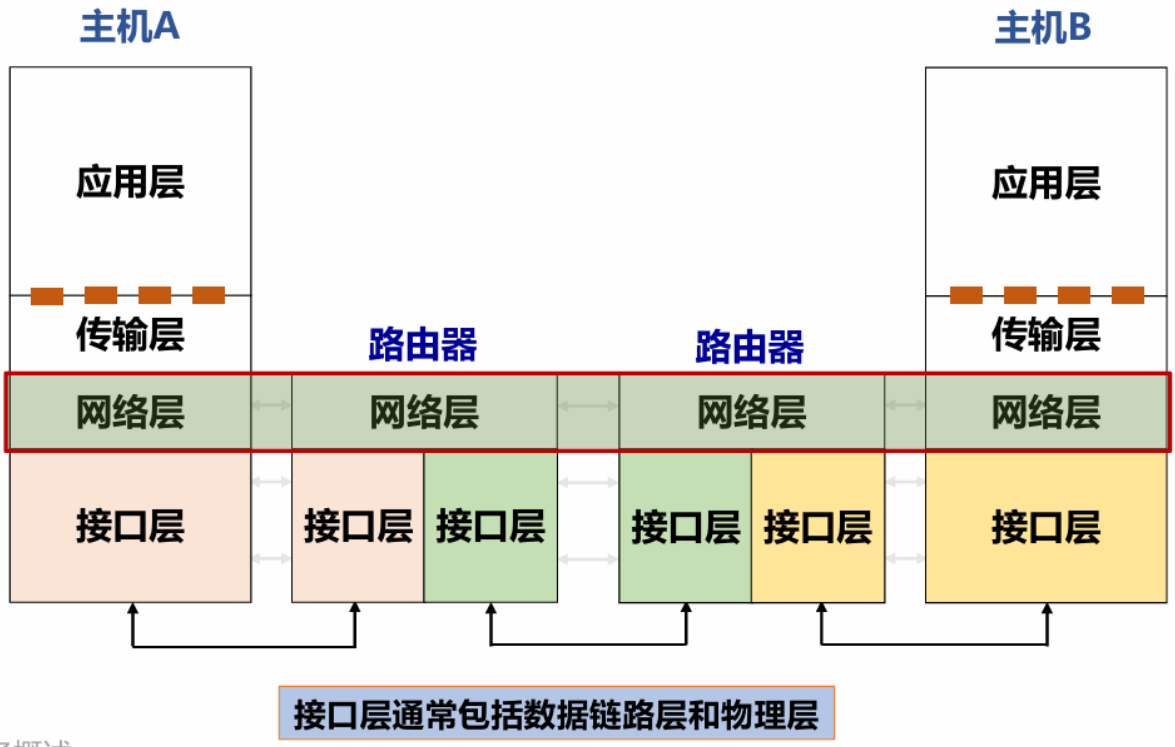
第五章 网络层

网络层服务



络层服务概述

实现：端系统间多跳传输

功能存在于每个主机和路由器中

- 发送端：将传输层数据单元封装在数据包中
- 接收端：解析接收的数据包中，取出传输层数据单元，交付给传输层
- 路由器：检查数据包首部，转发数据包

关键功能：**路由和转发**

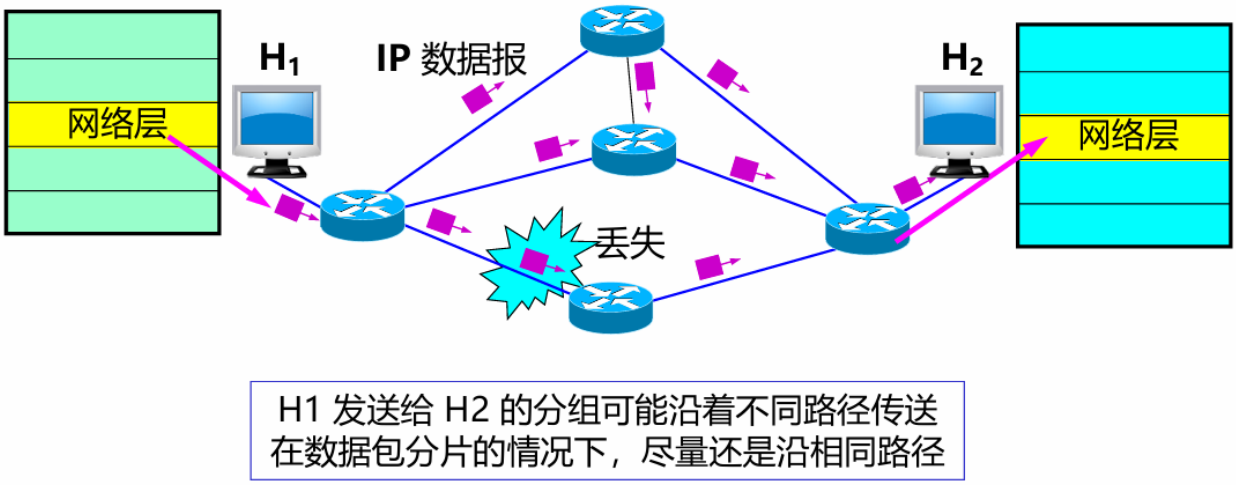
无连接服务：数据报

- 网络层向上只提供简单灵活无连接的、尽最大努力交付的数据报服务
- 发送分组时不需要先建立连接，每个分组独立发送
- 数据报独立转发，相同源-目的的数据报可能经过不同的路径

- 网络层不提供服务质量的承诺

缺点：传输网络不提供端到端的可靠传输服务：丢包、乱序、错误

优点：网络的造价大大降低，运行方式灵活，能够适应多种应用



面向连接服务：虚电路

当建立一个连接时，从源机器到目标机器之间的一条路径就被当作这个连接的一部分 确定了下来，并且保存在这些中间路由器的表中

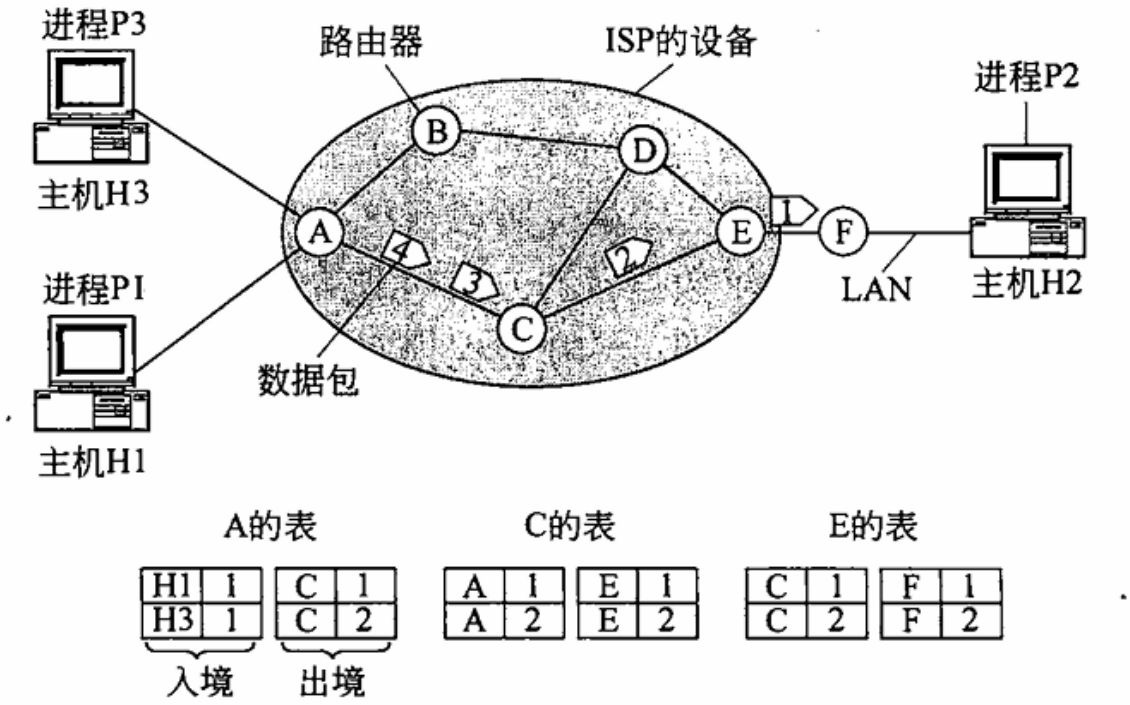


图 5-3　虚电路网络的路由过程

在路由器A上打标签，1代表H1发送的，2代表H3发送的

比较

虚电路到达发送端是有序的，而数据报是无序的

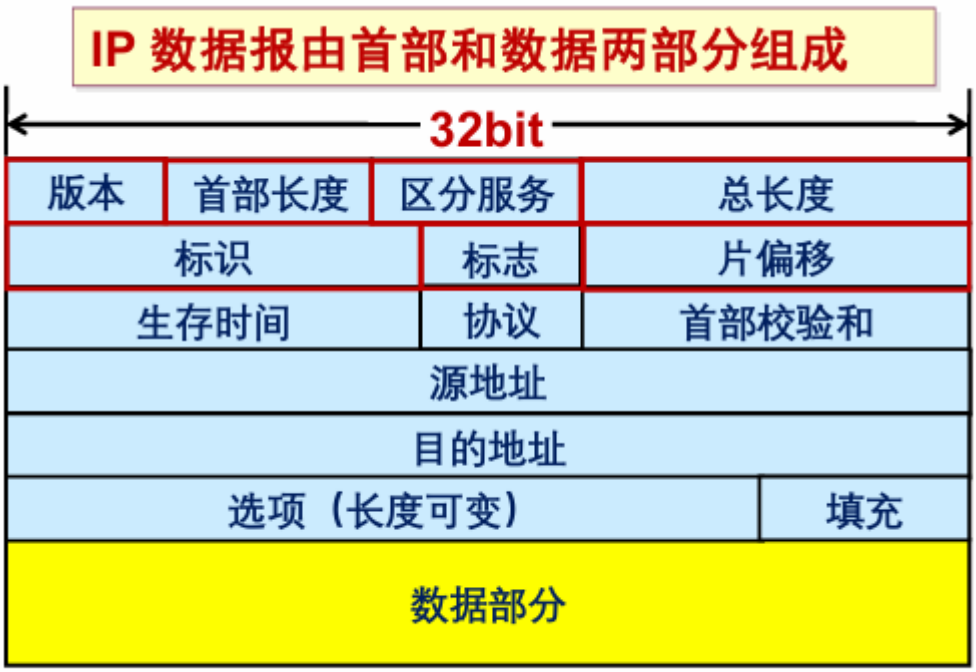
Internet网际协议

internet协议执行两个基本功能：寻址(addressing) 分片(fragmentation)

IPv4协议

- 版本：4bit，表示采用的IP协议版本
- 首部长度：4bit，表示整个IP数据报首部的长度
- 区分服务：8bit，该字段一般情况下不使用
- 总长度：16bit，表示整个IP报文的长度,能表示的最大字节为2^16-1=65535字节
- 标识：16bit，IP软件通过计数器自动产生，每产生1个数据报计数器加1；在ip分片以后，用来标识同一片分片
- 标志：3bit，目前只有两位有意义；MF，置1表示后面还有分片，置0表示这是数据报片的最后1个；DF，不能分片标志，置0时表示允许分片
- 片偏移：13bit，表示IP分片后，相应的IP片在总的IP片的相对位置

- 生存时间TTL(Time To Live)： 8bit,表示数据报在网络中的生命周期，用通过路由器的数量来计量，即跳数（每经过一个路由器会减1）
- 协议：8bit，标识上层协议（TCP/UDP/ICMP...）
- 首部校验和：16bit ，对数据报首部进行校验，不包括数据部分
- 源地址：32bit，标识IP片的发送源IP地址
- 目的地址：32bit，标识IP片的目的地IP地址
- 选项：可扩充部分，具有可变长度，定义了安全性、严格源路由、松散源路由、记录路由、时间戳等选项
- 填充：用全0的填充字段补齐为4字节的整数倍



数据报分片

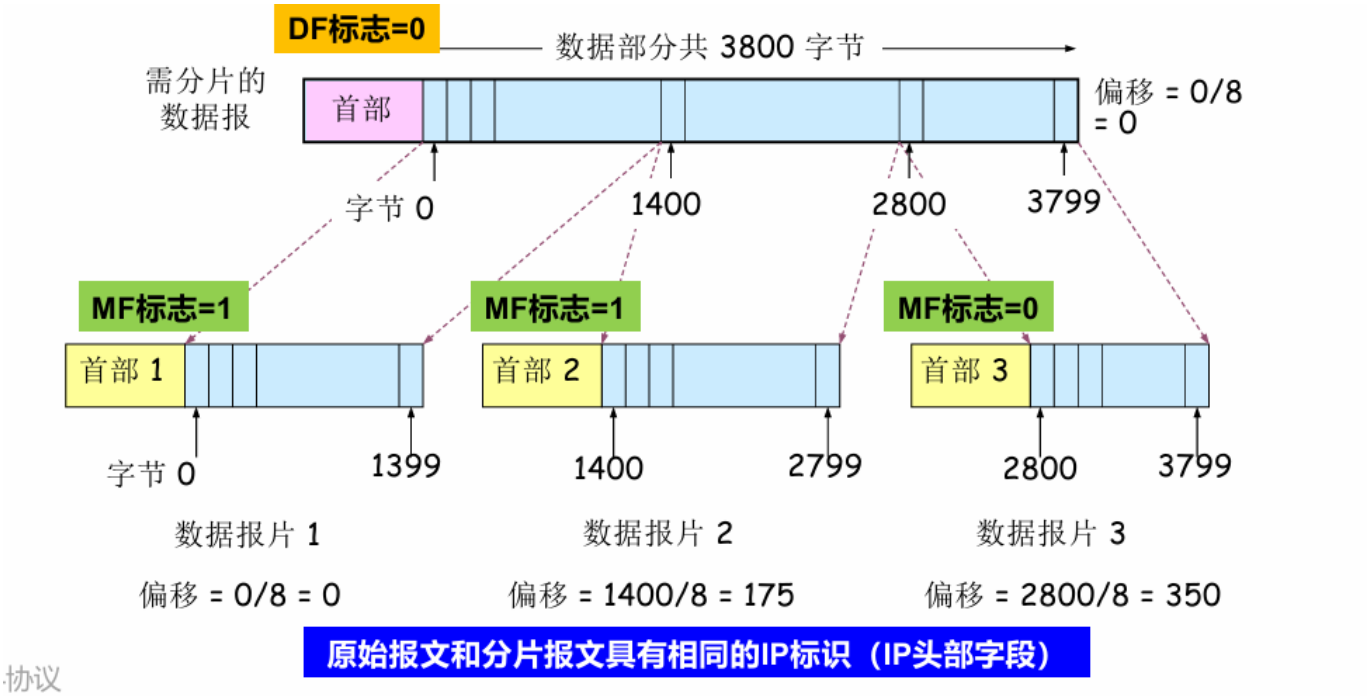
MTU（Maximum Transmission Unit），最大传输单元

分片策略：

- 允许途中分片：根据下一跳链路的MTU实施分片
- 不允许途中分片：发出的数据报长度小于路径MTU

重组策略：往往目的端重组

IPv4选择途中分片+目的端重组



协议

IP地址

IP地址，网络上的每一台主机（或路由器）的每一个接口都会分配一个全球唯一的32位的标识符

IP地址共分为A、B、C、D、E五类，A类、B类、C类为单播地址；书写采用点分十进制记法，其中每一段取值范围为0到255

A类：0~127

B类：128~191

C类：192~233

D类：224~239 组播

E类：240~255

特殊地址：

地址	用途
全0网络地址	只在系统启动时有效，用于启动时临时通信，又叫主机地址
网络127.0.0.0	指本地节点(一般为127.0.0.1)，用于测试网卡及TCP/IP软件，这样浪费了1700万个地址
全0主机地址	用于指定网络本身，称之为网络地址或者网络号
全1主机地址	用于广播，也称定向广播，需要指定目标网络
0.0.0.0	指任意地址
255.255.255.255	用于本地广播，也称有限/受限广播，无须知道本地网络地址

子网： 在一个网络内部需要分块，每一部分叫做一个子网。每一个子网内的主机的IP地址前n位都是相同的，称为前缀；将前缀的位全部置1，称为子网掩码

例子：

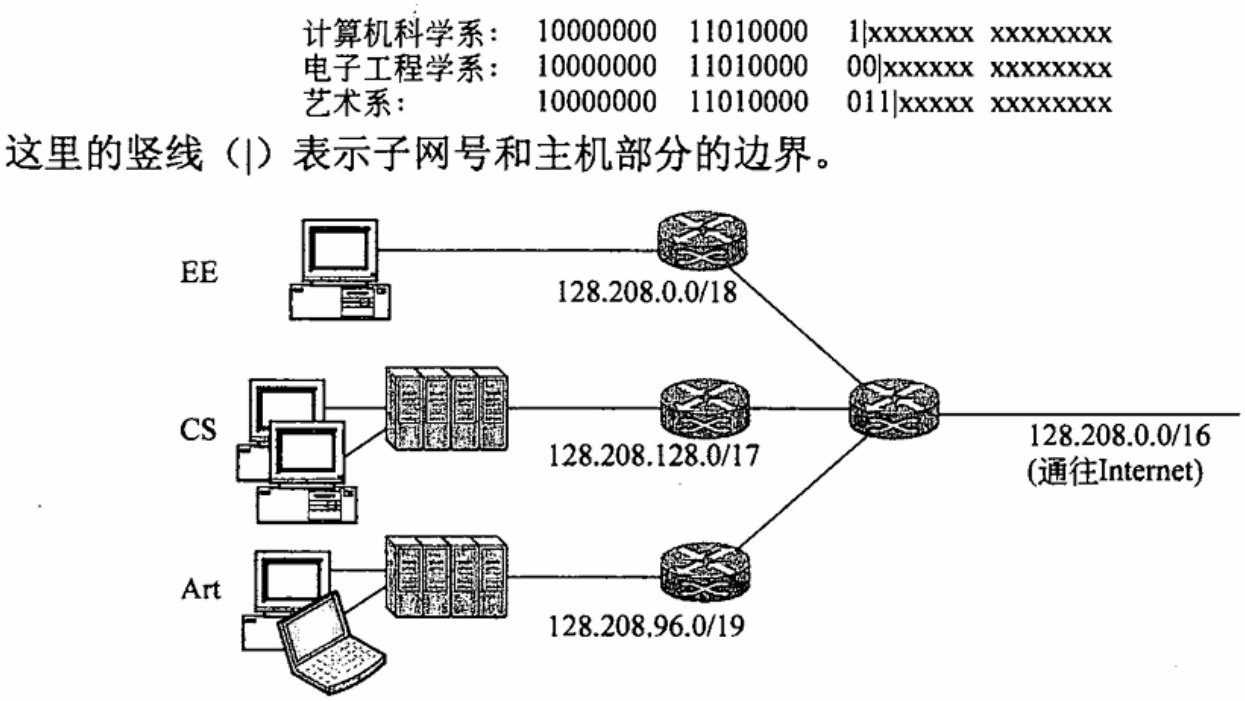


图 5-49 将 IP 前缀进一步细分为网络和子网

假如现在收到一个发往 IP 地址 128.208.2.151 的数据包，为了看它是否属于计算机科学系，我们把该目标地址与 255.255.128.0 进行 AND 操作，获得前 17 位（即128.208.0.0），并且检查它们是否匹配计算机系的前缀地址（即 128.208.128.0）。显然它们不匹配。然后再检查前 18 位，在与电机工程学系的子网掩码 AND 操作后，我们得到128.208.0.0，这恰好匹配电机工程系的前缀地址

无类域间路由 CIDR

在这里我们把多个小前缀的地址块合并成一个大前缀的地址块。这个合并过程称为路由聚合（routeaggregation），由此产生的较大前缀地址块有时称为超网（supernet），以便有别于地址块的分割。

有了地址聚合， IP地址可包含大小不等的前缀。同样一个IP地址，一台路由器把它当作/22 的一部分对待，而另一台路由器把它当作一个更大的/20一部分对待。这是因为每个路由器有相应的前缀信息。这个设计和子网划分协同工作，统称为无类域间路由（CIDR, Classless Inter-Domain Routing）

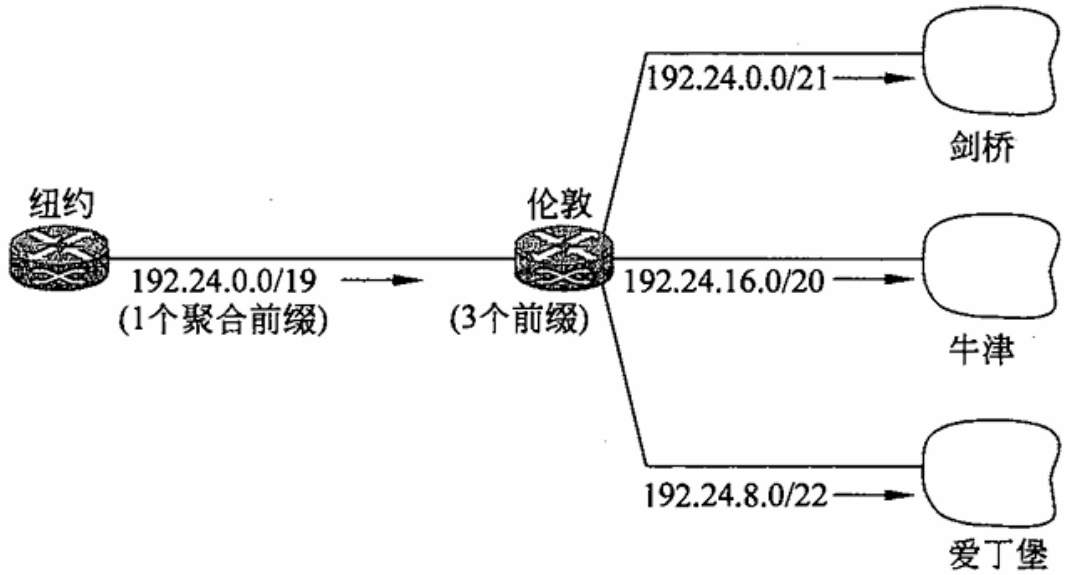


图 5-51　IP 前缀的聚合

如图中的例子，三个大学的子网并到伦敦的边缘路由器，再向其他地区传播时只需一个聚合前缀而不是三个小前缀，节省了路由表

图 5-52　纽约路由器中的最长匹配前缀路由

图52则是描述了**最长匹配前缀**。当一个数据包来到纽约时，查找与它匹配的前缀，可能左边右边都匹配就优先发往前缀更长的那边（/22），/22不匹配但/19匹配的发往/19

获取IPv4地址：静态、动态(DHCP)

DHCP 工作过程

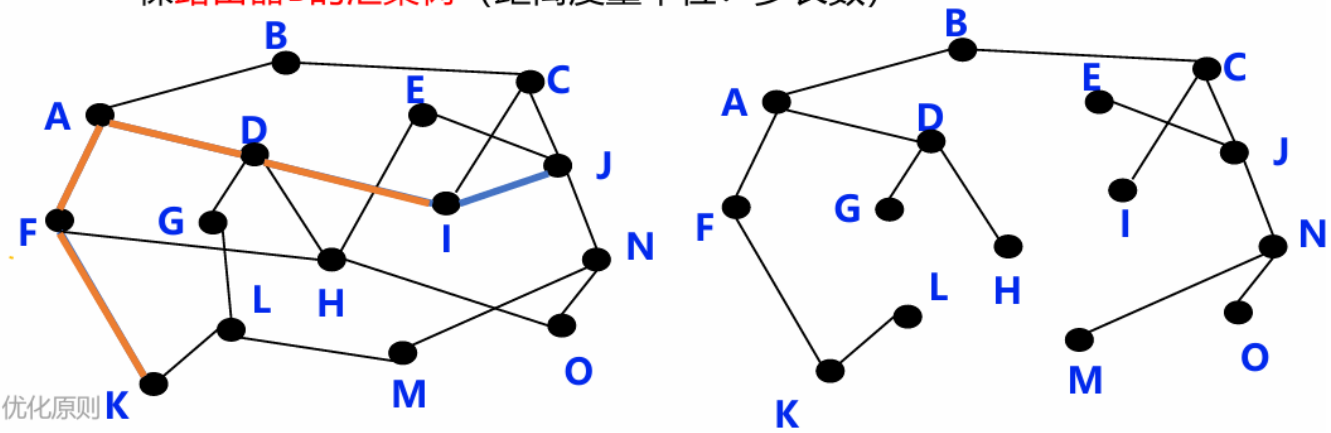
- DHCP 客户从UDP端口68以广播形式向服务器发送发现报文（DHCPDISCOVER）
- DHCP 服务器单播发出提供报文（DHCPOFFER）
- DHCP 客户从多个DHCP服务器中选择一个，并向其以广播形式发送DHCP请求报文（DHCPREQUEST）
- 被选择的DHCP服务器单播发送确认报文（DHCPACK）

路由算法

优化原则

从所有的源到一个指定目标的最优路径的集合构成了一棵以目标节点为根的树。这样的树称为汇集树（sink tree）

- 一棵路由器B的汇集树（距离度量单位：步长数）



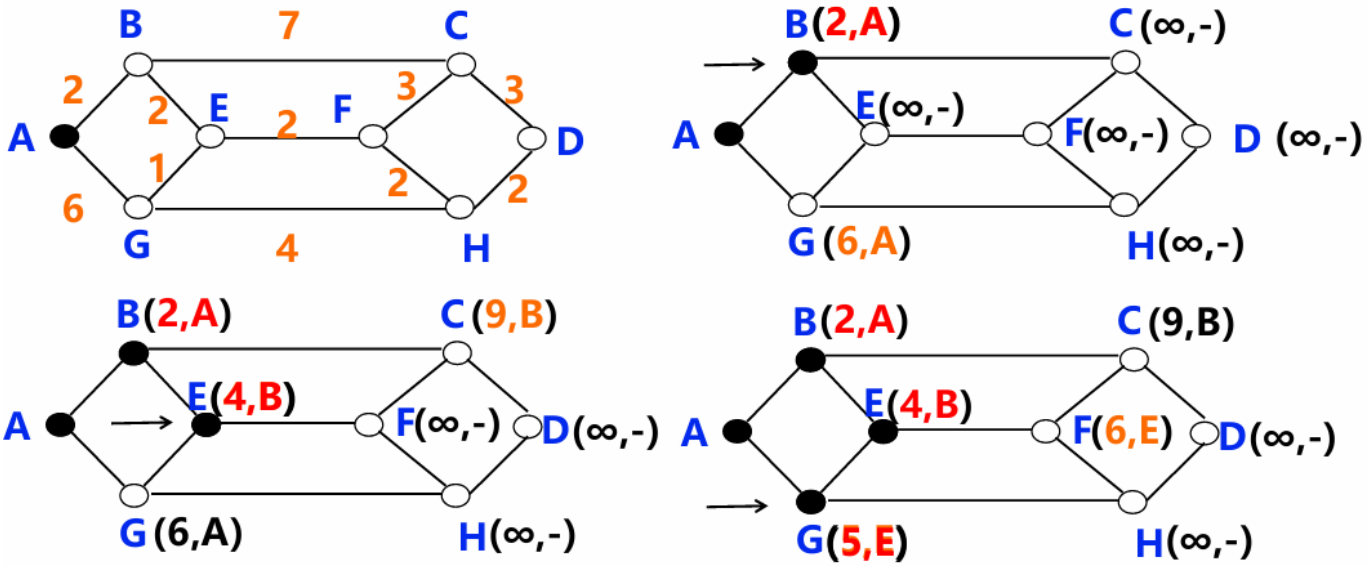
3.1 优化原则

最短路径算法

Dijkstra算法思想

建立网络图：节点表示路由器；边表示通信线路/链路；链路代价表示链路上的距离、信道宽度或通信开销等参数。根据算法在网络图上为某一对路由器找之间的最短路径

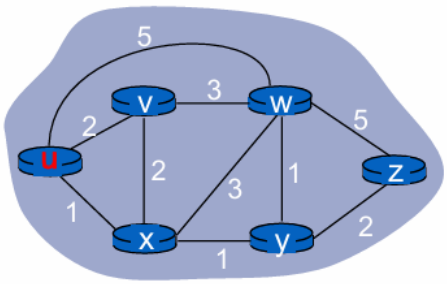
➢ 具体例子（A到D的最短路径过程）



距离向量路由

➢ Bellman-Ford 方程

- 假设$D_x(y)$ 是 从x到y最小代价路径的代价值
- 则：$D_x(y) = \min \{c(x,m) + D_m(y)\}$
  其中m为x的邻居, c(x,m)为m到X的距离



- 示例：
  - 已获知$D_v(z) = 5$, $D_x(z) = 3$, $D_w(z) = 3$
  - $D_u(z) = \min \{ c(u,v) + D_v(z), c(u,x) + D_x(z), c(u,w) + D_w(z) \}$
    $= \min \{2 + 5, 1 + 3, 5 + 3\}$
    $= 4$
- 计算出代价最小的节点，也就得到了对应转发项从u去往z，应从x转发

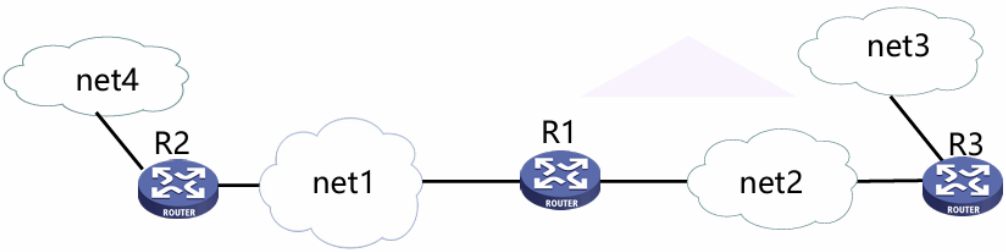
距离向量（Distance Vector）算法基本思想:

- 每个节点周期性地向邻居发送它自己到某些节点的距离向量
- 当节点x接收到来自邻居的新DV估计，它使用B-F方程更新其自己的DV：$D_x(y) = min\{c(x,v) + D_v(y)\}$,v代表x的每一个邻居
- 迭代这个过程，x到y的距离收敛到最小距离

具体实现:

➢ 路由器启动时初始化自己的路由表

- 初始路由表包含所有直接相连的网络路径，距离均为0



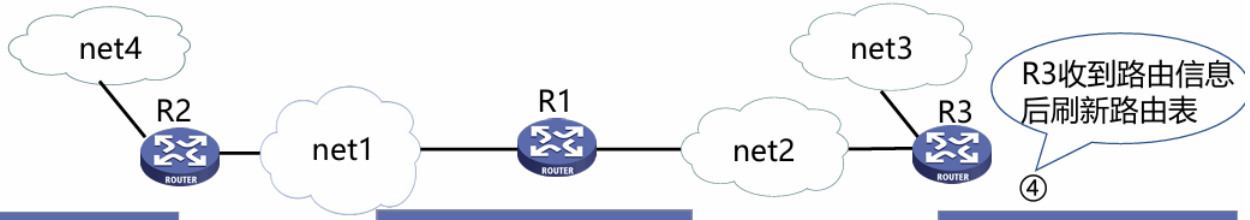
目的网络	距离	下一跳
net1	0	--
net4	0	--

目的网络	距离	下一跳
net1	0	--
net2	0	--

目的网络	距离	下一跳
net2	0	--
net3	0	--

➢ 路由器周期性地向其相邻路由器广播自己知道的路由信息

➢ 相邻路由器可以根据收到的路由信息修改和刷新自己的路由表



目的网络	距离	下一跳
net1	0	--
net4	0	--

①

R2将路由信息发送给R1

目的网络	距离	下一跳
net1	0	--
net2	0	--
net4	1	R2

②

R1收到路由信息后刷新路由表

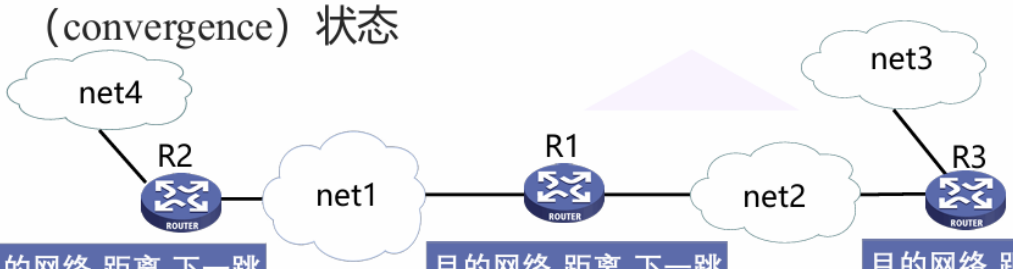
③

R1将路由信息发送给R3

目的网络	距离	下一跳
net2	0	--
net3	0	--
net1	1	R1
net4	2	R1

➢ 路由器经过若干次更新后，最终都会知道到达所有网络的最短距离

➢ 所有的路由器都得到正确的路由选择信息时网络进入“收敛”（convergence）状态



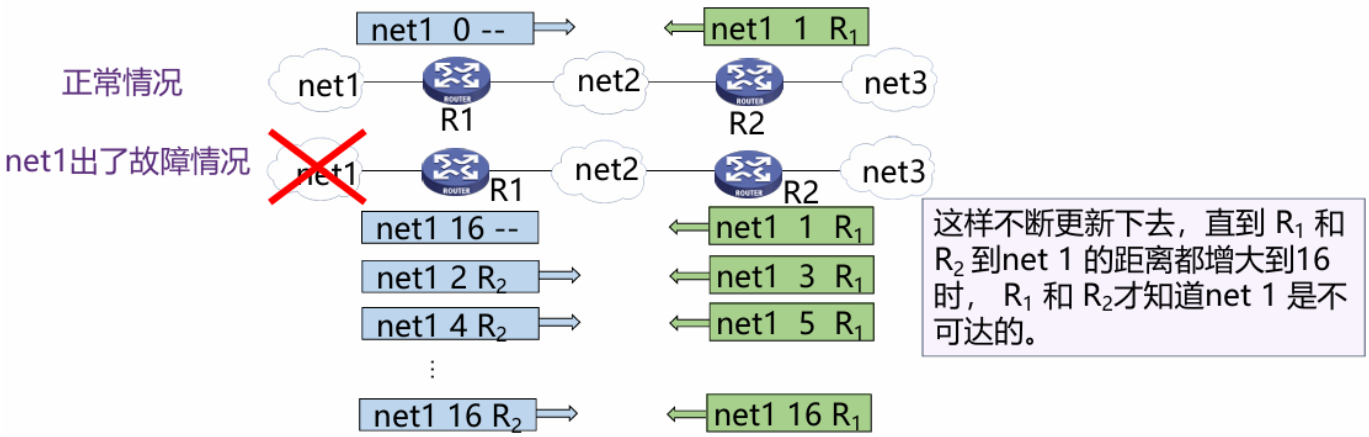
目的网络	距离	下一跳
net1	0	--
net4	0	--
net2	1	R1
net3	2	R1

目的网络	距离	下一跳
net1	0	--
net2	0	--
net4	1	R2
net3	1	R3

目的网络	距离	下一跳
net2	0	--
net3	0	--
net1	1	R1
net4	2	R1

可能会发生计数到无穷问题，如图中情况，net1损坏，但是R1接收了R2的消息，使其继续更新路由表，这样反复迭代直至两个R到net1的距离都到16才停止

➢ 计数到无穷问题（The Count-to-Infinity Problem）



➢好消息传播快，坏消息传播慢，是距离向量路由的一个主要缺点

链路状态路由

链路状态（Link State）路由可分为五个部分：

- 发现邻居，了解他们的网络地址：通过问候得知当前路由器要通过哪个网段连接哪个路由器
- 设置到每个邻居的成本度量：常用链路带宽做为代价值，反比
- 构造一个分组，分组中包含刚收到的所有信息(LSP)：发送方标识 序列号 年龄 邻居列表
- 将此分组发送给其他的路由器：当一个新分组到达时，路由器根据记录判断：

- 如果是新分组，洪泛广播
- 如果是重复分组，丢弃
- 如果是过时分组，拒绝

- 计算到其他路由器的最短路径。用dijkstra自己算出到所有其他节点的距离，构成路由表

➢ 距离向量和链路状态算法比较：

- 网络状态信息交换的范围
  - DV:邻居间交换
  - LS:全网扩散
- 网络状态信息的可靠性
  - DV:部分道听途说
  - LS:自己测量
- 健壮性:
  - DV:计算结果传递，健壮性差
  - LS: 各自计算，健壮性好
- 收敛速度：
  - DV: 慢,可能有计数到无穷问题
  - LS: 快

层次路由

网络非常多时会导致路由表过于庞大，不便于查找和存储，于是需要分层

自治系统（AS，Autonomous System）

自治系统内部使用IGP协议，外部使用EGP协议

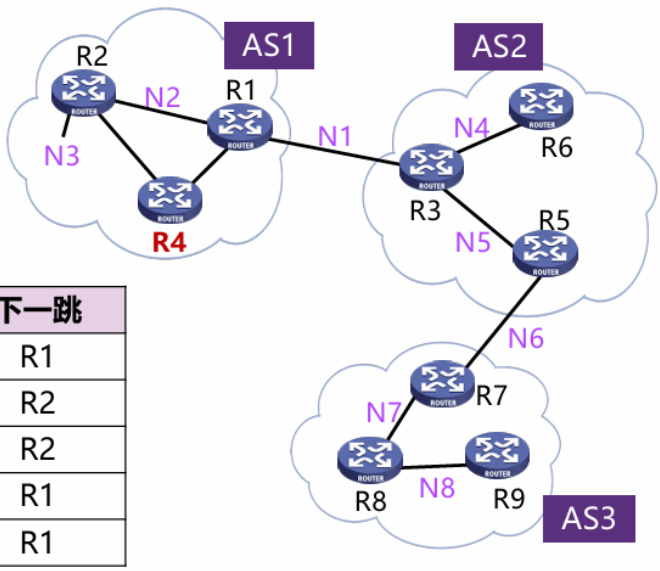
层次路由-效果

路由不分层的情况下
R4的路由表

目的地	下一跳
N1	R1
N2	R2
N3	R2
N4	R1
N5	R1
N6	R1
N7	R1
N8	R1

R4的层次表

目的地	下一跳
N1	R1
N2	R2
N3	R2
AS2内的所有网络	R1
AS3内的所有网络	R1



广播路由

广播（Broadcasting）：源主机同时给**全部目标地址**发送同一个数据包

例：服务器希望将视频广播给3个网络中的所有30个用户，有哪些办法?

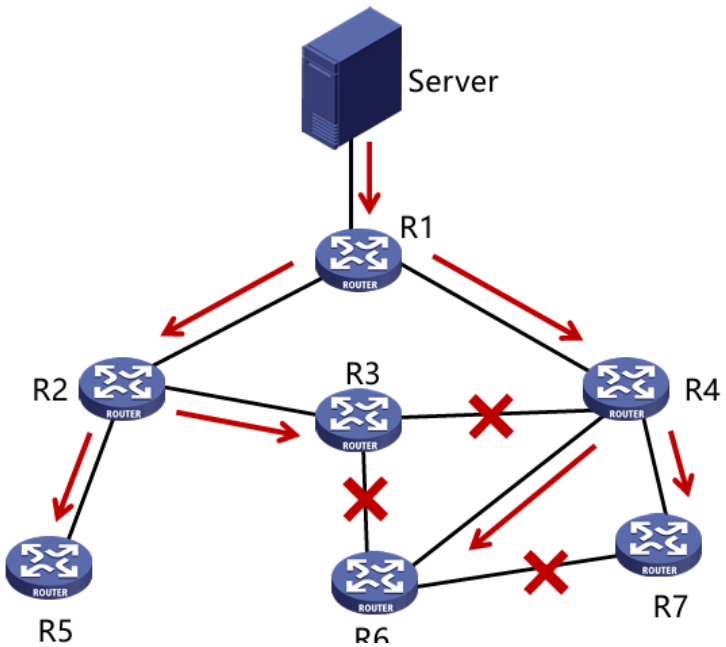
- 方法1：给每个主机单独发送一个数据包 效率低，服务器需要知道所有主机的IP
- 方法2：多目标路由，在需要转发的路由器线路复制一次该数据报 服务器需要知道所有主机的IP 难以实现
- 方法3：泛洪（flooding）一种将数据包发送到所有网络节点的简单方法 将每个进入数据包发送到除了进入线路外的每条出去线路 **无限制的泛洪可能导致广播风暴**

解决方法：**受控制的泛洪**： 序号控制泛洪、逆向路径转发

- 方法4：生成树（spanning tree）

➢ 方法4：生成树（spanning tree）

- 源节点向所有属于该生成树的特定链路发送分组
- 改进了逆向路径转发
- 没有环路
- 最佳使用带宽
- 最少副本，消除了冗余分组
- 一个路由器可以不必知道整颗树，只需要知道在一颗树中的邻居即可



组播路由

组播（multicasting）：源主机给网络中的**一部分目标用户**发送数据包

组播路由算法的目标： 为每个组建立多播转发树（到达该组所有成员的路径树）

生成树：

- 密集分布，基于源点树（source-based trees） 大型网络中，组播源很多时，路由器需生成多颗棵树，工作量巨大

- 链路状态路由算法：每个路由器针对组内的每个发送者构造一颗独立树，例如多播开放最短通路优先协议（Multicast Open Shortest Path First，MOSPF）
- 距离向量路由算法：逆向路径转发，修剪没有组成员的路由器，例如距离向量多播路由协议（Distance Vector Multicast Routing Protocol，DVMRP）、协议无关多播-稠密模式（Protocol Independent Multicast - Dense Mode，PIM-DM）

协议无关指的是与单播路由协议无关，即PIM不需要维护专门的单播路由信息

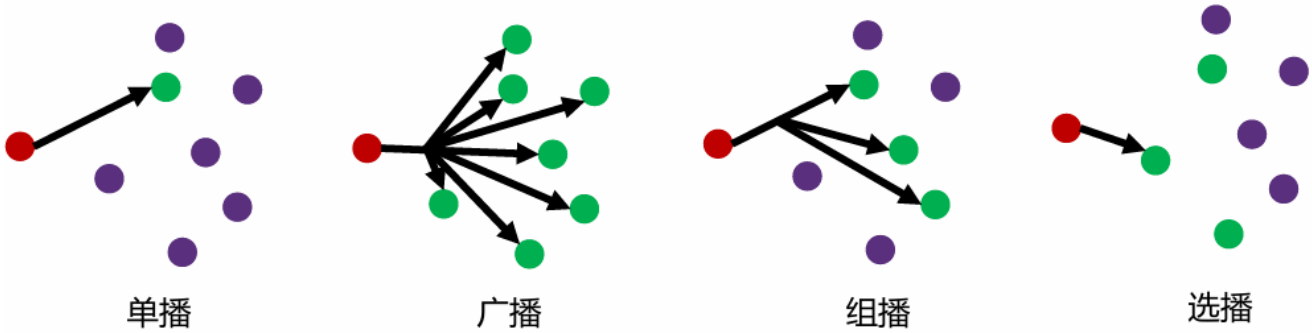
- 稀疏分布，基于核心树（core-based trees）

  设置一个核心，所有发送源发送的数据都要先到核心，再向需要接收的结点组播发送；问题：核心树可能无法达到最优：有些节点可能不需要经过核心就能直接与发送方连接，走核心反而更复杂

选播路由

将数据包传送给最近的一个组成员 在有多个服务器的情况下，用户希望快速获得正确信息，而不在乎从哪个服务器获得

- 选播（Anycast）
  - 将数据包传送给最近的一个组成员
  - 在有多个服务器的情况下，用户希望快速获得正确信息，而不在乎从哪个服务器获得
  - 与单播、广播、组播的区别



典型应用：DNS

Internet路由协议

OPSF-内部网关路由协议

OSPF（Open Shortest Path First）开放最短路径优先协议

采用分布式的链路状态算法 “链路状态”就是说明本路由器都和哪些路由器相邻，以及该链路的“度量”

基本思想

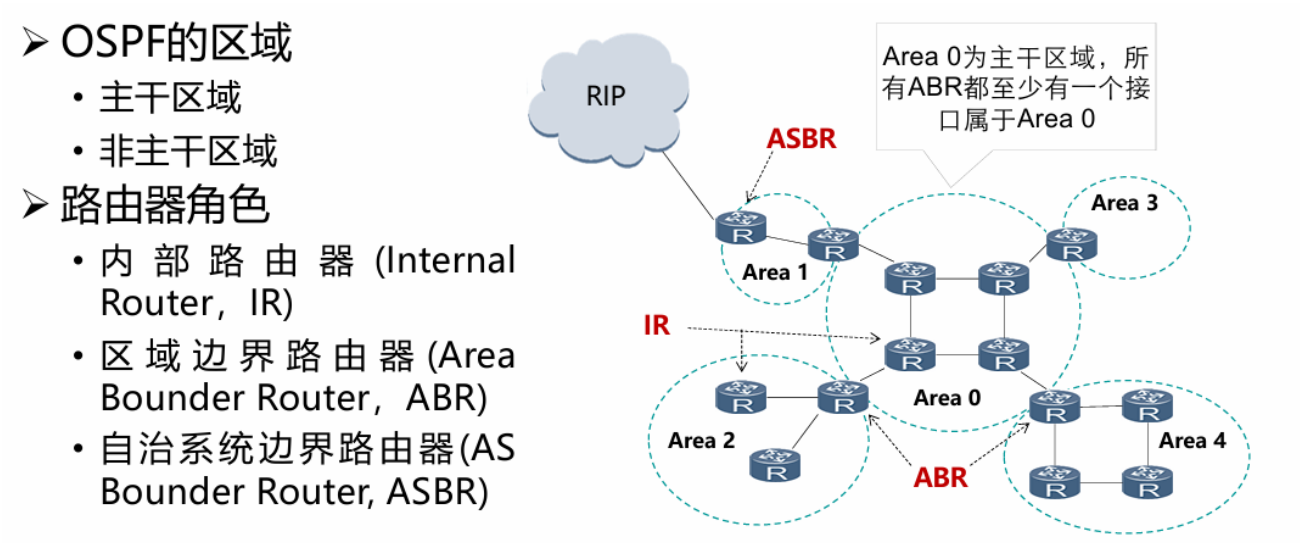
- 向本自治系统中所有路由器洪泛信息
- 发送的信息就是与本路由器相邻的所有路由器的链路状态
- 只有当链路状态发生变化时路由器才用洪泛法发送此信息

报文交互：

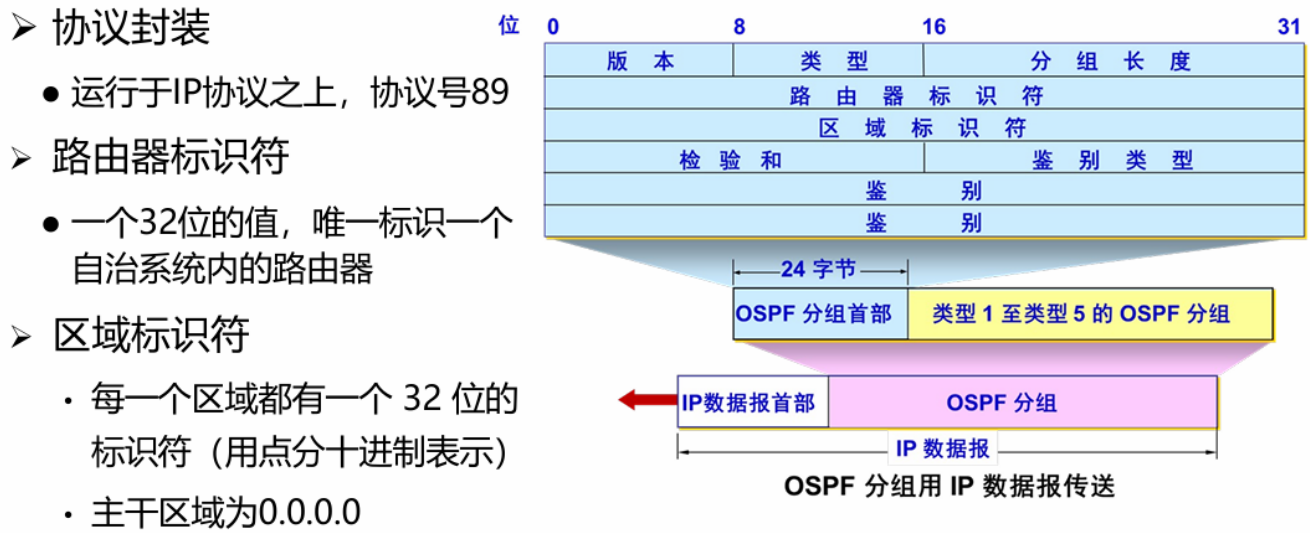
- 发现邻居-HELLO报文，检查参数，如果一致则形成邻居关系
- 数据库同步-DD报文，来进行主从路由器的选举和数据库摘要信息的交互
- 建立完全邻接关系：
    - LSR用于向对方请求所需的LSA
    - LSU用于向对方发送其所需要的LSA
    - LSACK用于向对方发送收到LSA的确认

OSTP中的**区域**：

- 详细的描述拓扑结构的链路状态信息仅在区域内传递，区域间传递的是抽象的路由信息
- 使用层次结构的区域划分，上层的区域叫做主干区域
- 划分区域可以缩小LSDB规模，减少网络流量



OSTP报文格式：



RIP-内部网关路由协议

路由选择协议RIP（Routing Information Protocol）是基于距离矢量算法的协议

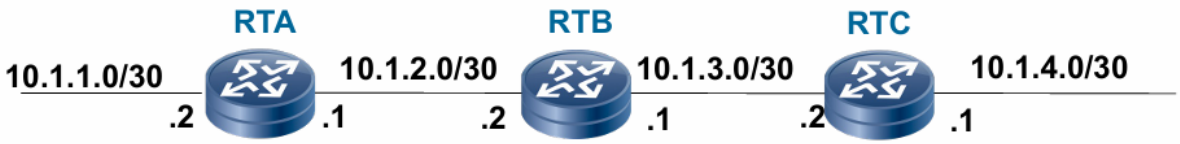
使用跳数来衡量距离，一个路径最多15个路由器

基本思想：

- 仅和相邻路由器交换信息
- 路由器交换的内容是自己的路由表
- 周期性更新：30s

工作过程：

➢ 初始化

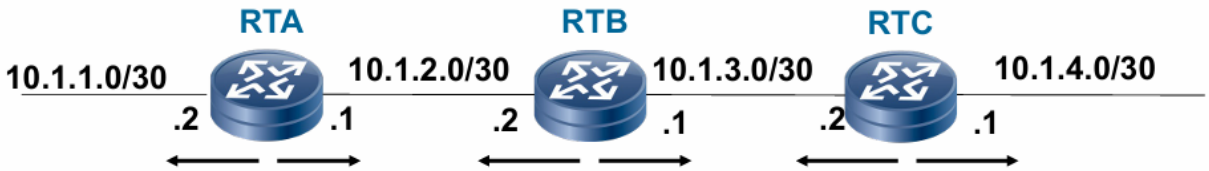


目标网络	下一跳	距离
10.1.1.0	– –	0
10.1.2.0	– –	0

目标网络	下一跳	距离
10.1.2.0	– –	0
10.1.3.0	– –	0

目标网络	下一跳	距离
10.1.3.0	– –	0
10.1.4.0	– –	0

➢ 周期性更新

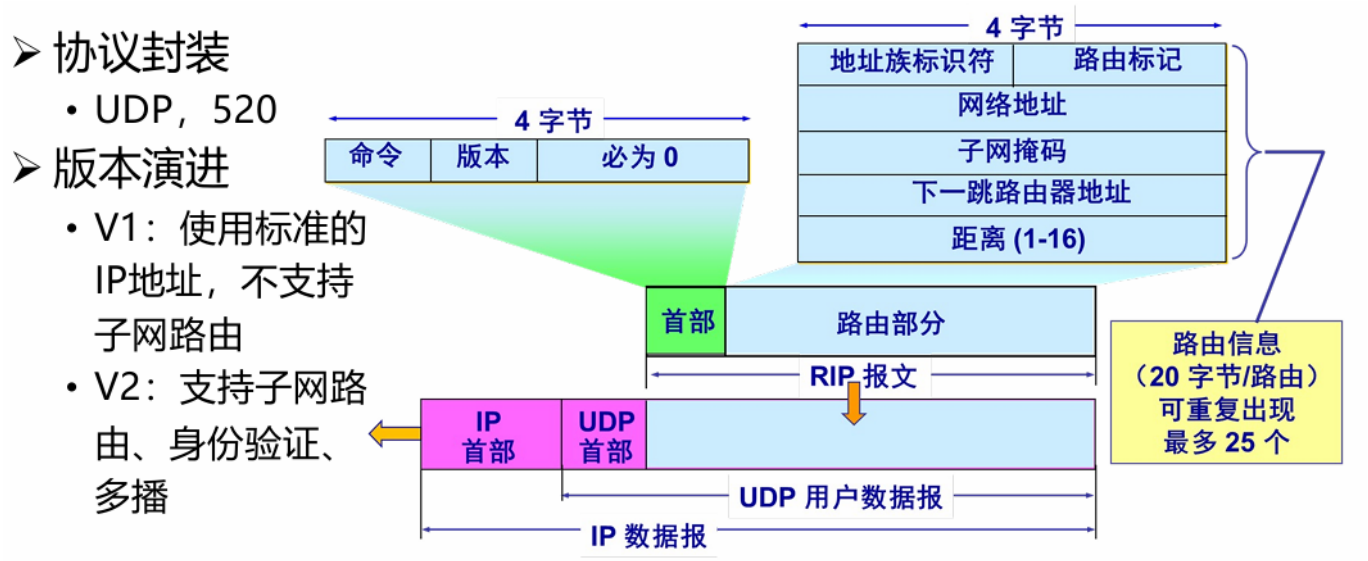


目标网络	下一跳	距离
10.1.1.0	– –	0
10.1.2.0	– –	0
10.1.3.0	10.1.2.2	1
10.1.4.0	10.1.2.2	2

目标网络	下一跳	距离
10.1.2.0	– –	0
10.1.3.0	– –	0
10.1.1.0	10.1.2.1	1
10.1.4.0	10.1.3.2	1

目标网络	下一跳	距离
10.1.3.0	– –	0
10.1.4.0	– –	0
10.1.2.0	10.1.3.1	1
10.1.1.0	10.1.2.1	2

报文格式：



BGP-外部网关路由协议

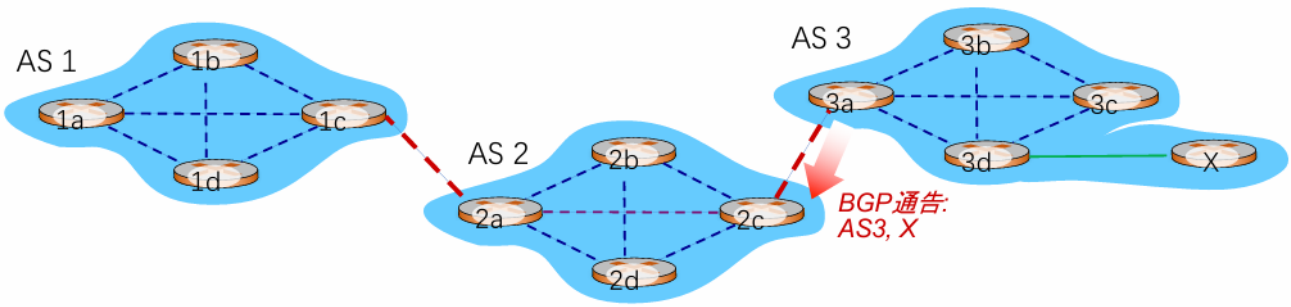
边界网关协议BGP (Border Gateway Protocol) 目前互联网中唯一实际运行的自治域间的路由协议

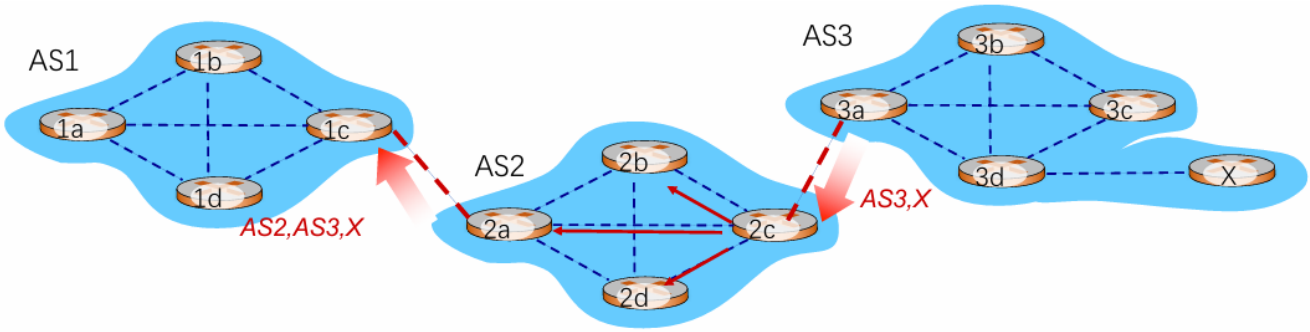
功能

- eBGP：从相邻的AS获得网络可达信息
- iBGP： 将网络可达信息传播给AS内的路由器
- 基于网络可达信息和策略决定到其他网络的“最优”路由

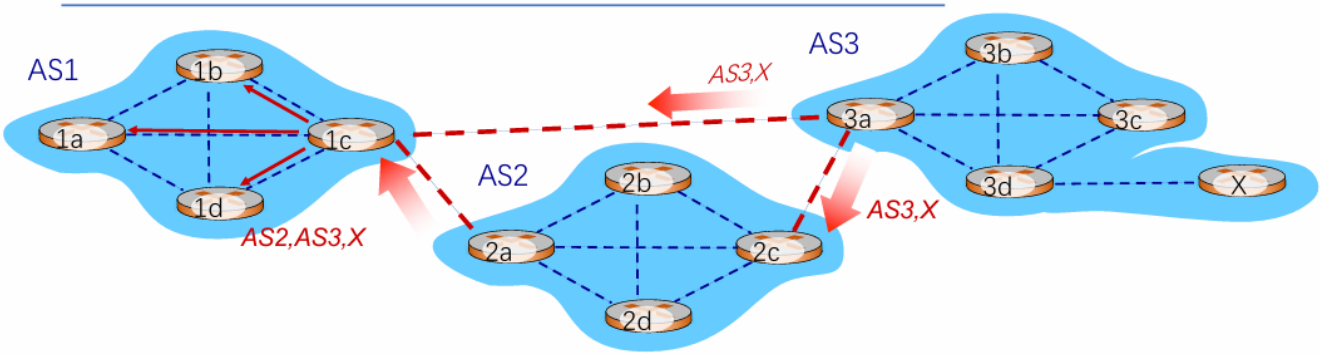
BGP会话过程：

- 例：当AS3的路由器3a向AS2的路由器2c通告路径AS3, X时AS3向AS2承诺它会向X转发数据包





- AS2的路由器2c从AS3的路由器3a接收到路径AS3, X
- 根据AS2的策略，AS2的路由器2c接受路径AS3, X，通过iBGP传播给AS2的所有路由器
- 根据AS2策略，AS2的路由器2a通过eBGP向AS1的路由器1c通告从AS3的路由器3a接收到路径AS2, AS3, X



路由器可能会学到多条到目的网络的路径:

- AS1的路由器1c从2a学到路径*AS2, AS3, X*
- AS1的路由器1c从3a学到路径*AS3, X*
- 由策略，AS1路由器1c可能选择路径*AS3, X,* 并在AS1中通过iBGP通告路径

标签交换和MPLS

MPLS (MultiProtocol Label Switching)全称是多协议标签交换

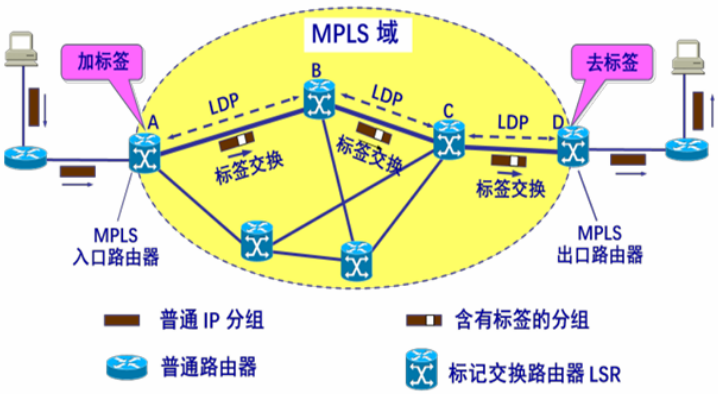
设计初衷为了提升查找速度

主要有以下三个方面的应用

- 面向连接的服务质量管理
- 流量工程，平衡网络负载
- 虚拟专用网VPN

标签交换和MPLS

- 标签交换路由器LSR
  - 支持MPLS的路由器
  - 具备标签交换、路由选择两种功能
- MPLS 域
  - 所有相邻的支持MPLS技术的路由器构成的区域
- 标签分配协议LDP
  - 用来在LSR之间建立LDP 会话并交换Label/FEC映射信息



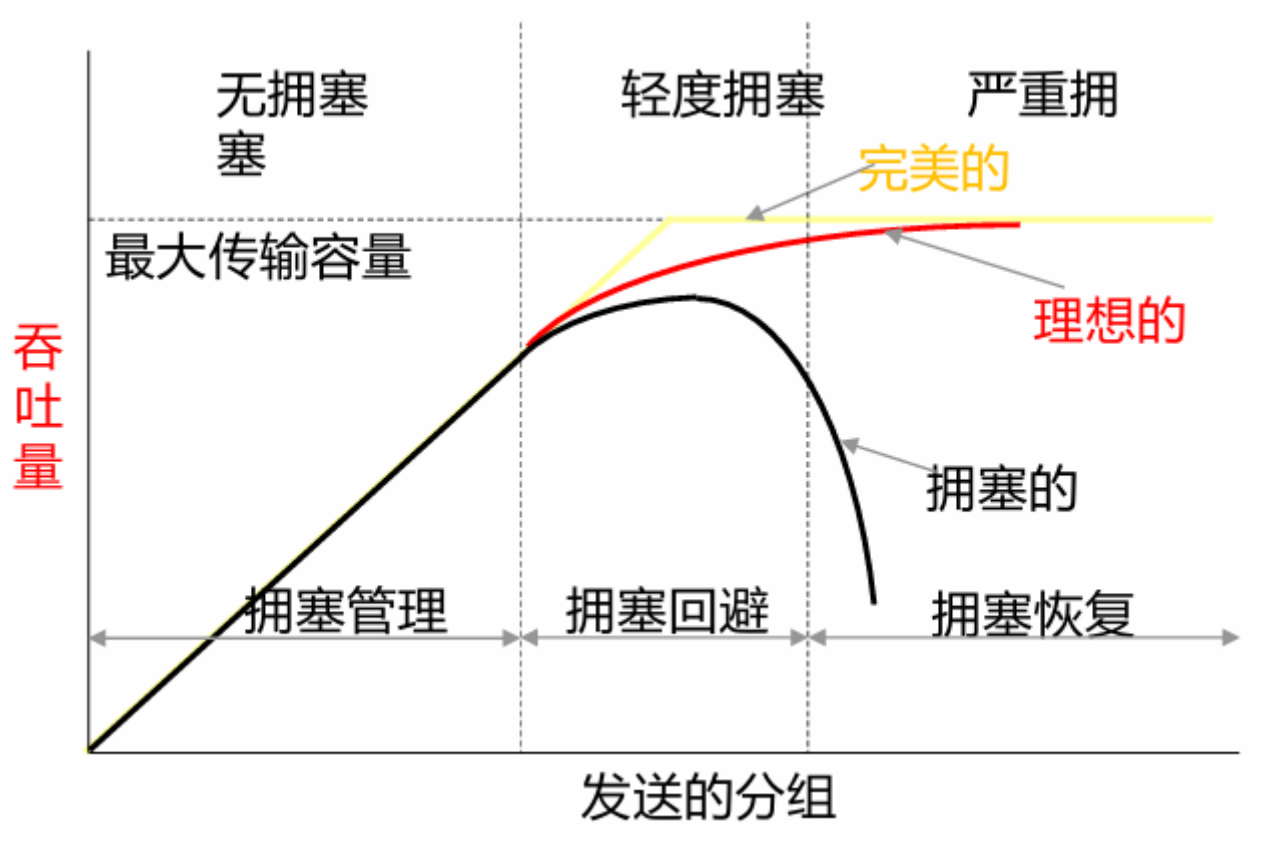
拥塞控制

拥塞：网络中存在太多的数据包导致数据包传输延迟或丢失，从而导致网络吞吐量下降

拥塞控制（congestion control）：需要确保通信子网能够承载用户提交的通信量，是一个全局性问题，涉及主机、路由器等多种因素

产生拥塞的原因：主机发送到网络的数据包数量过多，超过了网络的承载能力；突发的流量填满了路由器的缓冲区，造成某些数据包会被丢弃

简单来说就是发送方发的太多了



策略：开环控制（参数不能动态变化）、闭环控制（根据反馈动态变化）

方法：提高网络供给 流量感知路由 准入控制 流量调节 负载丢弃

流量感知路由

基本思想是绕开热门区域

计算路径权重时包含跳数、带宽、传输延迟、负载、平均排队延迟等参数；因为正常情况下会只根据跳数来选择最短路径

路由表可能会出现反复变化，从而导致**不稳定的路由**

流量调节

抑制包：发现拥塞的路由器给拥塞包的源地址发一个信息（抑制包）

逐跳抑制包：抑制包不发给源地址，而是发给来的路径上的路由器

显式拥塞通告（ECN）：在IP包头中记录数据包是否经历了拥塞

RFC2474中重新定义TOS域为包含一个6位的区分服务码点(DSCP)和2位未使用位；**RFC3168重新定义RFC2474中TOS域未使用的两位为ECN域**，包含如下值：

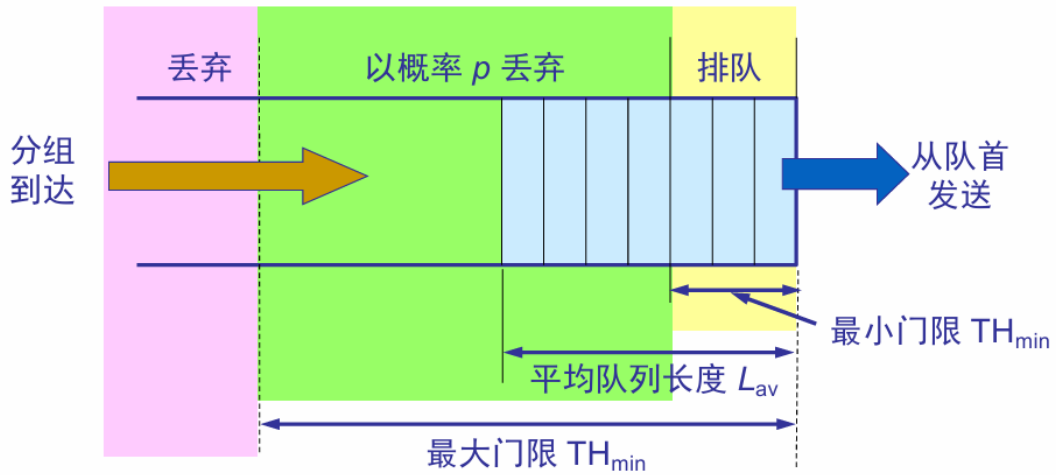
- 00：发送主机不支持ECN
- 01或者10：发送主机支持ECN
- 11：路由器正在经历拥塞

随机早期检测

随机早期检测 RED (Random Early Detection)

- 使路由器的队列维持两个参数，即队列长度最小门限 TH_{min} 和最大门限 TH_{max}
- RED 对每一个到达的数据报都先计算平均队列长度 L_{AV}
- 若平均队列长度小于最小门限 TH_{min} ，则将新到达的数据报放入队列进行排队
- 若平均队列长度超过最大门限 TH_{max}，则将新到达的数据报丢弃
- 若平均队列长度在最小门限 TH_{min} 和最大门限TH_{max} 之间，则按照某一概率 p 将新到达的数据报丢弃

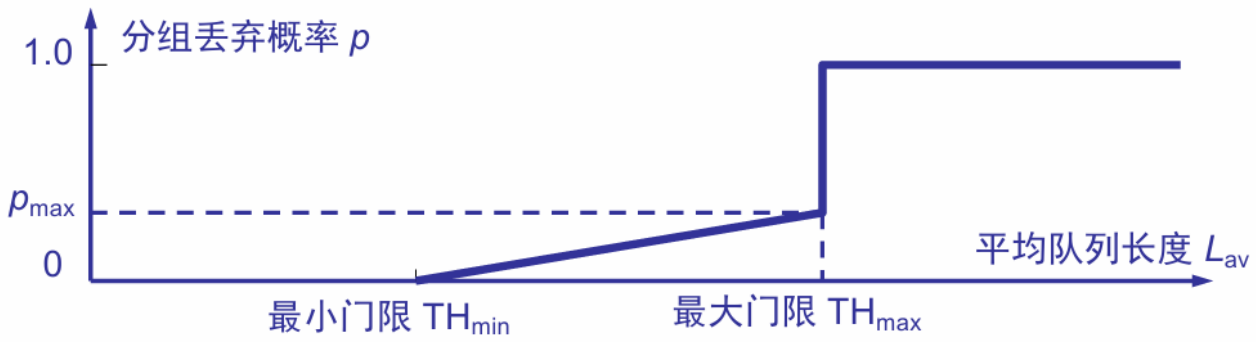
- RED 将路由器的到达队列划分成为三个区域



丢弃概率 p 与 TH_{min} 和 Th_{max} 的关系

- 当 $L_{AV} < Th_{min}$ 时，丢弃概率 $p = 0$
- 当 $L_{AV} > Th_{max}$ 时，丢弃概率 $p = 1$
- 当 $TH_{min} < L_{AV} < TH_{max}$ 时，$0 < p < 1$

  例如，按线性规律变化，从 0 变到 p_{max}



服务质量

网络服务质量（QoS, Quality of Service）QoS是网络在传输数据流时要满足一系列服务请求，具体可以量化为带宽、时延、抖动、丢包率等性能指标

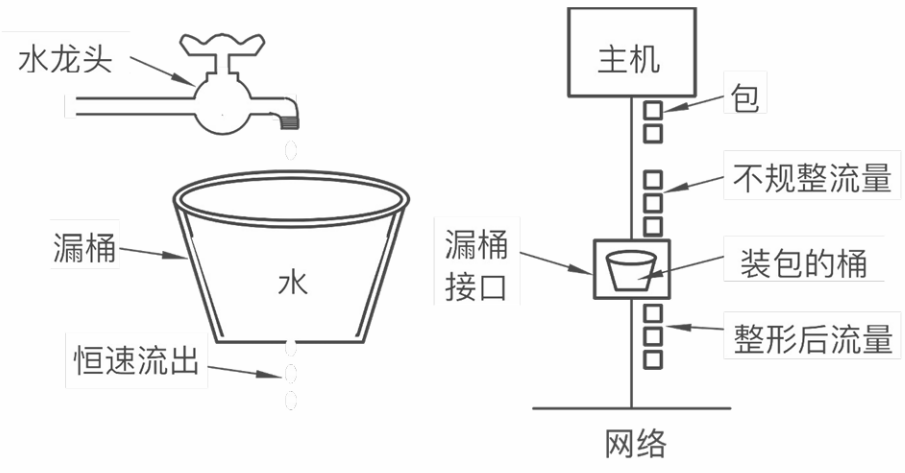
如何规范进入网络的流量?

流量整形(traffic shaping)：其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

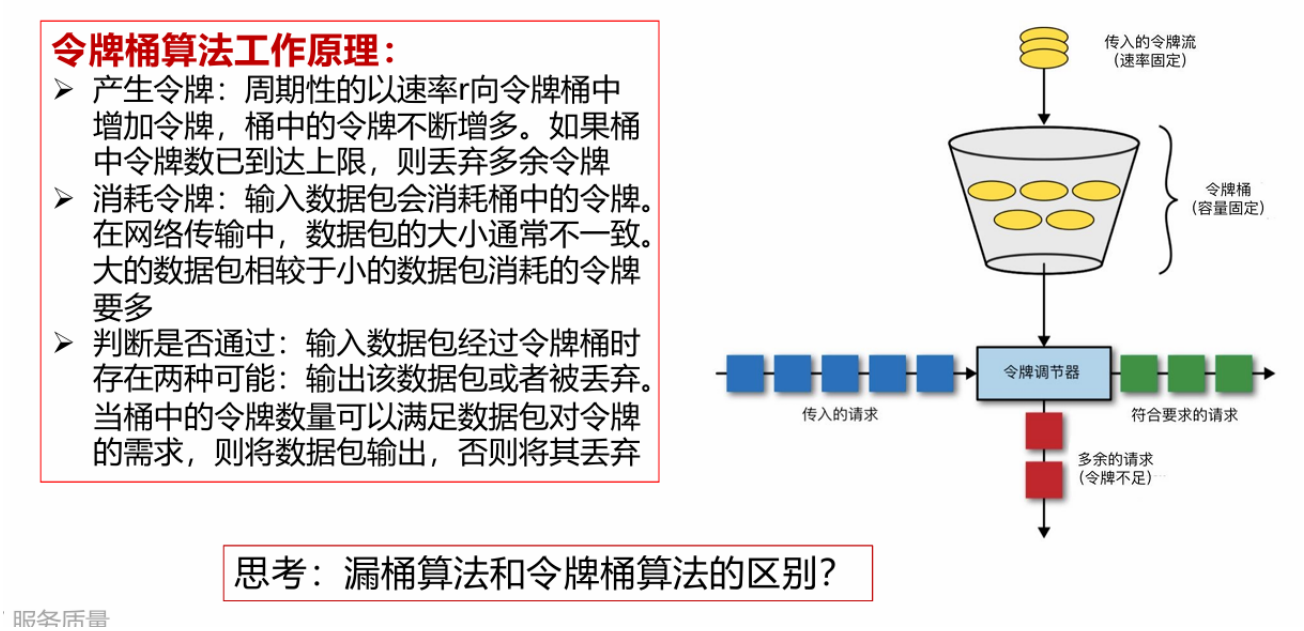
- 漏桶算法：通过它，突发流量可以被整形以便为网络提供一个稳定的流量

漏桶算法原理：

- 到达的数据包（网络层的PDU）被放置在底部具有漏孔的桶中（数据包缓存）
- 漏桶最多可以**排队b个字节**，漏桶的这个尺寸受限于内存。如果数据包到达的时候漏桶已经满了，那么数据包应被丢弃
- 数据包从漏桶中漏出，以常量速率（**r字节/秒**）注入网络，因此平滑了突发流量



- 令牌桶算法：用来控制发送到网络上的数据的数目，并允许突发数据的发送



为了保障性能如何在路由器预留资源?

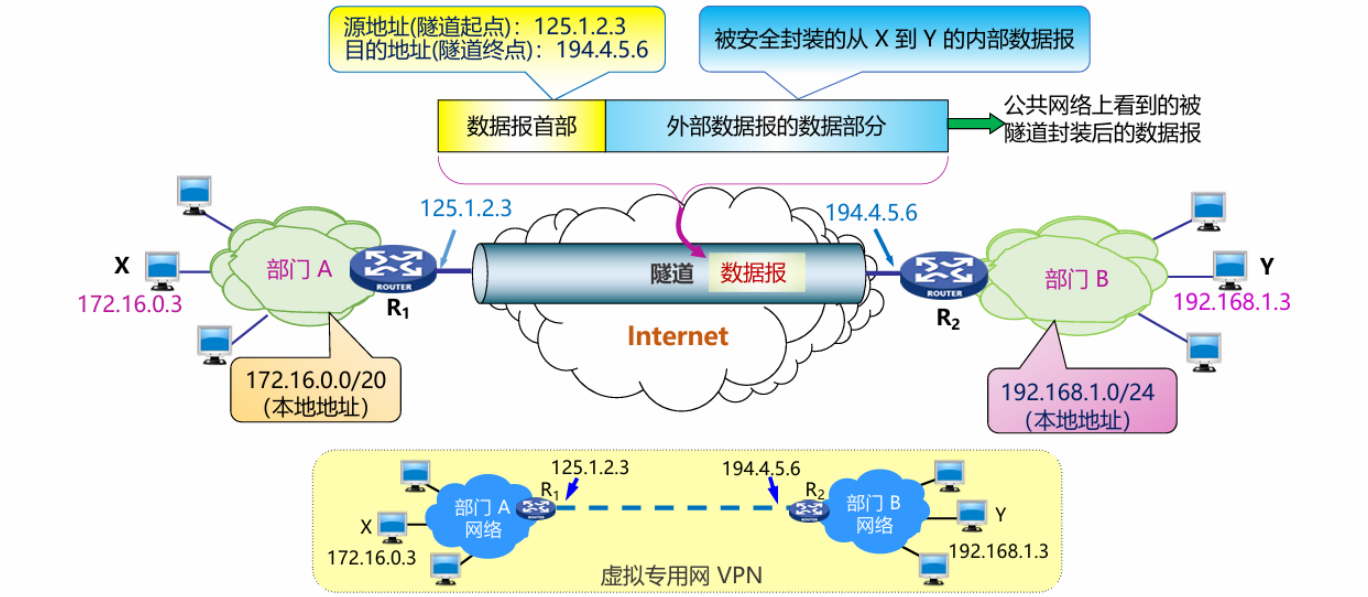
数据包调度：在同一个流的数据包之间以及在竞争流之间分配路由器资源的算法称为包调度算法，它负责分配带宽和其他路由器资源，负责确定把缓冲区中的哪些数据包发送到输出链路上

应用程序需要网络提供什么样的质量

综合服务(IntServ)、区分服务(DiffServ)

VPN

原理：VPN指利用公用网络架设专用网络的远程访问技术，通过隧道技术在公共网络上模拟出一条点到点的逻辑专线，从而达到安全数据传输的目的



8.2 VPN技术

IPv6

地址长度为128bit 地址空间数量约为3*10^38

冒分十六进制 每个x前面的0可省略，可把连续的值为0的x表示为“::”,且“::”只能出现1次

头部

长度固定段40字节，扩展段不一定

版本：4bit，协议版本号，值为6

流量类型：8bit，区分数据包的服务类别或优先级

流标签：20bit，标识同一个数据流

有效载荷长度：16bit ，IPv6报头之后载荷的字节数（含扩展头），最大值64K

下一个首部：8bit ，IPv6报头后的协议类型，可能是TCP/UDP/ICMP等，也可能是扩展头

跳数限制：8bit ，类似IPv4的TTL，每次转发跳数减1，值为0时包将会被丢弃

源地址： 128bit ，标识该报文的源地址

目的地址： 128bit ，标识该报文的目的地址

扩展头：可以有多个，每个扩展头都包含"下一个首部"字段（IPv6首部固定字段也有），可指向下一个扩展头类型，或指明传统上层协议类型（最后一个扩展头）：TCP/UDP/ICMP

IPv6分片机制：IPv6分组不能在传输途中分片，只在源端进行分片 设计了专门的分片扩展头，分片字段不存在基本IPv6头部中 支持Path MTU发现机制

IPv4-IPv6过渡：双栈技术 隧道技术 翻译技术