**华南理工大学《高级计算机网络》复习资料**

**[Part1].IPv6协议**

## 【IPv4的危机】

地址危机（最大的危机，但可通过NAT、CIDR、VLSM）、端到端业务模式无法实施、QoS和性能问题、配置复杂、安全问题、路由表的膨胀、移动性支持不够

# 【IPV4与IPV6地址空间】

IPV4：232 ≈ 43亿

IPv6：2128=3.4×1038 ≈ 340涧

# 【IPv6的改进之处】

简化的报头格式，地址扩展到128位，增强的安全性和服务质量，可以实现更高效得路由基础，提供了对移动特性得支

**【IPv9网络】**

每个y是32bit部分，如

。。。

**【TCP/IPv6参考模型】**



**【IPv6特点】**

地址及寻址

**全新的报文格式，高效的报头**

**全新的地址配置方式，即插即用**

更好的QoS支持

内置的安全性

**全新的邻居发现协议**

良好的扩展性

内置的移动性

端点分片

**【IPv6地址分类】**

**单播地址用于唯一标识支持 IPv6 的设备上的接口**。源地址必须是单播地址

* 单播地址（Unicast Address）
* 本地链路地址、环回地址、未指唯一本地地址、嵌入式Ipv4地址、全局单播地址、定地址
* 组播地址（Multicast Address）
* 任播地址（Anycast Address）

**【1.本地链路地址】**

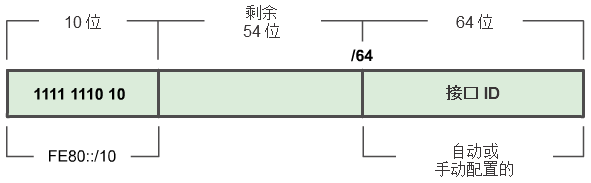
* 用在单一链路上
* 带有链路本地源或目的地址的数据包不转发到其它链路
* 支持IPv6的网络接口都需要有本地链路地址
* IPv4不会自动生成，IPv6主机会自动创建。

PS：**整个路由器挂下的全部（包括交换机）称为一个链路**

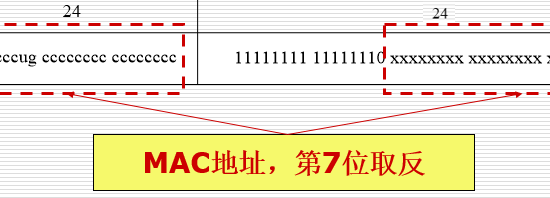
**【1.本地链路地址生成】**

只在同一本地链路中使用，范围：FE80::/64

后64位为EUI-64地址：



**EUI-64生成：**



**【1.本地链路地址用途】**

* 主机使用**路由器本地链路地址作为默认网关地址**。
* 路由器使用本地链路地址交换动态路由协议消息。
* **转发数据包时，路由器的路由表使用本地链路地址确定下一跳路由器**。

**【2.环回地址】**

\* 地址：::1/128

\* 主机用来将数据包发送给本身，**不能分配给物理接口**

\* 启用Ping命令，测试本地主机TCP/IP配置

**【3.未指定地址】**

地址：::/128

不能分配给接口，仅作为源地址

**【4.唯一本地地址】**

地址：从 FC00::/7 到 FDFF::/7

在一个站点内或有限站点数之间用作本地地址，在全局不具可路由性。

## 【6.可聚合全球单播地址】

地址：**前缀(FP)=001**，第一组16进制数范围：**2000~3FFF**

设计目标是聚合或汇总该地址以便产生有效路由基础结构

（相当于IPV4的公网地址）



* ISP商分配的前缀：/48
* Site拓扑：由组织机构划分子网
* 接口ID：64(全1可以分配给主机，全0分配给路由)
* 接口ID生成方法：EUI-64、随机生成、手工设置

## 【特殊单播地址】

**IPv6兼容地址**：0:0:0:0:0:0:w.x.y.z 或::w.x.y.z（双协议栈）

**IPv4映射地址**：0:0:0:0:0:FFFF:w.x.y.z 或::FFFF:w.x.y.z（IPv6网络中的IPv4节点表示，4->6）

# 【组播地址】



**Flags：**用来表示permanent（0000）或临时组播组（0001）

**Scope：**表示组播组的范围

**Group ID**：组播组ID

例子： FF02::1：链路范围所有节点组播地址

FF02::2：链路范围所有路由器组播地址

FF01::1、FF01::2：节点范围。。。

(**高八位不是全1(FF)的，即为单播地址**)

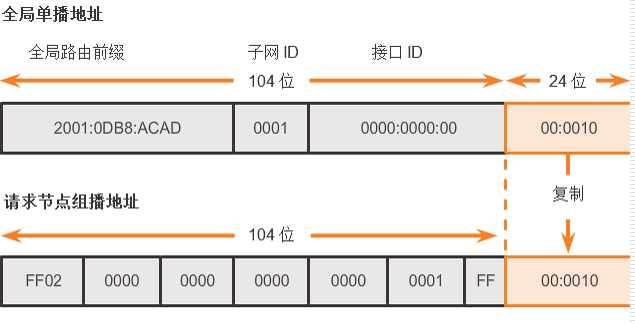
# 【请求节点组播地址】

**用于重复地址检测和获取邻居节点的链路层地址**

地址：前104位：FF02::1:FF/104

后24位：单播地址的后24位

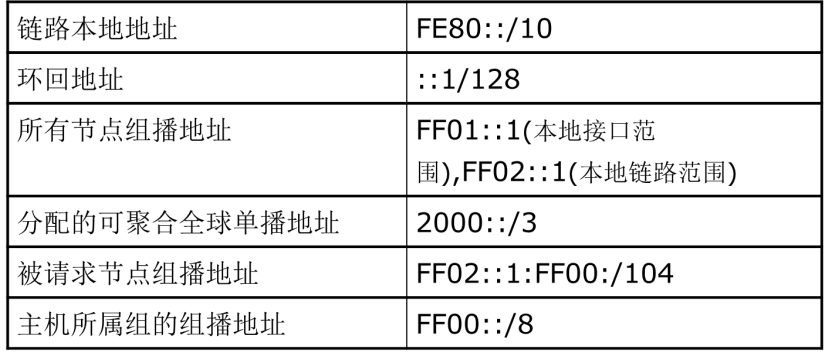
(凡是单播地址后24位相同的接口自动加入相应请求节点组播组)



# 【任播地址】

* 目标地址为任播地址的数据报将发送给**最近的一个接口**
* 适合于One-to-One of Many的通讯场合
* 从单播地址空间中分配，用于标识一组网络接口
* 仅能做目标地址，且仅分配给路由器

# 【一台主机的多IPv6地址】



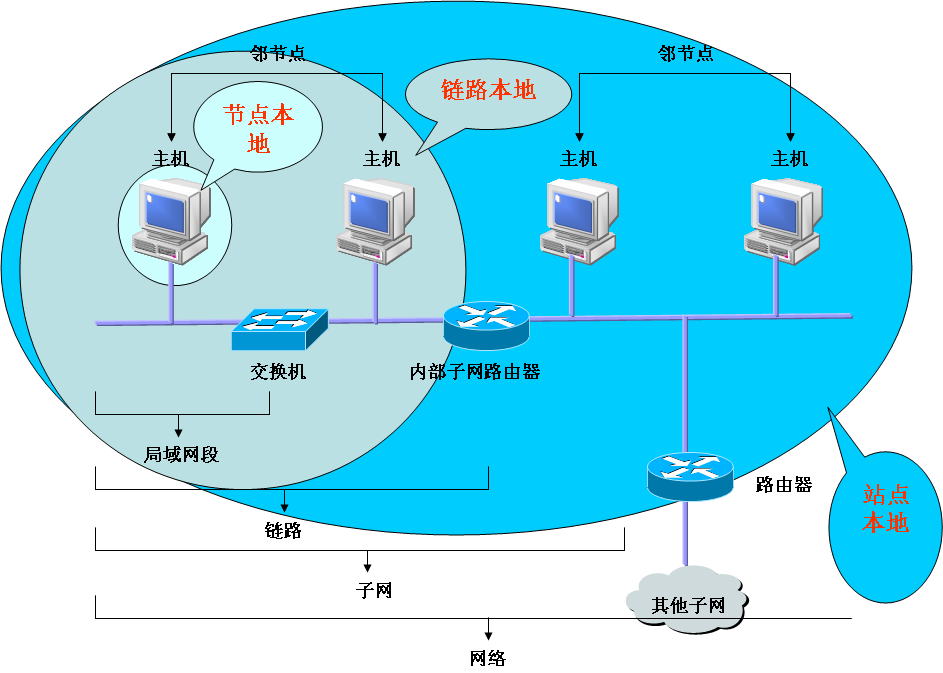
单播地址：

* 每个接口的链路本地地址
* 每个接口的单播地址（可以是一个站点本地地址和一个或多个可聚集全球地址）
* 回环（loopback）接口的回环地址（::1）

监听组播地址：

* 节点本地范围内所有节点组播地址（FF01::1）；
* 链路本地范围内所有节点组播地址（FF02::1）；
* 请求节点组播地址（如果主机的某个接口加入请求节点组）；
* 组播组组播地址（如果主机的某个接口加入任何组播组）

**【各类地址的应用范围】**



**【IPv6子网规划】**

* IPv4 子网划分是管理地址稀缺性
* IPv6 子网划分是根据路由器的数量及它们所支持的网络来构建寻址分层结构

（IPv6可以利用接口ID继续划分子网）

**[Part2].IPv6地址配置**

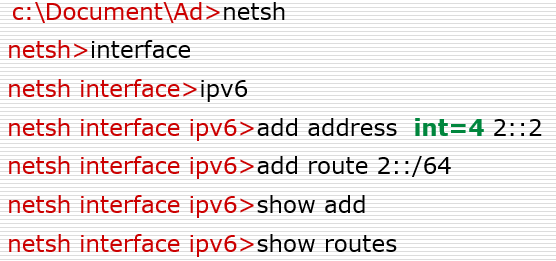
## 【配置方式】

手工配置

无状态地址自动配置（ND协议）

有状态地址自动配置（DHCPv6）

**【手工配置】**



**【无状态地址自动配置(SLAAC)】**

* 无须任何配置即可和外界通信，**真正的即插即用**（没有维护网络地址信息的服务器）
* 地址结构：64位前缀(和路由器前缀一致),64位EUI-64
* ND协议也即邻居发现协议，包括：Router Solicitation Router Advertisement 、Neighbor Solicitation 、Neighbor Advertisement 、Redirect (RS, RA, NS, NA, Redirect)

### 【SLAAC消息】

**两种重要的消息**

Router Solicitation（RS）：

主机向路由器发送请求报文，促使路由器发送Router advertisment消息（通过FF02::2）地址

Router Advertisement（RA）：

路由器向主机通告前缀等信息(通过FF02::1地址)

\* 两者都以ICMP报文的形式出现

## 【SLAAC三个机制】

**路由器发现（通过RS,RA发送消息）ppt68：**

\* 主机选择默认网关

\* 主机发现前缀，生成前缀列表

\* 参数发现：发现相关参数，如MTU，跳数限制、地址配置方式等

**重复地址检测（DAD）：**

\* **是节点确定即将使用的地址是否在链路上唯一的过程**

\* 所有单播地址，不管是自动配置还是手动配置，都必须要通过DAD

\* DAD机制通过ND中的NS/NA两种消息实现：

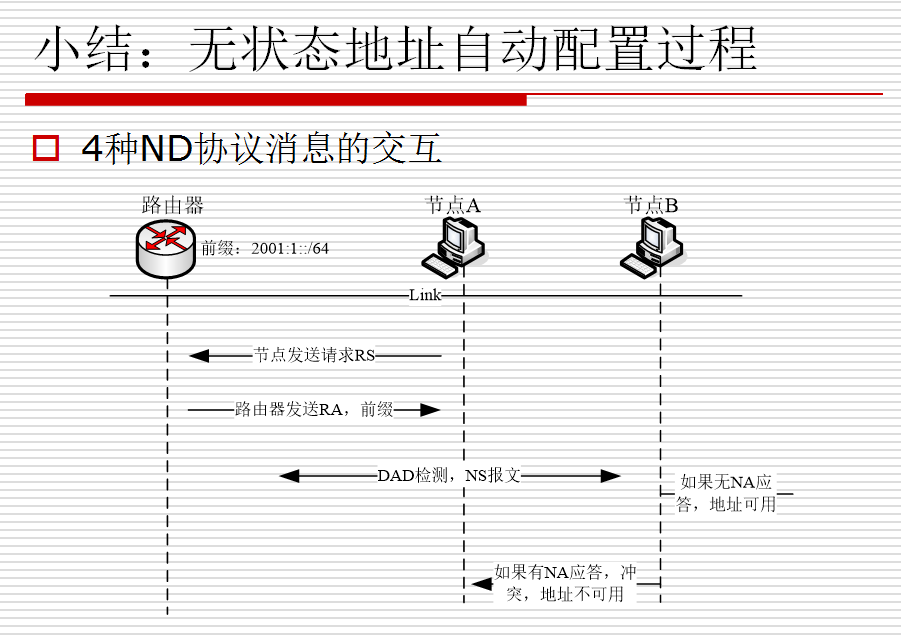
1.节点发送Neighbor Solicitation；

2.如果收到Neighbor Advertisment就证明地址重复

（a.接收者如果发现其中的Target域中的地址对它而言是tentative 的，则放弃使用这个地址；b.如果发现其中的Target域中的地址是一个它正在使用的地址，则发送Neighbor Advertisment消息）；

3**.尝试若干次**发送请求，都没有收到邻居通告，即可启用该地址

PS:一个地址在分配给一个接口之后且通过重复地址检测之前称为tentative地址，即试验地址。



**前缀重新编址（Redirect）：**

1.允许网络从以前的前缀平稳地过渡到新的前缀，提供对用户透明的网络重新编址能力

2.在前缀重新编址时，路由器会继续通告当前前缀， 只是优先时间和有效时间被减小到接近0，同时， 路由器开始通告新的前缀，这样，**链路中至少有两个前缀共存**

3.节点收到这样的RA，会发现当前前缀的生命周期较短，停止使用；同时开始用新的前缀配置接口， 并进行DAD，通过后获得新地址使用

## 【SLAAC注意事项】

\*为避免RS泛滥，节点启动时最多只能发送3 个RS

\*主机收到路由器的RA之后，自动设置默认路由器，建立

默认路由器列表、前缀列表及其它参数

\*路由器会主动周期性地发送RA（默认值200秒）

\*自动配置的IPv6地址在系统中有一个生存周期，跟优先

时间和有效时间有关：



1. 在Preferred Lifetime周期内的前缀生成的地址，任何上层应用都可不受限制地使用；
2. 在超过Preferred Lifetime 但未超过Valid Lifetime周期内的前缀生成的地址，正在使用该地址的上层应用可继续使用，但任何新的上层应 用不能使用这个地址；
3. 在超过Valid Lifetime周期内的前缀构造的地址，任何上层应用都不能使用该地址

其中，tentative地址为进行DAD之前的地址。一个链路本地地址的优先时间和有效时间是无限的，永不超时。

# 【RA消息中的M、O】

M=0，O=0：使用SLAAC(使用RA)

M=0，O=1：使用无状态DHCP：SLAAC和DHCPv6

（使用RA和DHCPv6服务器）

M=1，不含O：使用有状态DHCPv6(使用DHCPv6服务器)

**【前缀重新编址中的2个地址】**

在转换期间，节点有两个单播地址使用：

* 旧地址：基于旧的前缀，用以维持以前已经建立的连接；
* 新地址：基于新的前缀，用来建立新的连接

当旧的前缀的有效时间递减为0时，旧的前缀完全废止，此时，RA中只包含新的前缀

**【DHCP工作原理】**

DHCP客户发送请求广播

DHCP服务器单播应答

DHCP客户接收应答，获取IP等信息

**【DHCP运行方式】**

1. DHCP客户机在本地发送DHCPDISCOVER**广播**包；
2. DHCP服务器**单播**发送携带租约信息的DHCPOFFER包；
3. DHCP客户机确认租约信息并发送DHCPREQUEST**广播**包；
4. DHCP服务器**单播**送回DHCP ACK確认完成IP地址租用

**【DHCP地址分配】**

* 自动分配－由DHCP分配一个永久的IP。
* 手动分配－网络管理员预先安排分配，由DHCP转达。
* 动态分配－由DHCP分配具有租约期的IP。

**【有SLAAC为何需要DHCPv6】**

* 需要**动态指定DNS服务**时
* 当**不希望MAC地址成为IPv6地址的一部分时**
* 当需要良好的扩展性时

**【DHCPv6特点】**

1. 使用UDP来交换报文，546/547
2. 使用本地链路地址或其它机制获得的地址发送和接收DHCPv6报文
3. 没有广播，客户机只需发送给链路范围组播地址（FF02::1:2）
4. 取消了DHCPv4中的Discover和Offer消息

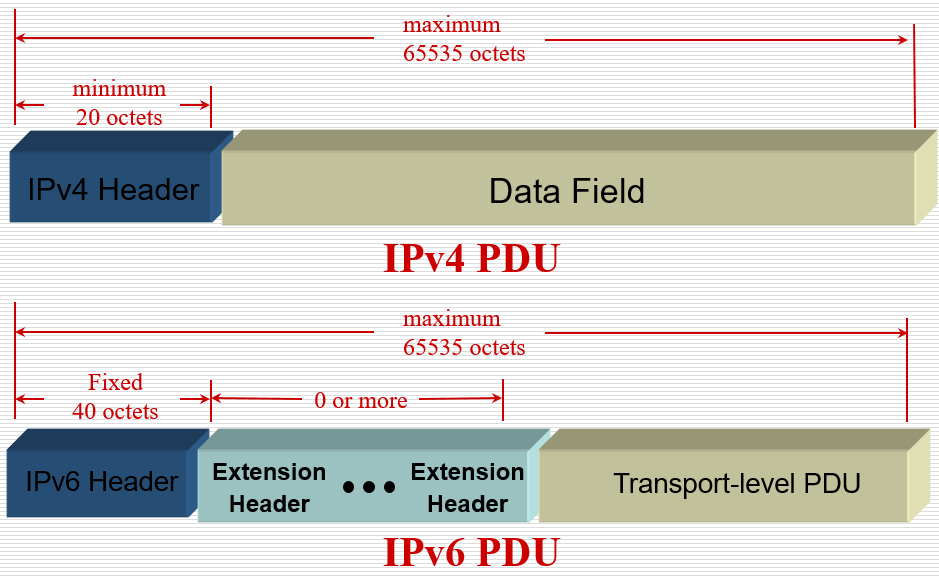
## 【DHCPv6获取地址和参数的典型过程】



DHCP和SLAAC是无状态全局单播地址配置的2种方式。

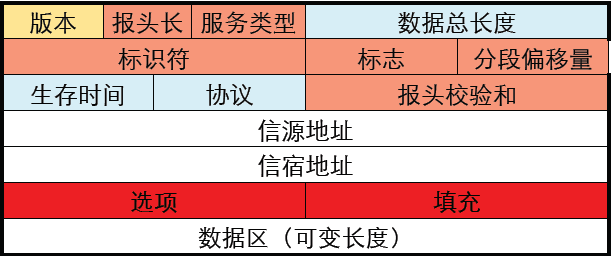
**[Part3].IPv6报文**

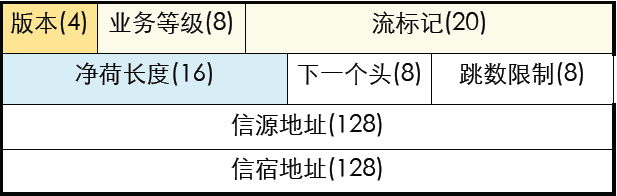
**【IPv4与IPv6报文】**



**IPv6扩展报头：**Pv6将一些IP层的可选功能实现在上层封装和基本IPv6基本头部之后的扩展头部中

主要的扩展报头：逐跳选项、路由报头、分段报头、认证报头、封装安全有效载荷报头、目标选项





**报文变化总结：**

修正项

地址：32位 -> 128位

Time to Live(生存时间) -> Hop Limit(跳数限制)

Protocol(协议) -> Next Header(下一个头)

Type of Service(服务类型) -> Traffic Class(业务等级)

删掉的项

Fragmentation fields(分段偏移量)

IP options(标志、标识符)

Header Checksum eliminated(报头校验和)

Header Length field (报头长)

Length field (数据总长度)

增加的项

Flow Label(流标识)

## 【扩展报头顺序】

逐跳选项报头 🡪目标选项报头1（当存在路由报头时，用于中间目标）🡪路由报头 🡪片段报头 🡪身份验证报头 🡪封装安全有效载荷报头 🡪目标选项报头2（用于最终目标）

**【ICMPv4与ICMPv6】**

* ICMPv6保留了ICMPv4的常用功能
* 如回声请求、抑制消息、重定向、参数错误等
* 还进行了很大的扩展（ARP、IGMP）
* 邻居发现
* 无状态地址配置
* 路径MTU发现（PMTU）

**【ICMPv6报文类型】**

**差错报文**，通告IPv6分组传输中出现的错误：

* 目标不可达、数据包超长、超时、参数问题

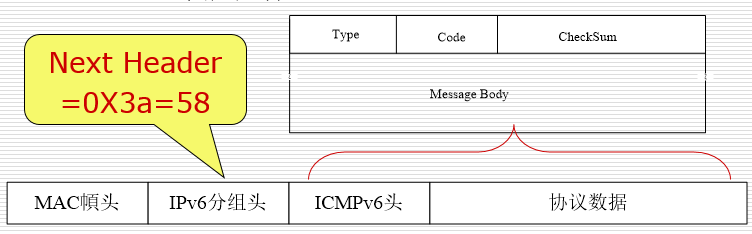
**信息报文**，提供诊断和附加的主机功能：

* MLD、ND (ARP,redirect…)、回声请求(128)和应答(129)

**【ICMPv6报文格式】**

差错报文：Type=0×××××××=0~127

信息报文：Type=1×××××××=128~255



**【ICMPv6三个应用】**

* Ping
* Tracert
* 第一个请求：HopLim=1
* 第一跳路由器收到，**发送超时消息**
* 得到第一跳路由器的信息
* 第二个请求：HopLim=2
* 第二跳路由器收到，**发送超时消息**
* 得到第二跳路由器的信息
* 。。。。。。
* 直到目的地，**目的机通常发送端口不可达报**
* PMTU发现

\* MTU：最大传输单元（载重能力）

\* IPv4网络可能由路由器承担分段；易遭受分片型攻击

\* IPv6规定分段不能由路由器承担，只能由发送方分段

\* 要求源节点，必须在发送数据前知道整个传送路径的最小MTU

PMTU发现原理（试探）



如，源机发送MTU=1500数据包，B返回超长消息，指定MTU=1400

## 【源和目的在同一链路的数据转发】

（路由：为一个数据分组找到一根能够到达目的地的路径）

\*类似于IPv4同一网段的数据路由

\*通过地址前缀判断是否同一链路：on-link

\*发起地址解析（不同于v4的ARP）

-独立于媒体层、利用三层安全机制、改广播为组播，降低了骚扰范围

\*由ND协议完成

### 【ND协议】



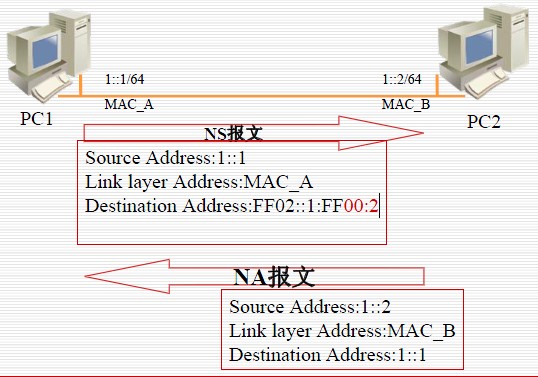
协议报文为ICMP报文，其中Type：RS(133)、RA(134)、NS(135)、NA(136)、Redirect(137)

# 【IPv6地址解析】

**地址解析的意义：**

**当两台主机需要直接通信时（链路层），需要根据目的主机的IP地址转为其对应的MAC地址，才能直接通信。**

1. 首先查找邻居缓存表，没有则进行地址解析（类似于ARP表）
2. 源主机发送组播NS报文，该报文的目的地址为目标IPv6地址所对应的请求节点组播地址（Solicited-node），在其中也包含了自己的链路层地址；
3. 目标主机收到NS报文后，就会了解到发送主机的IPv6地址和相应链路层地址；
4. 目标主机向源单播发送主机发送一个邻接点公告报文（NA），该报文中包含自己的链路层地址。（单播）



**【源和目的不在同一链路的数据转发】**

主机发给哪个路由器？（主机-路由器）：

1.具有一个合法的IPv6地址

2.一条目的包含PC2，下一跳RT1的路由

重定向：优化主机-路由器间的路由（ppt121）

路由器发给哪个路由器？（路由器-路由器）

核心：路由表(包括静态路由(IPv4/6：ip/ipv6 route)和动态路由)

**【DestinationCache（目的缓存表）】**

IPv6独有的一个数据结构，Show DestinationCache

* 初始时为空；
* 某个地址在表中查不到时，改查路由表，做on-link判断：
* 是on-link的，将目的地址本身加入DC表的nexthop域；
* 是off-link的，将路由表中的下一跳加入DC的nexthop域。

**【IPv4、IPv6共存技术】**

**双栈：**网络设备上运行IPv6/IPv4双协议栈

**隧道：**IPv6网络上承载IPv4分组，或相反

**翻译/转换：**地址、分组、端口的转换

## 【双栈技术】

* 适用于混合环境
* 基础设施设备，如路由器、交换机、公用服务器等，需要运行和支持双栈
* 非基础设施设备：单协议或双协议

（使用高优先级地址，IPv6失败再尝试IPv4）

## 【隧道技术】

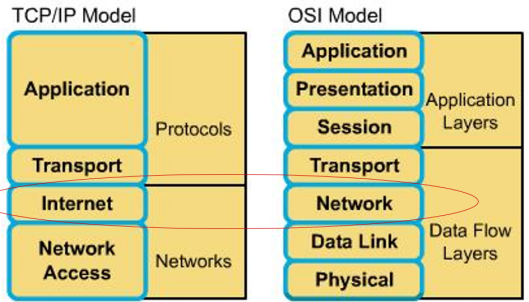
* **手动隧道**：事前配置
* **自动隧道**：创建和拆除都依赖当前网络条件
* IPv6分组作为 **载荷** 搭载到IPv4分组中，在这种情形下，IPv4分组头部的protocol=41。（v4转v6一样道理）

## 【翻译/转换技术】

* 从IPv4转换到IPv6，或反过来，不仅发生在网络层，还有传输层和应用层。
* **当双栈和隧道都无法使用的时候，才使用**；适用纯IPv4节点和纯IPv6节点间的通信

**[Part4].IPv4回顾**

## 【OSI、TCP/IP模型】



## 【地址分类】



## 【特殊IP地址】

32位全为0（0.0.0.0）：Cisco路由器指定的默认路由

32位全为1（255.255.255.255）：Flood(local) Broadcast

主机部分全为0（如172.16.0.0）：网络地址

主机部分全为1，（如172.16.255.255）：Direct Broadcast

127.0.0.0：Lookback Network

127.0.0.1：Lookback test

**[Part5].路由及DV**

## 【路由器的基本功能】

（网络层的主要功能：最主要的功能是将**分组**从源机经选定的路由送到目的机。这主要是路由器的功劳）

* 收到数据报后路由器的工作步骤

1. 打开数据报，读取头部信息
2. 确定目标网络地址
3. 查找路由表，重新打包后转发到相应的接口

上面的步骤完成了路由器的基本功能：

**路由选择:**转发表建立和维护的过程

* 1个或更多的路由选择协议
* 将路由信息注入到转发表中

**转发（交换）:** 将分组从入口移到出口的过程，依据：

* 转发表（路由表）
* 分组中携带的信息

# 【分组转发(交换)技术】

* Source routing（源路由）
* 分组携带路径
* Table of virtual circuits（虚电路）
* 穿越网络建立链接及状态
* 使用链接转发分组
* Table of global addresses (IP，数据报交换)
* 路由器维持到目的地的下一条
* 分组需要携带目的地地址

# 【源路由】

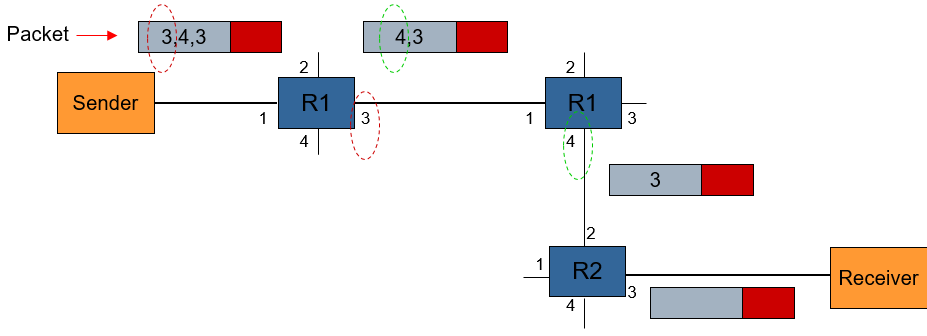
分组中列出整个路径，路由器处理的步骤：

1. 检查路径上的第一步
2. 从分组中剥离路径上的第一步
3. 将分组转发到剥离的那一步上

**优点：**可以很简单和快速地完成转发

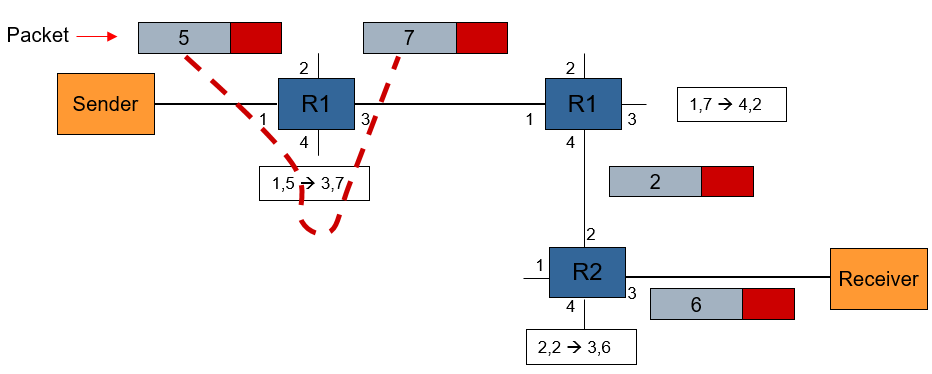
**缺点：**报头可变**、**源必须知道和发现拓扑

**应用：**Ad-hoc networks (DSR)



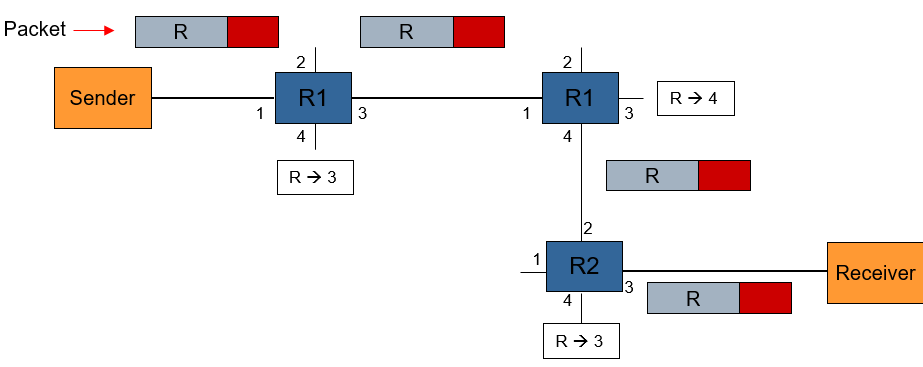
# 【虚电路】

* 连接建立过程
* 处理连接请求
* 每个路由器在本地链路上分配流ID
* 为每个流创建 入口-出口 映射
* 每个分组携带着链接ID
* 源发送分组到第一跳，并携带链接ID
* 路由器处理过程
* 查找流ID
* 用出口流ID置换分组中的流ID
* 转发到出口



* 优点
* 更加有效的查找 (simple table lookup)
* 链接建立的时候即可预留带宽
* 易于硬件实现
* 缺点
* 链接建立时仍需路由
* 错误恢复更复杂（重创链接）
* 典型应用
* ATM –固定的信元
* MPLS – 为IP网络设计的标签交换

## 【数据报交换】



路由器维持到目的地下一跳（维持路由表），分组需要携带目的地地址

* 优点
* 无状态：无差错恢复
* 缺点
* 每个路由器原则上要知道每个目的地，可导致超大路由表
* 到同一目的地的分组选取同样的路径

## 【路由表】

含网络地址、接口、计量值(跳数)、网络掩码、网关等信息

## 【路由】

根据数据包目的ip地址(目的网络地址)决定下一跳如何转发

**静态路由**：由网络管理员手工输入。

**动态路由**：通过路由选择协议自动适应网络拓扑或流量变化。

**【静态路由配置】**

1. 确定目的网络的地址和掩码，确定下一跳地址或转发数据包的本地接口。
2. 进入全局配置模式。
3. 使用ip route 命令添加路由。
4. 退出全局配置模式。
5. 使用copy running-config startup-config 将配置保存到nvram里面

第(3)步语法：ip route *prefix mask {address|interface} [distance]*

* 使用接口，路由器知道从哪里转发出去，更高效
* 使用地址需要第二次查找，以确定转发接口

# 【管理距离】

重要的管理距离：0、1、110、120

* 直连网络的管理距离为0
* 静态路由的管理距离为1
* 值越小，越可信、可靠，有多条路由选择时依据此选择
* 出于备份目的的静态路由，只有当其它动态路由失效时才起作用，可以配置它的管理距离，让他高过动态协议相应的管理距离

## 【静态路由使用场合】

* 拓扑长期不变，无须学习，可节约路由器学习维护开销
* 存根网络（stub），可配置默认路由
* 减小学习维护开销
* 缩减路由表的规模
* 避免错误地丢弃分组

## 【动态路由协议】

* 内部网关协议
* 路由信息协议（RIP）（典型DV协议）
* 开放最短路径优先（OSPF）（典型的LS协议）
* 内部网关路由协议（IGRP）
* 增强内部网关路由协议（E IGRP ）
* 外部网关协议
* 边界网关协议（BGP）

## 【DV工作原理】

* 每个路由器维护两个向量Di和Si来表示该点到网上所有节点的路径距离及其下一个节点
* 相邻路由器之间交换路径信息
* 各节点根据路径信息更新路由表

dij= Min{dix+ dxj, dij} ( x ∈邻居点 )

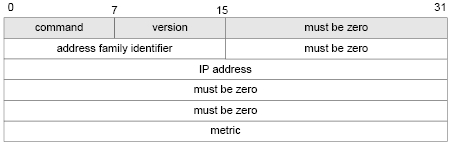
## 【DV优缺点】

* 优点：
* 算法简单
* 缺点：
* 交换的路径信息量大
* 路径信息传播慢，使得路径信息可能不一致。
* 收敛速度慢，存在无穷计算问题。
* 不适合大型网络

## 【RIP特点】

* RIP是一种距离矢量路由协议。
* RIP使用跳数作为路由选择的度量。
* 当到达目的网络的跳数超过15跳，数据包将被丢掉。
* 缺省地，RIP路由更新广播周期为30秒

## 【RIPv1报文格式】

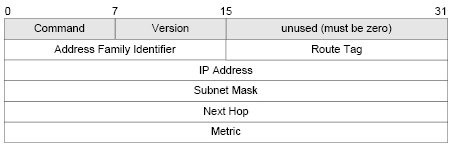


Command：报文类型，1为request，2为reponse

Version：RIP版本号

AFI：必须为2，表示为IP地址族

## 【RIPv2报文格式】



Router tag：外部路由标记

## 【RIP缺点】

* 超过15跳便无法到达
* 协议以跳数，即报文经过的路由器个数为衡量标准，并以此来选择路由，这一措施欠合理性
* 该路由协议应用到实际中时，很容易出现“计数到无穷大”的现象，这使得路由收敛很慢

## 【DV存在的问题】

* 表现
* 路由环路（routing loop）
* 计数到无穷问题
* 收敛慢的问题
* 原因：相信错误信息导致

## 【DV上述问题解决办法】

* 定义路径代价的最大值（Maximum）
* 超过该值就认为不可达，扔弃
* 提高收敛速度

**水平分割（Split Horizon）**

当路由器从某个网络接口大宋RIP路由更新报文时，其中不能包含从该接口获取的路径信息。

**毒性逆转（Poison Reverse）**

与水平分割类似，区别在于，包含从该接口获取的路径信息，但是距离设为无穷

**抑制定时器（Hold-Down Timers）**

在某邻居节点发生故障时，启动抑制计时器，在计时期内：

* 若节点down转为up，关闭计时器，保留原有信息。
* 若收到另一邻居关于这个节点的信息，且比原路径短，则关闭计时器，更新信息。
* 上述2种情况不发生，则更新路由为信宿不可达。

**触发更新（Triggered Updates）**

一旦发现某一些路由表项发生变化，就立即广播路由更新报文，而不必等待下一次刷新周期

**【RIP运行机制】**

* 当路由器启动RIP后，以组播向邻居发送信息请求
* 邻居收到请求信息，发送整个路由表作为响应；
* 此后，周期性交互消息（路由表）
* 路由器检测到路由变化时，组播更新消息给邻居

**【DV特点】**

* 采用最短路径准则，计算D信宿(距离，下站)；
* 每个站点只知道自己和邻居的局部信息，在自己的刷新周期到来时，根据邻居的路由变化重新启动算法；
* 算法的收敛速度慢（特别是对网络崩溃）造成全网信息的不一致，导致产生路径环，使计数至无穷大；
* 当路径环产生时，定义距离的最大值可防止算法进入死循环，解决计数至无穷大问题。

## 【RIPng】

* RIPng保留了RIP的主要特点：
* 距离矢量采用跳数，16跳为不可达
* 工作机制不变；
* 仍然采用水平分割、毒性逆转、触发更新等技术减少路 由环的发生
* 主要改变的地方：
* 组播代替广播：主机不再受骚扰
* 下一跳信息由单独的R Table Entry表示
* 安全考虑：不单独设置验证，由IPv6本身保证
* 只用于IP网络：不再支持其他网络协议

## 【RIPng的Request报文】

当路由器启动或更新时，发该类报文 （组播）

* 通用request报文：路由器启动时发送，只有一项RTE，且前缀和前缀长度都为0，量度为16，表示请求全部
* 指定request报文：用于网络诊断

# 【RIPng的Response报文】

对Request的回应：通常包含全部的路由信息

主动发送

* 周期性发送 （组播）
* 触发更新

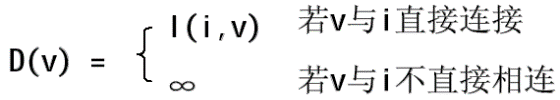
(通常，路由会对收到的Response报文进行有效性检查，如端口号，地址，通过了才会使用该信息来更新路由表)

**[Part6].OSPF**

# 【Dijistra算法】

1. 初始化

设节点i为源节点令N = {i}，对所有不在N中节点v，写出



1. 寻找一个不在N中的节点w，其D(w)值为最小，把w加入到N中，然后对所有不在N中的节点用[D(v)，D(w) + l(w,v)]中的较小的值去更新原有的D(v)值，即



1. 重复步骤2，直到所有的网络节点都在N中为止。

**【LS算法步骤】**

1. 各路由器主动测试所有与之相邻的路由器的状态，周期性地向相邻路由器发出简短的查询报文，询问相邻路由器当前是否能够访问，假如对方做出反应，则说明连接状态为UP，否则为DOWN
2. 各路由器周期性地向所有参与SPF的路由器广播其L-S信息，而不像DV算法那样只向相邻路由器发送信息
3. 路由器收到L-S报文后，利用它刷新网络拓扑图，将相应的连接状态改为UP或者DOWN。假如L-S发生了变化，路由器立即利用Dijikstra前向搜索算法，这个算法可以求出加权无向图中从某给定节点到目的节点最小耗费路由或最佳路由

**【LS优点】**

1. L-S信息向全网广播，各路由器都使用相同的原始数据进行路径计算，保证了各路由器L-S图的一致性。
2. 各路由器在本地计算路由计算，路由信息不会反过来对原路由器发生作用，使路由器的收敛性得到了保证
3. 广播的路由信息只与该路由器相连的链路数目和状态有关，比DV算法要少

**【LS与DV对比】**

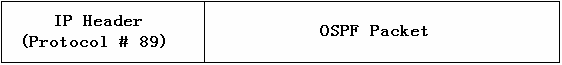
* DV算法原理清楚，实现简单，并且对路由器处理能力要求不高，在规模较小和拓扑结构变化不快的网络中获得了广泛的应用，而且由于成型较早，比较成熟。
* LS性能优越，适用于规模较大及拓扑变化较快的网络，但每一网络节点（包括主机和路由器）均需要实时形成全网的拓扑图以及自己为根节点的树，因此对CPU的处理能力提出较高的要求，同时由于路由信息是以广播的方式传播的，比较占网络带宽。

**【OSPF特点】**

* OSPF是一种基于开放标准的链路状态路由协议，是目前IGP中应用最广、性能最优的一个协议
* OSPF可以在大型网络中使用
* 无路由自环
* OSPF支持VLSM
* 使用带宽作为度量值（108/BW）
* 收敛速度快
* 通过分区实现高效的网络管理

## 【单区域OSPF】

直接封装在IP分组中，IP协议号是89



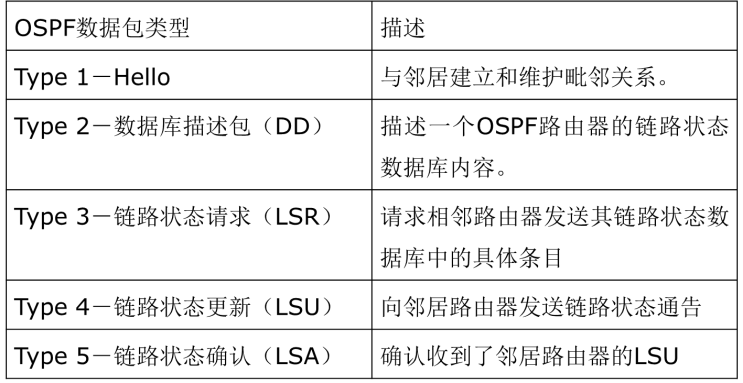
TTL=1：通常OSPF报文不转发，只被传递一条，即在IP报头的TTL值被设为1，但虚联接除外

### 【OSPF开销】



小于1则取为1

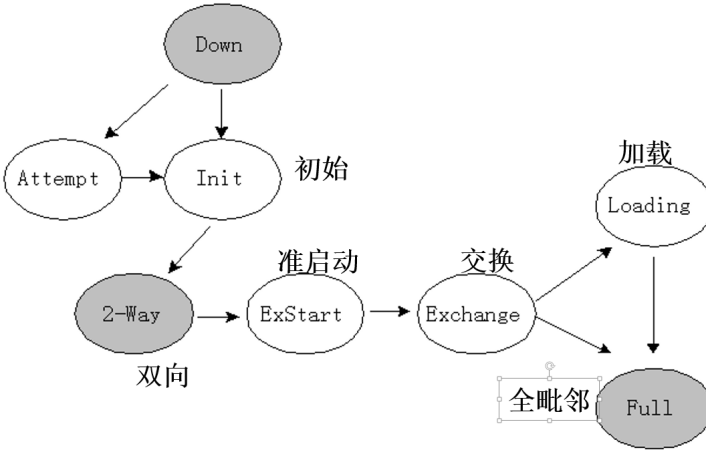
# 【OSPF分组类型】



**【OSPF的运行步骤】**

* 建立路由器毗邻关系
* 选举DR和BDR
* 发现路由
* 选择最佳路由
* 维护路由信息

# 【运行OSPF的路由器状态图】



**【选举DR和BDR的原因】**



同步的次数减少了（O（n）），减少了带宽的利用

* 路由器的角色：DR、BDR、DROther
* 路由器间的关系：Unknown、Neighbor、Adjacent

## 【DR选举】

选出优先级最高路由器，为防止频繁更换，选举结果在一定条件下保持不变。优先级为0不会当代表路由器，选举过程：

1. 如果一个或多个邻居竞选备份代表路由器，那么拥有最高优先级将被选为代表路由器。在出现多个最高优先级的路由器有的情况时，拥有最大ID的路由器将被选中。
2. 如果没有邻居竞选备份，…..最高优先级，…..ID最大。
3. 如果一个或者多个邻居竞选代表路由器，…..最高优先级，…..ID最大。
4. 如果没有邻居竞选代表路由器，备份将升级为代表。重新竞选备份路由器

## 【发现路由】

* 在exstart状态下交换DD报文
* 比较自己的数据库与收到的摘要是否相同
* 不相同，则请求完整LSA信息

（当链路是一个点到点联络员或一个虚拟链路时，在建立一个双向链接之后，就可以开始建立紧邻关系。而在网络端口，必须首先选举代表路由器及其备份。）

## 【选择最佳路由】

SPF算法、负载均衡

# 【维护路由信息】

* 触发更新，LSU
* Hello分组发送的时间间隔：缺省10秒
* Hello分组的失效间隔：缺省40秒
* 即使没有拓扑变化，LSA在条目过期（缺省30分钟）后，发送LSU，通告链路存活

# 【OSPF克服路由自环原因】

* 每一条LSA都标记了生成者（用生成该LSA的路由器的RouterID标记），其他路由器只负责传输，这样不会在传输的过程中发生对该信息的改变和错误理解。
* 路由计算算法是SPF，计算结果是一棵树，路由是树上叶子节点，从根节点到叶子节点是单向不可回复的路径。
* 区域之间通过规定骨干区域避免

# 【OSPF在大型网络中的问题】

* LSDB非常庞大，占用大量存储空间
* 计算最小生成树耗时增加，CPU负担很重
* 一点变化都会引发从头重新计算
* 网络拓扑结构经常发生变化，网络经常处于“动荡”之中
* 接口up down
* 路由器的增加删除

# 【路由器类型】

* 内部路由器 --- 路由器所有接口都在一个区
* 主干路由器 --- 所有接口都在主干区域的路由器
* 区域边界路由器(ABR) ---路由器接口分属不同区域
* 自治域边界路由器 (ASBR) --- 路由器至少有一个接口不属于本自治域/OSPF.

（Stub区域：边界区域，不接收外部路由信息）

# 【距离矢量与链路状态对比】

|  |  |
| --- | --- |
| **距离矢量** | **链路状态** |
| 通过邻居获得网络拓扑 | 直接获得整个网络拓扑 |
| 通过距离矢量计算 | 最短路径优先计算 |
| 周期性更新，收敛满 | 事件触发更新，收敛快 |
| 传递路由表格 | 传递链路状态信息 |

# 【OSPFv3与OSPFv2】

* OSPFv3保留了OSPFv2的主要工作机理
* 网络类型和接口类型
* 邻居发现和邻接（毗邻）建立机制
* 接口状态机和邻居状态机
* 基于LSDB计算路由
* LSA老化更新机制
* 泛洪机制(Flooding mechanism)
* 共五种协议报文: Hello, DD, LSR, LSU, LSAck
* 主要改变的地方

基于链路运行

* 在OSPFv2中，协议的运行是基于子网的，邻居之间形成邻接关系的条件之一就是两端的IP地址属于同一网段而且掩码相同。
* 而在OSPFv3中，协议基于链路运行，与具体的IPv6地址、前缀分离开来，即使同一链路上的不同节点具有不同的IPv6地址时，协议也可以正常运行。

取消了编址语义

* 在OSPFv2中，协议分组和LSA中的许多字段都是来自于网络上的某个IP地址，或掩码，或某个IP子网号。严重依赖IPv4。
* 在OSPFv3中，取消了上述编址性语义，而只保留协议运行必须的核心内容。ID依然保留32位，但只是一个编号，不再包含地址信息。

链路本地地址的使用

* 在OSPFv2中，每一个运行OSPF的接口都必须有一个全局的IPv4地址，协议的运行和路由的计算都依赖于它。
* 在IPv6中，每个接口都会分配本地链路地址（link-local address），OSPFv3使用了这个本地链路地址作为协议分组发送的源地址（虚连接除外），而且使用它作为路由的下一跳。
* 可以节省大量的全局地址，同时可以说协议的运行独立于IPv6，可以方便的扩展用于多种协议的路由。

使用专门的LSA来发布路由前缀信息

* 新增加了Intra-Area-Prefix-LSA，用于传递区域内路由前缀；新增加了Link-LSA，用于传递链路范围内的IPv6前缀。

## 【OSPFv3特点】

* 结构更加清晰

取消了协议报文的验证字段

明确的LSA泛滥范围

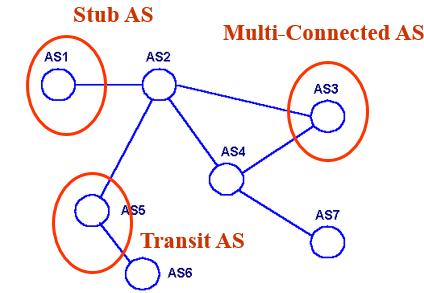
* 本地链路范围（Link-local scope）
* 区域范围（Area scope）
* AS范围（AS scope）
* 在LS\_Type中增加了专门的字段，方便传播控制
* 更好的可扩展性和适应性
* 在OSPFv2中，不同的实例必须运行在不同的链路上。
* 在OSPFv3中，明确的提供了对多实例的支持，在协议报文中增加了“instance ID”字段，同时在报文接收时对该字段进行判断，只有实例号匹配的报文才会处理，否则丢弃。这样，即使是同一链路也可以运行多个OSPF实例了，而且互相独立运行不会影响。

**[Part7].BGP**

## 【AS】

处于**相同技术管理下的路由器的集合**，**内部使用相同的IGP**（ *interior gateway protocol* ）和量度去引导分组，外部使用EGP（ *exterior gateway protocol* ）去引导分组到其他AS。

## 【BGP分类】



## 【BGP】

运行于AS之间，允许基于策略（policy-based）的路由选择，是一种距离矢量路由协议，但避免了环路（as之间采用aspath & as内部不转发路由）。通过TCP传输，端口号179，支持CIDR路由聚合。

## 【BGP如何避免路由环】

BGP 路由器记录下**全路径信息**，而不仅仅是路径代价（ keeps track of the exact path），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。

## 【BGP使用场合】

什么时候使用：

* 在多个AS间传输数据
* 两AS间存在多条路径
* 需要做路由策略和选择

什么时候不用：

* AS只有一个出口
* 所有出口指向一个ISP
* 路由性能不高，内存不高，CPU太慢，带宽不大

## 【BGP消息类型】

Open报文：打招呼，“你好，交个朋友吧”（协商参数）

Keepalive报文：我还活着，别不理我（30秒钟交换一次）

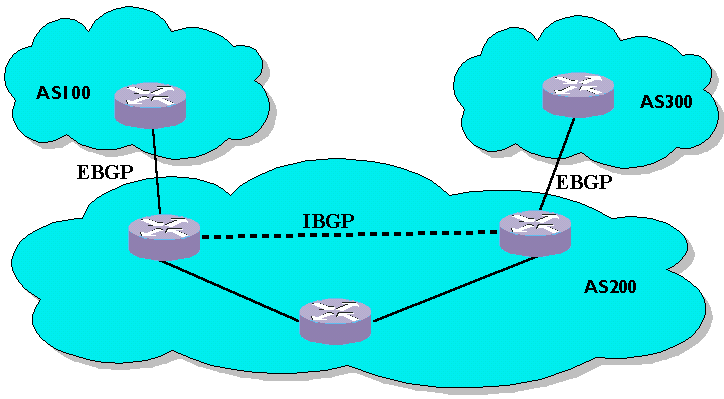
Update报文：有新闻（链路的变化）

Notification报文：我不跟你玩了(异常情况的通报，终止连接)

**【BGP工作机制】**

* 通过TCP建立BGP连接时，发送OPEN报文
* 连接建立后，如果有路由器**需要发送路由或路由发生变化**时，发送UPDATE报文
* 稳定后，周期发送KEEPALIVE报文，维持连接有效性
* 当本地BGP运行中发生错误时，发送NOTIFICATION报文通告BGP对端

**【BGP两种邻居】**



**【路由注入方式】**

动态路由注入：自动，配置简单，操控性差，可能不稳定

半动态路由注入：比动态路由具有更高的操控性

静态路由注入：操控性最强

**【路径属性】**

* Well-known mandatory 公认必遵
* Well-known discretionary 公认可遵（可选）
* Optional transitive 过渡（可转发）
* Optional nontransitive 非过渡的（不识别就不转发）

**【BGP路由的使用原则】**

前言：EBGP和IBGP指的是BGP间的关系

* 多条路径时，BGP 路由器只选最优的给自己使用
* BGP 路由器只把自己使用的路由通告给相邻体
* BGP 路由器从EBGP获得的路由会向它的所有BGP相邻体通告（包括EBGP和IBGP）
* BGP路由器从IBGP获得的路由不会向它的IBGP相邻体通告（避免内部产生环路）
* BGP 路由器从IBGP获得的路由是否通告给它的EBGP相邻体要依IGP和BGP同步的情况而定
* 连接一建立， BGP路由器将把自己所有的BGP路由通告给新相邻体（极大的考验和挑战！）

**【BGP重要属性】**

**路径属性**：施加在每条路由上，表征了路由的特征

**Origin属性（必遵）**：IGP优于Incomplete

**AS路径属性（必遵）**：描述了完整的路径，途经的AS

**Next Hop属性**：下一个AS、在同一个AS，Next hop值不会发生变化

**本地优先**属性（可选）：施加影响，引导流量

**MED属性（非过渡）**：除非配置了“always-compare-med”，其他时候永远只比较来自同一AS的路径、不跨AS传播，只影响它的上游AS、MED值越小，该条路由越优

**团体属性（过渡）**：便于制定路由策略

**【BGP过滤功能】**

可按照路由的IP地址过滤

可依照路由经过的AS-Path过滤

可以依照路由的属性过滤

可以依照路由到来的接口过滤

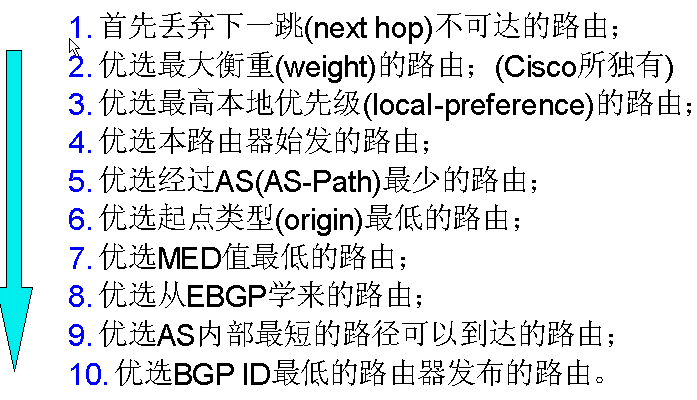
可以依照过滤的结果设置路由的属性

**【BGP选路规则】**

* 选AS\_PATH最短的那条
* 如果AS\_PATH距离相等，则优选本AS内到出口路由器最短的那根
* 如果还相等，选择Router\_ID（发送路由的路由器）最小的那根

**不是很明智，如下：根据属性来选路！！**

**【BGP根据属性来选路决策】**



**【BGP大型网络可能遇到问题】**

* 从IBGP收到的路由不会通告给其它IBGP 相邻体，所以，**AS内部的IBGP必须全连接**
* 路由表庞大：超过10万条（解决：BGP聚合）
* INGP相邻体过多，逻辑全链接不现实（解决：BGP联盟、BGP反射）
* 复杂网络中的路由变化频繁（解决：BGP衰减）

**[Part8].组播**

**【IP组播】**

IP组播是指在IP网络中将数据包以**尽力传送**（best-effort）的形式发送到网络中的**某个确定节点子集**，这个子集称为一个**组播组**（multicast group）

# 【组播基本思想】

* 源主机只发送一份数据，这份数据的目的地址为组播组地址；
* 组播组中的所有成员都可接收到同样的数据拷贝，并且只有组播组内主机（成员，目标主机）可以接收该数据

**【组播的优势】**

* 一对多通讯下，降低网络流量，提高网络通讯效率。由于每个分支只发送一份报文，所以网络规模（如用户数量）的增大不会额外增加网络的负担。
* 降低了骨干上的网络流量
* 降低了应用服务器的负担

**【组播缺点】**

* 传送不可靠！（尽力投递/best effort）
* 组播报文的复制开销！
* 路由器的资源消耗增加！
* 其它缺陷（可控可管性差）
* 用户管理
* 安全

## 【组播地址】

D类地址，1110开头，范围224.0.0.0~239.255.255.255

私人组播地址：239.0.0.0~239.255.255.255

# 【MAC组播地址】

IPv4：MAC前25位 +IPv4组播地址低23位

IPv6： 33:33：+ IPv6组播地址后32位

# 【组播基础】

* 组播转发树：有源树和共享树
* 逆向路径转发
* TTL阈值

# 【有源树和共享树】

相同点：

* 无回路，分支处复制

不同点：

* 有源树（最短路径树，为每个节点建立SPT）
* 占用内存较多O(S x G)，但路径最优，延迟最小
* 共享树（汇聚到RP，RP建立SPT）
* 占用内存较少O(G)，路径可能不是最优的，引入额外的延迟；有一个瓶颈

# 【逆向路径转发】

**作用**：**能防止环路（多播）**

**RBF**：路由器收到组播数据报文后，只有确认这个数据报文是从自己到源的出接口（单播）上到来的，才进行转发，否则丢弃报文

RBF检查：

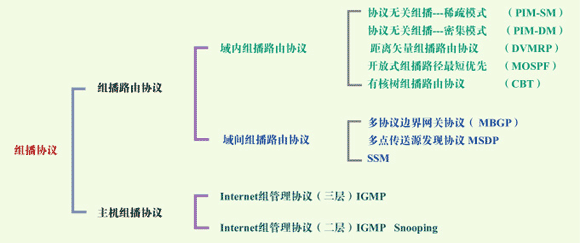
* 在单播路由表中查找 去往 组播报文源地址的路由
* 如果该路由的出接口就是组播报文的入接口，RPF检查成功；否则RPF检查失败

**简单来说就是根据去的数据路由表项来检查回来的包，确定去回在一线上。**

**【TTL阈值】**

* IP组播包被路由器转发的时候，IP头中的TTL值要减1
* 规则：只有**数据包的TTL值大于或等于接口的TTL**阈值时，路由器才能在出接口转发该数据包。
* **限制组播范围**

## 【组播路由协议分类】

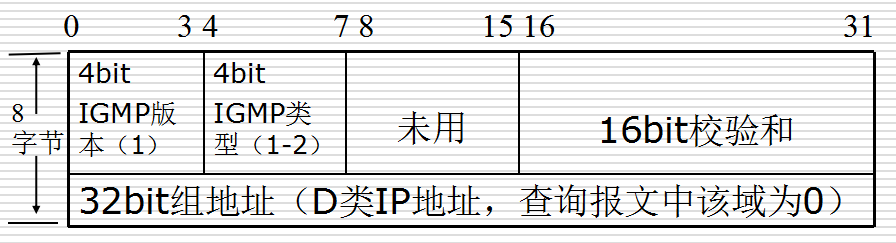


## 【组播协议：主机与路由器之间（IGMP）】

* **主机和直连路由器**之间建立、维护组播组成员关系。
* 不包括组播路由器之间的组成员关系信息的传播与维护。
* 所有参与组播的主机必须支持IGMP协议。

## 【IGMPv1】

* 报文格式



* 类型

1：成员关系查询（路由器发出）

2：成员关系报告（成员主机发出）

组地址：成员关系报告时填入**组播组地址**

工作方式：查询—响应

* + 查询是周期性的，默认值为60s一次

## 【报告抑制】

* 对于某一查询，当前子网中对于**每个组只需有一个**成员响应成员报告；这一过程叫响应抑制。
* 报告响应抑制步骤
* 主机收到IGMP成员关系查询后，主机对已经加入的每个组播组启动一个倒数计时器，计时器初始化为一个给定时间（在IGMPv1中为固定值10S）范围内的随机值。
* 当计时器计时值为0时，主机发送成员关系报告至与该计时器相关的组播组，以通知路由器本地网中有处于活动状态组播组接收者。
* 然而，当主机在它的倒数计时器达到0之前收到其他主机发送的某一成员关系报告，那么它就停止与成员关系报告组地址相同的倒数计时器的计时，这样就抑制了主机到这一组播组的成员关系报告。

# 【IGMPv1成员加入】

主机并不等待来自路由器的下一次成员关系查询，可主动向要加入的组播组发送成员关系报告表示加入

# 【IGMPv1成员离开】

没有定义离开机制，可以默默地离开。（最后一个成员离开后）**路由器设置了组播关联定时器，超时无组成员报告，则停止转发**。

### 【IGMPv2查询消息】

* 普通查询（组播地址段为零）
* 特定组查询：直接对单个组查询（组播地址段为正在查询的组播组地址）

## 【IGMPv2成员离开】

主动发送离开报告，离开延迟可以缩短到几秒钟

**【查询路由器选举】**

**IGMPv1**：依赖于路由协议

**IGMPv2**：

* 多访问网络上每个路由器假设自己为查询器并发出查询
* IP地址低（接口）的路由器被选为查询器
* **非查询路由器设置定时器，当超时没有收到查询器的周期查询，认为查询器出事了，重新选举**

**【IGMPv1与v2交互】**

* v2成员与v1路由器的交互
  + v2成员报告会被v1看作无效并且忽略
  + v1路由器不能识别v2报告，当v1路由器查询时，v2主机必须发送v1报告
* v2路由器与v1主机交互
  + V2路由器的查询可被V1的主机所处理，只是忽略第二个八位组的信息
  + v2路由器的特定查询不被v1主机理会
  + v2路由器必须忽略离开报告
* 路由器与路由器之间
  + 同一网段运行同一IGMP协议，假如存在IGMPv1路由器，则所有路由器手工配置为v1

**【IGMPv3】**

允许主机只收到**组播组内**某个**特定**信源的传输

**【IGMP三个版本】**

* IGMP v2：增加了**查询器路由器选择**/**特定组查询/离开组消息**及最大响应时间字段等扩展功能。
* IGMP v3：增加了对**特定(源,组)**的加入离开的支持，以提供对多播应用更好的支持。



### 【组播协议：路由器之间协议（PIM DM;PIM SM）】

* 密集模式（Dense-mode）
* 使用“推”（Push）模型
* 组播数据在整个网络的泛滥（Flood）
* 下游不想接收的话则剪枝（Prune）
* **泛滥、剪枝、泛滥、剪枝…周而复始** (通常3分钟折腾一次)
* 稀疏模式（Sparse-mode）
* 使用 “拉”（Pull）模型
* 组播数据只发送到有需要的地方
* 有显式的加入（Join）过程

### 【PIM】

* **协议无关组播**，支持所有的单播路由协议: 静态路由、RIP、IGRP、IS-IS、BGP、OSPF，总之，单播路由是什么都没关系。
* 特点
* 独立于单播协议
* 扩散和剪枝机制
* 无类

### 【PIM-DM运行机制】

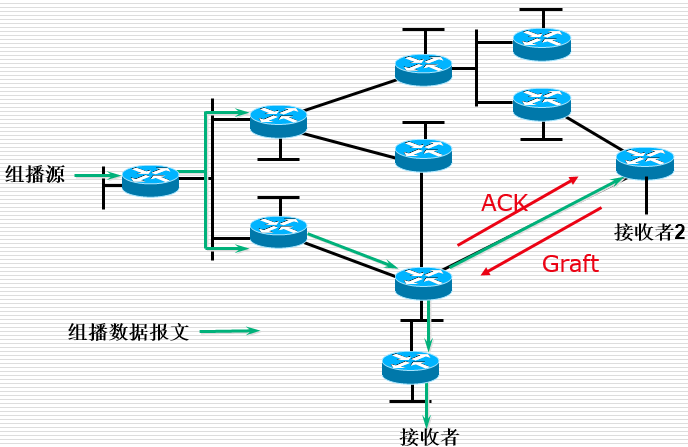
* 采用PUSH方式，将组播流量周期性扩散到网络中所有设备，建立和维护SPT(short path tree) （假设所有主机都需要接收组播数据）
* 周期性扩散、剪枝、嫁接过程，也成为：洪泛/剪枝/嫁接机制
* 使用逆向路径转发（RPF）机制：
* 先向网络泛滥(Flood)，然后根据组播组成员关系进行剪枝 (Prune)
* 使用断言机制来剪枝冗余数据流
* 适合于：
* **小规模的网络**
* 组播源和接收者比较靠近
* 源少，接收者多
* **数据流大且稳定**

# 【触发剪枝的条件】

* 信息到达PIM-DM路由器的非RPF点对点接口；
* PIM-DM路由器没有下游邻居，且所有叶网络上没有组成员；
* PIM-DM路由器所有接口上的下游邻居已经通过了剪枝表决

## 【PIM-DM嫁接】

**向RPF上游借口单播嫁接信息**，**迅速得到数据，不必等到下一个周期。**



# 【为什么需要断言机制】

* 网络拓扑多种多样
* **RPF检查能够防止组播流量出现环路，但无法避免出现组播流量重复和多份**

## 【断言机制实现】

* 路由器从其“出接口列表”(oiflist)中某个接口收到数据
* 只有其中一个路由器应该继续发送数据，以避免重复
* 路由器发送 “PIM Assert”消息
* 计算distance和 metric值
* 谁到源的路由最优谁获胜
* 如果distance和 metric相等，IP地址大的获胜
* 输的就停止转发 (剪枝接口)

## 【PIM-DM总结】

* 对于小型网络来说非常有效
* **优势:**
* 易于配置
* 实现机制简单（泛滥剪枝）
* **潜在问题：**
* **泛滥剪枝过程不够高效**
* 复杂的Assert机制
* 控制和数据平面混合

\*导致网络内部的所有路由器上都有(S, G)

\*可能会导致非确定性的拓扑行为

* **不支持共享树**

## 【PIM-SM】

* **支持共享树和源树**
* 假设没有主机需要接收组播数据，除非它们明确地发出了请求
* 稀疏组播的特点：
* 组成员所在的网络数相对少
* 组成员跨越的区域太大
* 带宽资源还没有富裕到可以忽略DM模式带来的消耗
* 适合于
* **大规模的企业网络**
* 接收者稀少
* 几乎是任何网络的优选方案（目前PIM-SM占主流）

# 【PIM-SM工作机制】

* 源只是简单地把数据包放在第一跳路由器（RP）上，接收者必须主动通知路由器
* 使用“汇聚点”(RP)，发送者和接受者在RP处汇聚
* 发送者的第一跳路由器把发送者注册到RP上（报个到，挂个号）
* 接收者的DR（直连网络上的负责人）为接收者加入到共享树 (树根在RP) —— **说到底是接收者加入了这个组，才需要逐跳记录路径信息**
  + 接收者发送加入消息，逐跳上行到RP，沿途路由器记录组播转发状态；
  + 接收者不想要组播数据时，发送剪枝消息，逐跳上行到RP，沿途路由器更新它的转发状态
* 跟DM本质的差别：路由器转发状态通过组播消息的抵达而建立或更新。

**PS：通常是每个RP负责每个组播组**

# 【自举路由器机制】

* 由管理员配置
* C-BSR （Candidate-Bootstrps Router）
* C-RP （Candidate-Rendezous Point）
* 选举后最终确定
* C-RP向BSR报告
* C-BSR发送自举报文

## 【源注册】

* **源的DR**将**组播数据**封装进一个注册消息，单播到RP；
* RP打开注册消息
* 发送（S，G）加入消息，沿途建立（S，G）状态
* 将组播数据在RPT上转发
* 停止注册时机：
* RP察觉到从源到RP的SPT树已经建立
* RP发送“注册停止”消息给源
* 若SPT树已建成，但没收到停止注册消息：
* 等待RP的注册停止消息
* 空注册消息和沿SPT的组播数据流并存
* 直到收到RP的注册停止消息
* 只沿SPT传播数据，停止空注册消息

# 【为什么进行PIN-SM SPT切换】

* 通过源树—RP—共享树的路径传输组播信息，走的路径可能不是最短的，浪费带宽。
* RP可能成为瓶颈。

## 【SPT切换条件】

最后一跳路由器（和接收者直连的路由器）一旦发现某个特定的**组播源的数据量超出了某个界**限(阈值)，马上向组播源发送（S，G）Join消息。

## 【共享树剪枝条件】

最后一跳路由器根据自己的状态表中的（\*，G）和（S，G）的入接口情况来判断是否发送剪枝消息（剪共享树），触发条件是：

* 在（S，G）的入接口上收到了相符合的组播数据 （源树已经建立） —— 即SPT已建立
* (S，G)上的初始化标志位被设置（创建时设置）
* 同时存在的（\*，G）和（S，G）的**入接口不同**

## 【PIM-SM总结】

* 对于稀疏和密集应用都很高效
* 优势:
* 数据流仅沿“加入”的分支向下发送
* 可以根据流量等条件动态地切换到源树
* 与具体的单播路由协议无关
* 域间组播路由的基础
* **能够以比PIM-DM更经济的方式建立SPT**

\*和MBGP、MSDP共同结合使用可以完成跨域的组播

**[Part9].QoS**

## 【Qos度量参数】

有效性(可靠性)、带宽(bandwidth,更确切的说:吞吐量)、延时(delay)、抖动(jitter)、丢包率(loss rate)

# 【导致服务质量问题的因素】

用户需要的服务（负载）几乎是无限的，承载服务的设施（资源）却是有限的，当负载>资源的时候，就会产生QoS问题：拥塞、路由器过载、延迟加大、抖动、丢包。

# 【综合服务】

* 以RSVP信令向网络提出业务流传输规格(FlowSpec)，并建立和拆除传输路径上的业务流状态。
* 主机和路由器节点建立和保持业务流状态信息。
* 以资源预留为核心（RSVP）

# 【资源预留协议（RSVP）】

* 发送者每30秒发送 PATH message （消息）
* 沿途路由器记录路径信息 （设置路径状态）
* 接收者每30秒回发 RESV message（消息）

# 【综合服务缺点】

* 所有的路由器都必须为每个流维护状态信息
* 完全的分布性导致极大的复杂性！
* 在低速连接上造成高负载的低效工作模式
* 扩展性和鲁棒性

## 【综合服务失败原因】

* 经济因素---部署代价过于昂贵
* 可扩展性差----每流状态难以维护
* 面向组播的预留策略难以实现
* 访问控制、认证、计费均实现困难

## 【差/区分服务的服务类】

* Expedited Forwarding (EF) (加速转发)
* Assured Forwarding (AF) (确保转发)
  + n queues × m drop preferences
* Default (Best Effort，BE)

## 【差分服务功能】

* 为不同的组（例如客户或企业中的部门）提供不同的服务级别
* 为提供给特定组或应用程序的网络服务设置优先级
* 发现和消除网络瓶颈区域以及其它形式的拥塞
* 监视网络性能并提供性能统计信息
* 控制进出网络资源的带宽

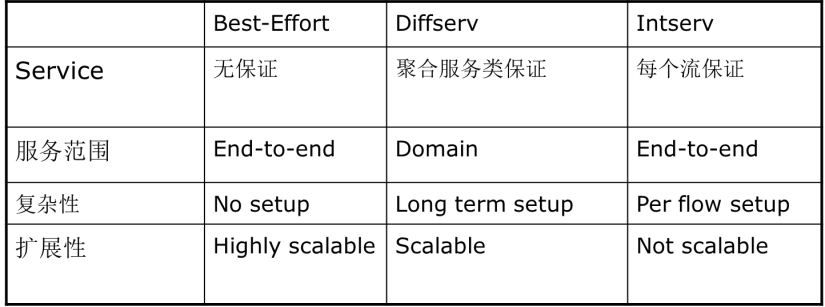
# 【Traffic class类别】

* Class Selector(CS)：aaa 000
* Expedited Forwarding(EF)：101 110
* Assured Forwarding(AF)：aaa bb0
* Default (BE)：000 000

# 【差分服务有点】

* 无需逐条信令（signalling）
* 无需单个流或单个用户状态
* 只有聚合的类别状态（ classification state）
* 转发行为较单纯（PHB）
* 只有在边界路由器上分类/调节较复杂
* 简单的分组处理（ 核心路由器）

# 【三种服务比较】



# 【标签交换】

在每个分组的前端加一个标签（Lable，tag），然后根据这个标签而不是根据目的地址进行路由，标签放在2.5层

## 【MPLS(多协议标签坏换)工作机制】

* 在MPLS网络入口的边沿路由器上打上标签
* 沿途路由器检查标签，根据标签值查找转发表，转发（替换新的标签）
* 在MPLS网络出口的边沿路由器上拆除标签，以后按照通常意义的三层路由转发
* 数据驱动或控制驱动建立转发表

## 【MPLS与普通标签交换区别】

* 聚合水平：FEC（转发等价类）
* 多标签
* 标签转发表由MPLS路由器建立

# 【队列技术】



## 【FIFO】

先进先出，队列太长太短都会产生问题。

# 【Priority Queuing】

**算法**：只要高优先级队列有分组，就一直从这里取。

**优点**：保证高优先级报文的带宽、延时和抖动

**缺点**：低优先级的报文可能得不到调度而“饿死”

# 【Custom Queuing】

* 0～16号队列，0号队列（系统）优先处理
* 可定义带宽（以字节数为单位），按照带宽从1～16号取分组转发（Round Robin调度），避免被“饿死”

# 【Weighted Fair Queuing】

* 分组流分类：相同的源IP、目的IP、源端口、目的端口、协议号、ToS值归为一个流 （五元组）
* 对于相同的标签和EXP域值的归为一个流
* 每个流分配到一个队列，采用hash散列，最多4096
* 队列数目可配，按优先级来分配带宽(体现权值W)

# 【WFQ公平性】

* WFQ给每个流分配的带宽可以是相同的：
* “W”（加权）跟IP优先级连在一起
* 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

# 【Class-Based WFQ】

* 类似CQ，可为每个队列保留最小带宽，与CQ不同，用户可配置实际占用的流量，而不是字节数
* 0～64号队列，所有不能归入流的都进入0号队列

# 【Low Latency Queuing】

* 在CBWFQ基础上增加1个或多个优先级队列来完成
* LLQ调度时首先检查低延时队列
* 低延时和非低延时队列之间的关系类似PQ，为防止“饿死”，设置低延时队列的带宽

**[CCNA 1.8].IP寻址**

# 【子网】

一个子网中，第一个主机地址是xxx001，最后一个主机是xxx110，网络地址是xxx000，广播地址是xxx111

## 【IPv4地址】

单播地址范围：0.0.0.0~223.255.255.255(中间含其他地址)

本地网络广播：255.255.255.255

定向广播：xxx.yyy.zzz.255/24

本地组播地址范围：224.0.0.0~239.255.255.255(中间含…)

全局组播地址范围：224.0.1.0~238.255.255.255

私有地址：10.0.0.0/8 ~ 10.255.255.255.255/8

172.16.0.0/12 ~ 172.31.255.255/12

192.168.0.0/16 ~ 192.168.255.255/16

环回地址：127.0.0.1(127.0.0.0/8中的地址都会环回)

本地链路地址：168.254.0.0/16 ~ 169.254.255.255

TEST-NET地址：192.0.2.0/24 ~ 192.0.2.255/24，仅教学使用

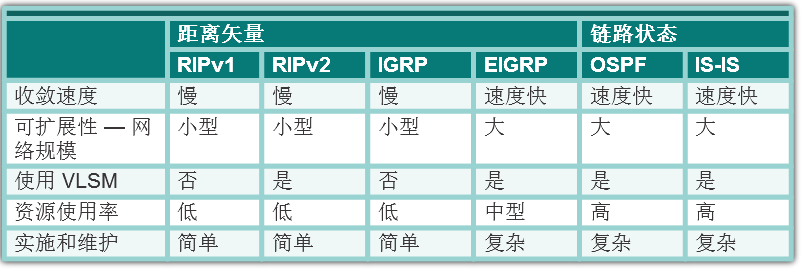
实验地址：240.0.0.0~255.255.255.254

## 【无类编址（CIDR）】

无类域间路由，不同于A,B,C类地址，可自选择网络位长度

**[CCNA 2.7].动态路由**

# 【路由协议分类】



## 【静态路由优缺点】

优点：

* 容易在小型网络中实施静态路由。
* 静态路由保持不变，这使它们很容易进行故障排除。
* 静态路由不发送更新消息，因此，几乎不需要开销。

缺点：

* 不容易在大型网络中实施。
* 管理静态配置非常耗时。
* 如果连接失败，静态路由无法重新路由流量。

## 【动态路由优缺点】

优点：

* 适用于需要多个路由的所有拓扑中
* 通常不受网络规模限制
* 如果拓扑发生变化，该协议会自动确定更好的路由。

缺点：

* 可能会使实施更加复杂。
* 不够安全，需要其他配置设置来确保安全。
* 根据当前拓扑进行路由
* 需要占用额外的CPU，RAM和链路带宽

# 【EIGRP】

数据结构区域，EIGRP创建和维护：

拓扑表、邻居表、路由表中的最佳路径。不含更新表

路由协议消息区域，EIGRP使用：

Hello，应答，更新，查询，确认

确定最佳路由，EIGRP使用：

DUAL算法（和IGRP一样； RIP：贝尔曼·福特）

# 【动态路由协议的运行过程】

1. 路由器通过其接口发送和接收路由消息。
2. 路由器与使用同一路由协议的其他路由器共享路由消息和路由信息。
3. 路由器通过交换路由信息来了解远程网络。
4. 如果路由器检测到网络拓扑结构的变化，路由协议可以将这一变化告知其他路由器。

# 【动态路由协议开销】

**RIP：**跳数

**OSPF：**开销（积累链路带宽）

**EIGRP**：延迟（数据包传输时间），带宽，负载（链路流量数量），可靠性（链路故障的可能性）

# 【RIPv2改进了RIPv1之处】

* 无类路由协议 - 它支持 VLSM 和 CIDR，因为它在路由更新中包含子网掩码。
* 提高效率 - 它将更新转发至组播地址 224.0.0.9，而不是广播地址 255.255.255.255。
* 减少路由条目 - 它支持所有接口上的手动路由总结。
* 安全 - 它支持身份验证机制以保证邻居之间路由表更新的安全。

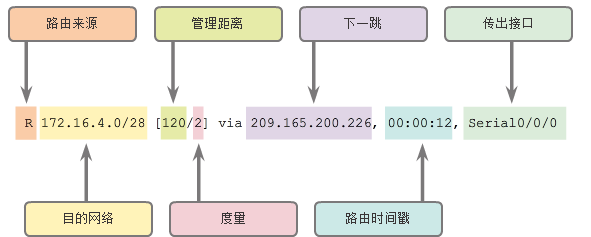
# 【链路状态路由协议工作过程】

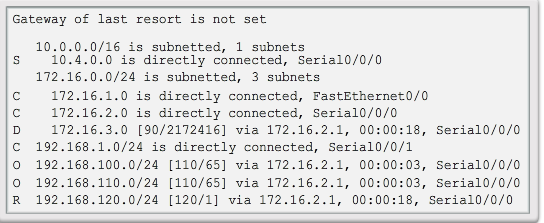
1. 每台路由器了解其自身的链路和与其直连的网络。 这通过检测哪些接口处于工作状态来完成。
2. 每台路由器负责“问候”直连网络中的相邻路由器。 链路状态路由器通过与直连网络中的其他链路状态路由器互换 Hello 数据包来达到此目的。
3. 每台路由器创建链路状态数据包 (LSP)，其包含与该路由器直接相连的每条链路状态。 这通过记录每个邻居的所有相关信息（包括邻居 ID、链路类型和带宽）来完成。
4. 每台路由器将 LSP 泛洪到所有邻居。 这些邻居将收到的所有 LSP 存储到数据库。 接着，它们将 LSP 泛洪给自己的邻居，直到区域中的所有路由器均收到这些 LSP 为止。 每台路由器会在本地数据库中存储邻居发来的 LSP 的副本。
5. 每台路由器使用数据库构建一个完整的拓扑图并计算通向每个目的网络的最佳路径。 就像拥有了地图一样，路由器现在拥有关于拓扑中所有目的地以及通向各个目的地的路由的完整地图。 SPF 算法用于构建该拓扑图并确定通向每个网络的最佳路径。

## 【链路状态发送LSP时机】

* 在初始启动该路由器上的路由协议过程期间（例如，路由器重启）。
* 每次拓扑结构发生变化时（包括链路断开或接通，或是相邻关系建立或破裂）

# 【路由表条目读取】







# 【路由表解释】

* S - 用于确定管理员手动创建的通往特定网络的路由。 这是一种静态路由。
* D - 用于确定使用 EIGRP 路由协议从另一台路由器动态获知的路由器。
* O - 用于确定使用 OSPF 路由协议从另一台路由器动态获知的路由器。
* R - 用于确定使用 RIP 路由协议从另一台路由器动态获知的路由器。
* C - 用于确定直连网络。 当某个接口配置了 IP 地址并激活时，将会自动创建直连网络。
* L - 表示这是本地路由。 为接口配置 IP 地址并激活时，会自动创建本地路由。
* 静态直连网络 管理距离为1

**[CCNA 2.8].OSPF**

# 【管理距离】



# 【多区域OSPF优点】

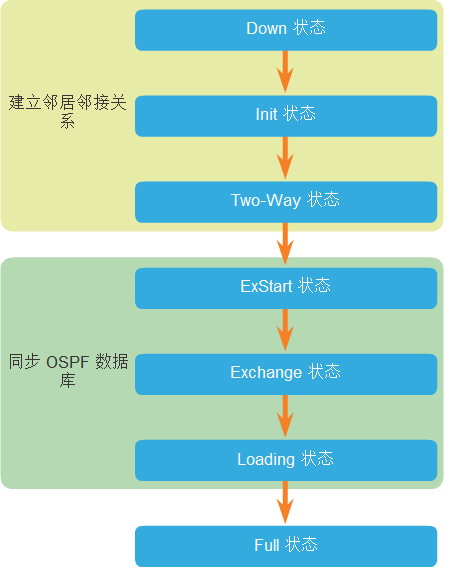
* **路由表减小** - 路由表条目减少，因为区域之间的网络地址可以总结。 默认情况下不启用路由总结。
* **链路状态更新开销减少** - 将处理和内存要求降到最低。
* **SPF 计算频率降低** - 使拓扑变化仅影响区域内部。 例如，由于 LSA 泛洪在区域边界终止，它使路由更新的影响降到最小。

### 【OSPF消息作用】

DBD：在路由器间检查数据库同步情况

LSU：发送所请求的特定链路状态记录

# 【OSPF状态】



# 【OSPF状态变化：建立邻居邻接关系】

当在接口上启用 OSPF 时，路由器将通过所有启用 OSPF 的接口转发包含其路由器 ID 的 Hello 数据包，接口从 Down 状态转变为 Init 状态，当路由器收到一个 Hello 数据包，且其路由器 ID 在邻居列表中，路由器将从 Init 状态转变为 Two-Way 状态，Two-Way 状态下执行的操作取决于相邻路由器之间的连接类型：

* 如果两个邻接的邻居通过点对点链路相互连接，则可立即从 Two-Way 状态进入数据库同步阶段。
* 如果路由器通过通用的以太网络相互连接，则必须选择指定路由器 DR 和 BDR。

# 【OSPF状态变化：同步OSPF数据库】

ExStart 状态下，每个路由器与其相邻的 DR 和 BDR 之间会形成主从关系，Exchange 状态下，主路由器和从属路由器交换一个或多个 DBD 数据包，收到新的链路状态条目后，路由器将转换为 Loading 状态，当所有 LSR 满足特定路由器的要求时，邻接路由器被视为已同步并处于 full 状态。

# 【OSPF工作流程】

1. **建立邻居邻接关系**- 启用了 OSPF 的路由器必须在网络中互相识别对方，才能共享信息。 启用 OSPF 的路由器将 Hello 数据包从所有启用 OSPF 的接口发送出去，以确定这些链路上是否存在邻居。 如果存在邻居，启用 OSPF 的路由器将尝试与该邻居建立邻接关系。
2. **交换链路状态通告**- 建立邻接关系之后，路由器会交换链路状态通告 (LSA)。 LSA 包含每个直连链路的状态和开销。 路由器将其 LSA 泛洪到邻居。 收到 LSA 的邻接邻居立即将 LSA 泛洪到其他直接连接的邻居，直到区域中的所有路由器收到所有 LSA。
3. **建立拓扑表**- 在收到 LSA 后，启用 OSPF 的路由器根据收到的 LSA 构建拓扑表 (LSDB)。 此数据库最终负责维护有关网络拓扑的信息。
4. **执行 SPF 算法**- 路由器将执行 SPF 算法。 图中的齿轮用于表示 SPF 算法的执行过程。 SPF 算法创建一个 SPF 树。