NHRP的协议请求与目的节点建立捷径，如果可行，该MPC在其入口表中记录下该信息，建立捷径VCC，通过该捷径VCC发送帧。对于使用捷径的分组，MPC从分组中去掉数据链路层(DLL)封装。

在出口点，MPC从其它MPC接收网络数据，对于通过捷径接收到的帧，该MPC加上适当的DLL封装把它们传送给上层协议。该DLL封装信息由MPS提供并存贮在出口缓存中。

MPS是路由器的逻辑成分，给MPC提供网络层转发信息，它包含NHRP中定义的完整的NHS。MPS与本地NHS和路由功能交互以回答入口MPC的MPOA请求，并给出口MPC提供DLL封装信息。

下面是ELAN内和ELAN间通信过程的简单描述。

ELAN内通信从一个MPOA主机或LAN主机到同一ELAN的另一MPOA主机或LAN主机，这些数据流使用ELAN做地址解析和数据传输。ELAN间通信从一个MPOA主机或LAN主机到不同ELAN的MPOA主机或LAN主机，短数据流使用缺省的路径，长数据流使用捷径，缺省的路径利用ELAN和路由器，捷径使用LANE和NHRP做地址解析和捷径。捷径是这样工作的：如果源节点和目的节点不在同一个MPS的管理域，入口MPS将MPOA解析请求翻译成NHRP解析请求，通过NHRP将该请求转发给出口MPS，当出口MPS收到出口MPC的回应后，它生成NHRP解析回应并把它发回给入口MPS，当入口MPC得到入口MPS的MPOA解析回应后，它与出口MPC之间就可以建立捷径了。

2、MPOA的优点和限制

MPOA从根本上将数据传送和路由计算分开，将功能分布到不同的设备，从而减少了参与路由计算的设备数目和端设备的复杂性。它可以以统一的方式支持二层和三层网络互连，因此保证了ATM环境中大规模的互连。它可以同时有效地处理突发数据和长期的数据流，但是，MPOA的复杂性有很大的争议。

IP交换

IP交换的目的是在快速交换硬件上获得最有效的IP实现，将非连接的IP和面向连接的ATM的优点互补。IP交换是标准的ATM交换加上连接于ATM交换机端口上的智能的软件控制器，即IP交换控制器。IP交换机将数据流的初始分组交给标准的路由模块(IP交换机的一部分)处理，当IP交换机看到一个流中足够的分组，认为它是长期的，就同相邻的IP交换机或边缘设备建立流标记，后续的分组就可以高速地标记交换，将缓慢的路由模块旁路。特别的IP交换网关或边缘设备负责从非标记分组向标记分组和分组到ATM数据的转换。

每个将现有网络设备连到IP交换机的IP交换网关或边缘设备在启动时建立一个到IP交换控制器的虚信道作为缺省的转发信道，从现有网络设备接收到分组时，边缘设备通过缺省转发信道将分组传送给IP交换控制器。

IP交换控制器执行传统的路由协议，如RIP、OSPF和BGP，将分组以正常的方式通过缺省转发信道转发给下一个节点，这可能是另一个IP交换机或边缘设备。IP交换控制器还执行数据流分类，它识别长期的数据流，因为这样的数据可以用ATM硬件的cut-through交换来优化，其余的通信仍然使用缺省的方式，即点到点的存贮转发路由。

当长期的数据流被识别，IP交换控制器要求上一节给之打标记，使用新的虚信道，如果源边缘设备同意，该数据流就通过新的虚信道流向IP交换控制器。下一节点也执行同一动作。当该流独立使用特殊的输入信道和

输出信道，IP交换控制器指示交换机建立适当的硬件端口映射，旁路路由软件和相关的处理开支。这个过程继续下去，该流的前面几个分组使从源边缘设备到目的边缘设备建立直接的连接。此设计使IP交换机以仅受交换引擎限制的速率转发分组。第一代IP交换机支持高达每秒5.3M分组的吞吐量。此外，因为不需要将ATM信元封装到中介IP交换机的IP分组中，IP网中的吞吐量也得到了优化。

Ipsilon给IETF提出了两种协议。通用交换管理协议(GSMP, RFC1987)允许IP交换机控制器访问交换机硬件并动态转变交换模式：存贮转发或cut-through。Ipsilon流量管理协议(IFMP, RFC1953)用于在边缘设备和IP交换控制器间交换控制信息并将IP流与ATM虚信道联系起来。

IP交换的一个重要特性是流的分类和交换在本地执行，而不是基于端到端的基础上，这保留了IP的非连接本质，并允许IP交换机绕过失效节点路由而不需要从源主机重新建立通道。

此外，流分类使IP交换同样有效地支持长期和突发数据。

然而，IP交换是基于流的，在大型网络中其伸缩性是值得质疑的，在很大的网络中流的数目可能最终超过可用的虚通道数。

有五家公司正式宣称支持Ipsilon的IP交换，它们是：Ericsson、General Datacomm、Hitachi America Ltd. 、NEC America Inc. 和DEC Ipsilon。它们试图使此技术成为事实上的标准--MPLS。

标记交换

另一个选择是Cisco公司的标记交换。标记交换网络包含三个成分：标记边缘路由器、标记交换机和标记分发协议。标记边缘路由器位于标记交换网络边缘的含完整3层功能的路由设备，它们检查到来的分组，在转发给标记交换网络前打上适当的标记，当分组退出标记交换网络时删去该标记。作为具有完整功能的路由器，标记边缘路由器也可应用增值的3层服务，如安全、记费和QoS分类。标记边缘路由器的能力不需要特别的硬件，它作为Cisco软件的一个附加特性来实现，原有的路由器可以通过软件升级具有标记边缘路由器的功能。

标记交换机是标记交换网络的核心。所谓标记是短的、固定长度的标签，使标记交换机能用快速的硬件技术来做简单快速的表查询和分组转发。标记可以位于ATM信元的VCI域、IPv6的flow label域或在2层和3层头信息之间，这使得标记交换可用于广泛的介质之上，包括ATM连接、以太网等。

标记分发协议提供了标记交换机和其它标记交换机或标记边缘路由器交换标记信息的方法。标记边缘路由器和标记交换机用标准的路由协议(如BGP、OSPF)建立它们的路由数据库。相邻的标记交换机和边缘路由器通过标记分发协议彼此分发存贮在标记信息库(TIB)中的标记值。

下面是标记交换网络的基本处理过程。

(1)标记边缘路由器和标记交换机用标准的路由协议识别路由，它们完全可以与非标记交换的路由器互操作。

(2)标记边缘路由器和交换机通过标记分发协议给用标准路由协议生成的路由表赋以标记信息并分发，标记边缘路由器接收标记分发协议信息并建立转发数据库。

(3)当标记边缘路由器收到需要通过标记交换网络转发的分组，它分析其网络层头信息，执行可用的网络层服务，从其路由表中给该分组选择路由，打上标记然后转发到下一节点的标记交换机。

(4)标记交换机收到带标记的分组，仅基于标记来进行交换，而不分析网络层头信息。

(5)分组到达出口点的标记边缘路由器，标记被剥除，然后继续转发。

在标记交换网络中，标记分发协议和标准路由协议可以用目标前缀标记算法集合起来，此算法可以在数据流穿过网络前在TIB中建立标记信息。这有两个意义。一个是流中的所有分组都可以被标记交换，即使是突发短数据也是如此；此外它是基于拓扑的，在每个源/目的分配一个标签。而在IP交换中只有长期数据流在一定数目的分组经过后才建立捷径。因此，标记交换比基于流的机制更有效地使用标签，避免了一个一个流的建立过程，这使之具有了公共因特网服务网络所需要的很好的伸缩性，在公共因特网中，流的数目是巨大的，其改变

速率也是很高的。

其他厂商也有类似的机制，如Cabletron的SFVN(Secure Fast Virtual Networking)、Cascade的IP Navigator、DEC的IP packet switching、Frame Relay Technologies的Framenet Virtual WAN switching和IBM的ARIS(Aggregate Route-based IP Switching)等。

Frame-Relay

Basic Introduction

frame relay概述（面向连接服务），

FR遵循ANSI T1.617、ITUTQ.933（3层）、Q.922（2层）规范。

它是一条虚电路。通过用**frame relay map** 语句来影射一个dlci号和一个IP地址。

或者用动态的方式，既**inverse arp**（有些象LAN的ARP解析）。

要注意dlci号只于本地有效。使用DLCI来标识CPE和FRSW之间的VC。

关于LMI

CISCO支持的有3种：

ansi t1.617。

ITU-T的q.933a。

cisco。默认是CISCO。

可以用命令**frame-relay lmi q933a**来修改LMI的类型，这里说明了LMI为Q933A的类型。

有关于LMI的状态有三种：

active(0x2)，说明是LMI是正常活跃的。

Inactive(0x0) 说明对端的ROUTER或者中心交换机到对端的ROUTER之间的链路有问题。表明到FR交换机的本地连接是可以工作的但是远端路有器到FR交换机的连接 不能工作。

DELETED(0x4)。说明在中心交换机上根本就没有配置这个DLCI号。表明路有器没有从FR交换机受到任何的LMI 或在CPE路有器和FR交换机之间没有任何业务。

LMI是用于CPE和FR交换机之间的一种信令标准，负责管理设备之间的连接以及维护连接的状态。支持存活机制，多播机制，全局编址和状态机制。从CISCO IOS11.2起LMI是可以配置的，但会尝试自动检测FRSW使用的LMI类型，发送一个或多个Full-Status请求来实现的。FRSW返回一种或多种LMI类型，Router将自己配置为最后收到的LMI类型。FR的帧由header and address area、frame check sequence、user-data portion组成。

FR是包交换网络，比电路交换更有效，带宽是共享的，成本比租用线路低。

LMI extensions are considered to be optional包括Multicasting（Providing network server with its local DLCI）、Simple flow control、Global addressing。也就是支持存活机制（检查数据传输是否顺畅）、多播机制（给DTE提供本地DLCI）、全局编址（赋予DLCI以全局意义）和状态机制（提供交换机所知的DLCI的当前状态）。

2：配置frame relay

首先在端口上要封装frame-relay的类型和IP地址（在同一网段）和BW

frame-relay封装的类型基本有两种cisco（默认）和IETF

默认情况下其他都不用设置DLCI号可以动态自动学习（IARP），命令要加广播。

配置静态的影射时候可以关掉INARP或者不关都可以。静态MAP时将阻止Inverse ARP解析。意味着分支路由器无法到达中心点和另一个分支。如果对方为非CISCO设备的时候使用ietf。命令如下：

frame-relay map ip 10.0.0.1 110 broadcast (ietf) ，IP是对方的IP，DLCI号是本地路由器到SW的DLCI号码。

动态学习到的它是自己自动的加上参数broadcast。加上这个参数可以在VC上跑多播和广播用来传输路由协议，

强调：DLCI号只具有本地意义（本地R到FRSW）。导致分支站点无法与其它分支站点连通的问题是IARP。

有几个VC一般就有几个本地意义上的DLCI号。但是他们走的是同一个物理链路~

hub-and-spoke的拓扑：没有全互连 VC情况：

spoke ROUTER：如果要传输数据给另一个 spoke ROUTER。那么DLCI号是同样的。~让HUB ROUTER来传。

Frame-relay interface-dlci dlci 指定本地DLCI值。

3：检验 frame-relay的配置

show inter s0 （显示出Line、Protocol、DLCI、LMI信息）

不同的LMI会有不同的DLCI号。

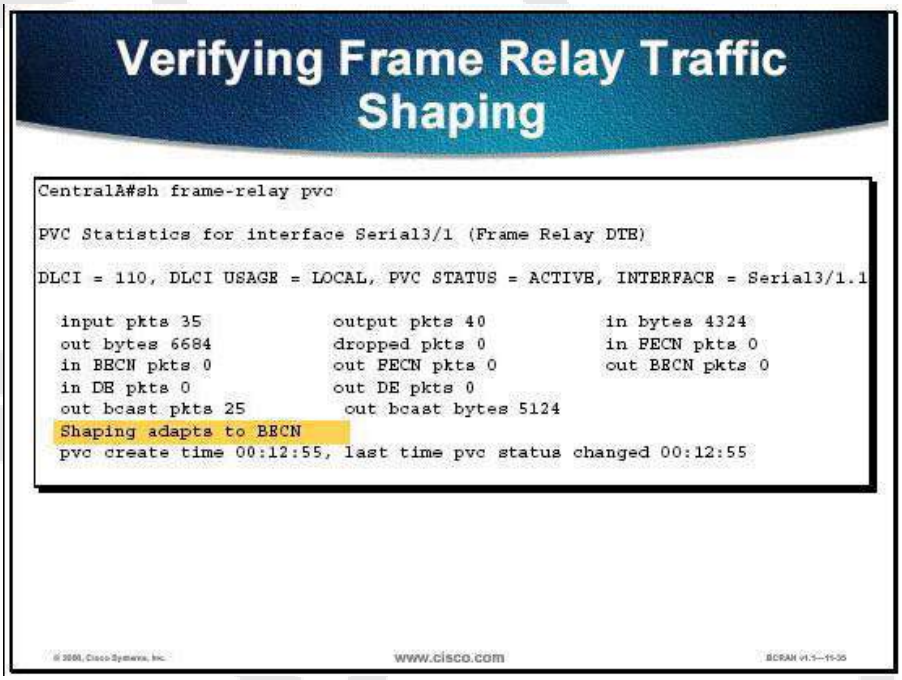
show frame-relay pvc 显示PVC状态和数据流统计信息，B(F)ECN信息、PVC的状态、LMI状态、FRTS信息。

Show frame-relay lmi和**Debug frame-relay lmi** 显示有关LMI信息，如本地Router和FRSW之间的交换状态消息等（0x0=inactive、0x2=active、0x4=delete）。

（out/in）status前者表示路由器发送的LMI状态请求，后者是FRSW的应答。Type(0/1)后者显示的是给FRSW的存活消息每10s一次，前者显示的是FR全状态消息，每60s发一次，这状态表示有活动的PVC连接，Router将发送IARP请求包。重新启用IARP，使用命令：**frame-relay inverse-arp**

show frame-relay map 显示远程协议地址、DLCI、静（动）态MAP和PVC状态。

注意:命令**clear frame-relay-inarp**可以清除动态学习到的影射关系。



frame-relay的子端口：

主要考虑路由更新的问题~

如果没有子端口可能会因为水平分割使路由更新不可能正确完成。

点到点的子端口在路由更新上是没有问题的！当然，每个子端口应该在一个独立的网段里。（虽然浪费IP）

多点子端口跟一个物理端口的情况是一样的，所有参与的接口都位于同一子网中。子接口配置IP物理接口不能在配置IP。点对点需独立网段，多点用于全互连（子接口缺省NBMA网络，节省IP，同网段）。

默认情况下FR接口上对IP 协议禁用了水平分割，但EIGRP例外。NBMA环境中存在水平分割问题子接口可以解决水平分割的问题，子接口类似将一个物理接口模拟成多个逻辑接口。

配置子端口：如果要改子端口的类型，必须先删除这个子端口，然后重新启动路有器，然后再指定同样那个子端口的类型。

在物理端口上不设置IP。并且封装frame-relay. 在子端口上配置IP。和dlci号frame-relay dlic 1111(这个配置在子端口里是必须的)

5：frame-relay 的流量整形traffic shaping

要了解4个名词：FECN。BCEN。BC承诺突发量。BE超额突发量，CIR承诺速率。CIR是一个承诺的速率。BC是突发量，不是速率。access rate是用户到ISP之间的物理线路上的速度。FRTS应用于VC。

FECN (C/R=0) and BECN (C/R=1)

T=BC/CIR、BC=T*CIR、CIR=T*BC

CIR：FRSW准许的数据传输速率。BE：FRSW试图传输的CIR之外的非承诺比特的最大数。BC：在任何承诺速率计算间隔（TC）内交换机准许（同意）传输的最大比特数。DE对过量数据流（超过CIR速率后接收到的，也就是BE）其DE位被设置。DE可丢弃指示（表示帧比其它帧重要程度低,DE=1说明正常，=0时说明数据量超过CIR）。

为什么采用流量整形。？如果ISP之间两段ROUTER的介入速度不一样那么在传输数据的时候在ISP出会产生瓶颈和丢包。可以采用BECN来通知发送方拥挤的信号（默认是禁用的）。FECN是告诉接受者拥挤的信号。

Average（平均） and peak（突发） rate（速率）

6：配置 FRAME-RELAY的流量整形（FRTS）

1， 建立一个MAP-CLASS，

2， 2定义流量整形的方法，

3， 3在接口上封装FRAME-RELAY，

4， 4在端口上应用MAP-CLASS

5， 5，开启流量整形，一般用于源端（发送端）。

4种情况下使用FRTS：

1)中心高速，分支低速；

2)单条物理线路承载到不同目的地的众多VC时；

3)若FR发生了拥塞，想让路由器将数据流拦住（Throttle）；

4)需要在同一条FR的VC上传输多种协议（IP、SNA、IPX）的数据流，并希望每种类型的数据流都能占用一定的BW。

配置例子：

interface s0

ip add 1.1.1.1

1) 全局模式下：**map-class frame-relay asdf** 定义一个帧中继的CLASS 名字为ASDF，区分大小写。

2) 映射类模式下：**frame-relay traffic-rate 32000 64000** 定义流量整形的方法，设置平均速度和最高速率分别32K和64K。（要让路由器根据BECN量动态调整其发送速率，

让流量整形使用BECN机制，使用命令：**frame-relay adaptive-shaping becn**。

要使用排队机制（不建议使用）可使用CQ或PQ，命令：**Frame-relay custom-queue-list number** 和**frame-relay priority-group number**。）在每个映射类中，只能使用一种排队机制，修改的话前面先加No。

3) 接口模式下：**frame-relay class kaka** 应用一个帧中继的CLASS ASDF，同接口上的所有VC关联起来。

4) 接口模式下：**encap frame-relay** 封装帧中继

5) 接口模式下：**frame-relay traffic-shaping** 开启帧中继流量整形。

用 **show frame-relay pvc** 来检验配置。**Show traffic-shape statistics**和**show traffic-shape**实例：

IPv6

Introduction

Pv6 早期被称为 IPng (next generation)目前是 IP 协议的最新版本。 IP 协议是一种网络层协议，采用 IP 协议构建的数据通信网络可提供高效的数据、语音和图像的传输服务。目前，在 Internet 上广泛采用的 IP 协议是 IPv4 版。随着 Internet 的迅猛发展，在充分享用了 IPv4 协议的简单高效的同时，人们也就意识到了 IPv4 的 32 位地址空间是不够的；因此，必须建立新的 IP 标准。

海量 IPv6 地址空间

IPv6 最根本的改变是提供了未来对全球范围内可确定的地址空间的需求。基于移动设备的应用，如：个人数字设备(PDAs)，移动电话,汽车，家庭网络和其他的移动数据通信设备都需要全球范围内可确定的地址。IPv6 将网络地址位数从 32 位扩展到 128 位，这代表着可以为地球上的任何需要联网的设备提供唯一确定的地址。正是因为有了全球范围内可确定的地址，IPv6 提供了全球范围内的地址可达，端到端的安全通信，以及对所有对地址有要求的应用和服务的支持。 除此之外，丰富的 IPv6 地址空间消除了网络中的 NAT(Network Address Translation)瓶颈，提高了网络效率。

IPv6 地址格式

由于 IPv6 的地址有 128 位长，比 32 位的 IPv4 地址表示起来复杂的多。目前，IPv6 的地址由一串 16 进制的数字表示，每 16 位之间用分号(:)隔开，格式如下： x:x:x:x:x:x:x:x.

下面是两个 IPv6 地址例子：

2001:0DB8:7654:3210:FEDC:BA98:7654:3210

1080:0:0:0:8:800:200C:417A

通常情况下 IPv6 地址会包含连续的 0。为避免 IPv6 地址表示的复杂性,连续的 0 可缩写为两个冒号表示(::)。Table1 lists IPv6 地址的缩写格式。 两个冒号可以用在 IPv6 地址的一部分。你可以在一个接口上配置多个地址，但只能一个 link-local 地址。

注意两个冒号(::) 只能在 IPv6 地址中出现一次，只能代表最长的连续的 0。

IPv6 地址中的十六进制数的表示是大小写不敏感的。

Table 1 压缩 IPv6 地址格式

IPv6 Address Type	Preferred Format	Compressed Format
Unicast	1080:0:0:0:8:800:200C:417A	1080::8:800:200C:417A
Multicast	FF01:0:0:0:0:0:0:101	FF01::101
Loopback	0:0:0:0:0:0:0:1	::1
Unspecified	0:0:0:0:0:0:0:0	::

Table1 中所列的 loopback 地址可以是一个网络节点送一个 IPv6 的数据包给自己。 这个 loopback 地址功能与 IPv4 的 loopback 地址 (127.0.0.1)一样.

注意 IPv6 的 loopback 地址不能配置在物理接口。 一个不管是原地址还是目的地址是 IPv6 loopback 地址的数据包必须停留在产生他的网络节点内，不能出现在网络链路上。IPv6 路由器不能转发代有 IPv6 loopback 地址的数据包，不管是原地址还是目的地址。

在 Table1 中的未定义的地址是不存在的 IPv6 地址.例如，一个刚刚进入网络的，正在初始化的节点可以用未定义的 IPv6 地址作为源地址，直到获得真正的 IPv6 地址为止.

注意未定义的 IPv6 地址不可以用作接口地址。未定义的 IPv6 地址不能用作在 IPv6 数据包的目的地址或出现在 IPv6 数据包头的路由信息内。

一个用 ipv6-prefix/prefix-length 的格式的 IPv6 地址前缀可表示一个连续的地址空间。其中，ipv6-prefix 部分必须是在 RFC2373 正式定义的以分号隔开的连续 16 进制数。前缀的长度是一个十进制的数值，他表示前多少位是网络 地址的前缀(网络标示)。 例如, 1080:6809:8086:6502::/64 是一个合法的 IPv6 前缀.

IPv6 地址类型 ：

单播 Unicast 地址

一个 IPv6 单播地址是用来标示一个网络节点的一个接口。一个数据包的目的地址是一个单播地址时，他将被送到以这个地址标示的网络接口。一个 IPv6 的路由器应支持下列单播地址类型:

Global aggregatable address:可汇聚的全球化地址。

Site-local address:区域内的地址 (proposal to remove by IETF)

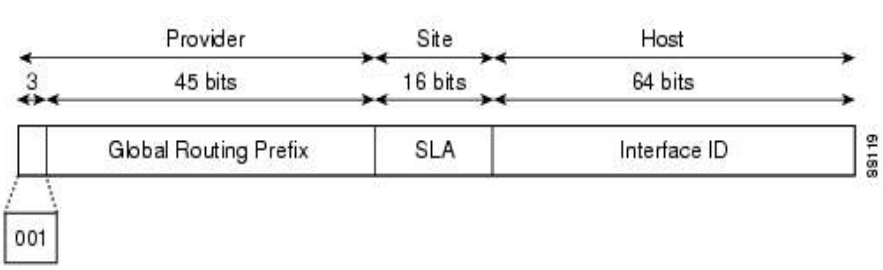
Link-local address: 链路接口地址

IPv4-compatible IPv6 address: 兼容 IPv4 的地址

可汇聚的全球地址

一个可汇聚的全球地址是一个带有全球化的可汇聚的 IPv6 前缀的地址。 这种结构化的全球化的单播地址可以进行严格的路由前缀的汇聚,这样就可以大大减少整个路由表的路由条数,减少路由的复杂度。可汇聚的全球 地址用来标示链路地址，可以进行多级的路由汇聚，一直到顶级的 Internet 服务提供商(ISPs).全球化的可汇聚的 IPv6 地址被定义为全球化的路由前缀, 子网络号,和接口地址。除了以二进制 000 开始的地址, 所有的全球化的 IPv6 单播地址都带有 64 位的接口地址。目前， 全球化的单播地址的分配范围从以二进制的 001 (2000::/3)开始。Figure1 表示了全球化的可汇聚的单播地的格式 。

Figure 1 全球化的可汇聚地址的格式



带有前缀从 2000::/3 (001) 到 E000::/3 (111)的地址需要带有从(EUI)-64 格式的 64 位接口地址。IANA (Internet Assigned Numbers Authority)将 2000::/16 IPv6 地址空间下发到各个区域的地址注册机构分发,如 APNIC 等。

在通常情况下,全球化的可汇聚的 IPv6 地址包含一个 48 位的全球路由前缀和 16 位的子网络号或称为区域级的可汇聚前缀(Site-Level Aggregator-- SLA)。在 IPv6 的全球化的可汇聚单播地址格式文件(RFC 2374)中指出， 全球化的路由前缀包括两层的层次化结构：顶层汇聚(Top-Level Aggregator--TLA) 和第二层汇聚(Next-Level Aggregator -- NLA)。现在，IETF 决定从 RFCs 把 TLS 和 NLA 的区域规定删除，这样可以使路由汇聚有更多的灵活性。在一些已经采用了 RFC 2374 中的汇聚方式的网络可继续沿用原有的方式。 一个 16 位的子网络标示被称作子网号，他可以被不同的组织机构用来区分自己内部的地址分配。一个子网号与 IPv4 的子网非常类似, 知识他可以有多达 65,535 个子网。

一个接口地址用来标示链路的接口。接口地址在一个链路上必须是唯一的 ,他们可以是 64 位接口地址中的任何一个，只要确保是唯一的。在多数情况下，接口地址可以是从链路层的地址中衍生而来。在全球化的可汇聚的 IPv6 地址中，接口地址必须是 64 位长，并且采用修改的 EUI-64 的格式.

修改的 EUI-64 的格式的接口地址可从下列的产生方法之一产生：

所有的 IEEE 802 的接口类型(如：以太网，FDDI 等)，前三个八进制数(24bits)从 MAC 地址中的 OUI (Organizationally Unique Identifier)复制；第四和第五个八进制数为 FFFE，最后的三个八进制数(24bits)复制 MAC 地址中的后 24 位。 最后，在构成的 64 位接口地址的第一个八进制数的第七位(U/L) bit 表示该接口地址是本地化的还是本地的：0 表示本地地址，1 表示全球唯一的接口地址。

对其他的接口类型(如：串行接口，loopback，ATM，Frame Relay，和 tunnel 接口—除 IPv6 overlay tunnel 的接口)，接口地址的构成采用 IEEE 802 接口类型相似的方法：MAC 地址来自于设备的 MAC 地址池中的第一个 MAC 地址。 当接口类型是 IPv6 overlay tunnels 时，接口地址是低 32 位的 IPv4 地址加上高位的全 0。

注意当接口类型是 PPP 时，两端的不同接口可能具有相同的 MAC 地址；这时，两端设备需要重新协商，重新选择接口地址，直到他们是唯一的。网络设备的第一个 MAC 地址用来构建 PPP 接口地址。

当网络设备不具有 IEEE802 类型接口， link-local 的 IPv6 地址按以下方式产生：

1.路由器请求 MAC 地址 (从路由器本身的 MAC 地址池)。

2.如果没有 MAC 地址可用，路由器的序列号可用做 link-local 地址。

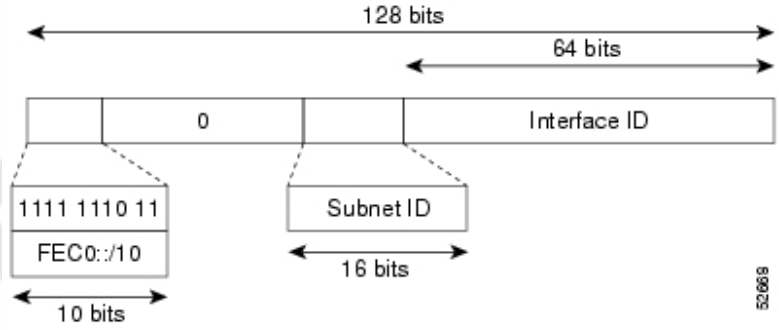
3.如果序列号不可用，路由器将采用 MD5 算法结合自己的名字创建 MAC 地址。

Site-Local 地址

一个 site-local 地址是一个带着前缀 FEC0::/10 和 16 位子网号加上 64 位的 EUI-64 格式的 IPv6 单播地址。Site-local 地址非常像 IPv4 中的私有地址空间 10.0.0.0/8；可以在一个区域内部进行路由的查找，而不需要全球化的唯一的地址前缀。 Site-local 地址可以被认为是私有地址，因为他只能在一个严格限制的区域内使用。Figure2 表示了 site-local 地址。

IPv6 路由器不可以将带有 site-local 的源地址或目的地址的数据包转发到区域之外。

Figure 2 Site-Local 地址 Format

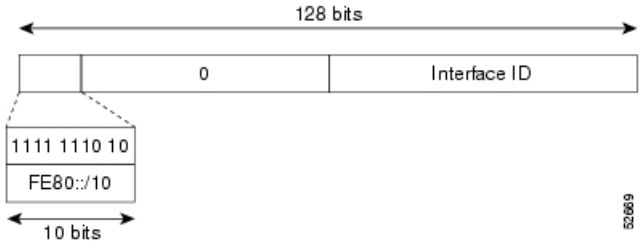


Link-Local 地址

一个 link-local 地址是一个带有 link-local 前缀 FE80::/10(1111 1110 10)的 IPv6 单播地址，他采用修改的 EUI-64 地址格式自动产生。 Link-local 地址可用于邻居发现协议（neighbor discovery protocol）和无状态自动配置进程。 在同一个链路上的节点可使用 link-local 地址来通信； 不需要 site-local 地址或全球化的单播地址。 Figure3 表示 link-local 地址的结构。

IPv6 路由器不能将带有 link-local 的源地址或目的地址的数据包转发到其他网络或链路。

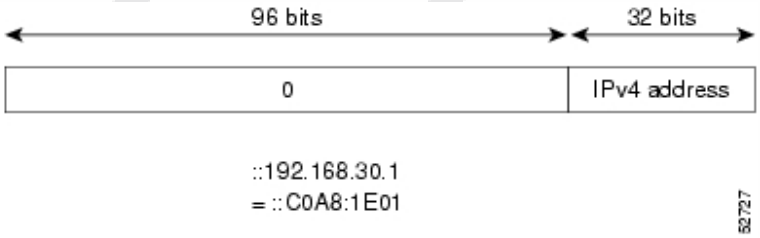
Figure 3 Link-Local 地址格式



与 IPv4 兼容的 IPv6 地址

一个 IPv4 兼容的 IPv6 地址是一个高 96 位全 0 的，低 32 位为 IPv4 地址的 IPv6 单播地址。这种 IPv4 兼容的 IPv6 地址可表示为 0:0:0:0:0:0:A.B.C.D 或::A.B.C.D. 整个 128 位的 IPv4 兼容的 IPv6 地址是将网络节点的 IPv4 地址直接植入 IPv6 地址的低 32 位。IPv4 兼容的 IPv6 地址用来在运行 IPv4 和 IPv6 双栈时使用自动的隧道技术（tunnels）。 Figure4 表示了 IPv4 兼容的 IPv6 地址 格式。

Figure 4 IPv4 兼容的 IPv6 地址格式



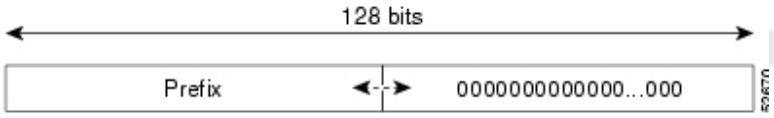
IPv6 地址类型：Anycast 地址

一个 anycast 地址是一个属于多个网络节点的接口的 IPv6 的地址。一个以 anycast 地址为目的地址的数据包将被转发到最近的网络接口；接口的远近由路由的计算结果来确定。Anycast 地址从地址的结构上来说同单播地址是不可分的，因为 anycast 地址存在于单播地址的地址空间。但一个单播地址可以被超过一个网络接口接收时，他就是一个 anycast 地址。Anycast 地址需要在网络节点上明确定义：这个地址是一个 anycast 地址。

注意 Anycast 地址只能被路由器使用，不能被主机使用。Anycast 地址不可以出现在 IPv6 数据包的源地址。

Figure5 表示了 anycast 地址的构成；Anycast 地址是由一个前缀跟上一连串的 0(接口地址)来构成。

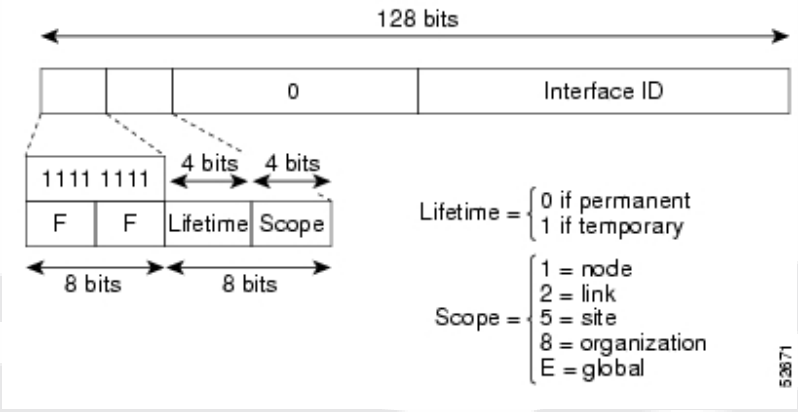
Figure 5 Subnet Router Anycast 地址格式



IPv6 地址类型：组播地址 Multicast

一个前缀 FF00::/8(1111 1111)的 IPv6 地址就代表它是一个 IPv6 组播地址。一个数据包的目的地址为组播地址时，他将被复制转发到多个网络的接收者。组播地址前缀的第二个八进制数代表组播地址的范围。这个八进制数被分为两部分：前四位为 0000 代表是一个永久性的组播地址，前四位为 0001 代表是暂时的组播 地址；后四位代表组播范围，目前定义了 1, 2, 5, 8, or E，分别代表不同的组播范围。如 Figure6 所示。比如：组播地址的前缀为 FF02::/16 代表是一个永久性的组播地址，在链路范围。

Figure 6 IPv6 组播地址格式



IPv6 的设备(主机和路由器)必须加入下列组播组(必须接收的相应的组播数据包):

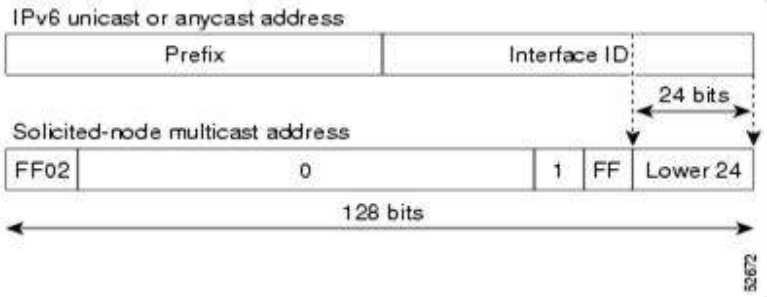
All-nodes 所有节点的组播组 FF02:0:0:0:0:0:0:1 (link-local 范围)

Solicited-node 节点请求组播组 FF02:0:0:0:0:1:FF00:0000/104 用来对应单播地址和 anycast 地址 。

IPv6 路由器必须加入所有路由器组播组 FF02:0:0:0:0:0:0:2 (link-local 范围).

节点请求的组播组地址（ solicited-node）是用来确定 IPv6 的单播地址或 anycast 地址的. IPv6 节点设备必须加入为地址和 anycast 地址对应的组播组。 当网络节点的 IPv6 单播或 anycast 地址的低 24 位再加上 IPv6 节点请求组播地址前缀 FF02:0:0:0:0:1:FF00:0000/104 就构成了这个节点的节点请求组播地址。 如 ： Figure7 。比如：当一个节点的 IPv6 地址为 2037::01:800:200E:8C6C 时，他的节点请求组播地址是 FF02::1:FF0E:8C6C。节点请求地址用来接收邻居节点的请求信息。

Figure 7 IPv6 节点请求组播地址格式



IPv6 包头格式

IPv6 包头经过改进，效率大为提高（如图所示）。 新的格式引入了扩展包头的概念，使支持可选项功能的灵活性有所增强。

IPv6 包头中的字段包括：

- Version（版本）：4 位因特网协议(Internet Protocol)版本号，值=6。
- Traffic Class（业务负载类别）：8 位业务负载类别字段，类似于 IPv4 的服务类型。
- Flow Label（流标号）：20 位流标号，用于确定服务质量方面附加控制的业务流。
- Payload Length（有效负载长度）：16 位无符号整数，IPv6 有效负载的长度。
- Next Header（下一包头）：8 位选择器，用于识别紧随 IPv6 包头之后的包头类型。
- Hop Limit（跳转限度）：8 位无符号整数，根据转发数据包的每个节点按 1 递减。 如果 Hop Limit 减至零，则数据包被丢弃。
- Source Address（源地址）：数据包始发方的 128 位地址。

• Destination Address（目标地址）：数据包预期接收方的 128 位地址。

IPv4 header

version	IHL	type of service	total length
identification		flags	fragment offset
time to live	protocol	header checksum	
source address			
destination address			
options			padding

IPv6 header

version	traffic class	flow label	
payload length		next header	hop limit
source address			
destination address			

IPv6 扩展包头

扩展包头在 IPv6 中为可选项。如果存在，扩展包头则紧随包头字段。IPv6 扩展包头具有以下特性：

• 它们按 64 位排列，其系统开销远远低于 IPv4 选项。

• 不像 IPv4 那样有大小限制。唯一的限制就是 IPv6 数据包的大小。

• 它们仅由目的节点处理。唯一的例外就是 Hop-by- Hop（逐段跳转）包头选项。

• 基本 IPv6 包头的 Next Header（下一包头）字段识别扩展包头。

当同一 IPv6 数据包内存在多个扩展包头时，其发生顺序如下：

• 逐跳(Hop-by-Hop)包头携带需由发送路径上的所有节点检验的信息。当逐跳选项存在时，则其始终紧 随基本 IPv6 包头之后。

• 目的(Destination)包头携带仅能由目的节点检验的附加信息。

• 选路(Routing)包头由源节点使用，以列出数据包通过路径到达其目的地所需的所有节点。

• 分段(Fragmentation)包头由源节点使用，以表明数据包已经被分为片段，适合在最大传输单元（MTU 大小）内使用。与 IP4 不同的是，在 IPv6 内，数据包分段与组装是通过端节点完成，而非通过路由器 完成，这进一步提高了 IPv6 网络的效率。

• 认证包头(AH)与封装安全有效负载(ESP)包头用于 IPSec 中，以提供安全服务，确保数据包的认证、 完整性和保密性。

IPv6 操作

相邻节点发现

相邻节点发现协议使 IPv6 节点和路由器能够判定相同网络上相邻节点的链路层地址，并发现和跟踪相邻的 路由器。IPv6 相邻节点发现方法使用 IPv6 ICMP（ICMPv6)消息与被请求的节点组播地址判定相同网络上 相邻节点的链路层地址，验证相邻节点的可到达性，并跟踪相邻的路由器。

当一个节点要判定相同本地链路上另一节点的链路层地址时，一个相邻节点的请求消息携带着发送者自身的链路层地址，在本地链路上被发送出去。目的节点在收到相邻节点的请求消息后，将在本地链路上使用其自 身的链路层地址发送一个相邻节点通告消息，以此来回复请求。在收到相邻节点的通告之后，源节点与目的 节点便可进行通信。相邻节点的通告消息也可在本地链路上一个节点的链路层地址发生改变时发出。

路由器发现

IPv6 路由器发现法使用路由器通告和请求消息发现本地链路上的路由器。路由器通告消息在 IPv6 路由器 的每个配置接口上定期发出，并回应来自链路上 IPv6 节点的路由器请求消息。当主机没有配置好的单播地 址时，便发送一个路由器请求消息，使主机能够迅速自行自动配置，而不必等待下一个列入计划的路由器通 告消息。

路由器通告包含或决定：

- 节点应该使用何种类型的自动配置——是无状态的还是有状态的。
- 节点应该放入 IPv6 包头的跳转(Hop)限度值。
- 节点应该用来形成单播地址的网络前缀。
- 所包含网络前缀的寿命信息。
- 节点在发送数据包中应该使用的最大传输单元(MTU)大小。
- 发端路由器是否应该用作默认路由器。 无状态的自动配置与 IPv6 节点的重新编号

无状态的自动配置使 IPv6 节点能够完成不用服务器的基本配置，并方便地进行重新编号。无状态的自动配 置使用路由器通告消息中的网络前缀信息作为节点地址前缀的/64。余下的 64 位地址由分配给结合

EUI-64 格式附加位的以太网接口的 MAC 地址获得。例如，带有以太网接口地址 0003B61A2061 的一个节

点，结合由路由器通告提供的网络前缀 2001:0001:1EEF:0000/64，将具有一个如

2001:0001:1EEF:0000:0003:B6FF:FE1A:2061 这样的 IPv6 地址。

IPv6 节点通过包含新、旧前缀的路由器通告消息，即有可能进行重新编号。旧前缀的寿命值减少，在仍以 旧前缀保持其连接完好的同时，即提醒节点使用新前缀。在此期间，节点有两个单播地址在同时使用。当 旧前缀不再使用时，路由器通告将只包含新前缀。

路径最大传输单元(MTU)

IPv6 路由器不处理数据包的分段。在需要时，数据包分段由数据包的发端节点或源节点处理。IPv6 使用

ICMP 错误报告来判定传输路径上的数据包大小是否与 MTU 大小相匹配。当一个节点通过 ICMP 错误报 告报告“数据包过大”时，源节点将缩小传输数据包的大小。该过程反复进行，直至传输路径上没有“数据 包过大”错误为止。这使节点能够动态发现特定数据路径上每个链路 MTU 大小之间的差异，并做出调整。

DHCPv6 与域名服务器（Domain Name Server，简称 DNS）

IPv6 除了支持无状态的自动配置以外，还通过 DHCPv6 支持有状态的配置。IPv6 节点具有一个在路由器 未被发现时通过 DHCP 服务器请求地址的选项。DHCPv6 的操作与 DHCPv4 大体类似，但是，DHCPv6 为其多数消息使用组播。IPv6 还引进了新的记录类型，以满足域名服务器中 IPv6 地址的需要。AAAA 记录，又称“四 A”，已由 IETF 推荐，用于将主机名映射到 IPv6 地址。

IPv6 部署

IPv6 技术相比 IPv4 技术而言具有许多优势。然而，人们一致认为，IPv6 部署的任何成功策略均需其在一定 持续时期内与 IPv4 共存。为了管理这种从 IPv4 到 IPv6 的长期复杂过渡，人们已制订出许多策略。以下 部分将对若干此类策略予以阐述。

双栈骨干网

在双栈骨干网部署中，网络中的所有路由器均同时保持 IPv4 与 IPv6 协议堆栈。应用程序在使用 IPv4 或

IPv6 之间进行选择，并由应用程序按照 IP 业务负载的类型与通信的特定需要选择正确的地址。

如今，双栈路由选择是具有需要两种协议的 IPv4 与 IPv6 组合应用程序网络基础架构的首选部署策略。然 而，该策略却有若干限制：网络中的所有路由器必须升级到 IPv6；路由器还需要双寻址方案、IPv4 与 IPv6 选路协议的双重管理以及 IPv4 与 IPv6 两个选路表所需的足够存储空间。

通过 IPv4 实现 IPv6 的隧道传输

通过 IPv4 实现 IPv6 的隧道传输方法是在 IPv4 数据包内封装 IPv6 业务负载，通过 IPv4 骨干网进行发送

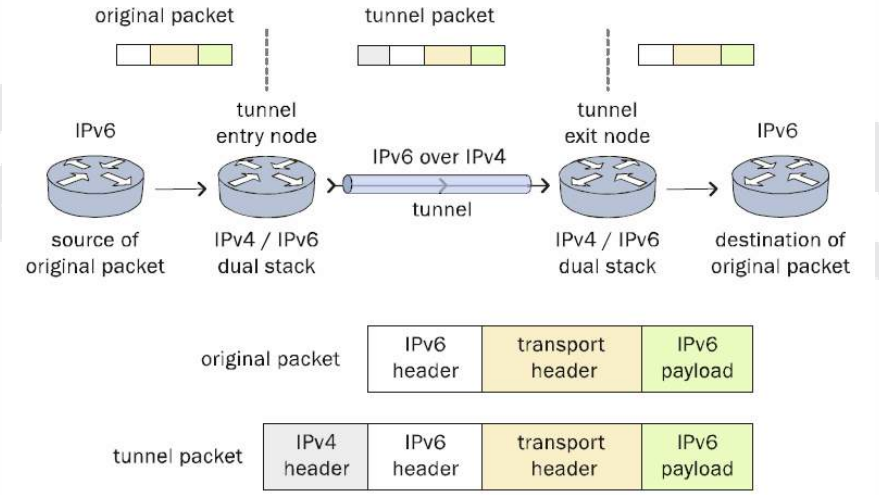
（如图所示）。这使“孤岛状”IPv6 终端系统和路由器能够通过现有 IPv4 基础架构进行通信。

如下文所述，有多种隧道传输机制可用于部署 IPv6（如图所示）。 人工配置隧道

如同 RFC 2893 所定义的那样，隧道的两个端点需要使用适当的 IPv6 和 IPv4 地址进行配置。坐落于端点 的边缘路由器，通常为双栈路由器，将按照配置转发通过隧道的业务负载。

通用路由封装（Generic Routing Encapsulation，简称 GRE）隧道

按照通过 IPv4 网络传输数据的定义，GRE 通过将需要传输的数据包封装在 GRE 数据包内，从而使一个网 络协议能够通过另一个网络协议进行传输。GRE 是通过隧道传输 IPv6 业务负载的一个理想机制。



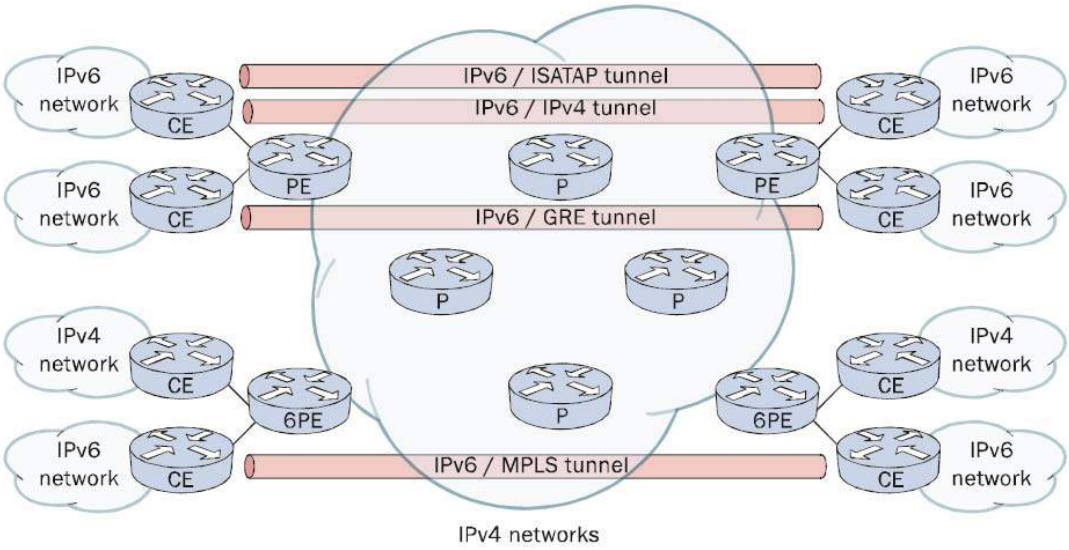
兼容 IPv4 的隧道或 6over4 隧道

如同在 RFC 2893 中定义的那样，这些隧道机制在兼容 IPv4 的 IPv6 地址上自动建立隧道。兼容 IPv4 的 IPv6 地址将最左侧的 96 位定义为零，后面跟随着一个嵌在最后 32 位中的 IPv4 地址。例如，0:0:0:0:0:0.64.23.45.21 是一个兼容 IPv4 的地址。

6to4 隧道

如同 RFC 3056 所定义的那样，6to4 隧道使用嵌在 IPv6 地址中的一个 IPv4 地址来确认隧道的端点，并自动建立隧道（如图所示）。

站内自动隧道寻址协议（Intra-Site Automatic Tunnel Addressing Protocol，简称 ISATAP）隧道



如同在 draft-ietf- ngtrans-isatap-16 中定义的那样，ISATAP 隧道传输非常类似于 6to4 隧道传输，但它是为 在本地站点或校园网中使用而设计的。ISATAP 地址包含 64 位网络前缀 0000:5EFE 以及一个确认隧道端 点地址

的 IPv4 地址（如图所示）。

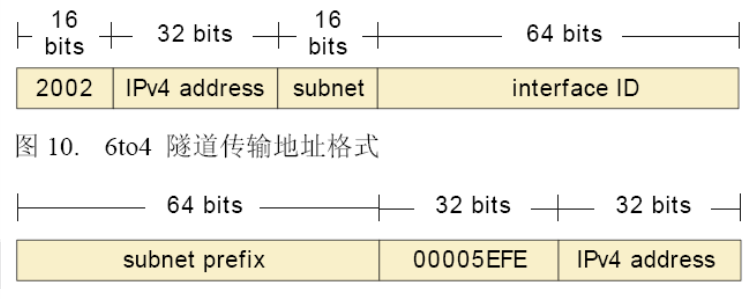


图 10. 6to4 隧道传输地址格式

多协议标记交换（Multi-Protocol Label Switching，简称 MPLS）隧道

使用 MPLS 技术，孤立的 IPv6 域能够通过一个 MPLS IPv4 核心网络在彼此间进行通信。因为 MPLS 转 发是基于标记的基础之上，而非基于 IP 包头本身，所以这种实施需要的骨干网基础设施升级大大减少，核 心路由器的重新配置也有所下降，从而为部署 IPv6 提供了一个非常经济合算的方法。此外，MPLS 固有的 VPN 与业务负载技术维护服务使 IPv6 网络能够通过支持 IPv4 VPN 与 MPLS-TE 的基础架构并入 VPN 或 外部网络。

Configuration

全局模式下 **Ipv6 unicast-routing** //打开 IPv6 转发功能

启用 CEFv6 **ipv6cef**

接口上配置 Ipv6 地址

在接口模式中：

Ipv6 address ipv6-addr/prefix-length [link-local]

Eg: **ipv6 addr 2001:0410::1/64**

ipv6 addr FE80::123:0456:0789:0abc link-local 覆盖路由器分配的默认本地地址

ipv6 addr fec0:0:0:9::1/128 配置回环口地址

ipv6 addr 2001:0410::1/64 eui-64

接口下可以不指定地址直接启用 ipv6 **if>#ipv6 enable**

Ipv6 unmumbered interface

修改 MTU 值 **ipv6 mtu 1492**

Show interface fa0/1

Show ipv6 interface fa0/1

Show ipv6 neighbors 察看 ipv6 的邻居

全局模式下 **ipv6 neighbor fec0::1:0:0:1:b fastethernet 0/0 000a.ebaa.0033** 静态添加 **ipv6** 邻居

Clear ipv6 neighbors 清除 **ipv6** 邻居

无状态自动配置

前缀公告

IPv6 addr 2001:0410:0:1::/64 eui-64

Ipv6 nd prefix 2001:410:0:1::/64 43200 43200

在接口上禁止路由器公告

Ipv6 nd suppress-ra

IPv6 ACL

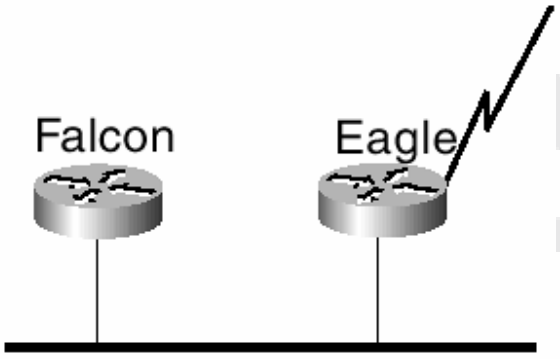
Ipv6 access-list access-list-name permit ipv6-prefix|any| host ipv6-addr log

IPv6-route

Static route

Ipv6 route ipv6-prefix {next hop| interface} distance

RIPng



Falcon

```
ipv6 unicast-routing
no ipv6 rip birdbath split-horizon
!
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 ipv6 enable
 ipv6 address FEC0::/64 eui-64
 ipv6 address FEC0::1:0:0:0:0/64 eui-64
 ipv6 address FEC0::2:0:0:0:0/64 eui-64
 ipv6 rip birdbath enable
  !
```

Eagle

```
ipv6 unicast-routing
no ipv6 rip birdbath split-horizon
!
!
interface Ethernet0
 no ip address
 no ip directed-broadcast
 ipv6 address FEC0::/64 eui-64
 ipv6 address FEC0::2:0:0:0:0/64 eui-64
 ipv6 address FEC0::3:0:0:0:0/64 eui-64
 ipv6 rip birdbath enable
!
interface Serial1
 ipv6 address FEC0::A:0:0:0:1/126
 ipv6 rip birdbath enable
```

OSPFv3

```
interface Ethernet7/0
 ipv6 address 2003:0:0:7::/64 eui-64
 ipv6 enable
 ipv6 ospf 1 area 1
 !
 interface Ethernet8/0
 ipv6 address 2003:0:0:8::/64 eui-64
 ipv6 enable
 ipv6 ospf 1 area 1
 !
 interface Ethernet9/0
 ipv6 address 2003:0:0:9::/64 eui-64
 ipv6 enable
```

```
ipv6 ospf 1 area 1
!
ipv6 router ospf 1
router-id 11.11.11.1
area 1 range 2003::/48
interface Ethernet0/0
ipv6 enable
ipv6 ospf 1 area 0
ipv6 ospf authentication ipsec spi 500 md5 1234567890ABCDEF1234567890ABCDEF
interface Ethernet0/0
ipv6 enable
ipv6 ospf authentication null
ipv6 ospf 1 area 0
```

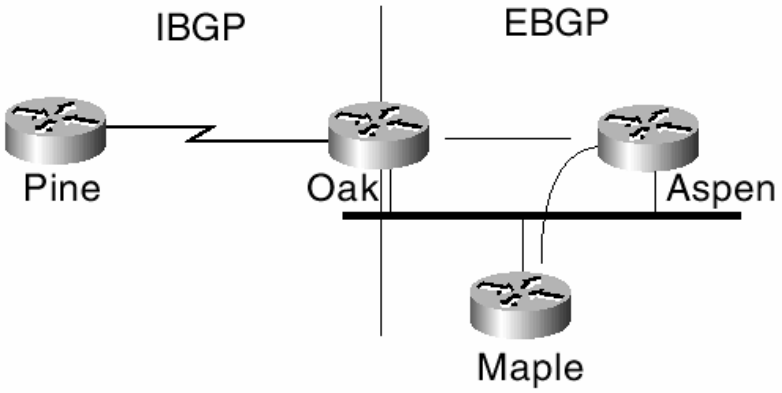
IS-IS

1. enable
2. configure terminal
3. router isis *area-name*
4. address-family ipv6 [unicast | multicast]
5. default-information originate [route-map *map-name*]
6. distance *value*
7. maximum-paths *number-paths*
8. summary-prefix *ipv6-prefix/ prefix-length* [level-1 | level-1-2 | level-2]
9. prc-interval *seconds* [*initial-wait*] [*secondary-wait*]
10. spf-interval [level-1 | level-2] *seconds* [*initial-wait*] [*secondary-wait*]
11. exit
12. interface *type number*
13. isis ipv6 metric *metric-value* [level-1 | level-2 | level-1-2]

EIGRP

```
ipv6 unicast-routing
interface e0
  ipv6 enable
  ipv6 eigrp 1
  no shutdown
!
ipv6 router eigrp 1
  router-id 10.1.1.1
  no shutdown
```

BGP4+




```
Oak
interface fastethernet 0
 description Oak to Aspen (e-bgp)
 ipv6 address 200A::2:0:0:0:1/64
!
interface serial 0
 description Oak to Pine (iBGP)
 ipv6 address 200A:0:0:10::1/124
!
interface serial 1
 description IGP link
 ipv6 address 200A:0:0:1::1/124
!
router bgp 100
 neighbor 200A::2:0:0:0:2 remote-as 300
 neighbor 200A:0:0:10::2 remote-as 100
!
 address-family ipv6
 neighbor 200A::2:0:0:0:2 activate
 neighbor 200A:0:0:10::2 activate
 network 200a:0:0:1::/124
 exit-address-family
Aspen
interface fastethernet 0
 description Oak to Aspen (e-bgp)
 ipv6 address 200A::2:0:0:0:2/64
!
router bgp 300
 neighbor 200A::2:0:0:0:1 remote-as 100
 !
 address-family ipv6
 neighbor 200A::2:0:0:0:1 activate
 exit-address-family
```

IPSec VPN

在两个或多个私人网络之间通过 INTERNET 传输数据。这里要考虑数据的机密性和完整性，以及验证用户身份。！他是通过公共网络来传输私人数据。

用 VPN 的好处：费用低廉，更高的灵活性，简单的控制管理，以及有通道的拓扑！

VPN 的网络主要包括：虚拟的网络和私有的网络。

虚拟主要是采用通道化的技术。而私有主要是采用加密的方法。

隧道技术：一个隧道就是一个点到点的连接。

隧道内可以承载多种不同的协议。比如 IP 或者 IPX 协议。

加密传输的数据。

VPN 有点到点的。点到多点的。移动用户到路由器，等。。

CISCO VPN 有一个完整的系统，解决方案。

VPN 的类型：按照远程介入有：客户初始化的也有 NETWORK ACCESS SERVER 初始化的。

按照站点到站点分有内部的和外部的。

加密：可以在多层加密。

应用层（SSH），传输层（SSL），网络层（IPSEC）。链路层。（主要介绍 IPSEC）

隧道协议：网络层：IPSEC。GRE。L2F。L2TP（RFC2661）。PPTP（主要介绍网络层）

L2TP 是一个层 2 的隧道协议，是是 L2F 和 PPTP 的结合。

GRE 是一个通用的路由的封装，不支持加密，只是构成一个隧道而已，可以和另外一些加密结合使用

IPSEC 构建一个加密的 IP 安全。

选择 3 层的 VPN 的隧道协议。

如果仅仅之后 IP 流量和单波就使用 IPSEC

如果有多中协议和广（多）播就使用 GRE 或者 L2TP。（这里加密并不是必须的）

一些术语：

tunnel 隧道。 加密和解密。加密系统，HASHING 哈西算法（是一个不可逆的算法）

验证，授权，KEY 的管理。CA（授权认证服务）

关于 IPSEC VPN 的术语：

AH：头验证 验证是必须的。

ESP：对有效数据进行验证和加密，在 RFC2406 中定义。 安全协议

IKE：INTERNET KEY 交换。在 INTERNET 上交换口令，可以不让其他人看到。保证口令的安全性。

ISAKMP(RFC2408)：Internet Security Association Key Management Protocol。

SA：安全观念

AAA：

TACACS+：

RADIUS：

digital certificate, A digital identification mechanism which establishes credentials issued by a certification authority.

2：CISCO IOS 加密系统：

密钥的管理：人工参与，安全 KEY 交换算法 IKE，CA 公共 KEY 交换（证书）。

加密：对称式的加密（DES，3DES。AES。）和非对称式加密（RSA），分公钥和私钥。

验证：MAC、HMAC

HASH 哈西算法：（SHA、MD5）

对称加密：加解密钥是一样的。

非对称加密：有共钥和私钥。加密用共钥，解密用私钥。保证了在传输过程中的口令的安全性，

如果在传输过程中口令被截取，那么他没有私钥，则不能进行解密。对称加密存在问题是 KEY 管理。

KEY(RFC2409) 的交换：

比如 ROUTERA 和 ROUTERB 交换 KEY。

ROUTERA 有私有值 XA 和共有值 YA。。。。ROUTERB 有私有值 XB 和共有值 YB

互相交换 YA 和 YB。

在 ROUTERA 方：K 等于 YB 的 XA 次放的模乘以 P

ROUTERB 方：K 等于 YA 的 XB 次放的模乘以 P。

这里 P 双方都是知道的！

这里的 K 应该相等。

HASHING 算法：

可以保证数据在传输工程中的数据的完整性，但是他不保证加密性。

具体是这样的：

本地的要发送的信息和共享的 KEY 经过 HASHING 算出一个值

再传输给远方。如果中间人要修改数据。那么 HASHING 的值会发生变化，这样对方就知道了数据被人改了。

3：IPSEC

加密以及验证，

ESP 可以做认证也可以做加密，加密的是内容，（不加密报头）

AH 包含了对头部的认证，一般两种方法可以结合使用通常。

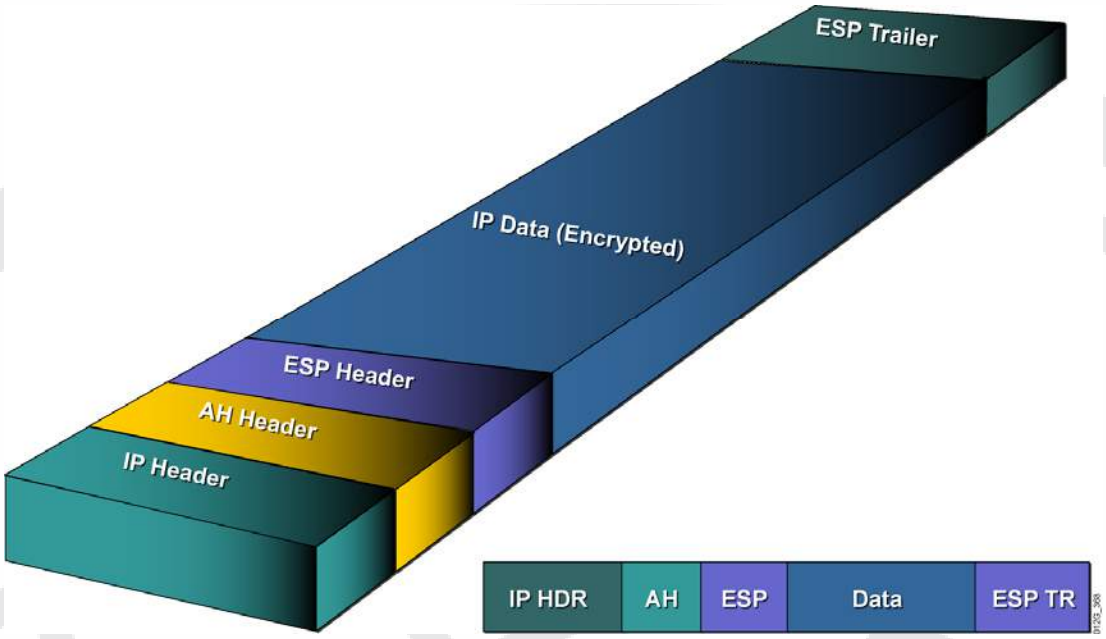
隧道模式传输模式：

区别：在隧道模式是支持多协议的，在隧道中用新的头封装了原来的数据，可以支持多种协议。对整个 IP 包提供安全保护，对 IP 包进行加密，然后再将加密后的包封装到另一个 IP 包中，再进行路由。只为高层（传输层及更高层）提供安全保护。IPSEC 网关为对等隧道中的其他主机提供 IPSEC 服务，终端主机不知道 IPSEC 被用来保护其数据流。IPSEC 为其他主机通过不可靠网络传输的数据流提供透明的保护。

传输模式：他对于原始数据的头部不做改变，而是在数据中增加 AH 和 ESP 的头来保证数据的安全性和完整性。

保护数据包的有效负载（Payload），不对原来的 IP 地址进行加密。原来的 IP 地址用于路由，ESP 传输模式用于主机之间。终端主机对其数据进行 IPSEC 封装，因此必须在终端主机上实现 IPSEC，应用端点也是 IPSEC 端点。

安全观念（SA）：两个设备之间进行安全信息交换的一个综合思想（概念），它是对等体或主机之间的策略约定。描述对等体如何使用IPSec安全服务来保护网络数据流。它安全地传输包所需的所有安全参数，实际上指定了IPSec的安全策略。SA总是包含规范。有些厂商也支持双向SA。对于每种封装协议（AH和ESP）和每种数据流向都需要一个独立的SA。VPN设备将其所有活动的SA都存储在一个本地数据库中，称之为SA数据库（SADB）。交换是数据。A single SA is negotiated by peers requesting secure communication（这句话不正确，因为router要协商出2个SA，因为有两个方向）。



IPSEC 的步骤：

A 发送敢兴趣流量到 B。这里敢兴趣流量是需要加密的数据。一般通过 ACL 来决定。

经过 IKE 的 SA 的协商。然后进行 IPSEC 的 SA 进行协商。

然后通过 IPSEC 的通道进行数据的交换。

IPSEC 的隧道形成。

配置 IPSEC 的任务

1，规划，准备，决定 IKE 和 IPSEC 的策略，

2，配置 IKE

3。配置 IPSEC

4 检验

show crypto isakmp policy 显示 IKE 策略。

Debug crypto isakmp，能够看懂输出的内容。

4：准备配置 IKE 和 IPSEC

准备 IKE 和 IPSEC 的策略。

决定 IKE 的策略。KEY 的分发思想和验证思想，IPSEC 对端的 IP 和 HOSTNAME，IKE 的策略。

保证两端配置一致。

决定 IPSEC。同样，两端配置应该一致。

5：配置 VPN

6.配置 IKE

步骤：crypto isakmp enable 启用/禁用 IKE。可直接在全局启用 ISAKMP。在不用 IPSEC 接口上要阻断 ISAKMP

Crypto isakmp policy priority 创建 IKE 策略。　　数据流，防止 DoS 攻击，可配置 ACL Deny 500 端口的语句。

Priority 取值 1-10000，1 最高。没有默认值需指定

配置 ISAKMP 身份 crypto isakmp identity {add|name}。没有解析名称的 DNS 时需指定 DNS 服务器 ip host 域名

Crypto isakmp key 配置共用密钥。　　。　　IP。

Show 检查 IKE 配置。

7 配置 IPSEC。

配置变换集：crypto ipsec transform-set mine esp-des。在 IKE phase2 快速模式下，要变换集协商。

配置全局 IPSEC SA 寿命：crypto ipsec security-association lifetie{second seconds| killobytes bytes}。

加密访问列表。

将扩展 ACL 用做加密 ACL

配置的加密 ACL。

加密映射的用途和参数。配置 IPSEC 加密映射：crypto map name seq-number （help match add。。）

应用加密映射到接口。Crypto map map-name

VPN 分两大类:远程接入 VPN 和场点到场点 VPN.

远程接入 VPN:　　安全地将远程用户连接到企业网络.

场点到场点 VPN:　　安全将企业或者公司分部分连接到企业网络.

远程接入 VPN 又分为两类:

客户发起的:远程用户通过使用客户端软件通过 ISP 共享网络建立的一条到企业网络的安全隧道.

网络接入服务器(NAS)发起的:远程用户拨入 ISP,.NAS 建立一条到企业私有网络的安全隧道,该隧道支持多个远程用户发起的会话,.

注意题目的词：The ISP-owned devices。两端都是自己的设备,就选 C.

场点到场点 VPN 又分外 An intranet VPN 和 An extranet VPN.An intranet VPN 主要是指所连的场点都是公司内部的办事处,分支等,是同一个公司的机构.而 An extranet VPN 是指连接客户,供应商,合作伙伴等.

在 phase 1,有 4 种:

1.选择密钥的分发方法.

2.选择验证方法.

3.确定 IPsce 对等体的 IP 地址和主机名.

4.确定对等体的 ISAMKP 的 policy.

在 phase 2,有 5 种:

1.选择 IPsec 的算法和参数也获得最佳的安全性和性能;

2.选择变化集.

3.确定 IP 对等题的细节.

4.选择 SA 的建立方式.（手动或 IKE 开始 SA）

5.确定要保护的数据流.

检验 IPSEC 配置：

show crypto isakmp policy：查看 ISAKMP 策略的参数。

Show crypto ipsec transform-set：查看配置的变换集。

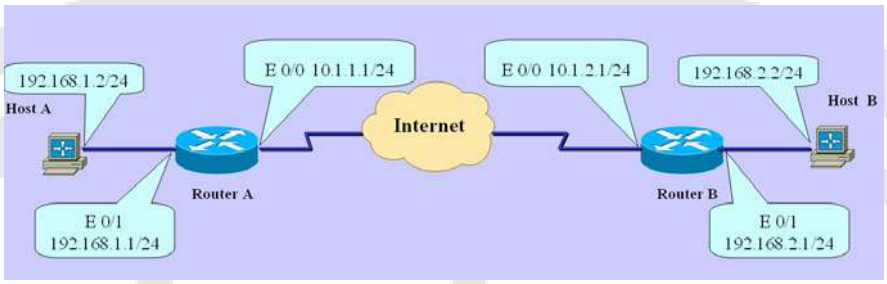
Show crypto map：查看加密映射配置。

Show crypto isakmp sa

Debug crypto [ipsec | isakmp]

Clear crypto sa| sa peer |sa map|sa entry

Clear [crypto] ipsec sa | sa peer |sa map|sa entry|isakmp sa。



上图AB间运行IPSec VPN ，配置如下

```
RouterA#show running-config

crypto isakmp policy 10
 hash md5
 authentication per-share
 crypto isakmp key nichole address 10.1.2.1
!
!
!
crypto ipsec transform-set nichole esp-des
!
crypto map nichole 20 ipsec-isakmp
 set peer 10.1.2.1
 set transform-set nichole
 match address 101
!
interface ethernet 0/0
 ip address 10.1.1.1 255.255.255.0
 ip access-group 100 in
 crypto map nichole
 no shutdown
!
interface ethernet 0/1
ip address 192.168.1.1 255.255.255.0
no shutdown
!
access-list 100 permit ahp host 10.1.2.1 host 10.1.1.1
access-list 100 permit esp host 10.1.2.1 host 10.1.1.1
access-list 100 permit udp host 10.1.2.1 host 10.1.1.1 eq 500 或(isakmp)
access-list 101 permit tcp 192.168.1.0 0.0.0.255 192.168.2.0 0.0.0.255
access-list 101 deny ip any any
!
RouterB#show running-config

crypto isakmp policy 10
 hash md5
 authentication per-share
 crypto isakmp key nichole address 10.1.1.1
!
!
!
crypto ipsec transform-set nichole esp-des
!
crypto map nichole 20 ipsec-isakmp
```

```
 set peer 10.1.2.1
 set transform-set nichole
 match address 101
!
interface ethernet 0/0
 ip address 10.1.2.1 255.255.255.0
 ip access-group 100 in
 crypto map nichole
 no shutdown
!
interface ethernet 0/1
ip address 192.168.2.1 255.255.255.0
no shutdown
!
access-list 100 permit ahp host 10.1.1.1 host 10.1.2.1
access-list 100 permit esp host 10.1.1.1 host 10.1.2.1
access-list 100 permit udp host 10.1.1.1 host 10.1.2.1 eq 500  或(isakmp)
access-list 101 permit tcp 192.168.2.0 0.0.0.255 192.168.1.0 0.0.0.255
access-list 101 deny ip any any
!
return
```

MPLS VPN

1 MPLS 提出的意义

传统的 IP 数据转发是基于逐跳式的，每个转发数据的路由器都要根据 IP 包头的目的地址查找路由表来获得下一跳的出口，这是个繁琐又效率低下的工作，主要的原因是两个：1、有些路由的查询必须对路由表进行多次查找，这就是所谓的递归搜索；2、由于路由匹配遵循最长匹配原则，所以迫使几乎所有的路由器的交换引擎必须用软件来实现，用软件实现的交换引擎和 ATM 交换机上用硬件来实现的交换引擎在效率上无法相 抗衡。

当今的互联网应用需求日益增多，对带宽、对时延的要求也越来越高。如何提高转发效率，各个路由器生产厂家做了大量的改进工作，如 Cisco 在路由器上提供 CEF（Cisco Express Forwarding）功能、修改路由表搜索算法等等。但这些修补并不能完全解决目前互联网所面临的问题。

IP 和 ATM 曾经是两个互相对立的技术，各个 IP 设备制造商和 ATM 设备制造商都曾努力想吃掉对方，想 IP 一统天下，或者 ATM 一家独秀！但是最终是这两种技术的融合，那就是 MPLS(Multi-Protocol Label Switching)技术的诞生！MPLS 技术结合和 IP 技术信令简单和 ATM 交换引擎高效的优点！

2 MPLS 技术的实现细节

2.1 标签结构

IP 设备和 ATM 设备厂商实现 MPLS 技术是在各自原来的基础上做的，对于 IP 设备商，它修改了原来 IP 包直接封装在二层链路帧中的规范，而是在二层和三层包头之间插了一个标签(Label)，而 ATM 设备制造商利用了原来 ATM 交换机上的 VPI/VCI 的概念，在使用 Label 来代替了 VPI/CVI，当然 ATM 交换机上还必修改信令控制部分，引入了路由协议，ATM 交换使用了路由协议来和其他设备交换三层的路由信息。

标签的结构如下：

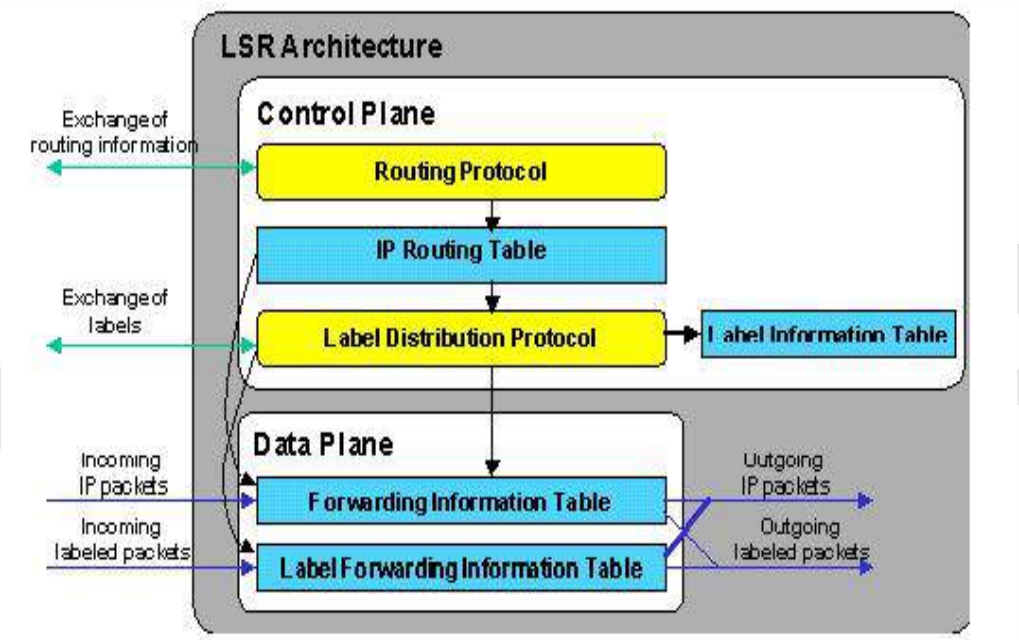
20 比特的 LABEL 字段用来表示标签值，由于标签是定长的，所以对于路由器来说，可以分析定长的标签来做数据包的转发，这是标签交换的最大优点，定长的标签就意味这可以用硬件来实现数据转发，这种硬件转发方式要比必须用软件实现的路由最长匹配转发方式效率要高得多！

3 比特的 EXP 用来实现 QOS

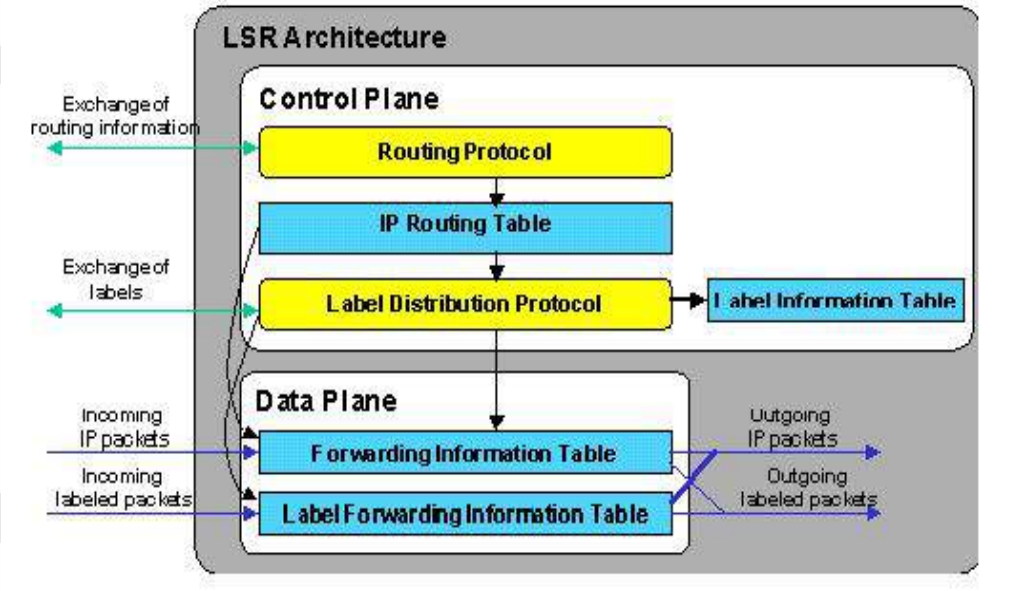
1 比特 S 值用来表示标签栈是否到底了，对于 VPN，TE 等应用将在二层和三层头之间插入两个以上的标签，形成标签栈。

8 比特 TTL 值用来防止数据在网上形成环路。

这样完整的带有标签的二层帧就成了如下形式：

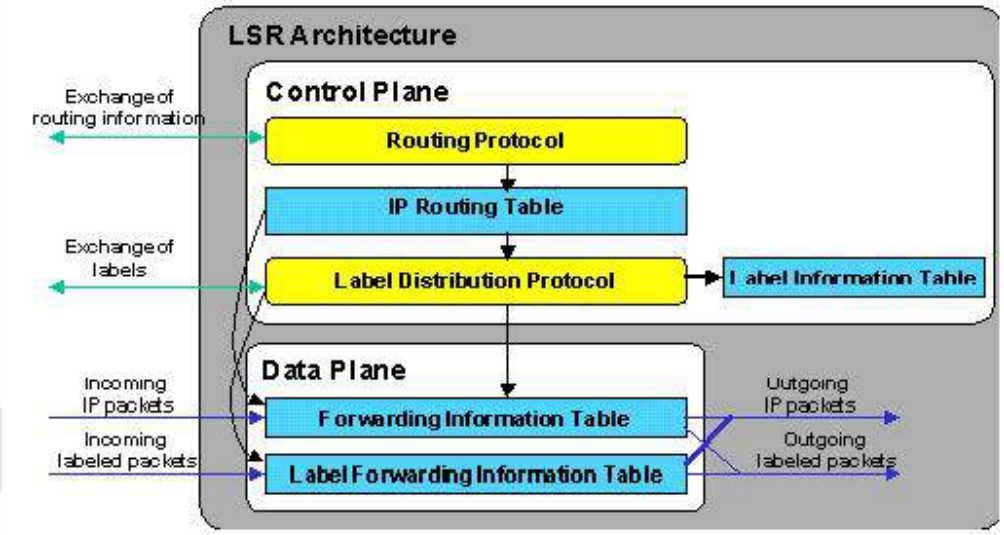


在 ATM 信元模式下，信元的结构如下形式：



2.2 LSR 设备的体系结构

通过修改，能支持标签交换的路由器为 LSR(Label Switch Router)，而支持 MPLS 功能的 ATM 交换机我们一般称之为 ATM-LSR。LSR 设备的体系结构如下：



LSR 的体系结构分为两块：

1. 控制平面(Control Plane)

该模块的功能是用来和其他 LSR 交换三层路由信息，以此建立路由表；和交换标签对路由的绑定信息，以此建 Label Information Table(LIB)标签信息表。同时再根据路由表和 LIB 生成 Forwarding Information Table(FIB)表和 Label Forwarding Information Table(LFIB)表。控制平面也就是我们一般所说的路由引擎模块！

2.数据平面(Data Plane)

数据平面的功能主要是根据控制平面生成的 FIB 表和 LFIB 表转发 IP 包和标签包。

对于控制平面中所使用的路由协议，可以使用以前的任何一种，如 OSPF、RIP、BGP 等等，这些协议的主要功能是和其他设备交换路由信息，生成路由表。这是实现标签交换的基础。在控制平面中导入了一种新的协议—LDP，该协议的功能是用来针对本地路由表中的每个路由条目生成一个本地的标签，由此生成 LIB 表，再把路由条目和本地标签的绑定通告给邻居 LSR，同时把邻居 LSR 告知的路由条目和标签帮定接收下来放到 LIB 表里，最后在网络路由收敛的情况下，参照路由表和 LIB 表的信息生成 FIB 表和 LFIB 表。具体的标签分发模式如下叙述。

2.3 标签的分配和分发

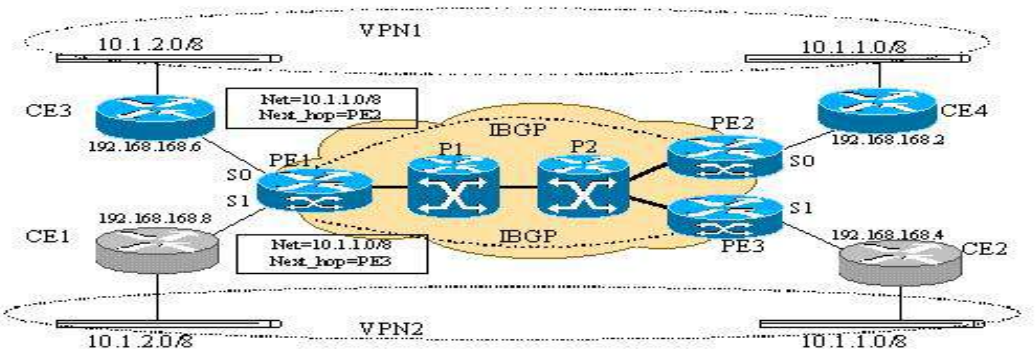
上面叙述到了，MPLS 技术是 IP 技术和 ATM 技术的融合。LSR 和 ATM-LSR 上实现标签的生成和分发是有点不同的。

2.3.1 包模式(Packet Mode)下的标签的分配和分发

对于实现包模式 MPLS 网络中，是下游 LSR 独立生成路由条目和标签的绑定，并且是主动分发出去的。

所有 LSR 上启动了 LDP 协议。以 LSR-B 为例，它已经通过路由协议获得网络 X 的路由了，一旦启动 LDP 协议，LSR-B 立即查找路由表,如果 X 网络的路由是由 IGP 路由协议学到的,则在 LIB 表中为通向 X 网络的路由生成一个本地标签 25,由于 LSR-B 和 LSR-A、LSR-C、LSR-E 形成了 LDP 邻居关系,所以下游 LSR-B 会主动给所有的邻居发送这个 X=25 的路由条目和标签的绑定!LSR-A、LSR-E、LSR-C 会把该路由条目和标签的绑定放置到本地的 LIB 表中，再结合本地的路由表，在 FIB 表中生成有关 X 网络的“网络地址->出标签”条目，在 LFIB 中生成有关 X 网络的“进标签->出标签”条目。所有的 LSR 上都如此操作。最终的结果使整个 MPLS 网络内部所有 LSR 上达到路由表、LIB 表、FIB 表、LFIB 表的动态平衡。

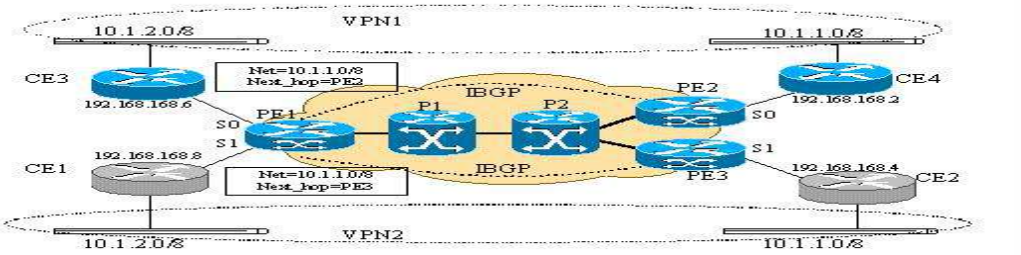
如果 LSR-A 接收到要去 X 网段的数据，由于 LSR-A 处在 MPLS 网络的边缘，必须查找 FIB 表，对接收到的 IP 包，做标签插入操作。对于 LSR-B，LSR-C 则纯粹是分析标签包，对包头的标签做转换，在转发标签包而已。数据到了 LSR-D，该边缘 LSR 会去掉标签包中的标签，再对恢复的 IP 包做转发！如下图：



2.3.2

信元模式(Cell Mode)下的标签分配和分发

在信元模式下，下游 ATM-LSR 接收到了上游 ATM-LSR 标签绑定请求后，下游受控分配标签，被动向上游分发标签。如下图

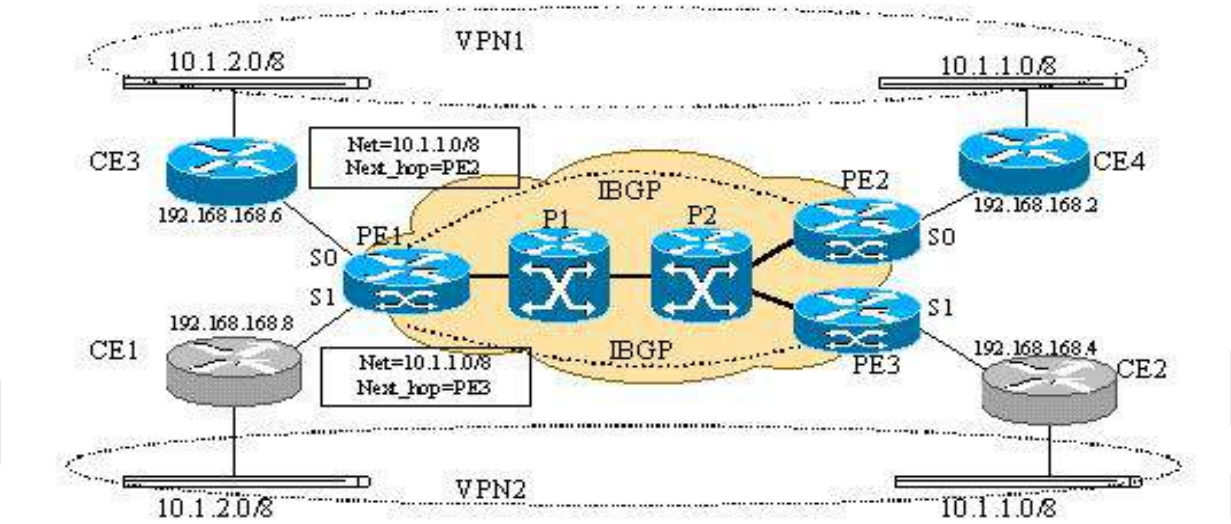


最上游的 LSR-A 向 ATM-LSR-B 发起对网络 X 的标签求情，ATM-LSR-B 再向 ATM-LSR-C 发请求，最后请求到达 LSR-D，LSR-D 生成本地对 X 网络的标签 1/37，把该标签告诉 ATM-LSR-C，C 做同样操作，这样一步一步到达 LSR-A。最终生成一条从 A->B->C->D 的 LSP(Label Switch Path)。这样如果 A 收到要到 X 网络的数据，A 就把 IP 数据包分割成带有标签的信元，通过 ATM 接口发送到 B，接下来 B 和 C 就纯粹做 ATM 信元的转发，到了 D 后再把信元组合成 IP 数据包，发向网络 X。

在此要强调的如果要组建以 ATM 交换机为核心的 MPLS 网络，那么在 ATM 网络的边缘必须设置路由器，原因在于 ATM 交换机只转发信元，无法处理用户数据 IP 包。当然上面也提到要在 ATM 交换机上实现 MPLS 功能，必须在 ATM 交换机的信令控制部分加入路由协议，而路由信息包往往是打在 IP 包中的，如 RIP，OSPF，BGP 等路由协议。ATM 交换机为了确保这些以 IP 包形式传递的路由信息能够在 ATM 交换机间传递，使用了专门的带外连接通道或者带内的管理 VC。

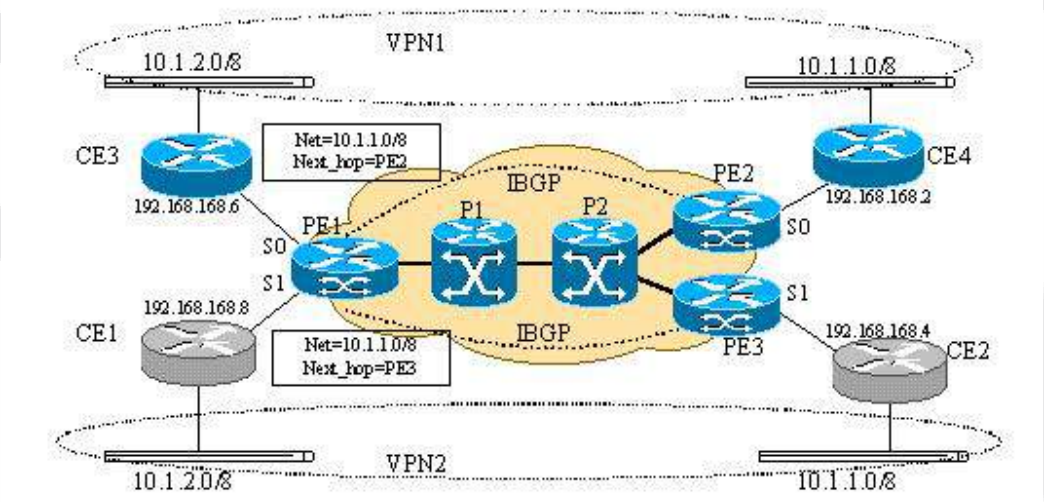
2.4 BGP 协议在 MPLS 网络中的特殊应用

上面提到 LSR 根据路由表分配标签时，只对从 IGP 协议获得的路由条目分配标签。原因何在？这是有特殊意义的！看下图：



整个 Transit AS 中启动 MPLS 交换。保证 ISP2 和 LSR-Border2 之间的网段发布到 Transit AS 内部的 IGP 路由协议中，对 ISP1 和 LSR-Border2 之间的网段也做同样的要求。前面提到过 LSR 为路由条目分配标签时，只对从 IGP 学来的路由分配标签，而网络 1.2.3.4 是被发布到 Transit AS 内部的 IGP 路由协议中了，可以肯定在 Border1 处是可以获得 Core1 告诉它有关 1.2.3.4 网络的标签 23。LSR-Border1，LSR-Border2 之间形成 IBGP 邻居关系，通过 BGP 协议，LSR-Border2 把从 ISP2 处学来的 10.0.0.0/8 这条路由告诉给 LSR-Border1，这条路由的下一跳地址是 1.2.3.4，这样一来让 LSR-Border1 得知要给网络 10.0.0.0/8 发送数据，先把数据发送到 1.2.3.4 这个网络来。1.2.3.4 被绑定了标签 23，所以在生成 FIB 表时，也给 10.0.0.0/8 这个网段绑定一个标签 23。这样，如果有数据从 ISP1 穿越 Transit AS 到达 ISP2，在 Border1 处就会给 IP 包插上 23 这个标签，把生成的标签包转发到 Core1，Core1 就只要分析标签头做标签包的转发就可以了！由于 Transit AS 内部核心路由器不必要运行 BGP 协议，这样一来，MPLS 网络的核心路由器就不会知道外部用户的路由，缩小了核心路由器的路由表，提高了搜索效率。大家也看到，由于打上了标签，IP 包头是不会在核心路由器被分析的，即使 IP 包头含有 10.0.0.1 这样的私有 IP 地址，也会因为只分析标签的原因被正常转发，这就是服务提供商提供 VPN 服务所追求的。当然在此必须重声，LSP 在整个 Transit AS 不能被断开，如果断开，标签包就恢复成 IP 包，而核心路由器是不含用户路由的，最终导致数据包的丢失。

BGP 在 MPLS 网络中的作用为我们提供了 VPN 服务打开了方便之门，但也应该意识到 VPN 服务两个最基本的要求是 1.用户可以独立规划 IP 地址；2.安全性非常重要！看下图：



以上为两个 VPN 实例，PE1(PE=Provider Edge device)上分别接了 CE1 （CE=Customer Edge device)和 CE3，但是 CE1 和 CE3 上

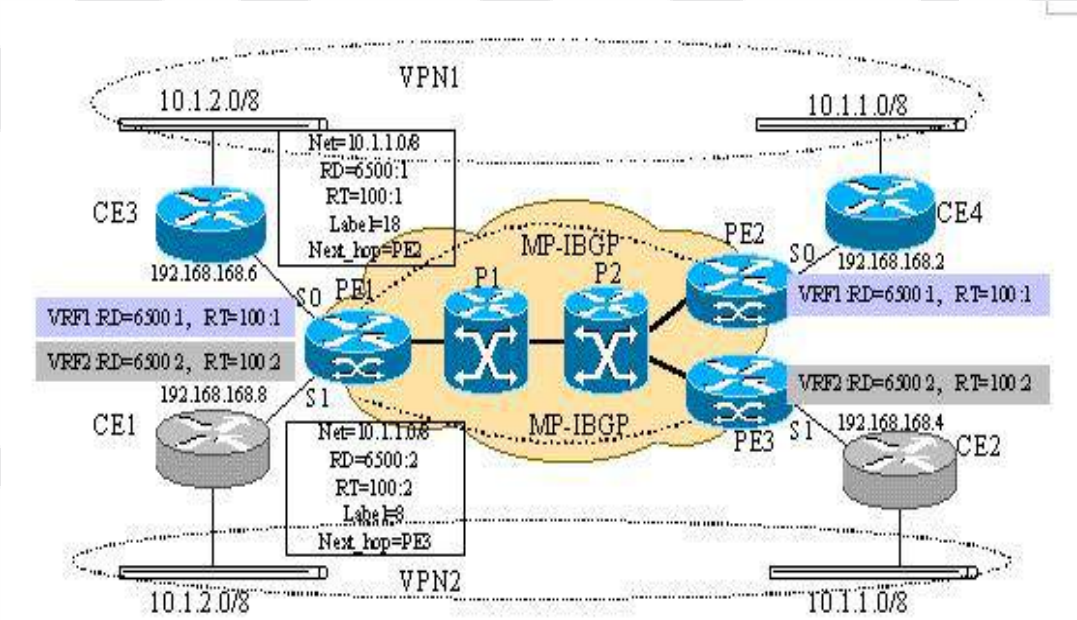
带到 IP 地址相同的网段 10.1.2.0/8，很明显如果不对 PE1 路由器做修改，PE1 只能认为往 10.1.2.0/8 的数据要么从 S0 出，要么从 S1 出,这样的话，不是 CE1 就是 CE3 就更本收不到从 PE1 发来的前往 10.1.2.0/8 网段的数据！

如果不对 BGP4 协议做修改，那么 PE2 和 PE3 发送给的 PE1 的有关 10.1.1.0/8 网络的路由更新就有可比性，PE1 最终会选择一条路由,认为或是 PE2 或者 PE3 是发送数据到 10.1.1.0/8 的必经路由器。这样如果 CE1 带的 10.1.2.0/8 网段上的主机给 10.1.1.0/8 网段上的主机发送数据时，可能会发到 CE4 所带的 10.1.1.0/8 的网段上，这样造成了数据泄露。

所以，为了使 LSR 能提供基于 MPLS 的 VPN 服务，还必须对此类设备做修改

3 基于 MPLS 的 VPN 实现

3.1 VPN 的历史



VPN 服务是很早就提出的概念，不过以前电信提供商提供 VPN 是在传输网上提供的覆盖型的 VPN 服务。电信运营商给用户出租线路，用户上层使用何种的路由协议、路由怎么走等等，这些电信运营商不管 。这种租用线路来搭建 VPN 的好处是安全，但是价格昂贵，线路资源浪费严重。

后来随着 IP 网络的全面铺开，电信服务提供商在竞争的压力下，不得不提供更加廉价的 VPN 服务，也就是三层 VPN 服务。通过提供给用户一个 IP 平台，用户通过 IP Over IP 的封装格式在公网上打隧道，同时也提供了加密等等的手段提供安全保障。这类 VPN 用户在目前的网络上数量还是相当巨大的！但是这类 VPN 服务因大量的加密工作、传统路由器根据 IP 包头的目的地址转发效率不高等等的原因不是非常令人满意。

MPLS 技术的出现和 BGP 协议的改进，让大家看到了另一种实现 VPN 的曙光。

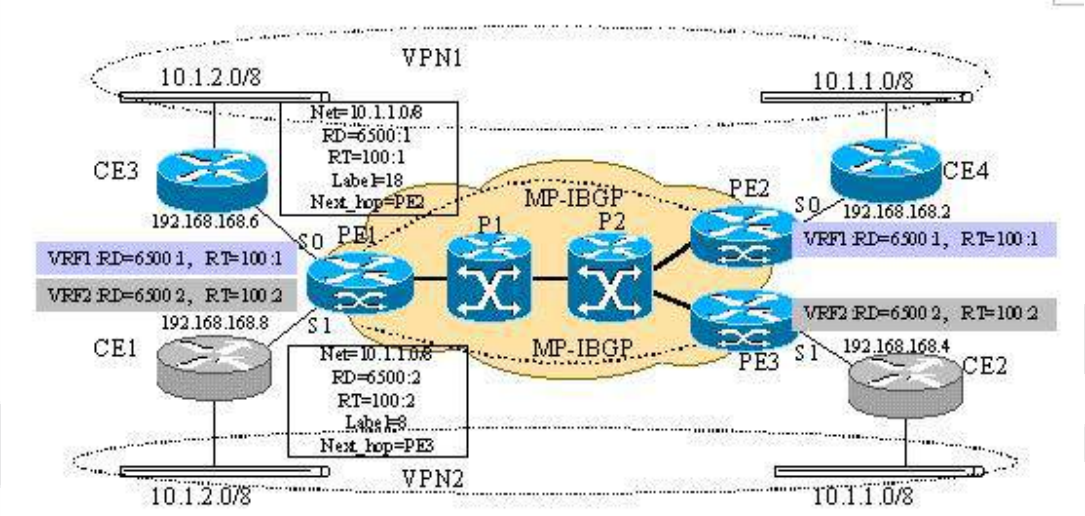
3.2 MPLS/VPN 体系结构

3.2.1 PE 路由器的改造和 VRF 的导入

为了让 PE 路由器上能区分是哪个本地接口上送来的 VPN 用户路由，在 PE 路由器上创建了大量的虚拟路由器，每个虚拟路由器都有各自的路由表和转发表，这些路由表和转发表统称为 VRF(VPN Routing and Forwarding instances)。一个 VRF 定义了连到 PE 路由器上的 VPN 成员。VRF 中包含了 IP 路由表，IP 转发表（也成为 CEF 表），使用该 CEF 表的接口集和路由协议参数和路由导入导出规则等等。

在 VRF 中定义的和 VPN 业务有关的两个重要参数是 RD(Route Distinguisher)和 RT(Route Target)。RD 和 RT 长度都是 64 比特。有了虚拟路由器就能隔离不同 VPN 用户之间的路由，也能解决不同 VPN 之间 IP 地址空间重叠的问题。

3.2.2 MP-BGP 协议对 VPN 用户路由的发布



正常的 BGP4 协议能只传递 IPv4 的路由，由于不同 VPN 用户具有地址空间重叠的问题，必须修改 BGP 协议。BGP 最大的优点是扩展性好，可以在原来的基础上再定义新的属性，通过对 BGP 修改，把 BGP4 扩展成 MP-BGP。在 MP-IBGP 邻居间传递 VPN 用户路由时打上 RD 标记，这样 VPN 用户传来的 IPv4 路由转变为 VPNv4 路由，这样保证 VPN 用户的路由到了对端的 PE 上，能够使对端 PE 区分开地址空间重叠但不同的 VPN 用户路由。例子如下：

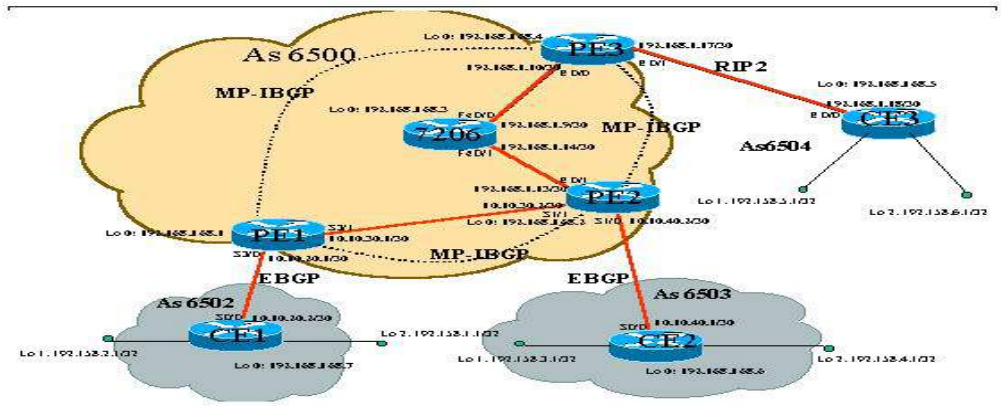
在 PE1、PE2、PE3 上分别配置 VRF 参数，其中 VPN1 用户的 RD=6500:1，RT=100:1，VPN2 用户的 RD=6500:2、RT=100:2。所有 VRF 可以同时 导入和导出所定义的 RT。

以 PE2 为例，PE2 从接口 S0 上获得由 CE4 传来的有关 10.1.1.0/8 的路由，PE2 把该路由放置到和 S0 有关的 VRF 所管辖的 IP 路由表中，并且分配该路由的本地标签，注意该标签是本地唯一的。通过路由重新发布把 VRF 所管辖的 IP 路由表中的路由重新发布到 BGP 表中,此时通过参考 VRF 表的 RD、RT 参数,把正常的 IPv4 路由变成 VPNv4 路由,如 10.1.1.0/8 变成 6500:1:10.1.1.0/8，同时把导出(Export)RT 值和该路由的本地标签值等等的属性全部加到该路由条目中去。通过 MP-IBGP 会话，PE2 把这条 VPNv4 路由发送的 PE1 处，PE1 收到了两条有关 10.1.1.0/8 的路由，其中一条是由 PE3 发来的，由于 RD 的不同，导致该两条路由没有可比性。MP-BGP 接受到该两条路由后的后继工作是：去掉 VPN4 路由所带的 RD 值，使之恢复 IPv4 路由原貌，并且根据各 VRF 配置的允许导入(Import)的 RT 值，把 IPv4 倒到各个 VRF 管辖的路由表和 CEF 表中，也就是说带有 RT=100:1 的 10.1.1.0/8 的路由倒到 VRF1 所管的路由表和 CEF 表中，带有 RT=100:2 的 10.1.1.0/8 的路由倒到 VRF2 所管辖的路由表和 CEF 表中。再通过 CE 和 PE 之间的路由协议，PE 把不同的 VRF 管辖的路由表内容通告的各自的相联的 CE 中去。

目前 PE 和 CE 之间可支持的路由协议只有四种 BGP、OSPF、RIP2 或者静态路由。

3.2.3 MPLS/VPN 中标签分组的转发

同过 MP-BGP 协议各个 VPN 用户路由器学习到正确的路由，现在看看如何转发用户数据的。



1.CE1 接收到发往 10.1.1.1 的 IP 数据包，查询路由表，把该 IP 数据包发送到 PE1。

2.PE1 从 S1 口上收到 IP 数据包后，根据 S1 所在的 VRF，查询对应的 CEF 表，数据包打上标签 8，注意该标签就是通过 MP-BGP 协议传来的。PE1 继续查询全局 CEF 表，获知要把数据发往 10.1.1.1，必须先发送到 PE2，而要发送到 PE2，则必须打上由 P1 告知的标签 2。所以该 IP 包被打上了两个标签。

3.P1 接收到标签包后，分析顶层的标签，把顶层标签换成 4，继续发送的 P2。

4.P2 和 P1 一样做同样的操作，由于次末中继弹出机制，P2 去掉标签 4，直接把只带有一个标签的标签包发送的 PE2。

5.PE2 收到标签包后，分析标签头，由于该标签 8 是它本地产生的，而且是本地唯一的，所以 PE2 很容易查出带有标签 8 的标签包应该去掉标签，恢复 IP 包原貌，从 S1 端口发出。

6.CE2 获得 IP 数据包后，进行路由查找，把数据发送到 10.1.1.0/8 网段上。

4 MPLS/VPN 配置实例

要提供 VPN 服务的前提是：服务提供商的网络必须启用标签交换功能，即把以前的数据网络升级为 MPLS 网络。然后具体配置 PE，PE 上的配置按六步走：

1.定义并且配置 VRF

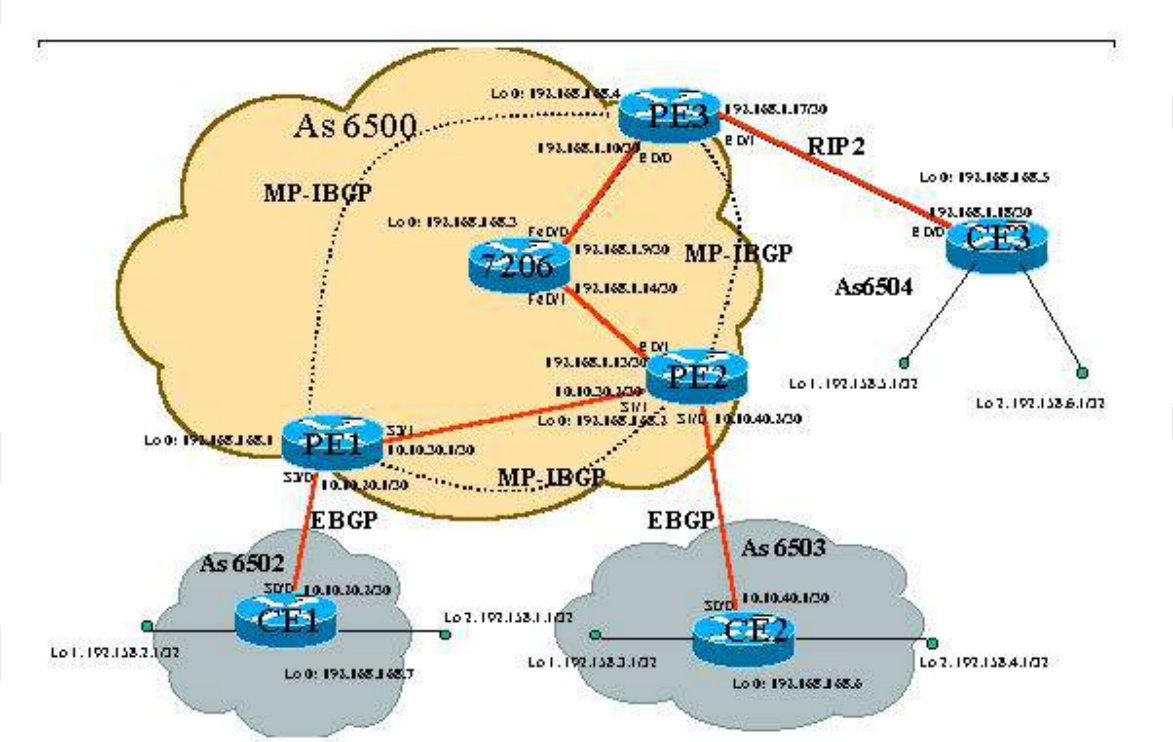
2.定义并且配置 RD

3.定义 RT，并且配置导入导出策略

4.配置 MP-BGP 协议

5.配置 PE 到 CE 的路由协议

6.配置连接 CE 的接口，将该接口和前面定义的 VRF 联系起来。



上图中 CE1、CE2、CE3 组成一个 VPN，其中 PE3 和 CE3 之间走 RIP2 协议，PE2 和 CE2 之间走 BGP 协议。整个 As 6500 中走 OSPF 协议。

PE3 的部分配置如下：

ip cef ——启用 CEF 转发功能

ip vrf Red ——定义一个 VRF ，名字为 Red

description For Red User Vpn

rd 6500:1 ——定义 RD 值为 6500:1

route-target export 6500:1 ——定义导出策略

route-target import 6500:1 ——定义导入策略

router rip ——配置 PE3 到 CE3 的路由协议 RIP2

version 2 !

address-family ipv4 vrf Red version 2 redistribute bgp 6500 metric 1———将 BGP 学到的路由从新发布的 RIP2 中，network 192.168.1.0 使 CE3 能学到同一 VPN 中的其他路由

no auto-summary

exit-address-family

router bgp 6500 ———配置 BGP 协议

no synchronization

no bgp default ipv4-unicast

bgp log-neighbor-changes

neighbor 192.168.168.2 remote-as 6500 ———和 PE2 建立邻居关系

neighbor 192.168.168.2 update-source Loopback0

no auto-summary !

address-family ipv4 vrf Red ————为 VPN 用户配置 IPv4 地址家族，使

redistribute rip metric 1 VRF Red 所管辖的路由表中的路由从新发布到 BGP 协议中去。

no auto-summary

no synchronization

exit-address-family !

address-family vpnv4 ————具体配置和 PE2 的关系，使 PE3 和 PE2 之间能交换 VPNv4 路由

neighbor 192.168.168.2 activate

neighbor 192.168.168.2 send-community both

no auto-summary

exit-address-family

interface Ethernet0/1 ————配置连接 CE3 的接口

ip vrf forwarding Red ————使该接口和前面定义的 VRF Red 联系起来

ip address 192.168.1.17 255.255.255.252

interface Ethernet0/0 ———配置联系到 7206 上接口

ip address 192.168.1.10 255.255.255.252

half-duplex

tag-switching ip ———在该接口上启用标签交换 !

PE2 上的部分配置如下：

ip cef ——启用 CEF 转发功能

ip vrf Red ——定义一个 VRF ，名字为 Red description For Red User Vpn

rd 6500:1 ——定义 RD 值为 6500:1

route-target export 6500:1 ——定义导出策略

route-target import 6500:1 ——定义导入策略

```
!
router bgp 6500 ——配制 BGP 协议
no synchronization
no bgp default ipv4-unicast
bgp log-neighbor-changes
neighbor 192.168.168.4 remote-as 6500
neighbor 192.168.168.4 update-source Loopback0
neighbor 192.168.168.4 next-hop-self
——这点在 PE-CE 之间路由协议为 BGP 时，一定要配置。
no auto-summary !
address-family ipv4 vrf Red
neighbor 10.10.40.1 remote-as 6504 ——配置和 CE2 之间的路由协议 BGP
neighbor 10.10.40.1 activate
no auto-summary
no synchronization
exit-address-family !
address-family vpnv4
neighbor 192.168.168.4 activate ——激活和 PE3 的 MP-IBGP 邻居关系
neighbor 192.168.168.4 send-community both
no auto-summary
exit-address-family !
interface Serial1/0 ——配置连接到 CE2 的接口
ip vrf forwarding Red ——把该接口和 VRF Red 联系起来
ip address 10.10.40.2 255.255.255.252
interface Ethernet0/1 ——配置连接到 7206 的接口
ip address 192.168.1.13 255.255.255.252
half-duplex
tag-switching ip ——在此接口上启用标签交换
interface Serial1/1 ——配置连接到 PE1 的接口
bandwidth 1544
ip address 10.10.30.2 255.255.255.252
encapsulation ppp
tag-switching ip ——在此接口上启用 MPLS 交换
```

IOS_Feature

Switch Configurations

configure SPAN on 3550 SW1, source interface is fa0/8, destination port is fa0/9.no need config the port secure information.

```
Interface fastethernet0/8
Switchport mode access
Interface fastethernet0/9
Switchport mode access
!
No monitor session all
Monitor session 1 souce interface fastethernet0/8 both
Monitor session 1 destination interface fastethernet0/9
!
Show monitor
```

xSW1与SW2的F0/23，F0/24要做一CHANNEL，只允许VLAN_BB(50)通过。

大家在做的时候注意要先把F0/23,F0/24DOWN掉，等把Channel-group 1配到接口下时再no shut.

否则接口会出现DOWN状态。如果这样，大家可以no shut把接口唤醒

```
Interface range fastethernet0/23-24
Shutdown
Switchport encap dot1q
Switchport mode trunk
Switchport trunk allowed vlan 50 //show inter switchport 确保只有50
Channel-group 1 mode desirable
No shutdown
!
```

Trunk as a simple way to load balance traffic create 2 separate isl trunks between sw1 and sw2 ussing these ports.Trunk all odd vlan over the trunk you created between the fa0/23 port and trunk all even vlans over the trunk on fa0/24. trunk only the existing vlans.(vlan 11-16)

```
Inter fa0/23 //在3550 vlan config里面有load sharing技术
Sw trunk enc isl
Sw mode trunk
Spanning-tree vlan 11 port-priority 16 //default priority 128
Spanning-tree vlan 13 port-priority 16 //11,13,15走23，12，14，16走24。有冗余
Spanning-tree vlan 15 port-priority 16
Inter fa0/24
Sw trunk enc isl
Sw mode trunk
Spanning-tree vlan 12 port-priority 16
Spanning-tree vlan 14 port-priority 16
Spanning-tree vlan 16 port-priority 16
```

Voice Vlan:现要接一个IP PHONE型号7960连接在fa0/15上，要求语音数据走在voice vlan 200上，数据走在vlan 3上。语音电话支持802.1Q

```
Mls qos
!
Interface fastethernet 0/15
Mls qos trust cos //由于后面改cos为1所以这句到时可以考虑删掉
Switchport mode access
Switchport voice vlan 200
Switchport access vlan 3
!
Show interface fa0/15 switchport
```

在fa0/15端口上收到所有的包，将cos改为1

```
Interface fa0/15
mls qos cos 1 //Cos value to assign to untagged frame
Switch priority extend cos 1 //将接到IP Phone的PC的Cos改成1---应该可以不加
mls qos cos override //override tag frames with specified in the mls qos cos [cos_vlue]
!
```

在fa0/17配置dot1x认证方式，要求使用Radius server,没给出Radius server地址要求最少语句实现

```
aaa new-model
aaa authentication dot1x default group radius //radius-server host X.X.X.X auth-port X key X
dot1x system-auth-control
!
Inter fa0/17
Switch mode access
Dot1x port-control auto
!
Show dot1x inter fa0/17
```

在fa0/17端口下做Mac-address-list ,deny Ethernet type 6000

```
Mac access-list extended deny6000
Deny any any etype-6000
Permit any any
!
Inter fa0/17
Mac access-group deny6000 in
!
//Show mac access-group inter fa0/17
```

创建Vlan2056，交换机必须是transparent mode fa0/2连接一台PC,配置使其16秒进入Forwarding状态

```
Inter fa0/2
Switchport mode access
Switchport access vlan 2
```

Spanning-tree vlan 2 forward-time 8 // how long each of the listening and learning states last before the interface begin forwarding.

//Show spanning-tree vlan 2

如果需要将所有状态都要经过

Spanning-tree vlan 2 forward-time 5

Spanning-tree vlan 2 max-age 6

Fa0/12连接外部public network drop的机器，这台机器的ip address=x.x.x.x; mac=0000.8333.3333.保证没有其他机器可使用这个端口

Interface fastethernet 0/12

Switchport mode access

Switchport port-security

Switchport port-security maxium 1

Switchprot port-security mac-address 0000.8333.3333

Switchport port-security violation protect（当此端口连接了一个未授权用户后的端口状态）

某一SERVER连接到VLAND上,它不停发送大量的UDP数据包造成网络拥塞,UDP的端口号为4000,为了减轻网络的负荷,要求对其进行流量限制最高速率为256Kbps 和normal burst 64000 bit,超出后丢弃弃, 此处注意超出多少后丢弃?

Mls qos

Access-list 100 permit udp 135.1.68.0 0.0.0.31 any eq 4000

!

Class-map cisco

Match access-group 100

!

Policy-map ccie

Class cisco

Police 256000 8000 conform-action transit exceed-action drop（符合的经过，超出的丢弃），有些版本的没有conform-action

Interface fa0/?

Service-policy in ccie

!

在SW1 的VlanA (20)和F0/6(连R6 的接口）配置Fallback Bridge，并设置只Forward mac address:1234.1234.1234, ABCD.ABCD.ABCD.

bridge 1 protocol vlan-bridge

no bridge 1 acquire //preventing the forwarding of dynamically learned stations

bridge 1 address 1234.1234.1234 forword

bridge 1 address abcd.abcd.abcd forward

interface fa0/6 // 端口必须是三层口

bridge-group 1

interface vlan 20

bridge-group 1

//show bridge 1

在SW1 的f0/4 上配置Cos 和round roubin queue 的一个Map。题中已经指定cos 队列中包含的ip 优先级.

Mls qos

Interface fa0/4

Wrr-queue cos-map 1 6 7

Wrr-queue cos-map 2 4 5

Wrr-queue cos-map 3 2 3

Wrr-queue cos-map 4 0 1

//Show mls qos interface fa0/4 queue

配置这个Que 的Depth，Que1，2，3，4 的比例是10%,10%,20%,60% 基本的是5(queue 1 是10%,即10%是5). （还没看到这里）

Interface fa0/4

Wrr-queue bandwith 5 5 10 30

Sw2的f0/1口,禁止组播

Interface fa0/1

Switchport block multicast //switchport block unicast

//Show interface fa0/1 switchport

首先在两台交换机sw1 和sw2上分别通过某种配置可以用来快速配置全部的fastethernet口（不包括ge口）

Define interface-range ccie fa0/1-24

配置两台交换机使它们的全部fastethernet口，不允许发送DTP和VTP traffic

Interface range macro ccie

Switchport mode access //VTP

Switchport nonegotiate //DTP

Fa0/8为了避免拥塞，在收到pause frame时停止发包

Inter fa0/8

Flow control receive on

//show interface fa0/8 flowcontrol

做某个vlan22的primary root ,修改几个spanning 参数选项 hello:4s,forward-time:12s,max-age:16s

Spanning-tree vlan 22 root primary

Spanning-tree vlan22 hello-time 4

Spanning-tree vlan 22 forward-time 12

Spanning-tree vlan 22 max-age 16

//show spanning-tree vlan 22

Vlan x 需要storm control multicast control level 40

Inter fa0/?

Storm-control multicast level 40.00

bpduguard 配置在sw1 的F0/16

int f0/16

spanning-tree bpduguard enable

show spanning-tree summary

Sw2的fa0/1上配置minimum-reserve levels（还没看到这里）

给level3=80 packets,level4=150,level8=170分别应用在queue 1,2,3

Mls qos

Mls qos mini-reserve 3 80

Mls qos mini-reserve 4 150

Mls qos mini-reserve 8 170

Inter fa0/1

Wrr-queue min-reserve 1 3

Wrr-queue min-reserve 2 4

Wrr-queue min-reserve 3 8

路由器feature配置

在未来会在vlanB里添加一台DRP server. Config R6 to support the distributed director agent function

I don’ t konw

DRP Server Agent

Description

The Director Response Protocol (DRP) is a simple User Datagram Protocol (UDP)-based application developed by Cisco Systems. It enables Cisco’s DistributedDirector product to query routers (DRP Server Agents) in the field for Border Gateway Protocol (BGP) and Interior Gateway Protocol (IGP) routing table metrics between distributed servers and clients. DistributedDirector, a separate standalone product, uses DRP to transparently redirect end-user service requests to the topologically closest responsive server. DRP enables DistributedDirector to provide dynamic, scalable, and "network intelligent" Internet traffic load distribution between multiple geographically dispersed servers.

DRP Server Agents are border routers (or peers to border routers) that support the geographically distributed servers for which DistributedDirector service distribution is desired. Note that, because DistributedDirector makes decisions based on BGP and IGP information, all DRP Server Agents must have access to full BGP and IGP routing tables.

Ip drp server

Ip drp access-group 1

Access-list 1 permit ip 135.1.26.0 0.0.0.255

//Show ip drp

Web-Cache Engine:

config on R5 let it can intercept the WWW traffic on f0/0 and redirect to cache-engine VLAN_B

This chapter describes how to configure your switch to redirect traffic to cache engines (web caches such as the Cisco Cache Engine 550) by using the Web Cache Communication Protocol (WCCP). WCCP is a Cisco-developed content-routing technology that you can use to integrate cache engines into your network infrastructure. The cache engines transparently store frequently accessed content and then fulfill successive requests for the same content, eliminating repetitive transmissions of identical content from web servers. Cache engines accelerate content delivery and ensure maximum scalability and availability of content. In a service-provider network, you can deploy the WCCP and cache engine solution at the points of presence (POPs). In an enterprise network, you can deploy the WCCP and cache engine solution at the regional site and the small branch office.

ip wccp web-cache // ip wccp {web-cache | service-number} [group-address groupaddress] [redirect-list

access-list] [group-list access-list] [password password]

group-address (mulicast address)

group-list Directs the router to use an access list to determine which cache engines are allowed to participate in the service group.

Redirect-list Directs the router to use an access list to control traffic redirected to this service group.

inter fa0/0

ip wccp web-cache redirect out //fundamental cofig :system management

//show ip wccp

在R1 配置SNMP 和rmon snmp community is “cisco” Trap host 150.100.1.252 当cpu 达到70%或者下降到40%的时候候, log and trap.间隔时间为1分钟钟, R1配置置:OID VIEW lsystem.57.0

Snmp-server community string cisco ro

snmp-server host 150.100.1.252 cisco

rmon alarm 1 OID VIEW lsystem.57.0 60 absolute rising-threshold 70 1 falling-threshold 40 2 owner ccie // absolute : Tests each MIB variable directly

rmon event 1 log trap cisco description “up to 70%” owner ccie

rmon event 2 log trap cisco description “down to 40%” owner ccie

//show rmon alarms

//show rmon event

//show snmp

在r5上配置snmp,给定150.100.100.100是snmp server,由于安全原因只允许这台服务器对r5进行snmp操作，通讯字符串是”CiscoWorks”要求有读写权限。R5只发送bgp trap,允许网管主机对r5进行重起操作，发送trap时要带着通讯字符串一起发送

Access-list 10 permit 150.100.100.100

Snmp-server community CiscoWorks rw 10

Snmp-server enable traps bgp //enable bgp traps in system

Snmp-server host 150.100.100.100 traps CiscoWorks bgp //only send bgp traps to host

Snmp-server system-shutdown

Config r3 as tftp server, r5 can get ios from it,use name 2600-ios-img, real image is c2600-js-mz.121-11c.img,don’t config r5

Tftp-server flash:c2600-js-mz.121-11c.img alias 2600-ios-img [access-list number] //config basic file transfer service 如果只是r5要考虑加access-list

Configure R1, R3 and R4’s frame relay interfaces so that so that broadcast flooding can be avoided. Use the following parameters:

Maximum transmission rate: 160 packets per second.

Maximum byte transmission rate of 240k bytes per second.

Limit the queue size to 80 packets.

Inter s0/0

Frame-re broadcast 80 240000 160 //frame-relay broadcast-queue [size] [byte-rate][packetrate]

R2's Se0/0 interface is experiencing excessive link flap causing peer routers in the network to recalculate best paths. This is consuming substantial amounts of system processing resources and cause routing protocols to lose synchronization with the state of the flapping interface.

Configure R2 flapping interface to be removed from the network until it stabilizes, thus improving convergence times and stability throughout the network by isolate link failures so that disturbances are not propagated.

Inter s0

dampening

//show dampening interface

//show interface dampening - summary of dampening parameters and status

Damping

value 1 多长时间内没变化后flap值衰减到500（半衰期值，默认是15 minutes）

value 2 当此值衰减到一定量后，将不被抑制（再使用值，默认是750）

value 3当此值达到一定量后，将被抑制（抑制值，默认是2000）

value 4 最大的抑制时间（默认是半衰期的4倍）

Upd broadcast management

Configure the ATM interface on R6 to forward incoming Mobile IP registration broadcast to 150.100.1.254 on R2's Backbone 2 Ethernet.

ip forward-protocol udp mobile-ip

Inter atm 3/0

Ip help-address 150.100.2.254

Configure HSRP on R3 and R4 for hosts on VLAN A: Carefully read the requirements:

There are 10 hosts on VLAN A。5 Windows PC's and 5 Unix workstations

The Windows PC's are configured to use 160.YY.10.50 as their default gateway

The UNIX workstations are configured to use 160.YY.10.100 as their default gateway

R3 is the preferred router for Windows users.

R4 is the prefered router for Unix users.

If the frame relay connection on R3 goes down, then R4 becomes the preferred router for all users。

If the frame relay connection on R4 goes down, then R3 becomes the preferred router for all users。

Once either R3 or R4 recover from a failure then the network must operate as outined above, I.e Each router must resume their original role。

R3:

Standby use-bia

Standby 1 ip 160.1.10.50

Standby 1 preempt

Standby 1 priority 105

Standby 1 track s0

Standby 2 ip 160.1.10.100

Standby 2 preemt

R4:

Standby use-bia

Standby 2 ip 160.1.10.100

Standby 2 track s0

Standby 2 priority 105

Standby 2 preempt

Standby 1 ip 160.1.1.50

Standby 1 preempt

//show standby all

在VLANC上的主机不想配置第二个网关，不允许使用HSRP，使得VLANC的主机优选R6的E0端口地址为网关，其次选R5的Fa0/0端口地址为网关。考题中使用的是best最高级，所以用最大和最小的优先级

ICMP Router Discovery Protocol (IRDP)

Router discovery allows the switch to dynamically learn about routes to other networks using IRDP. IRDP allows hosts to locate routers. When operating as a client, the switch generates router discovery packets. When operating as a host, the switch receives router discovery packets. The switch can also listen to Routing Information Protocol (RIP) and Interior Gateway Routing

Protocol (IGRP) routing updates and use this information to infer locations of routers. The switch does not actually store the routing tables sent by routing devices; it merely keeps track of which systems are sending the data. The advantage of using IRDP is that it allows each router to specify both a priority and the time after which a device is assumed to be down if no further packets are received.

Each device discovered becomes a candidate for the default router, and a new highest-priority router is selected when a higher priority router is discovered, when the current default router is declared down, or when a TCP connection is about to time out because of excessive retransmissions.

The only required task for IRDP routing on an interface is to enable IRDP processing on that interface. When enabled, the default parameters apply. You can optionally change any of these parameters.

R6:

Inter e0/0

Ip irdp

Ip irdp preference 2147483647

R5:

Inter fa0/0

Ip irdp

Ip irdp preference -2147483647

//Show ip ipdp fa0/0

VRRP VLAN D 上冗余网关使用VRRP，优选R5,要认证(另一说：是优选R6，看题意决定！)

R5

Inter fa0/1

Vrrp 1 ip 1.1.65.1

Vrrp 1 priority 105 //default 100

Vrrp 1 authentication text cisco //preempt默认enable

R6

Inter fa0/1

Vrrp 1 ip 1.1.65.1

Vrrp ip authentication text cisco

//show vrrp [brief|group] show vrrp interface

ip accounting

记录R5 的bri0 口所有进出(IN & OUT)的流量，基于任何源任何目的,以字节统计（ip accounting）；从ATM BB 过来的流量不容许使用isdn 链路，并且记录被拒掉的流量的子节数。

Ip access-list extend ccie

deny ip 198.2.3.0 0.0.0.255 any

deny ip 4.1.1.0 0.0.0.255 any

permit ip any any

inter bri 0

ip accounting //自动记录所有进出流量

ip accounting access-violations //provide infor identifying ip traffic that fail access-list

Enables IP accounting with the ability to identify IP traffic that fails IP access lists.

ip access-group ccie out //isdn发出去的流量

//show ip accounting [checkpoint][output-packet][access-violation]

R5作为DHCP Server,要求domain-name:cisco.com给以太网分配ip address,DNS-server:150.100.1.50,150.100.1.51,永不释放ip address 指定网关YY.YY.14.1

Service dhcp

No ip dhcp conflict logging

Ip dhcp excluded-address 1.1.14.1

Ip dhcp excluded-address 1.1.14.2

Ip dhcp excluded-address 1.1.14.5

Ip dhcp pool ccie

Network 1.1.14.0 /24

Domain-name cisco.com

Dns-server 150.100.1.50 150.100.1.51

Default-router 1.1.14.1

Lease-time infinite

R4和sw2要和r3同步，同步后r4和sw2的strutum is 3使用authentication.It can be synchronize only if there is path between routers

R3

Ntp master 2

Ntp source loopback 0

Ntp authentication

Ntp authentication-key 1 md5 cisco

Net trunsted-key 1

R4

Ntp server 1.1.3.3 key 1 source loopback 0 //指定哪个端口接受ntp信息

Ntp authentication

Ntp authentication-key 1 md5 cisco

Ntp trusted-key 1

//show ntp status show ntp associations

On R4,send logging information to 150.100.2.250,logging information includes emergency,alerts and critical only.The logging facility is local 6

Logging 150.100.2.250

Logging facility local6

Logging logging trap critical //emer 0,alerts 1,critical 2,向下兼容。也可以使用数字

NAT (3 Points)

在R1 向BB1 隐藏YY.YY.0.0 的地址，不得建立新的端口，不能使用静态路由，可以使用1YY.0.0.1/24 这个地址，地址池可以使用此网段地址，而且其他路由器还能Ping 通150.100.1.254 这个地址。

int loop0

ip add 1YY.0.0.1 255.255.255.0 sec /*这个初始化时就设好了*/

inter s0

ip nat inside

inter s1

ip nat inside

inter e0/0

ip nat outside

ip nat pool ccie permit 1yy.0.0.2 1yy.0.0.254 prefix 24

ip nat inside souce list 10 pool ccie

access-list 10 permit yy.yy.0.0 0.0.255.255

access-list 5 permit 1yy.0.0.1 0.0.0.255

route-map c2r permit 10

match ip address 5

route-map r2o deny 10

match ip address 5

route-map r2o permit 20

!

router rip

redistribute connected route-map c2r //将1yy.0.0.1通告进rip使bb1可以看到但题目说r1 不发路由给BB1，distribute-list 10 out e0/0 //只将1yy.0.0.1通告给bb1

!

Router ospf 1

Redistribute rip metric 200 metric-type 1 sub route-map r2o //不将1yy.0.0.1通告给ospf

!

//show ip nat statistics

//show ip nat translations

tunnel

R6与R3之间要建一TUNNEL,使SW2 在trace SW1的VLANA地址时为:

trace YY.YY.23.7

YY.YY.66.6 YY.YY.3.3 YY.YY.23.7

R6:

Inter tunnel 0

Ip unnumber e0/0 //yy.yy.66.6

Tunnel source loopback 0

Tunnel destination y.y.3.3

Access-list 110 permit udp any y.y.23.7 0.0.0.0

Route-map ccie

Match ip address 110

Set interface tunnel 0 //set ip next-hop ip-address

inter e0/0

ip policy route-map ccie

R3 :

Inter tunnel 0

Ip unnumber loopback 0

Tunnel source loopback 0

Tunnel destination yy.yy.66.6

QoS

Wred Weighted Random Early Detection

config random detect on s0/1 of R4. Config critical information with that lowest queue size is 32, maximum queue size is 2000 and dropping rate is one every 50 packets.

Inter s0/1

Random-detect //不需要在后面再加precedence

Random-detect precedence 5 32 2000 50

//show queueing interface s0/1, show queueing random-detect

R3来的VLAN_A(135.1.37.0)的数据流进入OSPF区域时做分类,使得WWW数据流具有 Precedence为5,其它数据流 Precedence 为3,在S0/0的OUT方向做做!

Access 110 permit tcp 135.1.37.0 0.0.0.255 any eq www

Class vlana

Match access-group 110

Policy-map ccie

Class vlana

Set ip precedence 5

Class class-default

Set ip precedence 3

Inter s0/0

Service-policy out ccie

//show policy inter s0/0

frame relay link effitiancy

在R1上设置,使从BB1来的数据流在通过S0/1进入网内时,要作为最小的 Low-latency的Real-Time的数据流流,R1与R3之间的Frame-relay在 512Kbps Bandwidth及640 bytes frame size时性能最好.考题意思是real-time流量要得到low-latency服务

frame-relay ip rtp priority

To reserve a strict priority queue on a Frame Relay permanent virtual circuit (PVC) for a set of Real-Time Transport Protocol (RTP) packet flows belonging to a range of User Datagram Protocol (UDP) destination ports, use the frame-relay ip rtp priority map-class configuration command. To disable the strict priority queue, use the no form of this command.

http://www.cisco.com/univercd/cc/td/doc/product/software/ios122/122cgcr/fqos_r/qrfcmd2.htm#wp1089893

Map-class frame-relay ccie

Frame-relay fragment 640

Frame-relay traffice-rate 512000

Fram-relay ip rtp priority 16384 16380 100 //frame-relay ip rtp priority [start-udp][udp-range][bandwidth kbps] //reserve a strict priority queue on a Frame Relay PVC for a set of Real-Time Transport Protocol (RTP) packet flows belonging to a range of User Datagram Protocol (UDP) destination ports

Inter s0/1

Frame-relay traffice-shaping //起LLC必须加fr traffic shaping&Frame fragmentation

Frame-relay class ccie

Frame-relay map ip x.x.x.x xxx broad rtp header-compression active //qos_feature link eff里

CBWFQ

Allocate 4Mbps in the Ethernet 0/0 interface on r6 for traffic coming from Backbone

2, Your solution should apply to all possible traffic from Backbone 2 on Fast Ethernet 1/0。

Class-map bb2

Match input-interface e0/0

Policy-map ccie

Class bb2

Bandwidth 4000000

Class class-default

Inter s0/0

Service-policy in ccie

Configure R1 so that any received IP taffic sourced from VLAN_B will be forwarded to R4

with an IP precedence of 3。

Access-list 1 permit 1.1.12.0 0.0.0.255

Route-map ccie permit 10

Match ip add 10

Set ip precedence 3

Set ip next-hop 1.1.11.4

Inter fa0/0

Ip policy route-map ccie

CBWFQ

来自195.1.1.0/24 的流量占用15%，来自195.1.2.0/24 的流量占用30%，假设VOICE 流量

IP Precedence 值为Critical, 他们占用45%的带宽，其他应用占用剩余的带宽，不要为剩余流量创建Class.

Access-list 10 permit 195.1.1.0 0.0.0.255

Access-list 11 permit 195.1.2.0 0.0.0.255

Class-map flow1

Match access-group 10

Class-map flow2

Match access-group 11

Class-map voip

Match ip precedence 5

!

Policy-map ccie

Class flow1

Bandwidth percent 15

Class flow2

Bandwidth percent 30

Class voip

priority percent 45 //set LLC for CBWFQ

!

Interface s0/0

Max-reserved-bandwidth 100 // The percent argument specifies the maximum percentage of the total interface bandwidth that can be used

Service-policy output ccie

//show policy inter s0/0

Performance tune

VLANB 的客户抱怨到VLANBB1 的基于WEB OR FTP 访问的应用较慢，而且明显R3 的带宽占用率没有满负荷。尽管可能需要进一步调整，使返回的数据包，当发生拥塞时，较普通数据被drop 的概率为低。请先给出一个合适的解决方法。R1连BB1,R1和R3通过s0/0用frame-relay连接

R1

Access 110 permit tcp 150.100.1.0 0.0.0.255 eq www 142.1.30.0 0.0.0.255

Access 110 permit tcp 150.100.1.0 0.0.0.255 eq ftp 142.1.30.0 0.0.0.255

Access 110 permit tcp 150.100.1.0 0.0.0.255 eq ftp-data 142.1.30.0 0.0.0.255

!

Route-map ccie permit 10

Match ip add 110

Set ip precedence 5

Route-map ccie permit 20

!

inter e0/0

ip policy route-map ccie

Inter s0/0

Random-detect

dscp

在R6 上从vlanC 来的路由分类 Class1：match cs5 size200－1500

Class2：match cs5 size＝500

第一个class 要求：当dscp 为cs5 及包的大小在200-1000 之间时，ip precedence 为

immediate;

第二个class 要求：当dscp 为cs5 及包的大小为500 时，ip precedence 为priority

Access-list 10 permit ip 1.1.65.0 0.0.0.255

Class class1

Match access-group 10

Match dscp cs5

Match packet length min 200 max 1500

Class class2

Match access-group 10

Match dscp cs5

Match packet length min 500 max 500

Policy-map ccie

Class class1

Set ip precedence immediate

Class class2

Set ip precedence priority

Inte e0/0

Service-policy in ccie

R5 的S0 口帧中继拥塞管理，ECN 的percentage of maximum queue size=60%,不要用FR 的流量整形

Inter s0

Frame-relay congestion-management //要不要frame-relay switching

Threshold ecn 60 // When the output interface queue reaches or exceeds the ECN excess threshold, all Frame Relay DE bit packets on all PVCs crossing that interface will be marked with FECN or BECN

Class-based pacet marking

在R1上做配置，使从vlan_a的user进入的www traffic的ip precedence为3，其他所有traffic是precedence 1.要求不能使用CAR和PBR

Access-list 110 permit tcp 1.1.15.0 0.0.0.255 any eq www

Class-map www

Match access-group 110

Policy-map ccie

Class www

Set ip precedence 3

Class class-default

Set ip precedence 1

Inter e0/0

Service-policy input ccie

Class-based WFQ

Configure R4 so that if congestion between 9 to 10 AM user on VlanD will have 20% of the bandwidth reserved for web traffic to server 199.172.11.11 on vlan_bb1 and 20% for telnet to all device within your network topology. At other times no bandwidth should be reserved percentage are based on the available interface bandwidth.

Time-range 9to10

Periodic daily 9:00 to 10:00

Access 110 permit tcp 1.1.13.0 0.0.0.255 199.172.11.11 0.0.0.0 eq www time-range 9to10

Access 111 permit tcp any any eq 23

Class-map www

Match access-group 110

Class-map telnet

Match access-group 111

Policy-map ccie

Class www

Bandwidth remaining percent 20 // Amount of guaranteed bandwidth, based on a relative percent of available bandwidth

Class telnet

Bandwidth remaining percent 20

Inter s0

Service-policy output ccie //不用max-reserved-bandwidth 因为没说要使用100%接口带宽

Dicard Eligible

The frame-relay between R1 and R6 experimenting heavy congestion this should result in ospf lost neighbor.configure r1 and r6 so that frame-relay provider does not drop any ospf packt during congestion.

R1:

Access-list 110 deny ospf any any

Access-list 110 permit ip any any

Frame-relay de-list 1 protocol ip list 110

Inter s0/0

Frame-relay de-group 1 [dlci]

R6:

Access-list 110 deny ospf any any

Access-list 110 permit ip any any

Frame-relay de-list 1 protocol ip list 110

Inter s0/0

Frame-relay de-group 1 106

//show frame pvc 106

R6-r1(601) 的帧中继上因为存在着语音流量，为了保证语音的质量，所以把数据流量超过40K分数据段给丢弃掉 ，语音流量不要管

R6/R1:

int s0

frame-relay traffic-shapping

int S0.1

frame-relay interface-dlci 601

class Frf1.1

vofr //起fragment11对语音不分段，超过队列的分段数据丢弃

map-class frame Frf1.1

frame frag 40

frame-relay fair-queue //默认关闭 When Frame Relay fragmentation is enabled, weighted fair queueing is the only queueing strategy allowed.

NBAR

R6的atm3/0口上做inbound的http下载的限速，限速100k 下载的image扩展名(expand name)包括gif,jpeg,jpe的文件

Class-map image match any

Match protocol http url *.(jpg|jpeg|gif)

Policy-map nbar

Class image

Police 100000

!

Inter atm3/0

Ip nbar protocol-discovery // Protocol discovery provides an easy way to discover application protocols transiting an interface so that QoS policies can be developed and applied

Service-policy input nbar

FR traffic shaping

要求：在R3的FR端口

1 isp对超过48kbps的流量加de标志

2 isp 对速率64kbp以上的流量会drop

3 tc=125ms

4 发生拥塞时，收到becn后降低速率，直到16kbps

R3

Map-class frame-relay ccie

Frame-relay cir 48000 //超过cir的加de位

Frame-relay bc 6000 //Tm=125 ms Bc=48000*1/8

Frame-relay be 2000 //64000/8=8000 be=8000-6000

Frame-relay minicir 16000

Frame-relay adaptive-shaping becn

CQ

P2P 1% bandwidth,max packet 512byte;use udp port range 1000-1500;Ftp 10%bandwidth;max packet 1k byte;

Access-list 101 udp any any port range 1000 1500

Access-list 102 tcp any any eq ftp

Access-list 102 tcp any any eq ftp-data

Queue-list 1 protocol ip 1 list 101

Queue-list 1 protocol ip 2 list 102

Queue-list 1 queue 1 byte-count 512

Queue-list 1 queue 2 byte-count 5120

Queue-list 1 queue 3 byte-count 45568

Inter fa0/0

Custom-queue-list 1

!

//show queueing custom

Priority queue

The priority queue should be done on R2 R4 R5 R6的s0口，并将ip precedence 5的流量放到high

Access-list 110 permit ip any any precedence 5

Priority-list 1 protocol ip high list 110

Inter s0

Priority-group 1

//show queue priority

CAR

网内的计算机要运行BB1那边的一个服务器150.100.1.240的程序,使用udp端口5000-6000。要求对此应用进行速率限制，基本速率是3Mbps,normal burst rate is 200kbps,excess burst rate is 300kbps.符合这个限制的以高优先级传送，不符合的数据则把数据包优先级设置为普通并进行best-effort转发。除此以外的数据。基本速率是800kbps,normal burst rate is 100kbps,excess burst rate is 150kbps.此数据包优先级为普通，如果超过Be则要求drop在R1 E0上做

Access-list 110 permit udp any 150.100.1.240 0.0.0.0 range 5000 6000

Inter e0/0

Rate-limit output access-group 110 3000000 25000 37500 conform-action set-prec-transmit 5 exceed-action set-prec-transmit 0

Rate-limit output 800000 12500 18750 conform-action set-prec-transmit 0 exceed-action drop

R5 和 R6 frame-relay run the RTP head compression. All the VC should be compressed,but only when the arriving package be compressed you can run the compression .And it needn’ t TCP head compression

R5

Inter s1

Frame-relay ip rtp header-compress

R6:

Inter s0

Frame-relay ip rtp header-compression passive //因为要所有的PVC压缩。所以不能加在dlci上

//Show frame-relay ip rtp header-compression

Security

Tcp intercept

VlanA有PC对BB1的服务器150.100.1.240进行syn flooding close, the tcp connection even the legitimate which open after 2.5 minuts.In TCP intercept,the original file said the server is 150.100.1.24, but in his lab,the attack is it.

Access-list 110 permit tcp 148.1.X.0 0.0.0.255 150.100.1.240

Ip tcp intercept list 110

Ip tcp intercept mode intercept

Ip tcp intercept connection-timeout 150

//show tcp intercept connections show tcp intercept statisics

ICMP flooding

怀疑ICMP 来自R5 与R2 连接的帧中继链路上,use access-list 日志跟踪出发送大量无效数据报的攻击源地址。access name ICMPTRACKER，your cannot deny any normal packets.

R5:

Ip access-list extended ICMPTRACKER

Permit icmp any any log-input

Permit ip any any

Inter s0/0

Ip access-group ICMPTRACKER in

要求只能YY.YY.28.0 网段telnet 至R4，用户名：joe，密码：joeshowbgp，telnet 上来后自动显示show ip bgp summary 后断掉

Access-list 110 permit tcp 1.1.28.0 0.0.0.255 any eq 23

Username joe password joeshowbgp

Username joe autocommand show ip bgp summary

Line vty 0 4

Login local

Access-class 110 in

DF(DATA FRAGMENT)

管理员发现有大量的设置了ＤＦ位的超大的数据包攻击BB2 上某个web－server,ip 为150.100.2.4, 非初始分段包到此服务器拒绝，初始包到此服务器允许.

Access-list 110 deny ip any 150.100.2.4 0.0.0.0 fragments

Access-list 110 permit ip any any

Inter s0/0

Ip access-group 110 in

1. Time-Ranges ACL. At weekdays, from 8:00 to 18:00, don’t permit any www traffic on VLANB,and at weekend, form noon to 16:00, don’t permit any UDP traffic. Configured on R5 for it.

2. 在R5 的F0/0 做access-list, 过滤RFC1918 规定的三个网段的私有地址和127.0.0.0/8 这个地址。

R5:

Clock timezone GMT +8

Clock set xx:xx:xx day month year //可以考虑不加

time-range weekdays

period weekdays 8:00 to 18:00

time-range weekend

period weekend 12:00 to 16:00

access-list 110 deny tcp 135.1.57.0 0.0.0.255 any eq www time-range weekdays

access-list 110 deny upd any any time-range weekend

access-list 110 deny ip 10.0.0.0 0.255.255.255 any

access-list 110 deny ip 127.0.0.0 0.255.255.255 any

access-list 110 deny ip 172.16.0.0 0.15.255.255 any

access-list 110 deny ip 192.168.0.0 0.0.255.255 any //没说哪个是目的哪个是源，需不需要any

access-list 11p permit ip any any

inter fa0/0

ip access-group 100 in

ICMP/Smurf with CAR

在R6上的F0/0下,对ICMP的smurf进行inbond的速率限制制,题目要求用CAR, 限制Smurf最高速率为256kbps, normal-burst 8Kbps(要看清此处是BPS,还 是BYTE)

Access-list 110 permit icmp any any eq echo

Access-list 110 permit icmp any any eq echo-reply

Inter fa0/0

Rate-limit input access-group 110 256000 1000 conform-action transmit exceed-action drop

在R2以内的区域为clean区域，从R2到 BB1之间的区域是 Dirty区域，在R2的子接口上配置：两个域之间都允许routing和ICMP包通过，外出包允许TCP、UDP，进入的包不允许有其它访问.(fa0/0是单臂路由，fa0/0.1 rip连bb1----150.100.1.0 fa0/0.2是eigrp------133.1.12.0)

当你配完以上的访问列表后你发现从R2上不再能TELNET到R1上,请解决此问题

R2

Ip access-list extended outband

Permit icmp any any

Permit eigrp any any

Permit tcp any any eq bgp

Permit tcp any any reflect ccie

Permit udp any any reflect ccie

Ip access-list extended inbound

Permit eigrp any any

Permit tcp any any eq bgp

Permit icmp any any

Evaluate ccie

Inter fa0/0.1

Ip access-group inbound in

Ip access-group outbound out

Ip access0list extended inbound

Permit tcp 133.1.12.1 eq 23 any //reflexive acc不支持从本地路由器发起的

RFC1918 Filtering（没有MICROSOFT 的，还要求不Filter 10.0.0.0）.

Configure an appropriate ACL on R2’s Backbone1 interface to prevent spoofing attacks originate from RFC1918 addresses and the Microsoft reserved looback address。

All attacks must be logged 。

Non-spoofed packets should be processed without logging。

Access-list 110 deny ip 172.16.0.0 0.15.255.255 any log-input

Access-list 110 deny ip 192.168.0.0 0.0.255.255 any log-input

Access-list 110 permit ip any any

Inter fa0/0

Ip access-group 110 in

Ip Spoofing

Setup R5 such that it only forwards the packets from the Frame Relay network which have a source address found in its routing table。Log packets that do not meet this requirement。

Access-list 100 deny ip any any log-input

Interface s0/0

Ip verify unicast reverse-path 100 // If an ACL is specified in the command, then when (and only when) a packet fails the Unicast RPF check, the ACL is checked to see if the packet should be dropped (using a deny statement in the ACL) or forwarded (using a permit statement in the ACL).

只允许ip address为yy.yy.0.0/16和150.100.x.x/16的host telnet 到sw2上

Access-list 1 permit 1.1.0.0 0.0.255.255

Access-list 1 permit 150.100.0.0 0.0.255.255

Line vty 0 4

Access-class 1 in

Support engineers who come in through vlan_d must be given occasional access to the rest of your yy.yy.0.0 net they will telnet from sw1 to r4 e0/0 and give their crendentials once this is done

they will have unrestricted access to the rest of the network .Use local authentication on r4 allows unrestricted access for 10 minutes and 2 minutes idle.make sure the existing feature of this link(connectivity,route)is not compromise.User name is ccie,password is cisco

R4:

Username ccie password cisco

Username ccie autocommand access-enable timeout 2

Access-list 101 permit tcp 1.1.13.0 0.0.0.255 1.1.13.4 0.0.0.0 eq 23

Access-list 101 permit ospf any any //icmp要不要考虑，连通性问题怎么解决

Access-list 101 permit icmp 1.1.13.7 0.0.0.0 any

Access-list 101 dynamic-list ccie timeout 10 ip 1.1.13.7 0.0.0.0 1.1.0.0 0.0.255.255

Inter e0/0

Access-group 101 in

Line vty 0 4

Login local

ATM configuration

接口IP ADDRESS=192.3.YY.1/24

对端IP ADDRESS=192.3.YY.254/24

End System ID=4000000000YY.00

Backbone 的SNAP 地址是xx.xxxx.xx.xxxxxx.xxxx.xxxx.xxxx.xxxx.x400.01yy.00

要求Ping 通Backbone 地址，并接收EIGRP 路由。

Inter atm3/0

Ip address 192.3.1.1 255.255.255.0

Pvc 0/5 qsaal

Pvc 0/16 ilmi

Atm esi-address 400000000011.00

Svc ccie nsap xx.xxxx.xx.xxxxx.xxxx.xxxx.xxxx.xxxx.x400.0101.00 //Once you specify a name for an SVC, you can reenter interface-ATM-VC configuration mode by simply entering the svc name command

(config-if-atm-vc)# Protocol ip 192.3.1.254 broadcast

(config-if-atm-vc)#encap aal5snap //default

//show inter atm 3/0

//show atm vc

R6 have atm3/0 ip address is 192.168.YY.1/24. Use PPP over ATM. The other end ip address is 192.168.YY.254 with vpi 0. pvc number is 100+YY. From your routers, you can ping the interface.

Inter atm3/0

Pvc 0/101

Encap aal5ciscoppp virtual-template 1

Inter virtual-template 1

Ip address Ip address 192.168.1.1 255.255.255.0 //需要encap ppp么？

//show inter atm 3/0

//show atm vc

Configure ATM on R6 as follows:

1) Your IP address for the ATM cloud is 192.1.YY.1/24

2) There is a PVC configured between your router and the ATM backbone router. Your end of the PVC is VPI=0 and VCI = '100+YY', where 'YY' is your rack number. For examples on Rack 1 your VCI is 101, on Rack2 it's 102, etc.

3) 题目要求不能用明显的映射命令(explicit map),即使用默认的inverse-arp

4) You must be able to ping the remote ATM router of 192.1.YY.254.

Inter Atm 3/0

Ip address 192.1.1.1 255.255.255.0

Pvc 0/101 // Inverse ARP is enabled by default when you create a PVC using the pvc command

(config-if-atm-vc)#broadcast // Sends duplicate broadcast packets for all protocols configured on a PVC

(config-if-atm-vc)#Encap aal5snap

//show inter atm 3/0

//show atm vc

Atm-svc:ip address of the local side is 192.3.yy.1,remote side is 192.3.yy.254. atm address of the local side is xxxxxxxxxxxxxxxxxxxxxxxxxxxx.xx, nsap address of the remote side is 47xxxxxxxxxxxxxxxxxxxxxxNN.00 这里的NN是你所在的rack号码的16进制表示的值。例如rack07的NN是07。Rack10的NN是0A

After finish this config,you must be able to ping the ip address of the remote side.

Don't use rfc1577 classical arp.

Don't use sub-interface

Inter atm3/0

Ip address 192.3.1.1 255.255.255.0

Pvc 0/5 qsaal

Pvc 0/16 ilmi

Atm esi-address xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx.xx

Svc ccie nsap address 47xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx01.00 //Once you specify a name for an SVC, you can reenter interface-ATM-VC configuration mode by simply entering the svc name command

(config-if-atm-vc)# Protocol ip 192.3.1.254 broadcast

(config-if-atm-vc)#no protocol ip arp

(config-if-atm-vc)#encap aal5snap //default

//show inter atm 3/0

//show atm vc

IPv6 base

在R1(lo0,e0/0.2),R2(lo0 fa0/0.1 fa0/0.2 s0),R3(s0 lo0 fa0/0) R5(fa0/0 lo0 s0/0.2)上启用IPV6,采用Site-local地址，采用相应接口的IPv4的第三个八位字节作为子网地址，IPv4的第四个八位字节作为主机地址。

（注意ipv6的子网位是第4个16位字节）

IPV6 RIP

在R1,R2, R3,R5上启动IPV6 RIP

R1:

Ipv6 unicast-routing

Inter loopback 0

Ipv6 address fec0:0:0:1::1/64

Ipv6 router rip cisco

!

Inter e0/0.2

Ipv6 address fec0:0:0:C::1/64

Ipv6 router rip cisco

!

Ipv6 router rip cisco

ipv6 unicast-routing

ipv6 router rip cisco

interface Loopback0

ipv6 addr fec0:0:0:2::2/64

ipv6 rip cisco enable

```
interface Ethernet0/0.2
ipv6 addr fec0:0:0:c::2/64
ipv6 rip cisco enable
interface Ethernet0/0.1
ipv6 addr fec0:0:0:17::2/64
ipv6 rip cisco enable
interface serial0/0
ipv6 addr fec0:0:0:eb::2/64
frame-relay map ipv6 FEC0:0:0:eb::5 205 broadcast
frame-relay map ipv6 FEC0:0:0:eb::3 205 broadcast
frame-relay map ipv6 R3-link_local 205 broadcast
frame-relay map ipv6 R5-link_local 205 broadcast
ipv6 rip cisco enable
R3
ipv6 unicast-routing
ipv6 router rip cisco
interface Loopback0
ipv6 addr fec0:0:0:3::3/64
ipv6 rip cisco enable
interface Ethernet0/0
ipv6 address FEC0:0:0:17::3/64
ipv6 rip cisco enable
interface serial0/0
ipv6 address fec0:0:0:eb::3/64
frame-relay map ipv6 fec0:0:0:eb::5 305 broadcast
frame-relay map ipv6 fec0:0:0:eb::2 305 broadcast
frame-relay map ipv6 R2-link_local 305 broadcast
frame-relay map ipv6 R5-link_local 305 broadcast
ipv6 rip cisco enable
R5
ipv6 unicast-routing
ipv6 router rip cisco
interface Loopback0
ipv6 addr fec0:0:0:5::5/64
ipv6 rip cisco enable
interface serial0/0.1
ipv6 address fec0:0:0:eb::5/64
frame-relay map ipv6 fec0:0:0:eb::2 502 broadcast
frame-relay map ipv6 fec0:0:0:eb::3 503 broadcast
frame-relay map ipv6 R2-link_local 502 broadcast
```

frame-relay map ipv6 R3-link_local 503 broadcast

ipv6 rip cisco enable

//link-local的地址通过show ipv6 inter 来看

ipv6

r4 e0 2068:yy:yy:44::4/64

r1 e0 2068:yy:yy:11::1/64

要求不再分配其他的ipv6地址，在r1和r4之间作隧道，运行ripng，使其能相互ping 通

并且，隧道的endpoints要足够的stable

r1:

ipv6 unicast-routing

ipv6 router rip cisco

int e0/0

ipv6 add 2068:yy:yy:44::4/64

ipv6 rip cisco enable

int tunnel0

ipv6 unnumbered e0/0

tunnel source lo0

tunnel destination yy.yy.4.4

tunnel mode ipv6ip auto-tunnel

ipv6 rip cisco enable

r4

同r1