计算机网络

- 主讲：肖林

内容

- 第1章 概述
- 第2章 物理层
- 第3章 数据链路层
- 第4章 介质访问控制子层
- **第5章 网络层**
- 第6章 传输层
- 第7章 应用层

第5章　网络层

- 网络层的设计问题
- 单个网络中的路由算法
- 网络层的流量管理
- 服务质量和应用QoE
- 网络互联
- 软件定义网络
- 因特网的网络层

5.1 网络层的设计问题

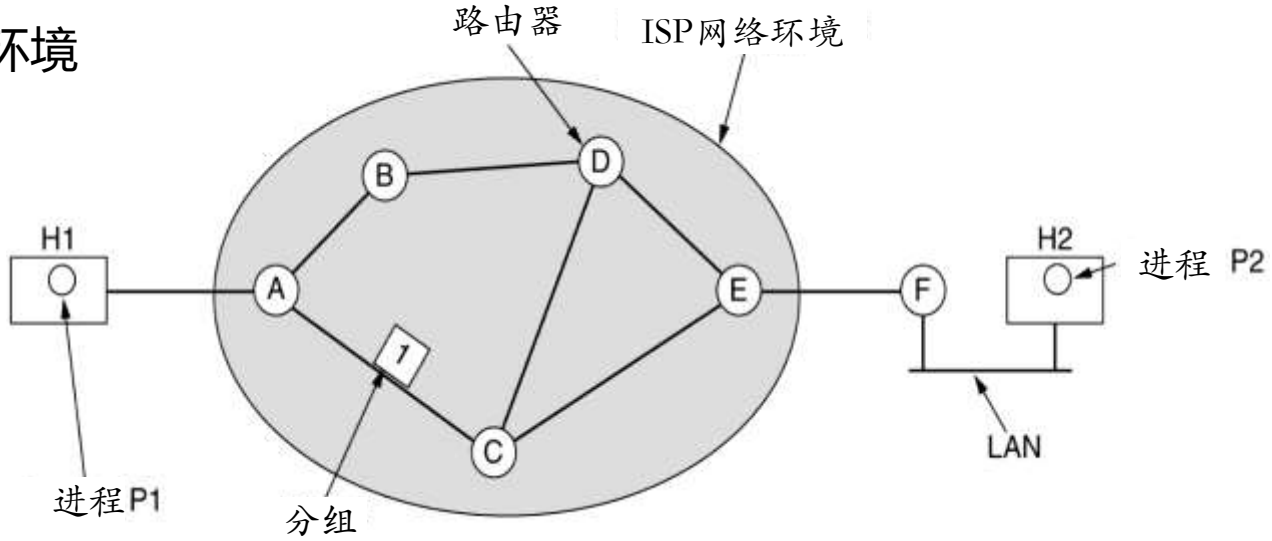
- 网络层问题概述
- 存储转发分组交换
- 为传输层提供的服务
- 无连接服务的实现
- 面向连接服务的实现
- 数据报子网与虚电路子网

网络层问题概述

- 网络层是处理端到端传输的最低层
- 网络层要解决的关键问题是了解通信子网的拓扑结构，选择路由
- 网络层设计的有关问题
  - 为传输层提供服务
  - 传统电信的观点：通信子网应该提供可靠的、面向连接的服务 (如ATM)
  - 因特网的观点：通信子网无论怎么设计都是不可靠的，因此网络层只需提供无连接服务 (如IP)
- 网络层为接在网络上的主机所提供的服务可以有两大类
  - 无连接的网络服务（数据报服务datagram）
  - 面向连接的网络服务（虚电路服务virtual circuit）

存储转发分组交换

- ISO 定义（网络层）
  - 网络层，为一个网络连接的两个传输实体间交换网络服务数据单元，提供功能和规程的方法，使传输实体独立于路由选择和交换的方式
- 网络层协议的环境



为传输层提供的服务

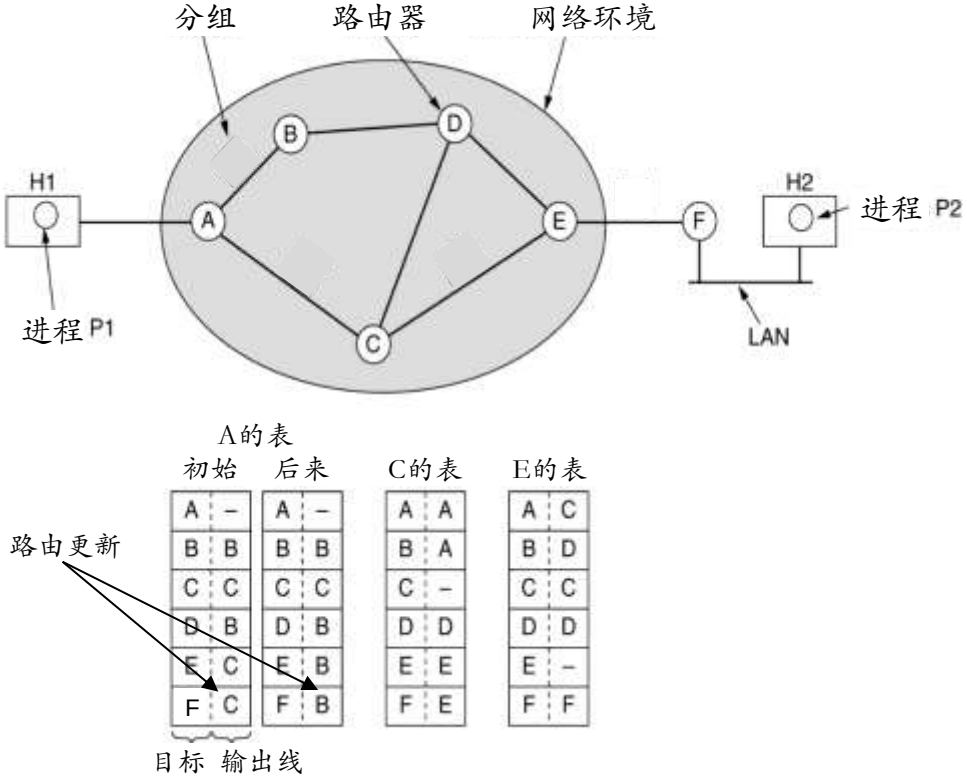
- 服务目标
  - 服务与路由器技术无关
  - 路由器的数量、类型和拓扑结构是隐蔽的
  - 传输层可用的网络地址统一编址，并且跨越LAN和WAN
- 服务规范的两大阵容
  - Internet组织：网络不可靠，主机完成差错控制和流量控制（端到端）
  - 电话公司：面向连接的可靠服务，注重服务质量

IP中的面向连接（MPLS与VLAN）

- 服务方式
  - 面向连接/无连接
  - 可靠/不可靠

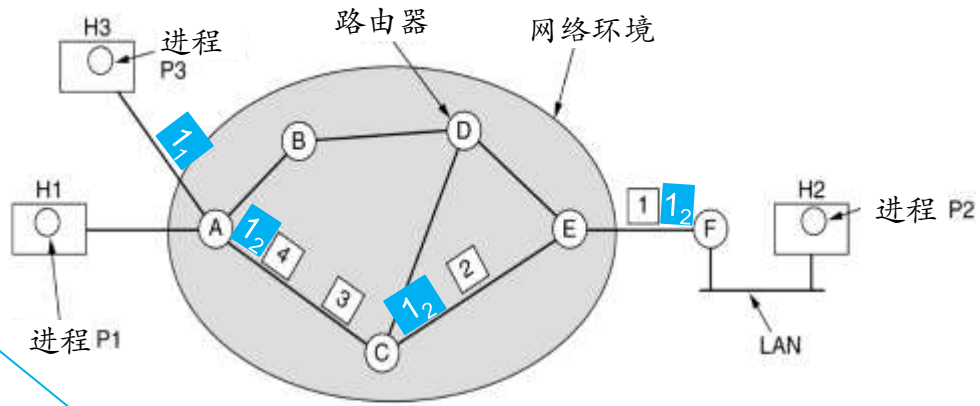
无连接服务的实现

- 数据报网络的路由过程
  - 进程P1发送长消息给进程P2
  - 目标路由器为F
  - 长消息拆分为4个数据包（分组）
  - 路由过程
    - 分组1、2、3使用初始表
    - 分组4使用更新表
  - 路由算法
    - 管理路由表并进行路由选择
- IP协议：无连接网络服务范例



面向连接服务的实现

- 虚电路网络的路由过程
  - 选择一条固定的路径传送消息，该路径信息保存在途经路由器的表中
  - 主机H1(P1)建立与H2(P2)的连接
    - 连接记录在路由器A、C、E的表中
  - 主机H3(P3)建立与H2的连接
    - 路由器A避免冲突，设为C标号2
    - 每个分组均包含虚电路标号
    - 途经路由器时替换为出境标号



- MPLS：多协议标签交换
  - 20位连接标识符（标签）代表同源和目标的电路
  - ISP用来为超大流量建立长期连接

虚电路与数据报网络的比较

- 创建时间(建立)与地址解析（寻址）时间
  - 虚电路需要在建立连接时花费时间，一次建立虚电路号，重复使用
  - 数据报则在每次路由时过程复杂，单独建立，随时调整
- 路由器内存要求的表空间数量
  - 虚电路方式，路由器保存简短虚电路号，需要维护虚电路的状态信息
  - 数据报方式，每个分组都携带完整的目的/源地址，浪费带宽
- 可靠性与拥塞控制
  - 虚电路方式很容易保证服务质量QoS(Quality of Service)，适用于实时操作，但比较脆弱；拥塞控制容易
  - 数据报不太容易保证服务质量，但是对于通信线路的故障，适应性很强；拥塞控制困难

两种服务的思路来源不同

- 虚电路服务的思路来源于传统的电信网
  - 电信网负责保证可靠通信的一切措施
  - 电信网的结点交换机复杂而昂贵
- 数据报服务力求使网络生存性好，并且对网络的控制功能分散
  - 只能要求网络提供尽最大努力的服务
  - 可靠通信由用户终端中的软件（即TCP）来保证

数据报服务和虚电路服务都各有一些优缺点

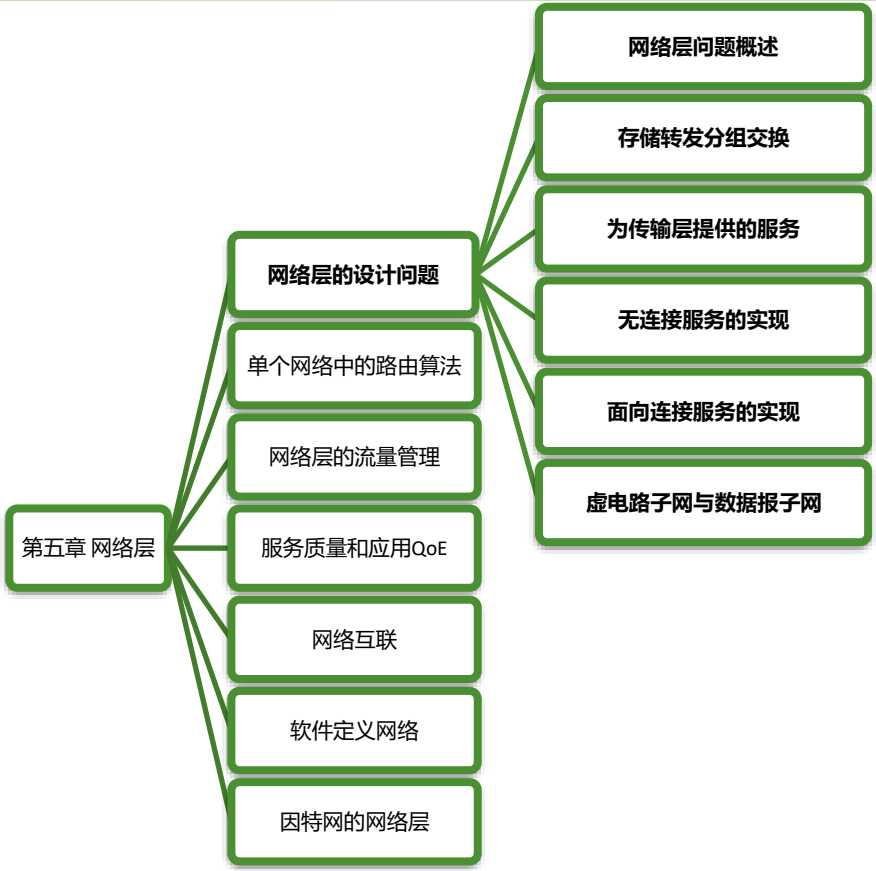
- 网络上传送的报文长度，在很多情况下都很短，用数据报既迅速又经济
- 在使用数据报时，主机承担端到端的差错控制和流量控制
- 数据报服务对军事通信有其特殊的意义；当某个结点发生故障时，后续的分组就可另选路由，因而提高了可靠性
- 在使用数据报时，每个分组必须携带完整的地址信息
- 数据报服务还很适合于将一个分组发送到多个地址(即广播或多播)

- 若用虚电路，为了传送一个分组而建立虚电路和释放虚电路就显得太浪费网络资源了
- 在使用虚电路时，分组按顺序交付，网络可以负责差错控制和流量控制
- 但在使用虚电路时，结点发生故障就必须重新建立另一条虚电路
- 在使用虚电路的情况下，每个分组不需要携带完整的目的地址，而仅需要有个很简单的虚电路号码的标志，这就使分组的控制信息部分的比特数减少，因而减少了额外开销

虚电路与数据报的比较

	虚电路网络	数据报网络
电路建立时间	有(需要)	无(不需要)
寻址	每个分组包含简短VC号	每个分组包含全部源和目的地址
状态信息	每条VC路由器保留状态	不需要
路由方式	建立VC时选择路由，所有分组遵循该路由	每个分组被单独路由
路由器失败影响	经过路由器的VC全部终止	故障时处理的分组丢失
服务质量	若有足够的资源，容易	难
拥塞控制	容易	难

本章导航与要点


第五章 网络层
网络层的设计问题
单个网络中的路由算法
网络层的流量管理
服务质量和应用QoE
网络互联
软件定义网络
因特网的网络层
网络层问题概述
存储转发分组交换
为传输层提供的服务
无连接服务的实现
面向连接服务的实现
虚电路子网与数据报子网

广域网的基本概念

- 当主机之间的距离较远时，例如，相隔几十或几百公里，甚至几千公里，局域网显然就无法完成主机之间的通信任务。这时就需要另一种结构的网络，即广域网
- 关注点
  - 即使是覆盖范围很广的互联网，也不是广域网，因为在这种网络中，不同网络的“互连”才是其最主要的特征
  - 广域网是单个的网络，它使用结点交换机连接各主机而不是用路由器连接各网络
  - 结点交换机在单个网络中转发分组，而路由器在多个网络构成的互联网中转发分组
  - 连接在一个广域网（或一个局域网）上的主机在该网内进行通信时，只需要使用其网络的物理地址即可

广域网中的分组转发机制

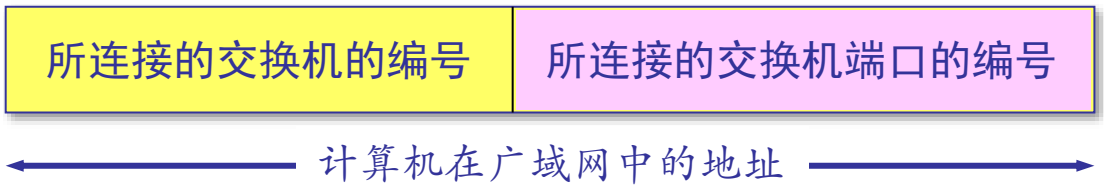
- **转发** (forwarding) 和**路由选择** (routing) 这两个名词的使用在过去有些混乱，现在的文献倾向于将它们区分开来
  - 转发是当交换结点收到分组后，根据其目的地址查找**转发表**(forwarding table)，并找出应从结点的哪一个接口将该分组发送出去
  - 路由选择是构造**路由表**(routing table)的过程
  - 路由表是根据一定的路由选择算法得到的，而转发表又是根据路由表构造出的

转发和路由选择

- 路由选择协议负责搜索分组从某个结点到目的结点的最佳传输路由，以便构造路由表
- 从路由表再构造出转发分组的转发表。分组是通过转发表进行转发的
- 为了使讨论更简单些，可以不严格区分“转发”和“路由选择”，也不一定使用“转发表”这一名词
  - 在转发分组时可以不说“查找转发表”，而说“查找路由表”

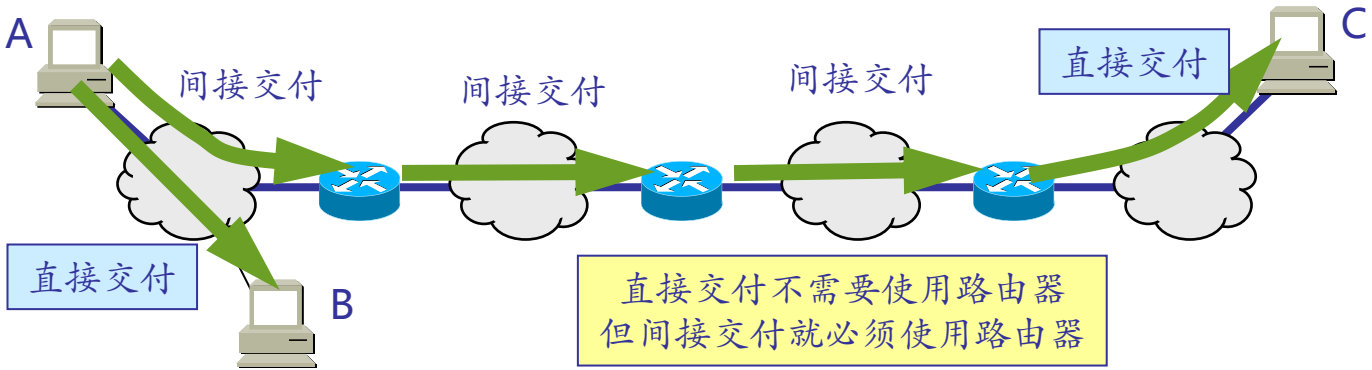
在结点交换机中查找转发表

- 局域网采用了**扁平地址结构**(flat addressing)
  - 对局域网，这种结构非常方便
- 广域网中一般都采用**层次地址结构**(hierarchical addressing)
  - 最简单的层次结构地址举例
    - 用二进制数表示的主机地址划分为前后两部分
    - 前一部分的二进制数，表示该主机所连接的分组交换机的编号
    - 后一部分的二进制数，表示所连接的分组交换机的端口号，或主机的编号

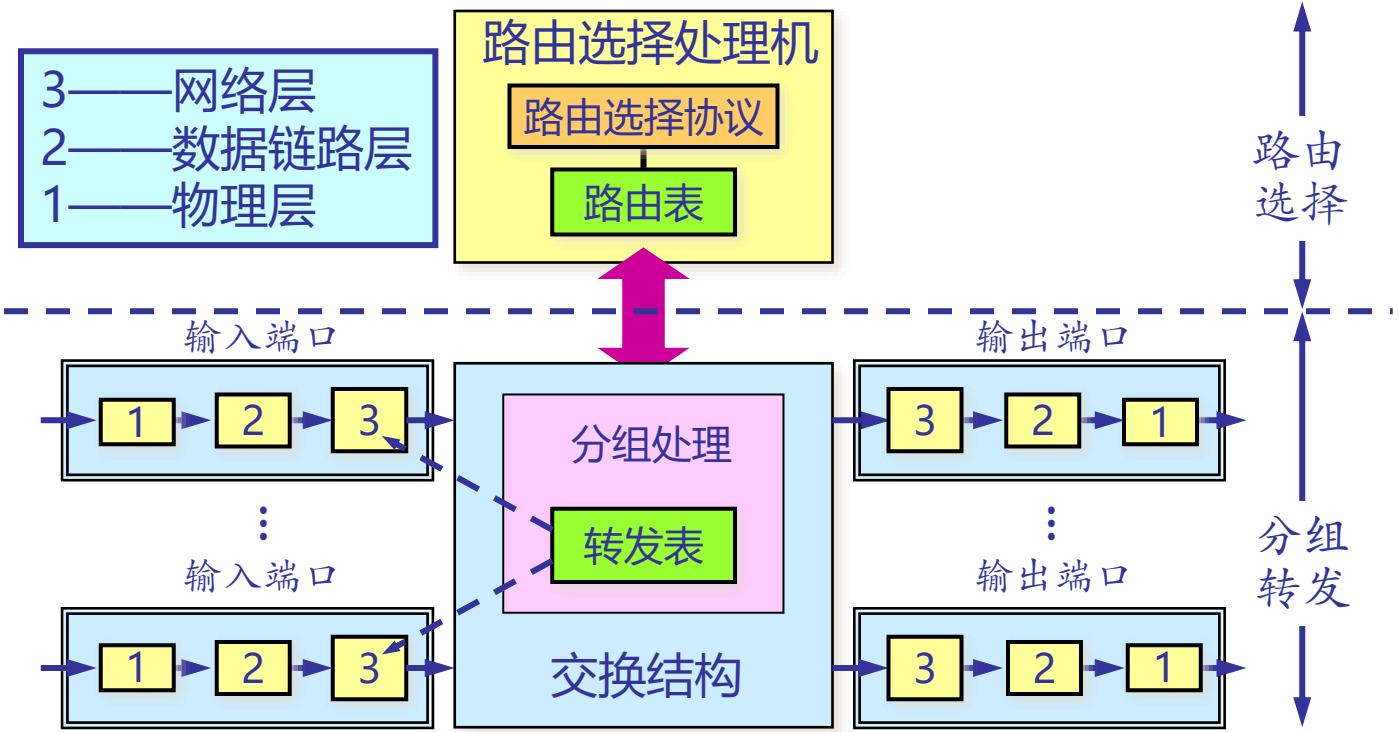


路由器在网际互连中的作用

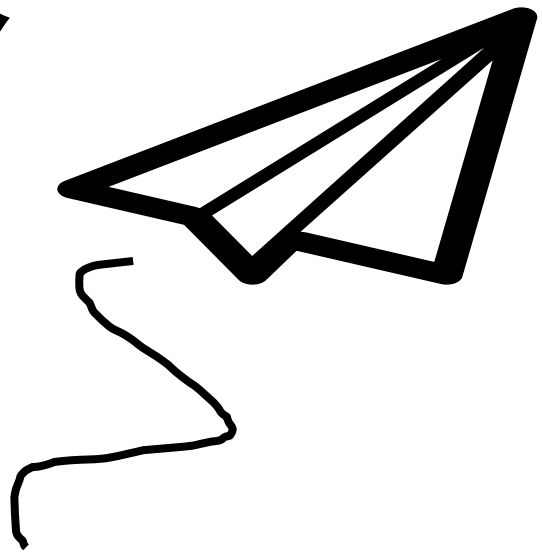
- 路由的构成
  - 当主机A 要向另一主机B 发送分组时，先要检查目的主机B 是否与源主机A 连在同一个网络上
  - 如果是，就将分组**直接交付**给目的主机 B 而不需要通过路由器
  - 但如果目的主机与源主机 A 不是连接在同一个网络上，则应将分组发送给本网络上的某个路由器，由该路由器按照转发表指出的路由将分组转发给下一个路由器；这就叫作**间接交付**



典型的路由器的结构


3——网络层
2——数据链路层
1——物理层
路由选择处理机
路由选择协议
路由表
路由选择
输入端口
输出端口
1
2
3
分组处理
转发表
交换结构
分组转发

本节课程结束



5.2单个网络中的路由算法

- 优化原则
- 最短路径算法
- 泛洪算法
- 基于流量的路由选择算法
- 距离矢量算法
- 链路状态路由算法
- 其他路由问题

优化原则

- 基本原理
- 公平性与有效性
- 最优化原则与汇集树

基本原理

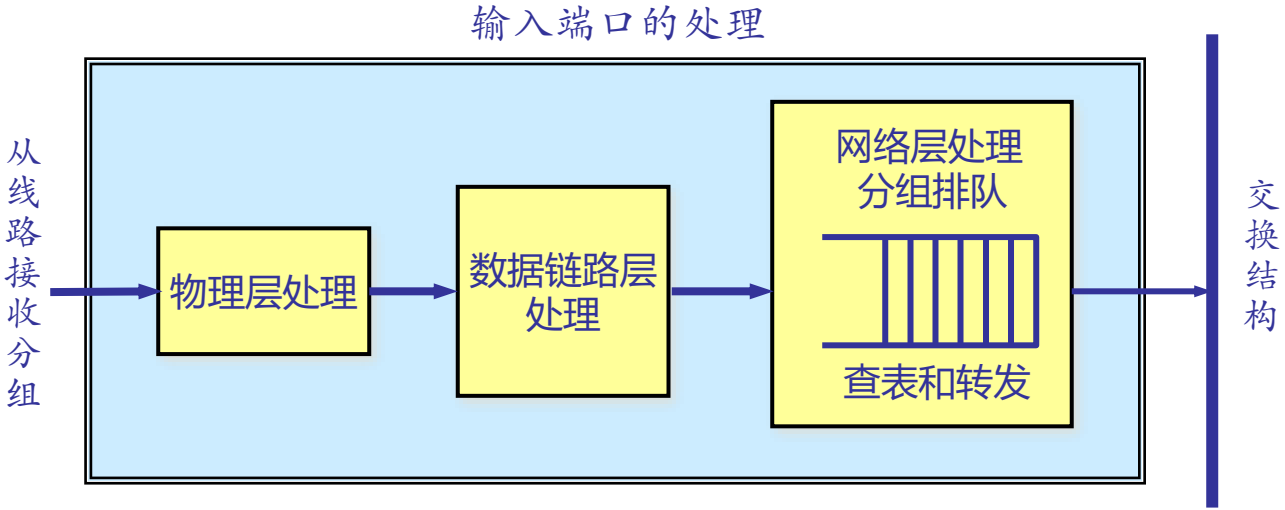
- 路由算法是网络层软件的一部分
  - 采用数据报方式，每个分组都要做路由选择
  - 采用虚电路方式，只需在建立连接时做一次路由选择(又称会话路由)
  - 路由与转发

路由器的“转发”和“路由选择”

- 转发就是路由器根据转发表将用户的 IP 分组从合适的端口转发出去
- 路由选择则是按照分布式算法，根据从各相邻路由器得到的关于网络拓扑的变化情况，动态地改变所选择的路由
- 路由表是根据路由选择算法得出的；而转发表是从路由表得出的
- 在讨论路由选择的原理时，往往不去区分转发表和路由表的区别
- 分组丢弃与输入输出队列
  - 若路由器处理分组的速率赶不上分组进入队列的速率，则队列的存储空间最终必定减少到零，这就使后面再进入队列的分组由于没有存储空间而只能被丢弃
  - 路由器中的输入或输出队列产生溢出是造成分组丢失的重要原因

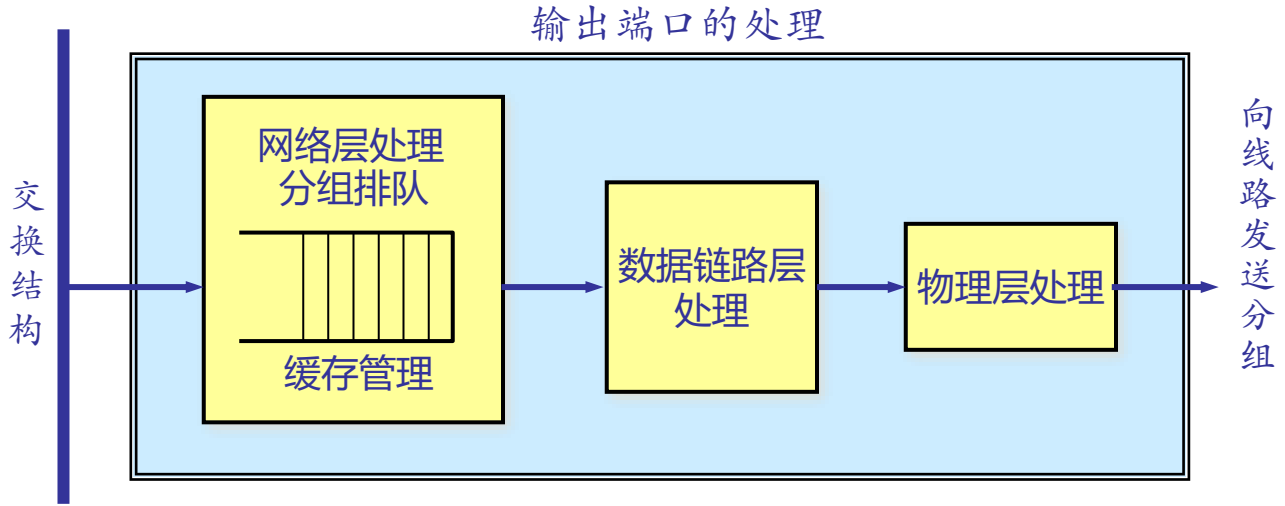
输入队列

- 输入端口对线路上收到的分组的处理
  - 数据链路层剥去帧首部和尾部后，将分组送到网络层的队列中排队等待处理；这会产生一定的时延



输出队列

- 输出端口将交换结构传送来的分组发送到线路
  - 当交换结构传送过来的分组先进行缓存。数据链路层处理模块将分组加上链路层的首部和尾部，交给物理层后发送到外部线路

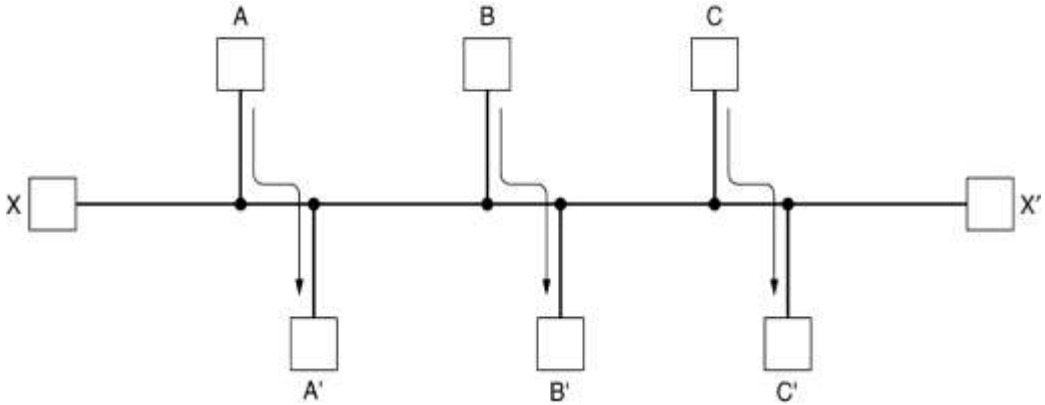


基本原理(2)

- 路由算法应具有的特性
  - 正确性（correctness）
  - 简单性（simplicity）
  - 鲁棒性（robustness）
  - 稳定性（stability）
  - 公平性（fairness）
  - 有效性（optimality）

公平性与有效性

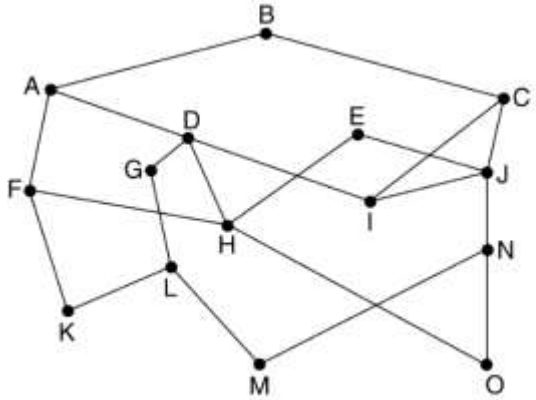
- 公平与最优的冲突



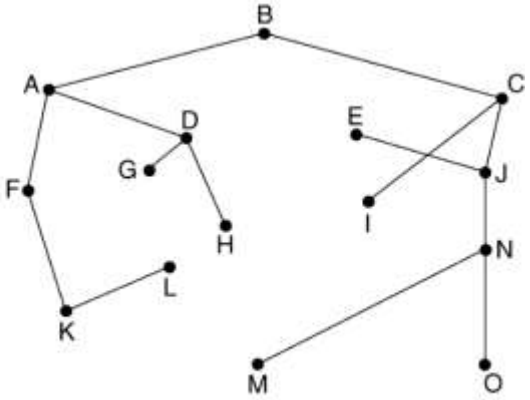
- 路由算法分类
  - 非自适应算法：静态路由算法
  - 自适应算法：动态路由算法

最优化原则与汇集树

- 最优化原则（optimality principle）
  - 若I---J---K为最佳路由，则其中的J---K也是最佳路由
- 汇集树（sink tree）
  - 从所有的源结点到一个给定的目的结点的最优路由的集合，形成了一个以目的结点为根的树，称为汇集树
  - 路由算法的目的是找出并使用汇集树



(a) 一个网络



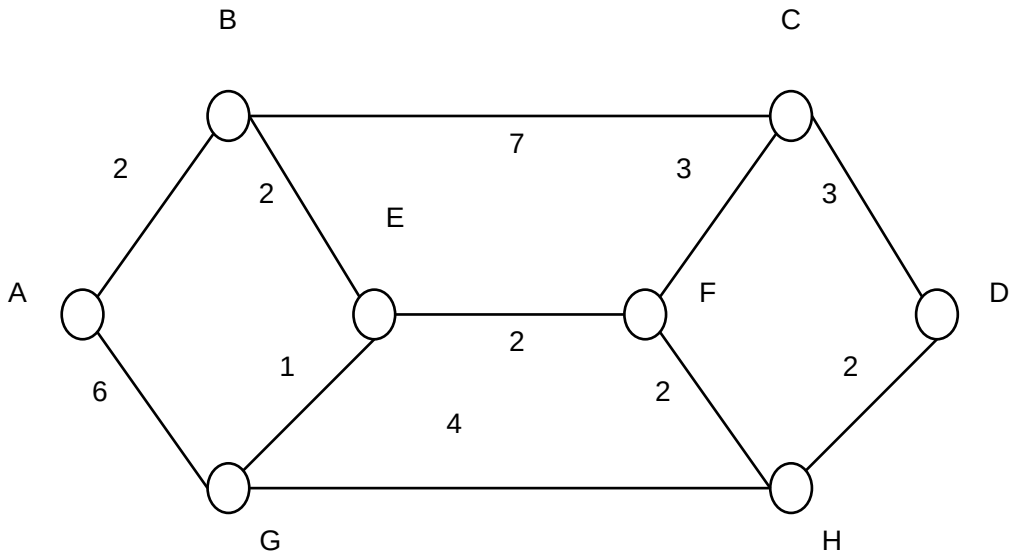
(b) 路由器B的汇集树

最短路径算法

- 基本思想
  - 构建网络的拓扑图，图中的每个结点代表一个路由器，每条弧代表一条通信线路
  - 为了选择两个路由器间的路由，算法在图中找出最短路径
- 测量路径长度的方法
  - 结点数量
  - 地理距离
  - 传输延迟
  - 平均队列长度
  - 距离、信道带宽等参数的加权函数

图的最短路径

- 图中A至D的最短路径

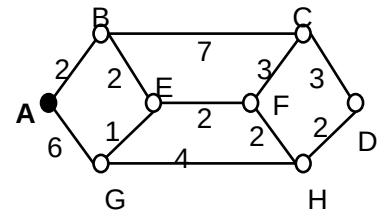


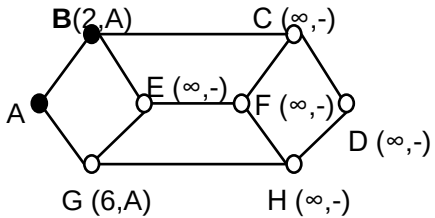
Dijkstra算法

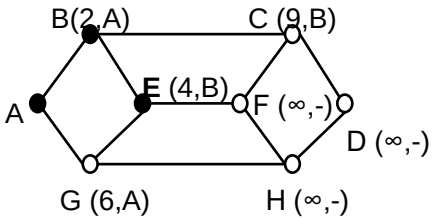
- 算法

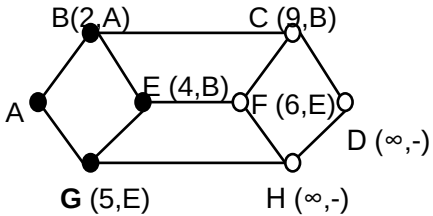
1. 每个结点，用从源结点沿已知最佳路径到本结点的距离来标注，标注分为临时性标注和永久性标注
2. 初始时，所有结点都为临时性标注，标注为无穷大
3. 将源结点标注为0，且为永久性标注，并令其为工作结点
4. 检查与工作结点相邻的临时性结点，若该结点到工作结点的距离与工作结点的标注之和小于该结点的标注，则用新计算得到的和重新标注该结点
5. 在整个图中查找具有最小值的临时性标注结点，将其变为永久性结点，并成为下一轮检查的工作结点
6. 重复第4、5步，直到目的结点成为工作结点

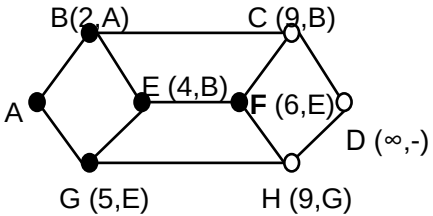
最短路径算法(Dijkstra)图解

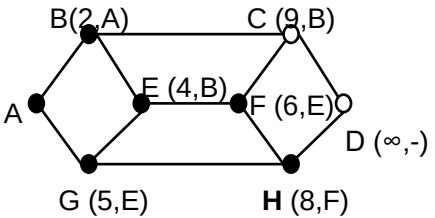


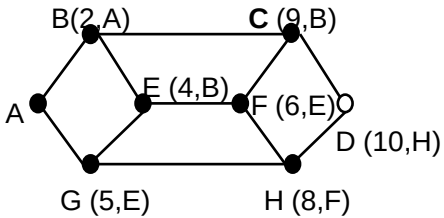


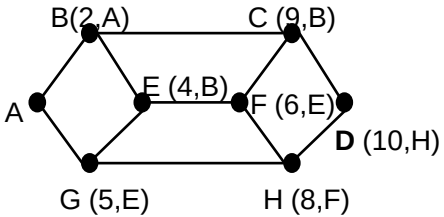


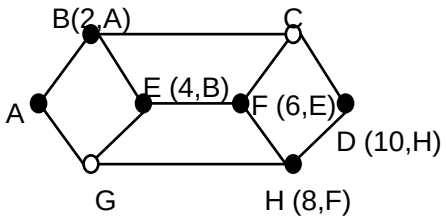








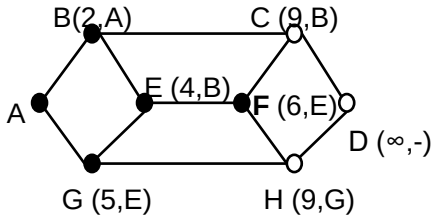




ABEFHD：10

最短路径算法证明

- 证明
  - 假设有比ABEF更短的路径A...XF（F为永久标注）
  - X为永久标注
    - XF已被试探过
  - X为临时标注
    - X标注 >= F标注， A...XF不可能更短
    - X标注 < F标注，则F不会在X之前先成为永久标注



最短路径算法程序

```
#define MAX_NODES 1024                  /* maximum number of nodes */
#define INFINITY 1000000000             /* a number larger than every maximum path */
int n, dist[MAX_NODES][MAX_NODES];/* dist[i][j] is the distance from i to j */

void shortest_path(int s, int t, int path[])
{ struct state {                        /* the path being worked on */
      int predecessor;                  /* previous node */
      int length;                       /* length from source to this node */
      enum {permanent, tentative} label; /* label state */
  } state[MAX_NODES];

  int i, k, min;
  struct state *p;

  for (p = &state[0]; p < &state[n]; p++) {  /* initialize state */
      p->predecessor = -1;
      p->length = INFINITY;
      p->label = tentative;
  }
  state[t].length = 0;  state[t].label = permanent;
  k = t;                                /* k is the initial working node */
```

最短路径算法程序 (2)

```
do {                                    /* Is there a better path from k? */
    for (i = 0; i < n; i++)             /* this graph has n nodes */
        if (dist[k][i] != 0 && state[i].label == tentative) {
            if (state[k].length + dist[k][i] < state[i].length) {
                state[i].predecessor = k;
                state[i].length = state[k].length + dist[k][i];
            }
        }

    /* Find the tentatively labeled node with the smallest label. */
    k = 0; min = INFINITY;
    for (i = 0; i < n; i++)
        if (state[i].label == tentative && state[i].length < min) {
            min = state[i].length;
            k = i;
        }
    state[k].label = permanent;
} while (k != s);

/* Copy the path into the output array. */
i = 0;  k = s;
do {path[i++] = k; k = state[k].predecessor; } while (k >= 0);
}
```

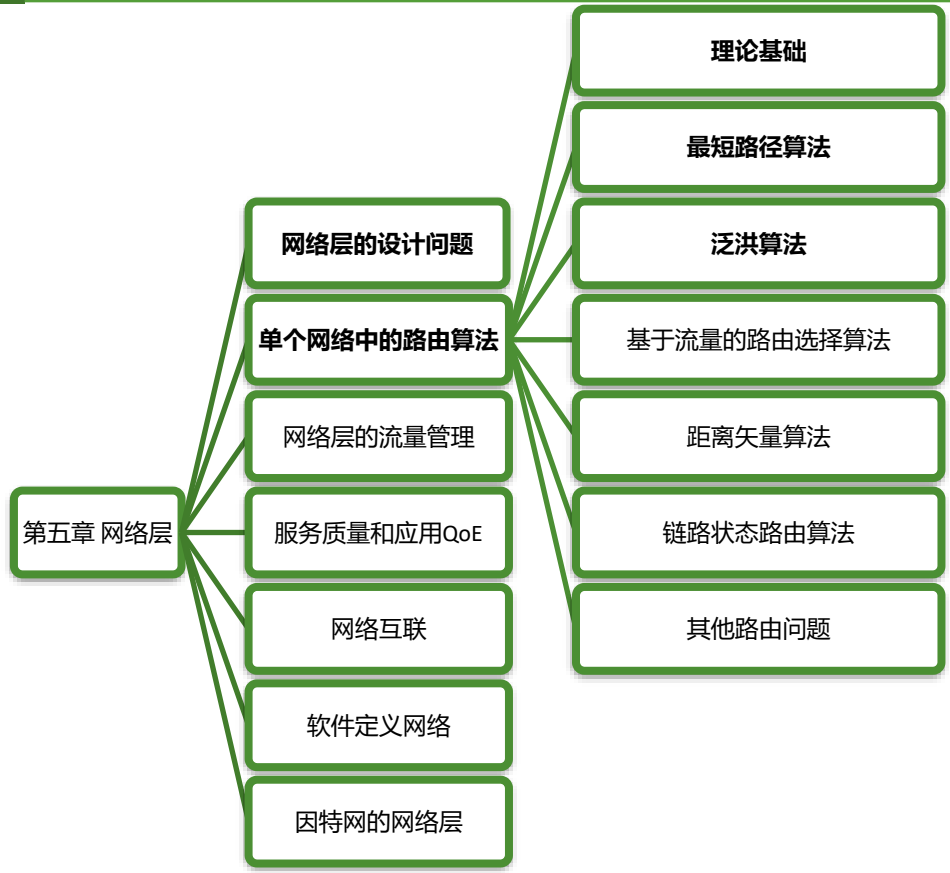
泛洪算法

- 基本思想
  - 向分组到来线路以外的所有线路发送
- 主要问题
  - 洪泛产生大量重复分组
- 解决措施
  - 分组头包含传递站点计数器
    - 如超过网络直径则丢弃
  - 纪录分组扩散路径
    - 记录源路由分组序号，以及最小有效序号k，序号小于k的分组直接丢弃

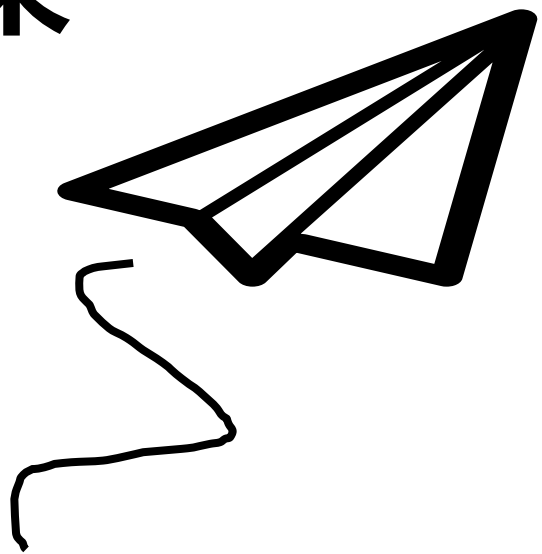
泛洪算法的改进

- 选择性扩散法
  - 选择相临方向线路发送
- 应用情况
  - 路由器和线路的资源过于浪费，实际很少直接采用
  - 具有极好的健壮性，可用于军事应用
  - 作为衡量标准，评价其它路由算法

本章导航与要点


第五章 网络层
网络层的设计问题
单个网络中的路由算法
网络层的流量管理
服务质量和应用QoE
网络互联
软件定义网络
因特网的网络层
理论基础
最短路径算法
泛洪算法
基于流量的路由选择算法
距离矢量算法
链路状态路由算法
其他路由问题

本小节课程结束



基于流量的路由选择

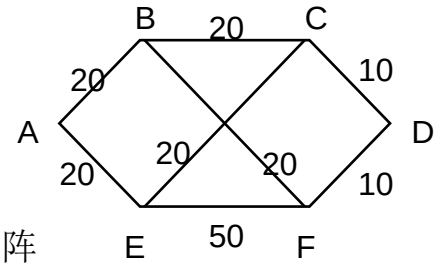
- 基本思想
  - 既考虑拓扑结构，又兼顾网络负荷
  - 前提：每对结点间平均数据流是相对稳定和可预测的
  - 根据网络带宽和平均流量，可得出平均分组延迟，因此路由选择问题归结为，寻找产生网络最小延迟的路由选择算法
  - 提前离线（off-line）计算

网络最小延迟的路由选择算法

- 所需信息
  1. 网络拓扑结构
  2. 通信量矩阵Fij
  3. 线路容量矩阵Cij
  4. 路由选择算法
- 最小延迟的路由选择算法
  - 平均延时 $T = 1/(\mu C-\lambda)$
  - 其中，容量 C (b/s)，数据到达率 λ (帧/秒)，平均帧长 1/ μ (比特/帧)

最小延迟路由选择算法计算

1. 拓扑结构



2. 通信量矩阵

Fij	A	B	C	D	E	F
A		9 AB	4 ABC	1 ABFD	7 AE	4 AEF
B	9 BA		8 BC	3 BFD	2 BFE	4 BF
C	4 CBA	8 CB		3 CD	3 CE	2 CEF
D	1 DFBA	3 DFB	3 DC		3 DCE	4 DF
E	7 EA	2 EFB	3 EC	3 ECD		5 EF
F	4 FEA	4 FB	2 FEC	4 FD	5 FE	

3.线路容量矩阵

1/μ= 800

i	线路	λi (分组/s)	Ci (kb/s)	μCi (分组/s)	T (ms)	权值 Q
1	AB	14	20	25	91	0.171
2	BC	12	20	25	77	0.146
3	CD	6	10	12.5	154	0.073
4	AE	11	20	25	71	0.134
5	EF	13	50	62.5	20	0.159
6	FD	8	10	12.5	222	0.098
7	BF	10	20	25	67	0.122
8	EC	8	20	25	59	0.098

平均延迟时间

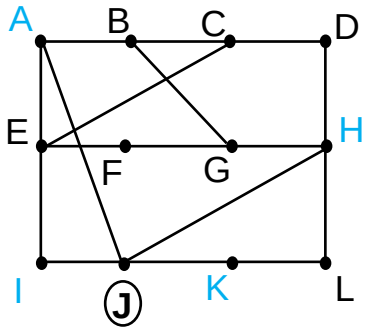
=T1*Q1+…+Ti*Qi+…T8*Q8 = 86ms

距离矢量算法

- 最初用于ARPANET，被RIP协议采用
- 基本思想：每个路由器维护一张表，表中给出了到每个目的地的已知最佳距离和线路，并通过与**相邻路由器**交换距离信息来更新表
  - 索引表：以网络中其它路由器为表的索引，表项包括两部分：到达目的结点的最佳输出线路，以及到达目的结点所需时间或距离
  - 交换信息：每隔一段时间，路由器向所有邻居结点发送它到每个目的结点的距离表，同时它也接收每个邻居结点发来的距离表
  - 计算更新：邻居结点X发来的表中，X到路由器i的距离为X_i，本路由器到X的距离为m，则路由器经过X到i的距离为$X_i + m$。根据不同邻居发来的信息，计算$X_i + m$，并取最小值，更新本路由器的路由表（注意：本路由器中的老路由表在计算中不被使用）

距离矢量算法图解

- 例题



计算J到G的新路由：

	A	I	H	K
到G的延迟	18	31	6	31
到J的延迟	8	10	12	6
J→G	26	41	18	37

G	18	H

延时　线路

↓	A	I	H	K
A	0	24	20	21
B	12	36	31	28
C	25	18	19	36
D	40	27	8	24
E	14	7	30	22
F	23	20	19	40
G	18	31	6	31
H	17	20	0	19
I	21	0	14	22
J	9	11	7	10
K	24	22	22	0
L	29	33	9	9
到J的延迟：	8	10	12	6

J从四个邻居收到的向量

无穷计算

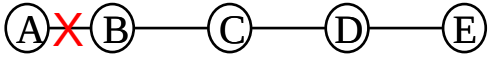
- 算法缺陷

A上网的消息传递

交换次数				
0	∞	∞	∞	∞
1	1	∞	∞	∞
2	1	2	∞	∞
3	1	2	3	∞
4	1	2	3	4

稳定性好（迅速收敛）

A下网的消息传递



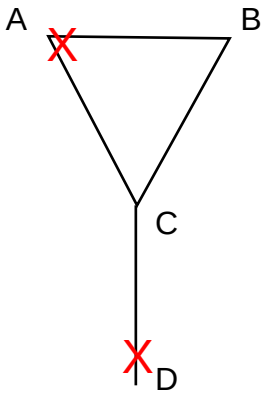
交换次数				
0	1	2	3	4
1	3	2	3	4
2	3	4	3	4
3	5	4	5	4
4	5	6	5	6
5	7	6	7	6
…				
∞	∞	∞	∞	∞

稳定性差（反应异常迟缓）

水平分裂算法

- 到X的距离不向X的邻居报告
  - A下网消息传递

交换次数	B	C	D
0	1	1	2
1	∞	∞	2
2	∞	∞	∞
3	∞	∞	∞



  - D下网消息传递

    不成功

交换次数	A	B	C
0	2	2	1
1	2	2	∞
2	3	3	∞
3	4	4	∞

链路状态路由算法

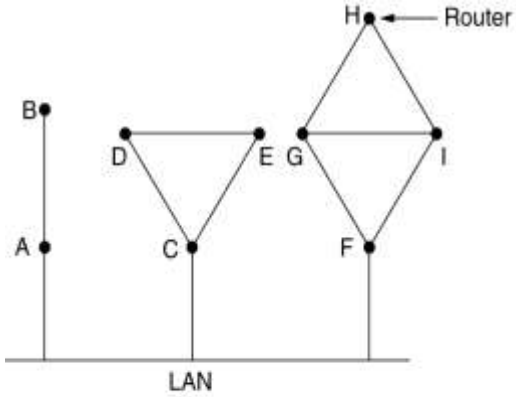
- 距离向量路由算法的主要问题
  - 选择路由时，没有考虑线路带宽
  - 路由收敛速度慢
- 链路状态路由算法思路

  已应用于OSPF、IS-IS

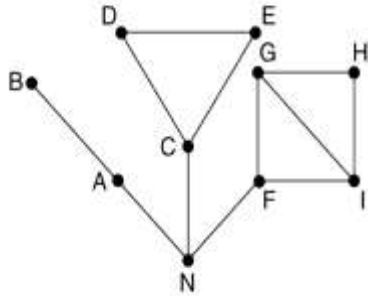
  1. 发现邻居节点，了解其网络地址
  2. 测量线路开销，到邻居节点的距离或成本度量值
  3. 组装链路状态分组
  4. 发送链路状态分组，接收其他所有链路状态分组
  5. 计算最短路径

发现邻居节点

- 路由器启动后，通过发送HELLO分组发现邻居结点
  - 邻居节点收到HELLO分组后，返回应答说明自己是谁
- 两个或多个路由器连在一个LAN时，引入人工结点
  - 图中，如果引入AC、AF、CF等两两互联，会增加拓扑复杂性



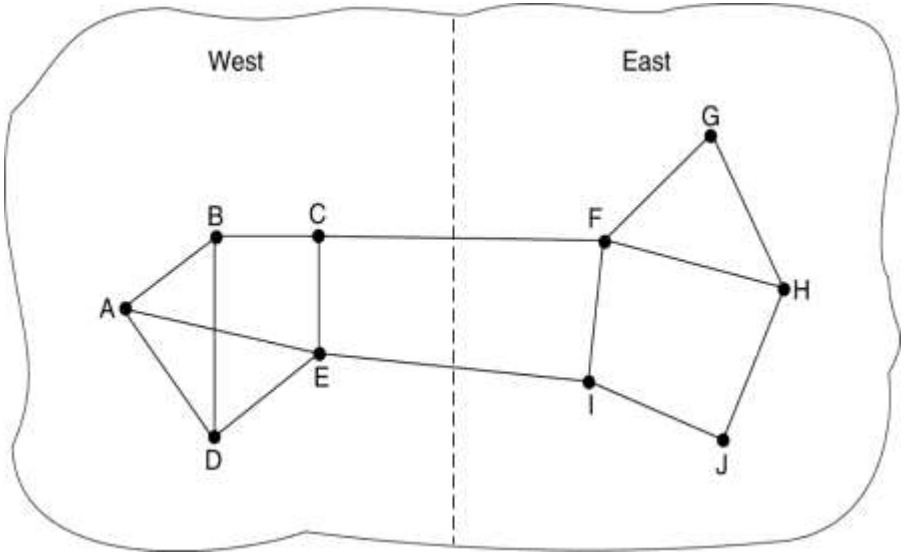
(a) 包含一个广播LAN的网络



(b) 网络(a)的模型图

选定LAN上的一个**指定路由器**，作为N来运行路由协议

设置链路成本

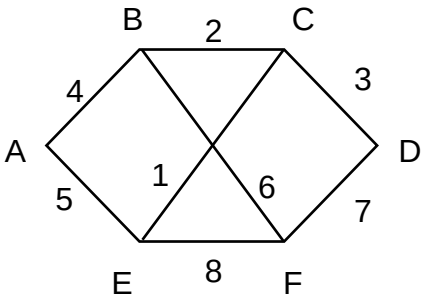


- 需要每条链路的距离或成本度量
  - 到邻居的成本取带宽值
  - 成本与链路带宽成反比

    如1G=1，100M=10
- 若地理分散，链路延迟作为成本
  - 线路延迟估算：往返时间除2
  - 一种直接的方法是：发送一个要对方立即响应的ECHO分组，来回时间除以2即为延迟

构造链路状态分组

- 分组以发送方的标识符开头，后面是序号、年龄和一个邻居结点列表
- 列表中对应每个邻居结点，都有发送方到它们的延迟或开销
- 链路状态分组定期创建，或发生重大事件时创建
  - 周期性创建
  - 事件驱动



链 路 状 态 分 组

A	
序号	
年龄	
B	4
E	5

B	
序号	
年龄	
A	4
C	2
F	6

C	
序号	
年龄	
B	2
D	3
E	1

D	
序号	
年龄	
C	3
F	7

E	
序号	
年龄	
A	5
C	1
F	8

F	
序号	
年龄	
B	6
D	7
E	8

分发链路状态分组

- 泛洪法发布
  - 基本思想
    - 洪泛链路状态分组，为控制洪泛，每个分组包含一个序号，每次发送新分组时，序号加1
    - 路由器记录信息对：<源路由器，序号>
    - 当一个链路状态分组到达时，检查序号，重复分组被丢弃；若序号比路由器记录中的最新序号小，则认为过时，也将丢弃；如果是新序号，则分发，并记录该序号
  - 问题
    - 序号绕回，会产生混淆
    - 路由器崩溃，将丢失记录的序号
    - 序号出错

      0000 0000 0000 0000 0000 0000 0000 0100

      0000 0000 0000 0001 0000 0000 0000 0100
  - 解决办法
    - 使用32位序号；2^32 > 137x365x24x60x60
    - 在每一分组的序号之后，增加年龄字段

      每秒年龄减1，减到0则丢弃（将混乱时间压缩到最短）

      每个路由器也递减年龄字段，保证分组不会无限制生存

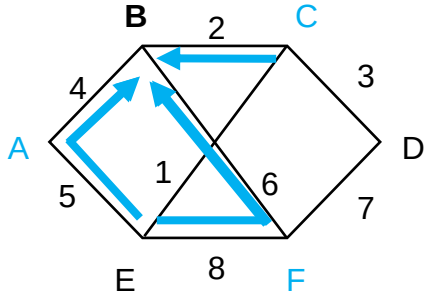
分发链路状态分组(2)

- 算法的改进
  - 收到分组后保留一段时间，以便同源分组再次收到，一并处理
  - 所有链路状态分组都确认，防止线路故障导致出错
  - B的分组缓冲区示例
    - A和F各发来一个链路状态分组
    - E的同一链路状态分组收到两次，一个经过EAB

      另一个经过EFB
    - C发来一个链路状态分组
    - 从D发来的链路状态分组有两个

      一个经过DCB，<D,20,59>

      另一个经过DFB，<D,21,59>



B的分组缓冲区

源	序号	年龄	发送标志 A	发送标志 C	发送标志 F	ACK标志 A	ACK标志 C	ACK标志 F	数据
A	21	60	0	1	1	1	0	0	
F	21	60	1	1	0	0	0	1	
E	21	59	0	1	0	1	0	1	
C	20	60	1	0	1	0	1	0	
D	21	59	1	1	0	0	0	1	

计算新路由

- 路由器积累全部链路状态分组
- 可构造完整网络拓扑图
- 运行Dijkstra算法，计算新路由
- 资源占用
  - n个路由器，k个邻居
    - 距离矢量算法：需要k+1个n表项(<延迟，出境线路>)的路由表，所需内存与 2kn 成正比
    - 链路状态路由：除了需要存储n(链路状态分组)*k(邻居)*2(二元组)以外，还需构造网络拓扑图，并运行Dijkstra算法，所需内存与计算资源都超过距离矢量算法
- 潜在风险：个别路由器失败的影响

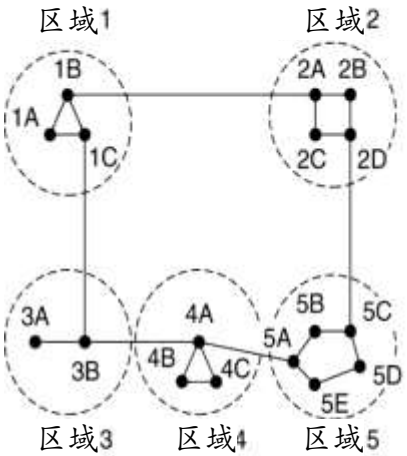
其他路由问题

- 网络内部的层次路由算法
- 广播路由算法
- 多播路由算法
- 任播路由算法

层次路由

- 网络规模增长带来的问题
  - 路由器中的路由表增大
  - 路由器为选择路由而占用的内存、CPU时间，以及占用的网络带宽增大
- 分层路由
  - 分而治之的思想
  - 将路由器划分为域（regions）、簇（clusters）、区（zones）和组（groups）…
  - 下页的示例中，路由表由17项减为7项
- 分层路由带来的问题
  - 路由表中的路由不一定是最优路由

层次路由示例



(a)

1A的完整路由表

目标	线路	跳数
1A	–	–
1B	1B	1
1C	1C	1
2A	1B	2
2B	1B	3
2C	1B	3
2D	1B	4
3A	1C	3
3B	1C	2
4A	1C	3
4B	1C	4
4C	1C	4
5A	1C	4
5B	1C	5
5C	1B	5
5D	1C	6
5E	1C	5

(b)

1A的层次路由表

目标	线路	跳数
1A	–	–
1B	1B	1
1C	1C	1
2	1B	2
3	1C	2
4	1C	3
5	1C	4

(c)

分级

- 域、簇、区、组
  - 例：

路由器数	级数	簇	区	组	表项数
720	1			720	720
	2		24	30	53
	3	8	9	10	25

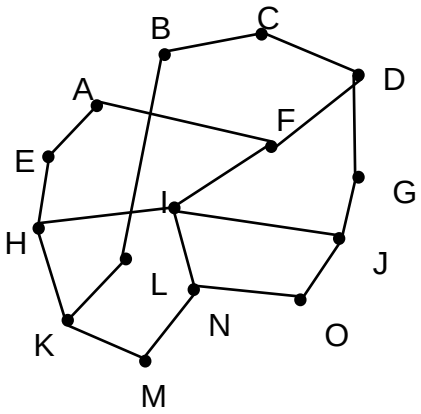
- 级数
  - N个路由器最优级数为：ln N
  - 每个路由器表项数为：e ln N

广播路由

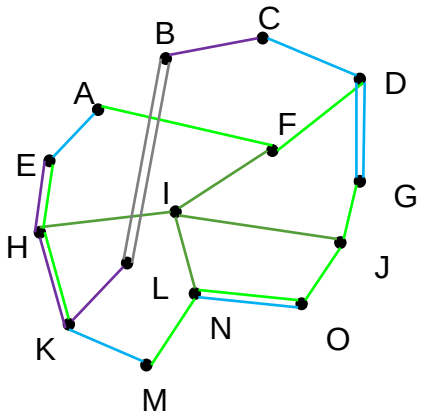
- 广播
  - 将分组发往所有目的地
- 广播路由选择算法
  1. 重复一 一发送
  2. 多目标路由选择
     - 分组含有按发送线路分组的目的地清单
  3. 泛洪法
  4. 逆向路径转发
     - 广播分组来自发往广播源的线路，是则转发，否则丢弃
  5. 利用源路由器的生成树
     - 生成树：包括网络的所有路由器，但不包含回路

逆向路径转发

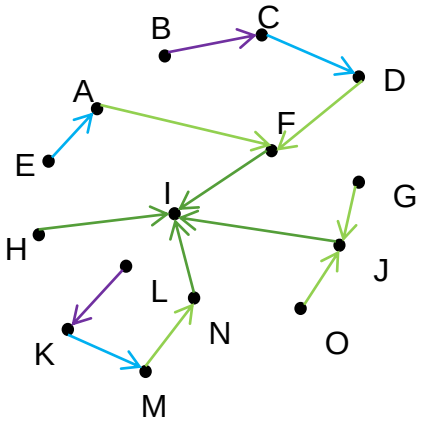
通信网络



转发路径



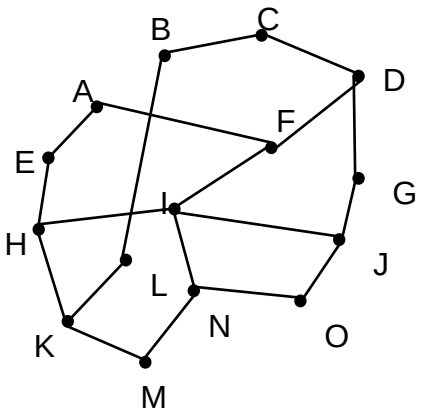
I的汇集树



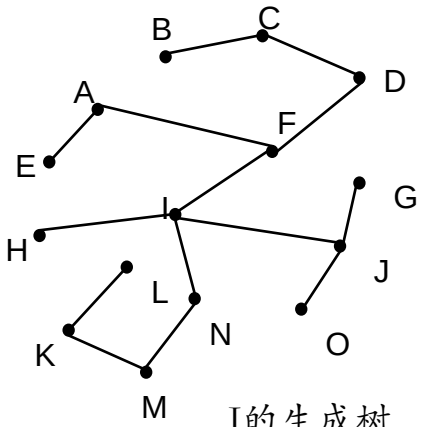
路由器I	第1跳(4)	第2跳(12)	第3跳(18)	第4跳(22)	第5跳(24)
到达节点	F,J,N,H	A,D; G,O; O,M; K,E	E; C,G; D; N; K	H;B;L;H	L;B
转发节点	F,J,N,H	A,D,G,O,M	E,C,K	B,L	
丢弃节点		O,E,K	G,D,N	H,H	L,B

逆向路径转发改进

- 只发生成树
  - 转发的分组数/跳数，由24/5变为14/4

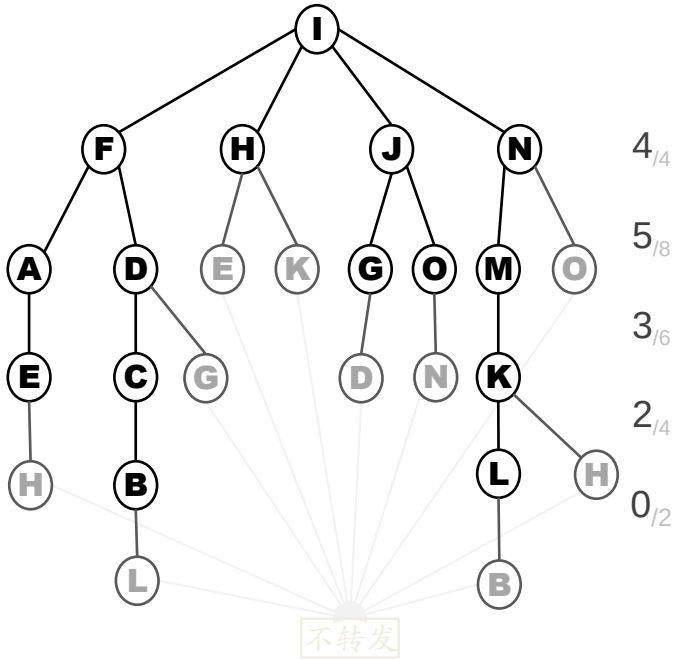


通信网络



I的生成树

逆向路径转发构成的树(14/24)



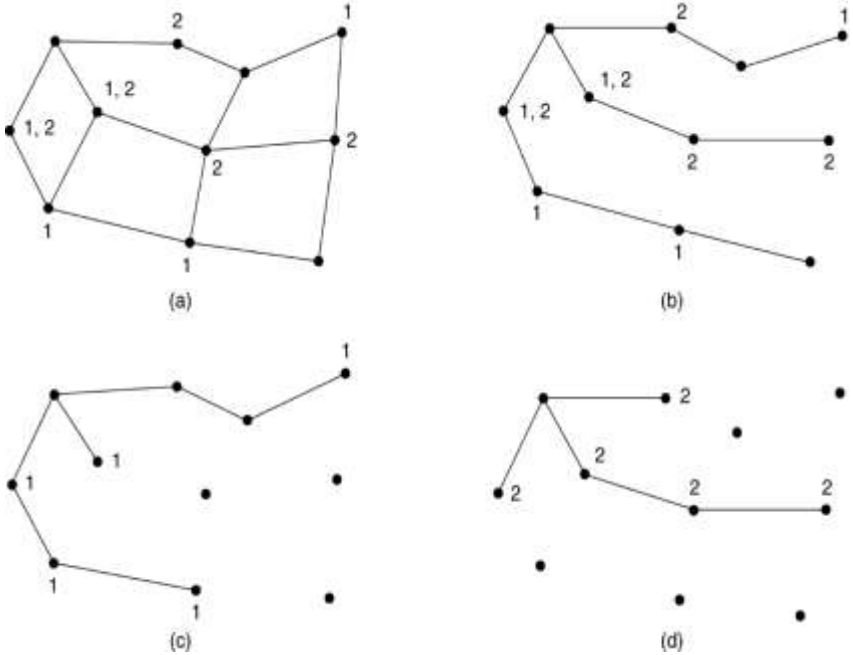
多播路由

- 多播：
  - 小组中广播，密度大则广播
- 网络实例
  - b：汇集树(10)；c、d：多播树
- 算法
  - 修剪生成树

    链路状态路由直接采用汇集树裁剪

    距离矢量算法在逆向路径转发中发送PRUNE

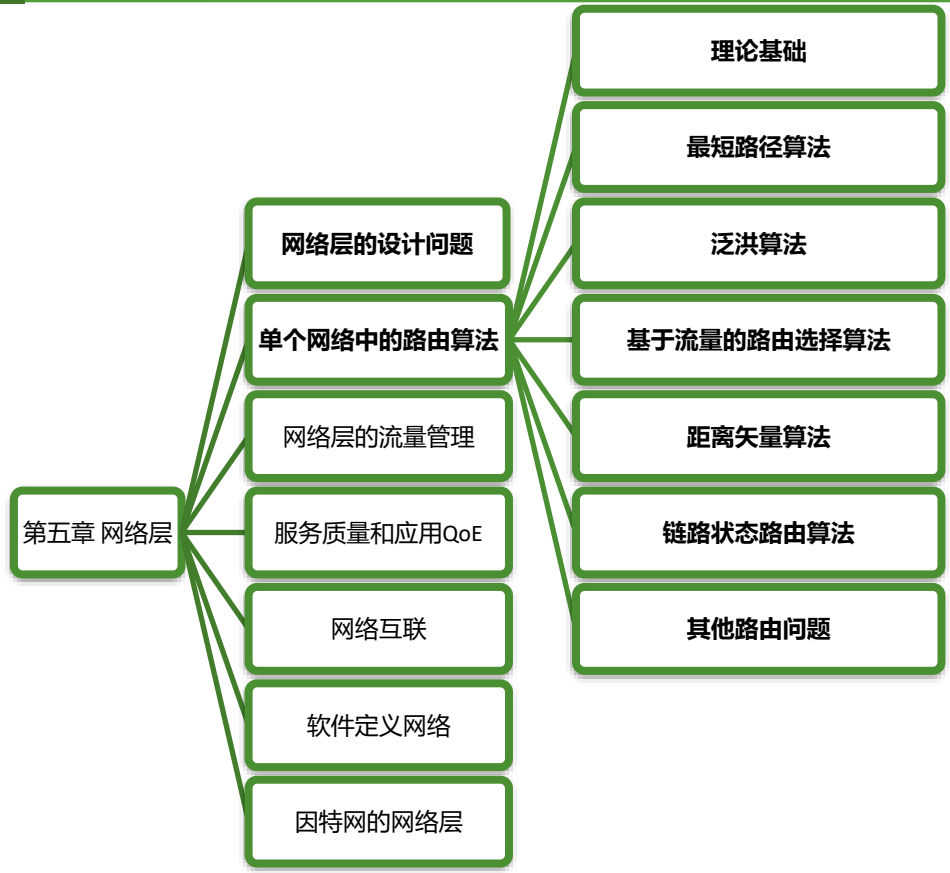
    网络有n个组，每组平均m个成员，共存n*m棵树
  - 核心基本树：每个组只计算一棵生成树，根成员处于组的中心部位



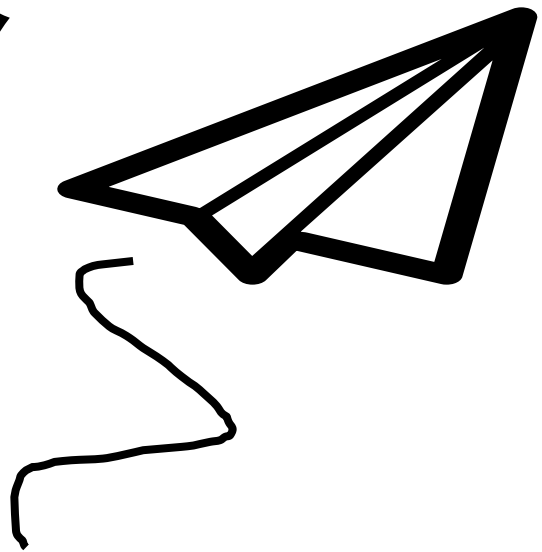
任播路由

- 数据分组传递模型
  - 单播、广播、多播
- 任播
  - 数据分组传递给组成员，重点是收到分组，而不是谁发送分组
  - 报时或内容分发，多信息源
- 任播路由
  - 利用普通的距离矢量和链路状态路由

本章导航与要点


第五章 网络层
网络层的设计问题
单个网络中的路由算法
网络层的流量管理
服务质量和应用QoE
网络互联
软件定义网络
因特网的网络层
理论基础
最短路径算法
泛洪算法
基于流量的路由选择算法
距离矢量算法
链路状态路由算法
其他路由问题

本节课程结束

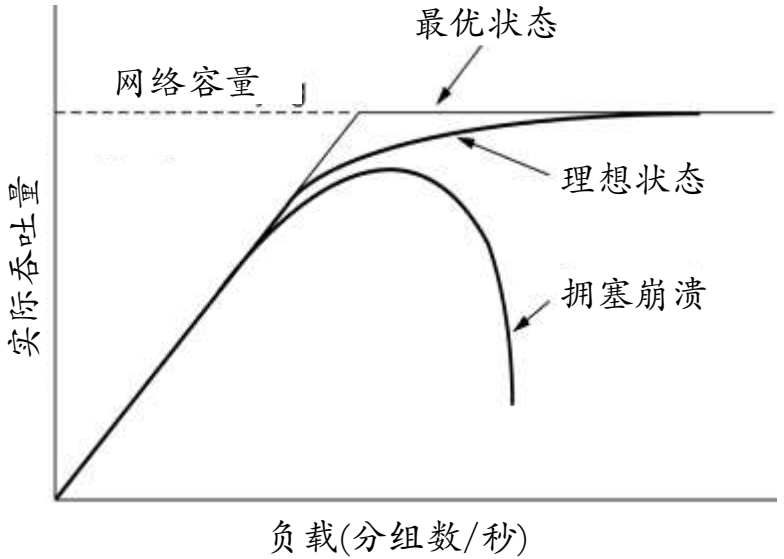


5.3 网络层的流量控制

- 流量管理的必要性：拥塞
- 流量管理方法
  - 流量感知路由
  - 准入控制
  - 负载脱落
  - 流量整形
  - 主动队列管理
  - 抑制分组

流量管理的必要性：拥塞

- 拥塞
  - 网络中有太多的分组时，延迟和丢失导致性能降低，这种情况称为**拥塞**
- 拥塞产生的原因
  - 多个输入对应一个输出，队列缓存不足
  - 低于线速率的路由器
  - 低带宽线路
- 解决办法
  - 针对某个因素的解决方案，只能缓解局部区域，甚至可能仅仅是转移了影响性能的瓶颈
  - 需要全面考虑各个因素，最后的办法只能卸下负载或建立更快的网络



拥塞与流量控制

- 拥塞崩溃
  - 路由器内存：队列溢出与无限内存
  - 网络瓶颈：线路带宽与低效路由器
- 拥塞与流量控制
  - 拥塞控制（congestion control）
    - 拥塞控制需要确保网络能够承载用户提交的通信量，是一个全局性问题，涉及主机、路由器等很多因素
  - 流量控制（flow control）
    - 流量控制与点到点的通信量有关，主要解决快速发送方与慢速接收方的问题，是局部问题，一般都是基于反馈进行控制的

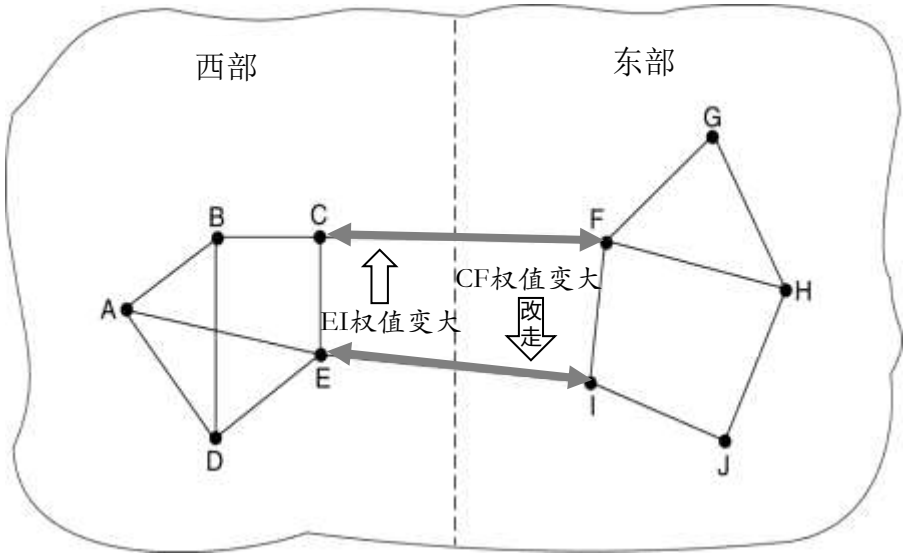
流量管理方法

- 拥塞控制方法与时间尺度

- 控制办法
  - 动态增加资源
  - 定制路由，充分利用；流量感知路由
  - 准入控制
  - 闭环控制：判断、调节、处理、补救（负载脱落）

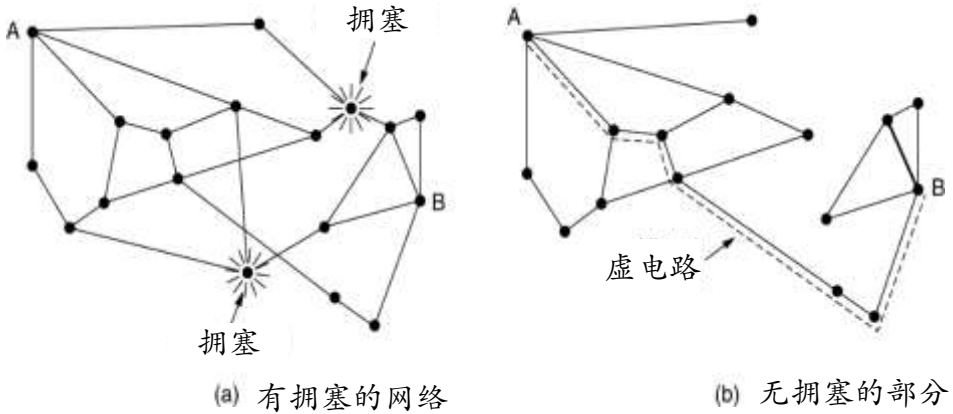
流量感知路由

- 路由算法改进
  - 原来的路由算法忽略负载，只考虑带宽和传输延迟
  - 新路由算法既能适应拓扑结构变化，又能适应负载变化
- 简单的链路权重函数
  - 缺陷：路由摇摆动荡(右图)
- 考虑负载的改进
  - 多路径路由
  - 流量缓慢迁移



准入控制

- 许可控制
  - 拥塞后不再建立任何虚电路
- 变通
  - 新的虚电路绕过有问题的区域
  - 在删除拥塞后的网络上建立虚电路
- 资源预留
  - 建立虚电路时，网络与主机达成协议，并根据协议在虚电路上为此连接预留资源
- 流量特性
  - 约束瞬间突发流量与平均速率，引入漏桶与令牌桶



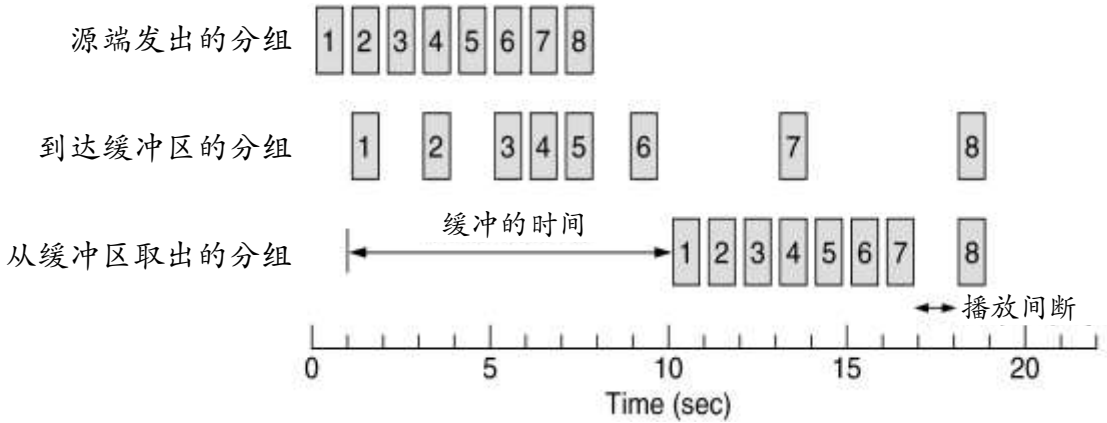
(a) 有拥塞的网络　　(b) 无拥塞的部分

负载脱落

- 载荷脱落
  - 葡萄酒策略：文件传输
  - 牛奶策略：多媒体
- 优先级
  - 价格：低优先级便宜
  - 通信量整形：空闲时低优先级先发送
  - 虚电路带宽：可超过虚电路建立时确定的极限值，但标为低优先级
- 随机早期检测
  - 拥塞之前处理更有效，隐形通知
  - 队列超过阀值，随机丢分组

流量整形

- 基本思想
  - 造成拥塞的主要原因是网络流量通常是突发性的
  - 强迫分组以一种可预测的速率发送
  - 在ATM网中广泛使用
- 缓存平滑输出流



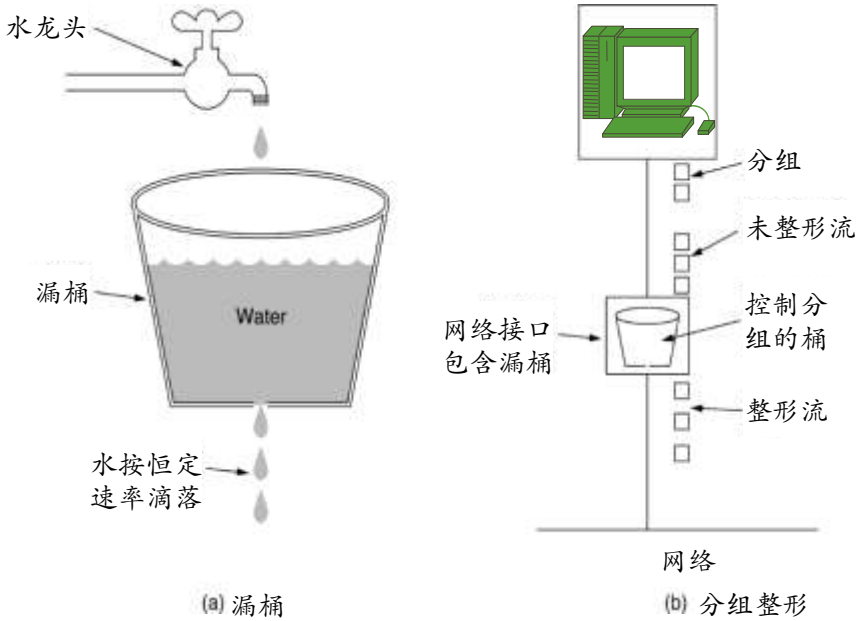
漏桶算法

- 漏桶算法（The Leaky Bucket Algorithm）
  - 将用户发出的不平滑的数据分组流转变成网络中平滑的数据分组流
  - 可用于固定分组长的协议，如ATM；

    也可用于可变分组长的协议，如IP，

    使用字节计数；
  - 无论负载突发性如何，漏桶算法强迫

    输出按平均速率进行，不灵活。



令牌桶

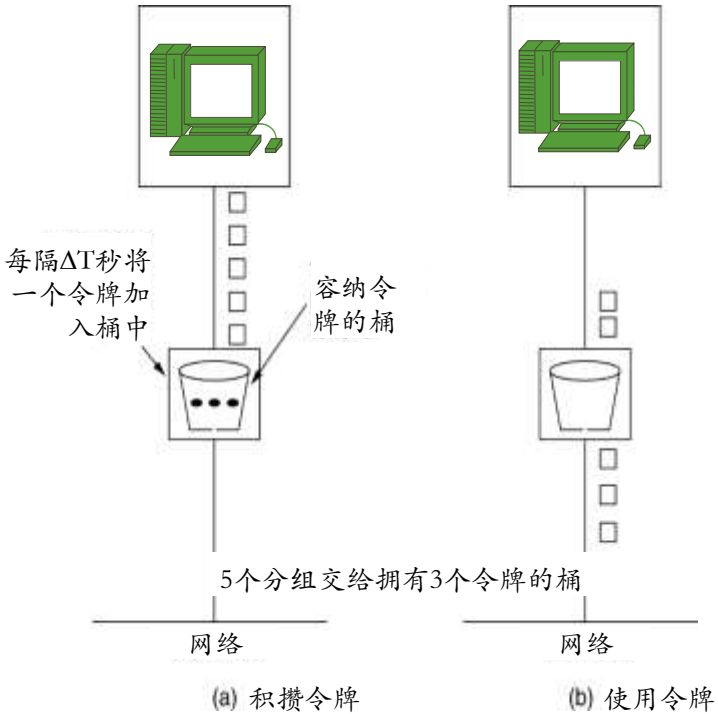
- 令牌桶算法（The Token Bucket Algorithm）
  - 漏桶算法不够灵活，因此加入令牌机制
  - 基本思想：漏桶存放令牌，每 ΔT 秒产生一个令牌，令牌累积到漏桶上界时就不再增加；分组传输之前必须获得一个令牌，传输之后删除该令牌
- 令牌桶算法与漏桶算法的区别
  - 流量整形策略不同
    - 令牌桶算法允许空闲主机积累发送权 (最大为桶的大小)，以便以后发送大的突发数据
    - 漏桶算法不允许
  - 令牌桶中存放的是令牌，桶满了丢弃令牌，不丢弃数据包；漏桶中存放的是数据包，桶满了丢弃数据包

令牌桶图示

- 令牌桶算法
  - 参数
    - 突发时间：S 秒
    - 令牌桶容量：B 字节
    - 令牌到达速率：R 字节/秒
    - 最大输出速率：M 字节/秒
  - 最大速率发送突发持续时间
    - 突发S秒能够发送的数据总量

      $$B + RS = MS$$

    - 因此　　$S = B / (M - R)$



漏桶与令牌桶示例

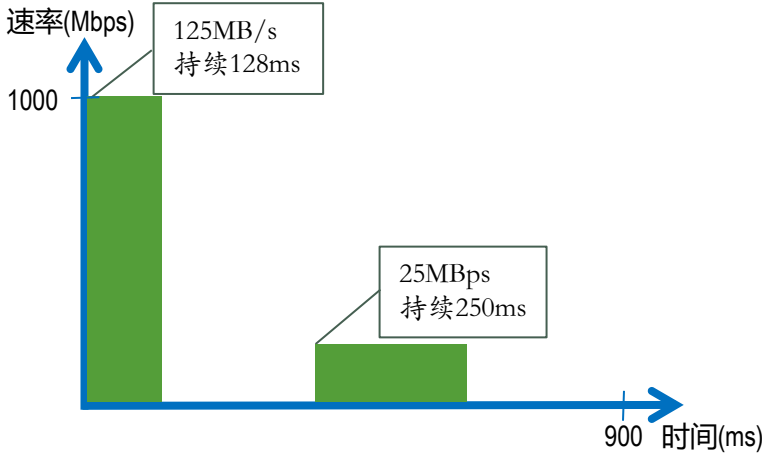
- 示例
  - 主机产生数据：突发16000KB，速率1000Mbps，

    之后，速率200Mbps持续250ms (数据量6250KB)
  - 路由器、网络链路

    最大速率： 1000Mbps； (换算：B=8b，K=1000)

    1000Mbps = 125MBps，200Mbps = 25MBps
  - 主机直接输出，参见书P307(a)

    16000KB / 125MBps = 128ms



漏桶与令牌桶示例 (2)

- 示例
  - 主机产生数据：突发16000KB，速率1000Mbps，

    之后，速率200Mbps持续250ms (数据量6250KB)
  - 路由器

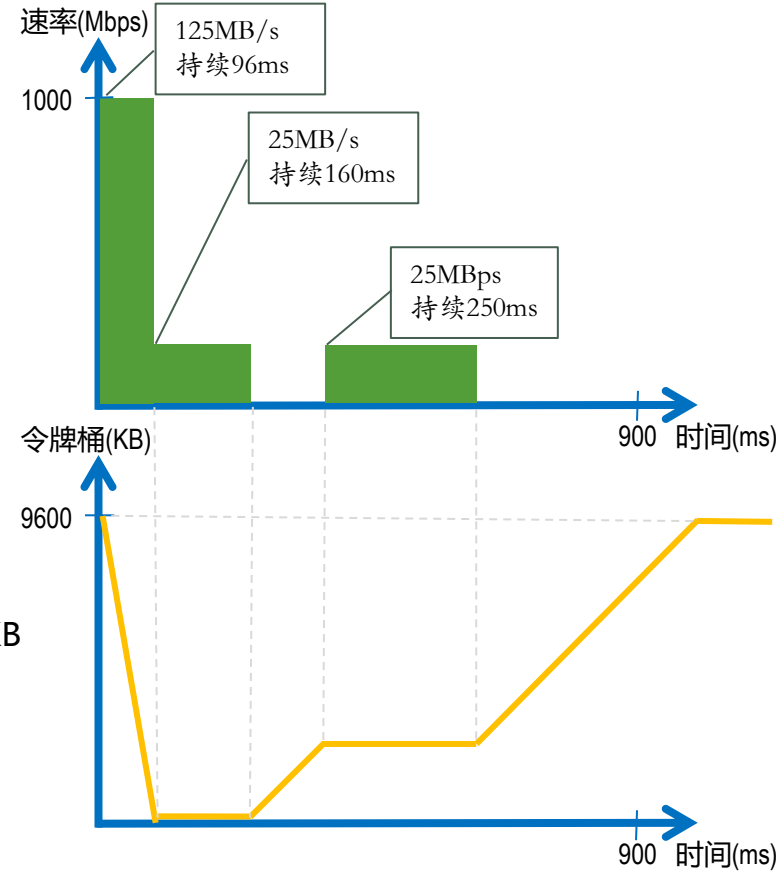
    缓冲区大小9600KB，最佳速率： 200Mbps
  - 主机端增加令牌桶，参见书P307(b)和(e)

    B = 9600KB， R = 200Mbps，M = 1000Mbps

    全速发送时间：S = B / (M - R) = 96ms

    剩余数据量：16000KB – 125MBps*96ms = 4000KB

    常速发送时间：4000KB / 25MBps = 160ms
  - 令牌桶容量线



漏桶与令牌桶示例 (3)

- 示例
  - 主机产生数据：突发16000KB，速率1000Mbps

    之后，速率200Mbps持续250ms (数据量6250KB)
  - 路由器

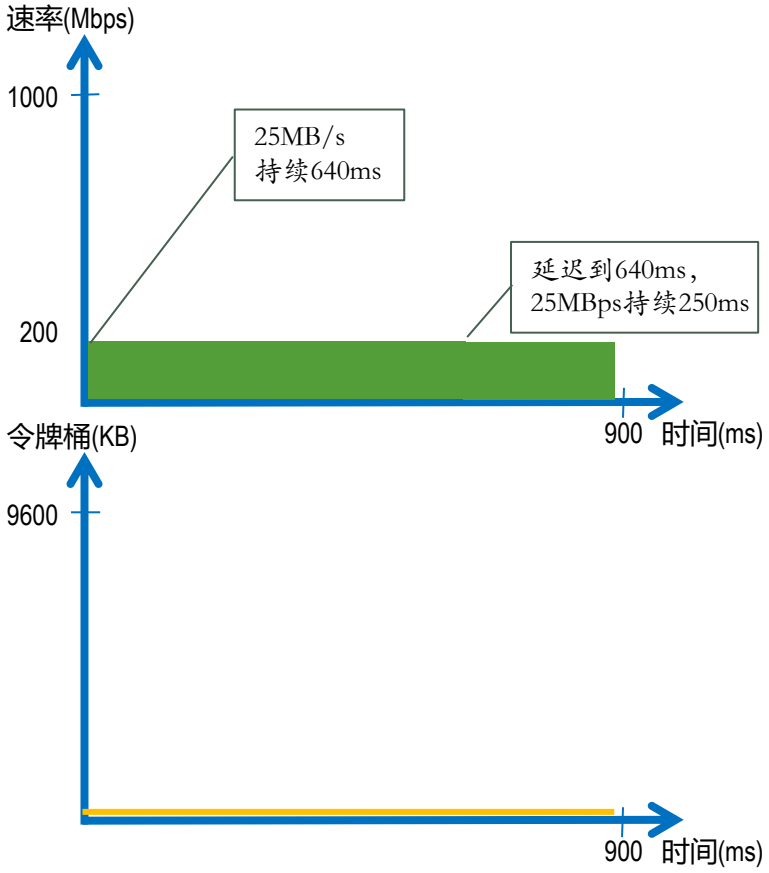
    缓冲区大小9600KB，最佳速率： 200Mbps
  - 主机端令牌桶容量为0，参见书P307(c)和(f)

    极端例子，流量完全平滑，不允许任何突发

    常速发送时间：16000KB / 25MBps = 640ms

    第二批数据在主机中排队等待到640ms之后发送

    总持续时间：640ms + 250ms = 890ms



漏桶与令牌桶示例 (4)

- 主机产生数据

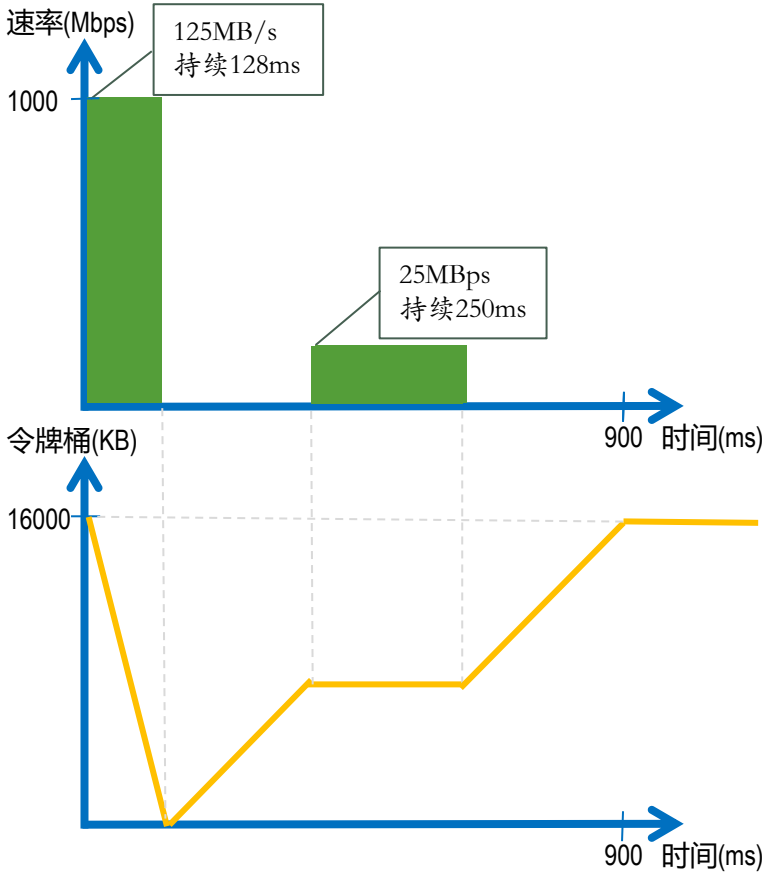
  突发16000KB，速率1000Mbps

  之后，速率200Mbps持续250ms (数据量6250KB)

- 令牌桶容量为最大时，桶容量线

  B = 16000KB， R = 200Mbps，参见书P307(d)

- 令牌桶的实现：计数
  - 计数器为离散值， 每嘀嗒ΔT秒，加 R/ΔT 字节
  - 若分组大小一致，桶容量也可以分组数来计数
- 不希望峰值降得过快：采用两个令牌桶
  - 第一个桶表述流量特征：平均速率，少量突发
  - 第二个桶降低突发峰值，如M= 500Mbps，B=0



主动队列管理

- 拥塞避免
  - 监视何时何地发生拥塞
    - 输出线路利用率
    - 缺少缓冲区空间而丢失分组的比例（超时和重发分组的数量）
    - 平均队列长度（排队延迟估计d ， s表示瞬时队列长度样值）

$$d_{new} = \alpha\, d_{old} + (1 - \alpha)\, s$$

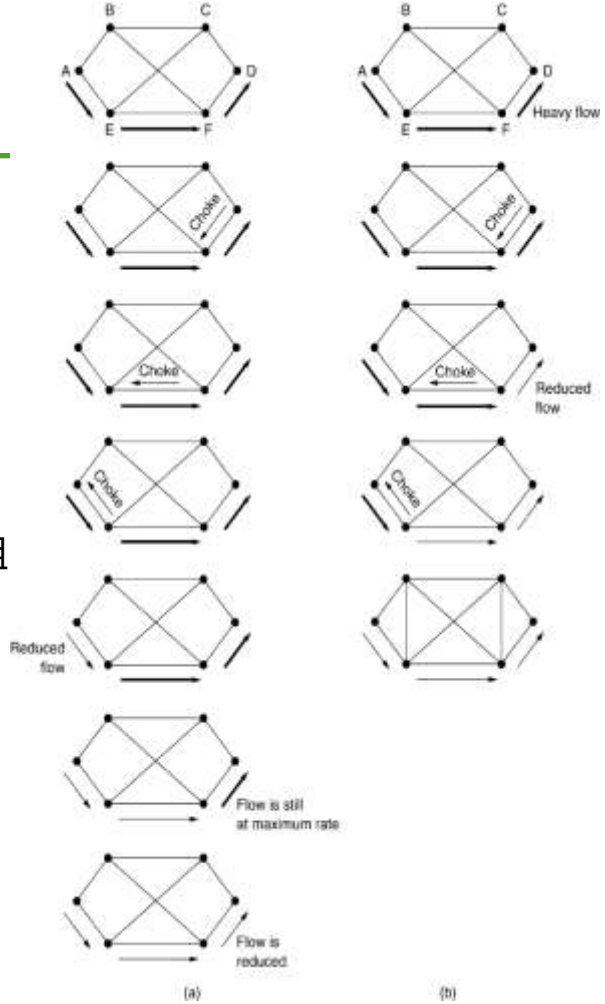
  - 将信息传递给可行动者
    - 拥塞路由器向信流源发送一个分组
    - 分组保留拥塞位或字段；证实分组询问拥塞
- 随机早期检测
  - 主机调整策略参数；数据率减半（有线网非常可靠，TCP丢包按拥塞处理）

抑制分组

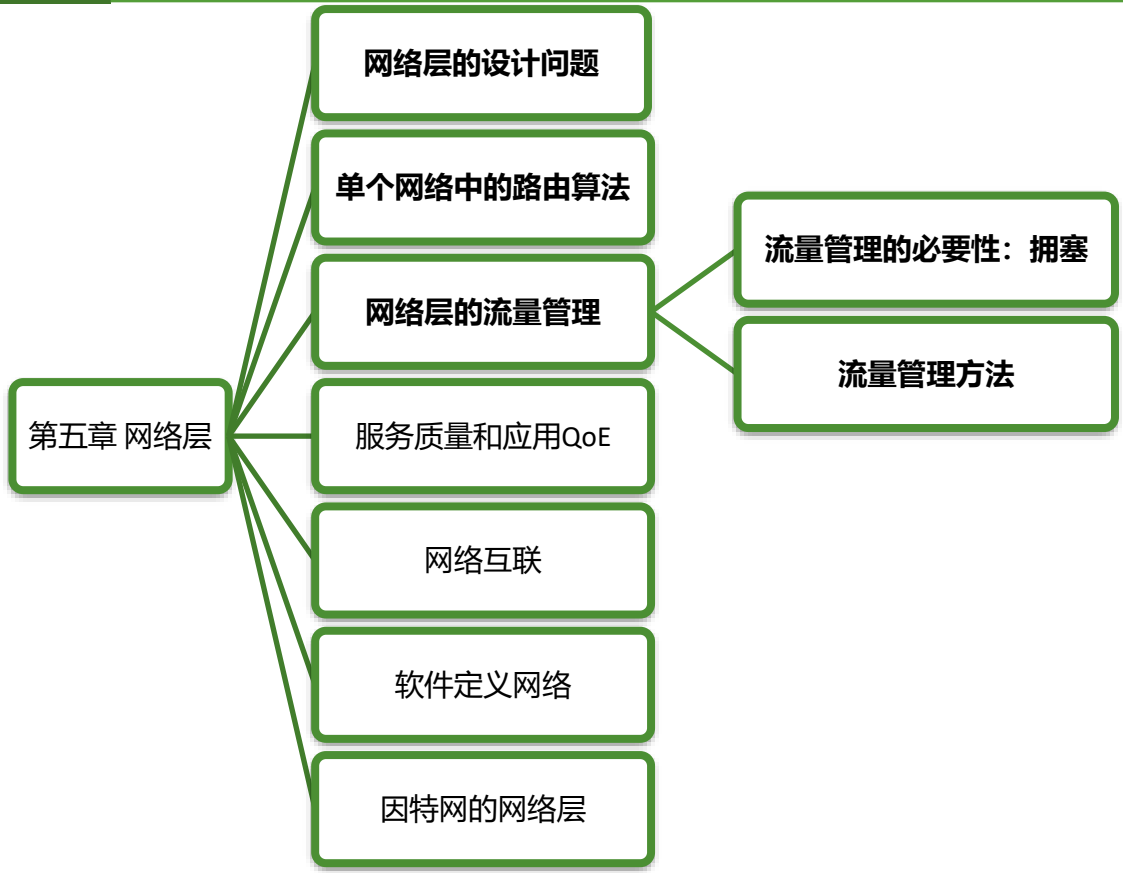
- 抑制分组 (Choke Packets) 基本思想
  - 路由器监控输出线路及其它资源的利用情况，超过某个阈值，则此资源进入警戒状态
  - 每个新分组到来，检查它的输出线路是否处于警戒状态
  - 若是，则向源主机发送抑制分组，分组中指出发生拥塞的目的地址。同时将原分组打上标记（称为显示拥塞通知，为了以后不再产生抑制分组），正常转发
  - 源主机收到抑制分组后，按一定比例减少发向特定目的地的流量，并在固定时间间隔内忽略指示同一目的地的抑制分组。然后开始监听，若此线路仍然拥塞，则主机在固定时间内减轻负载、忽略抑制分组；若在监听周期内没有收到抑制分组，则增加负载
  - 通常采用的流量增减策略是：减少时，按一定比例减少，保证快速解除拥塞；增加时，以常量增加，防止很快导致拥塞

逐跳抑制分组

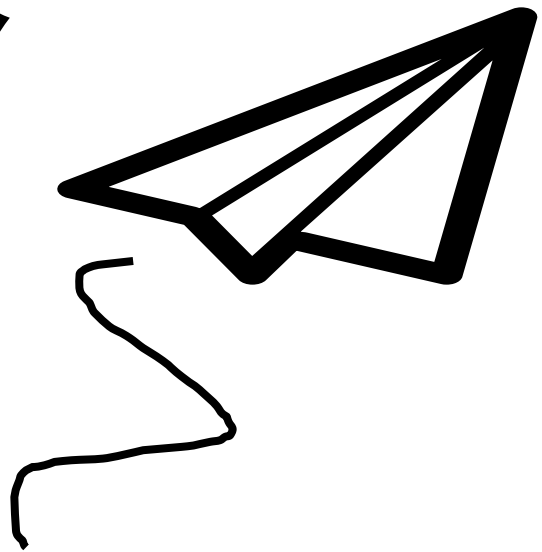
- 显示拥塞通知
  - 警告位
  - 应答通知
- 逐跳抑制分组（Hop-by-Hop Choke Packets）
  - 在高速、长距离的网络中，由于源主机响应太慢，抑制分组算法对拥塞控制的效果并不好，可采用逐跳抑制分组算法
- 基本思想
  - 抑制分组对它经过的每个路由器都起作用
  - 能够迅速缓解发生拥塞处的拥塞
  - 上游路由器要求有更多的缓冲区



本章导航与要点


第五章 网络层
网络层的设计问题
单个网络中的路由算法
网络层的流量管理
流量管理的必要性：拥塞
流量管理方法
服务质量和应用QoE
网络互联
软件定义网络
因特网的网络层

本节课程结束



5.4 服务质量和应用QoE

- 应用需求
- 过度配置
- 分组调度
- 综合服务
- 区分服务

应用需求

- 源端发给目的端的一系列分组称为一个**流**
- 每个流的需求总结为服务质量 (QoS，Quality of Service)

应用	带宽	延迟	抖动	可靠性
电子邮件	低	低	低	中等
文件共享	高	低	低	中等
Web访问	中等	中等	低	中等
远程登录	低	中等	中等	中等
音频点播	低	低	高	低
视频点播	高	低	高	低
电话	低	高	高	低
视频会议	高	高	高	低

过度配置

- 足够容量网络
  - 服务质量，堆量解决（如电话）
  - 成本太高，流量预期稳定
- 确保服务质量（小容量网络）要解决的问题
  - 应用对网络的要求?
  - 进入网络的流量如何规范?
  - 路由器如何预留资源以保证性能?
  - 网络如何安全地增加流量?

分组调度

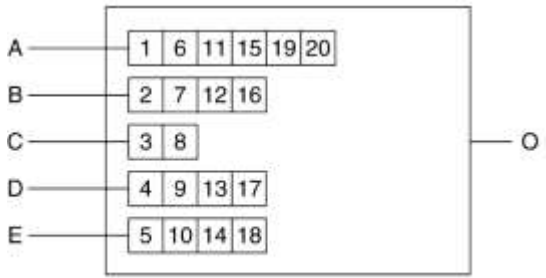
- 分组调度算法
  - 资源预留
    - 带宽
    - 缓冲区
    - CPU周期
  - 处理方法
    - 先入先出

      (FIFO，First-In First-Out)
    - 公平队列
    - 加权公平队列

公平队列的轮询机制

公平队列

- 公平队列（Fair Queueing）算法
  - 路由器的每个输出线路有多个队列
  - 路由器循环扫描各个队列，发送队头的分组
  - 所有主机具有相同优先级
  - 一些ATM交换机、路由器使用这种算法
- 一种改进
  - 对于变长分组，由逐分组轮讯改为逐字节轮讯



(a)

分组	完成时间
C	8
B	16
D	17
E	18
A	20

(b)

加权公平队列

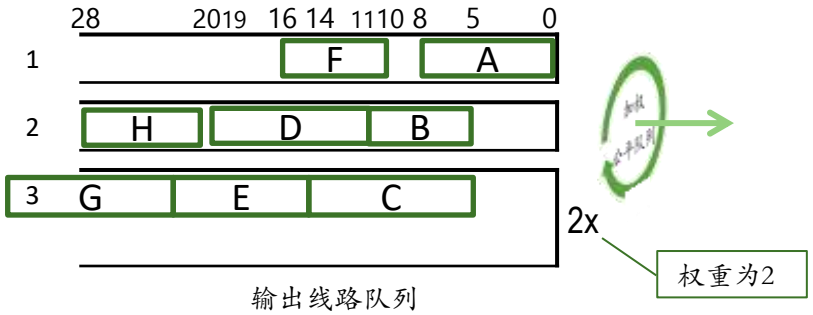
- 加权公平队列（Weighted Fair Queueing）算法
  - 给不同主机以不同的优先级（权值等于虚电路或流的数目）
  - 优先级高的主机在一个轮讯周期内获得更多的时间片
    - 例，分组到达顺序：A、B|C、D|E、F、G、H
    - 相对发送顺序为：A、B、C、F、D、E、H、G

WFQ结束时间：

$$F_i = \max(A_i, F_{i-1}) + L_i / W$$

A_i到达时间，F_i完成时间，L_i分组长度

分组	到达时间	长度	出发时间	完成时间	完成顺序
A	0	8	0	8	1
B	5	6	5	11	2
C	5	10	5	15	3
D	8	9	11	20	5
E	8	8	15	23	6
F	10	6	10	16	4
G	11	10	23	33	8
H	20	8	20	28	7



思考题：如果H到达时间为19或21，完成时间是多少？

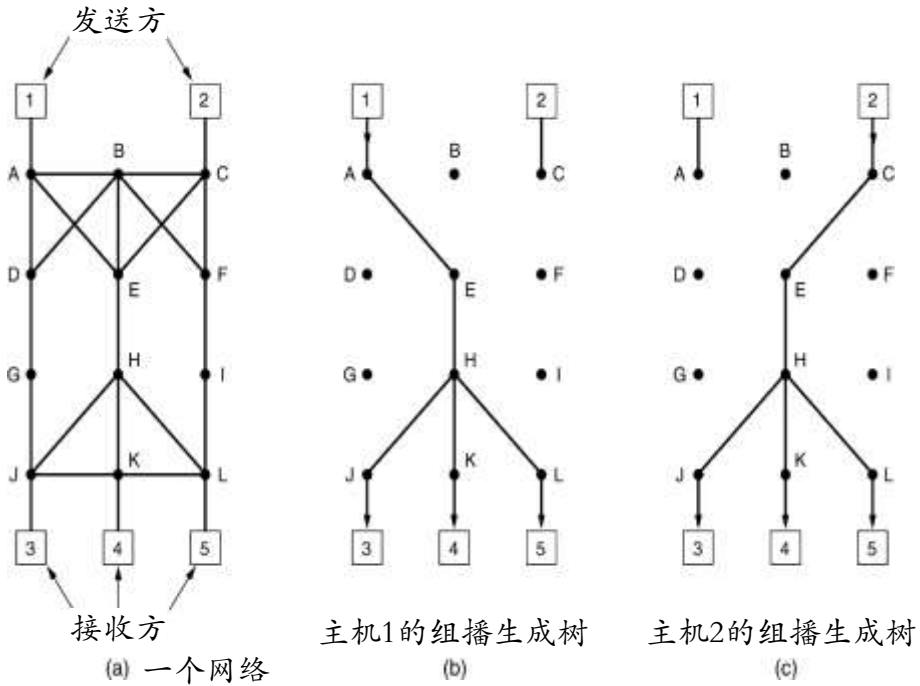
结合应用

- 服务质量
  - 服务质量保证通过许可控制过程来建立
  - QoS路由：选择的路径满足带宽需要
- 流规范
  - 一个数据流的发送方、接收方和网络三方认可的、描述发送数据流的模式和希望得到的服务质量的数据结构，称为流规范
  - 对发送方的流规范，网络和接收方可以做出三种答复：同意、拒绝、其它建议

参数	单位
令牌桶速率	字节/秒
令牌桶容量	字节
峰值速率	字节/秒
最小分组大小	字节
最大分组大小	字节

综合服务

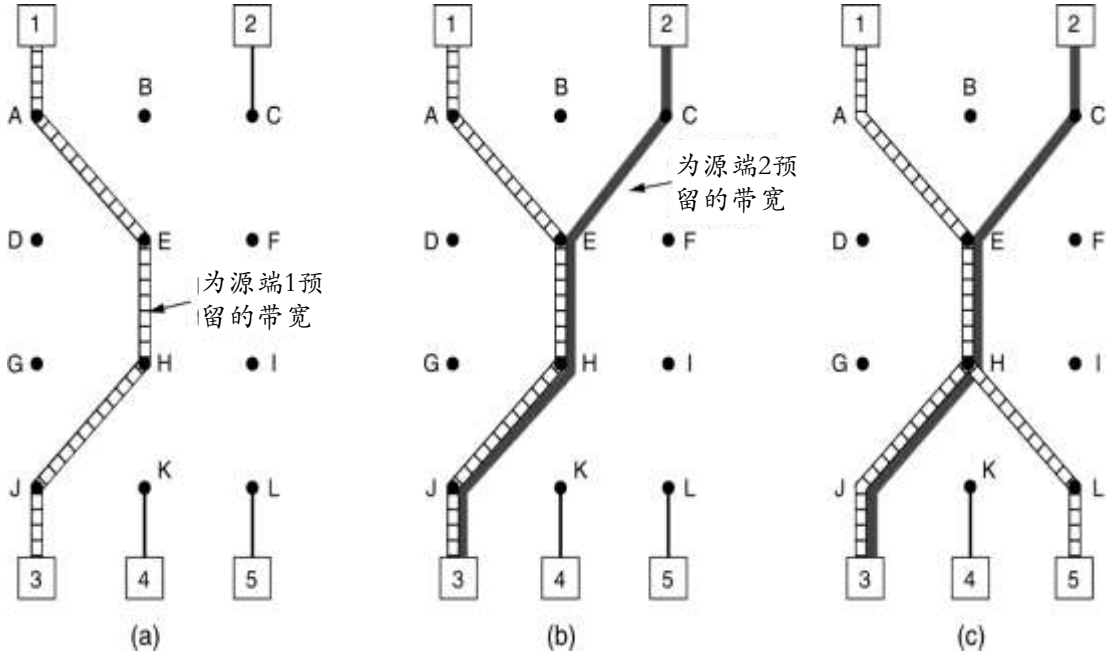
- 综合服务主要针对流式媒体的单播和组播应用
- RSVP资源预留协议
  - 使用基于生成树的组播路由
  - 接收方沿树发送预留消息
  - 每一跳都预留必要带宽
  - 一路带宽都预留直至发送方



RSVP资源预留

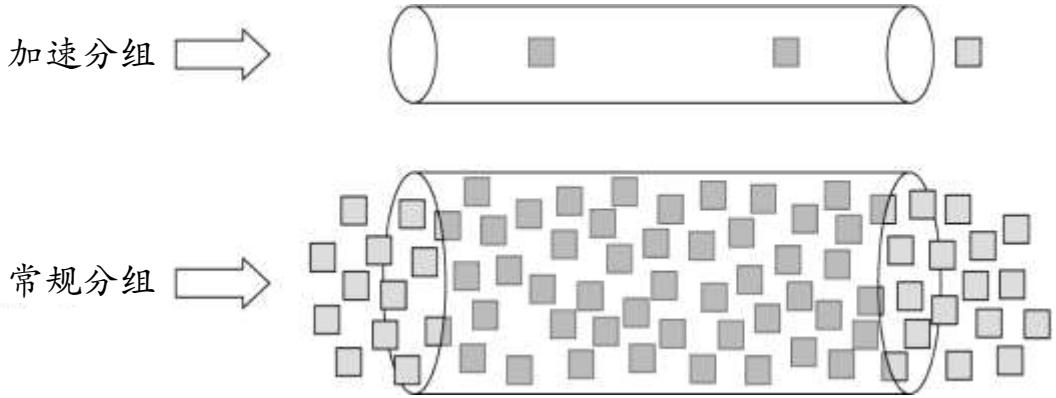
- 资源预留例子
  - a) 主机3向主机1请求一条信道
  - b) 再向主机2请求第二条信道
  - c) 主机5向主机1请求一条信道
- 路径优化
  - 如果3和5不改变数据源1

    A-E-H可共享



区分服务

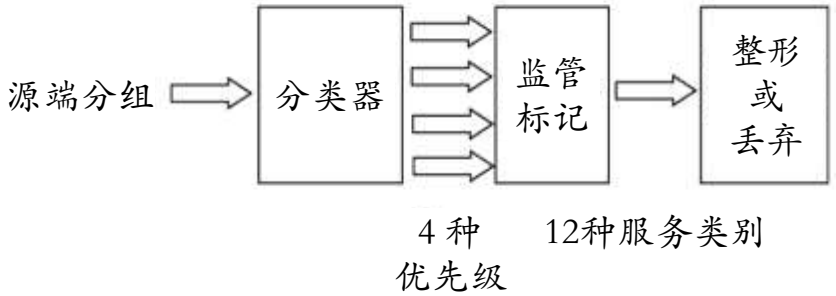
- 基于类别的服务质量
  - 不同于基于流的服务质量
- 加速转发
  - 分组分为加速与常规两类，并作相应标记

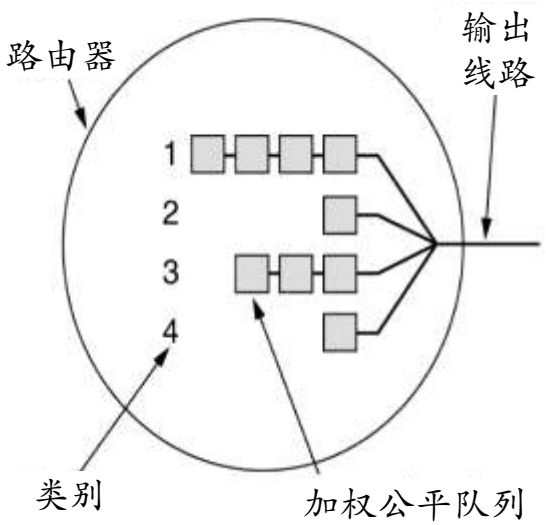


确保转发

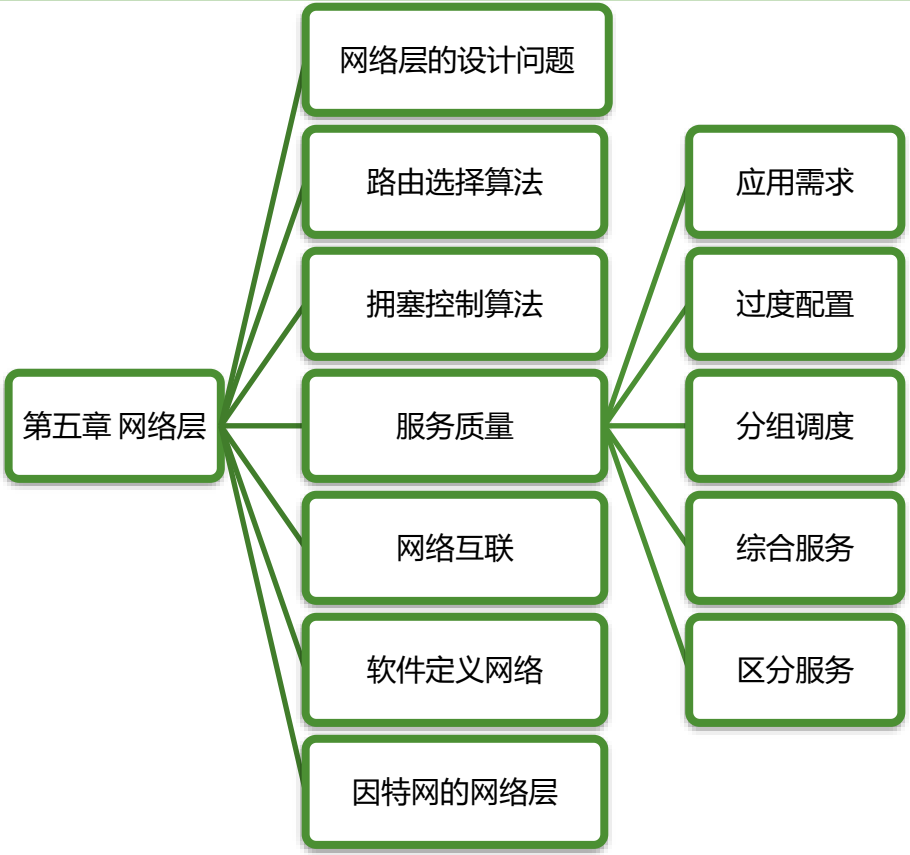
- 确保转发
  - 规定4种优先级（权重不同）
  - 定义拥塞控制的3种丢弃概率

    符合小突发量，超过小突发量，超过大突发量

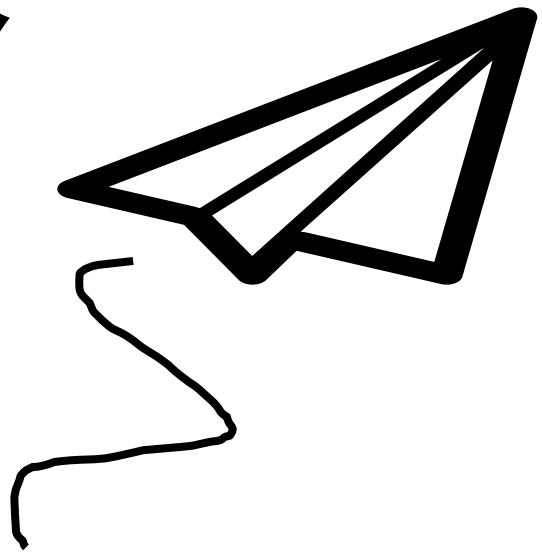




本章导航与要点


第五章 网络层
网络层的设计问题
路由选择算法
拥塞控制算法
服务质量
网络互联
软件定义网络
因特网的网络层
应用需求
过度配置
分组调度
综合服务
区分服务

本节课程结束



5.5 网络互联

- 网络互联概述
- 网络的不同
- 异构网络的连接
- 跨异构网络的连接
- 互联网路由
- 支持不同分组长度：分段

网络互联概述

- 网络的多样性和复杂性
  - 同质网络：网络各层协议相同
  - 不同范围的网络：PAN、LAN、MAN、WAN等
  - 不同技术的网络：802.3、802.4、802.5、802.11、802.16、卫星网等
  - 现有系统重用的数据网络：固定电话网、移动电话网、有线电视网、电力线网等
- 互联网与因特网
  - 互联网(internet)：不同网络的互联
  - 因特网(Internet)：互联网特例，最大的互联网

网络的不同

项 目	说 明
提供的服务	面向连接服务和无连接服务
寻址	寻址范围大小，层次或扁平
广播	是否广播和组播
分组大小	不同网络大小不同
有序性	有序或无序传输
服务质量	提供或缺乏；种类的不同
可靠性	丢包的不同级别
安全性	隐私规则、加密等
参数	不同的超时值、流规范等
计费	不收费或按连接时间、分组数、字节数等收费

异构网络的连接

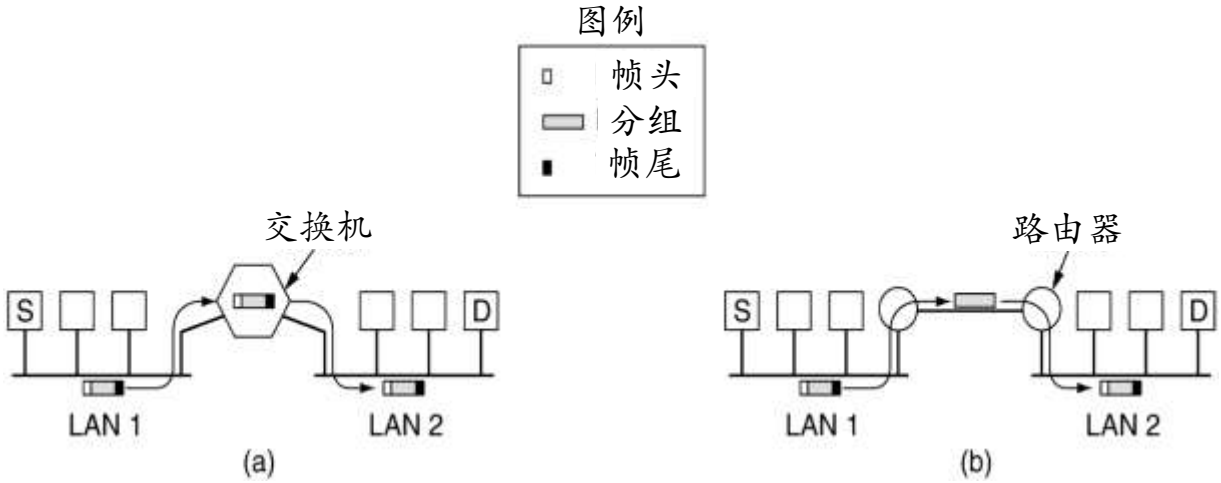
- 网络互联设备
- 互连网络与虚拟互连网络
- 级联虚电路
- 无连接网络互连

网络互联设备

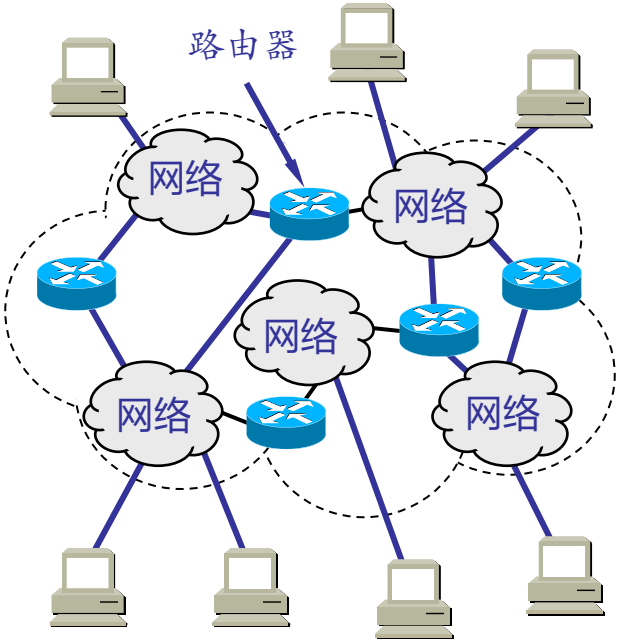
- 集线器与交换机（参见第四章PPT-p133）
  - 当中继系统是中继器或网桥时，一般并不称之为网络互连，因为这仅仅是把一个网络扩大了，而这仍然是一个网络
- 网关
  - 网关由于比较复杂，目前使用得较少
- 路由器
  - 互联网一般是指用路由器进行互连的网络
    - 由于历史的原因，许多有关 TCP/IP 的文献将网络层使用的路由器称为网关

路由器与交换机

- 两个以太网的连接
  - 交换式连接：交换机不识别分组，使用MAC地址
  - 路由连接：路由器从帧中提取分组、分组地址（IP地址）



互连网络与虚拟互连网络



(a) 互连网络

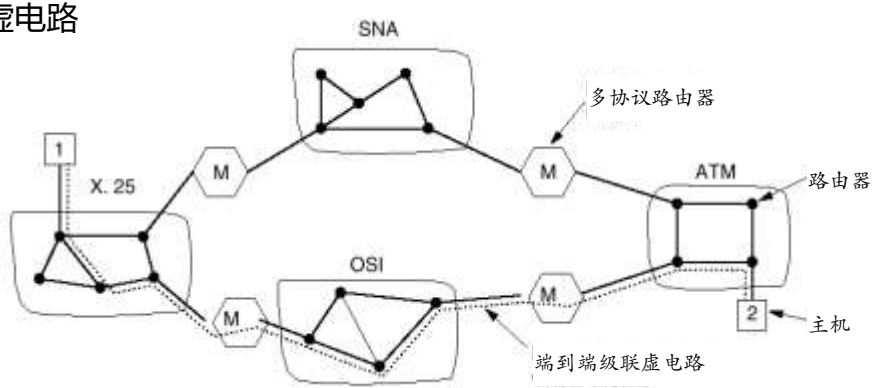
(b) 虚拟互连网络

虚拟互连网络

- 虚拟互连网络的含义
  - 虚拟互连网络就是逻辑互连网络，它的意思就是互连起来的各种物理网络的异构性本来是客观存在的，但是我们利用统一的(IP)协议，就可以使这些性能各异的网络从用户看起来好像是一个统一的网络
- IP网络
  - 使用 IP 协议的虚拟互连网络可简称为 IP 网
- 使用虚拟互连网络的好处
  - 当互联网上的主机进行通信时，就好像在一个网络上通信一样，而看不见互连的各具体的网络异构细节

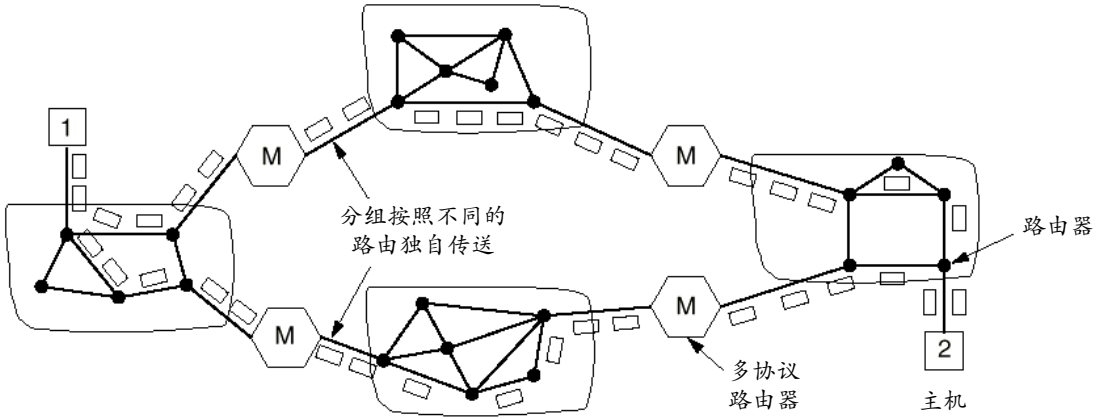
级联虚电路

- 级联虚电路（Concatenated Virtual Circuits）工作过程
  - 级联虚电路工作过程与虚电路网络工作过程相似
  - 建立连接
    - 当目的主机不在子网内时，则在子网内找一个离目的网络最近的路由器，与之建立一条虚电路
    - 该路由器与外部网关建立虚电路
    - 该网关与下一个子网中的一个路由器建立虚电路
    - 重复上述操作，直到到达目的主机
  - 传输数据
    - 同一连接的分组沿相同虚电路按序号传输
    - 网关根据需要转换分组格式和虚电路号
  - 拆除连接



无连接网络互连

- 无连接网络互连（Connectionless Internetworking），工作过程
  - 无连接网络互连的工作过程与数据报网络的工作过程相似
  - 每个分组单独路由，提高网络利用率，但不能保证分组按顺序到达
  - 根据需要，连接不同子网的多协议路由器做协议转换，包括分组格式转换和地址转换等

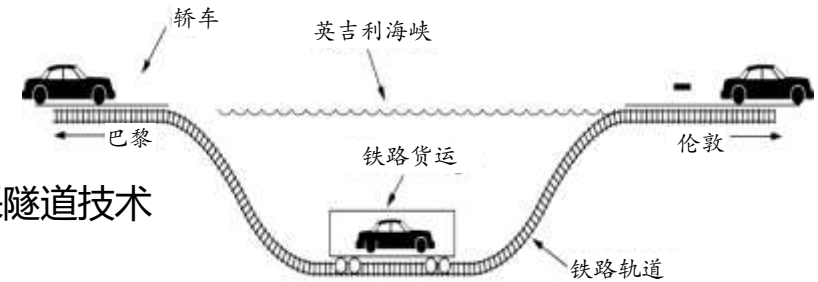


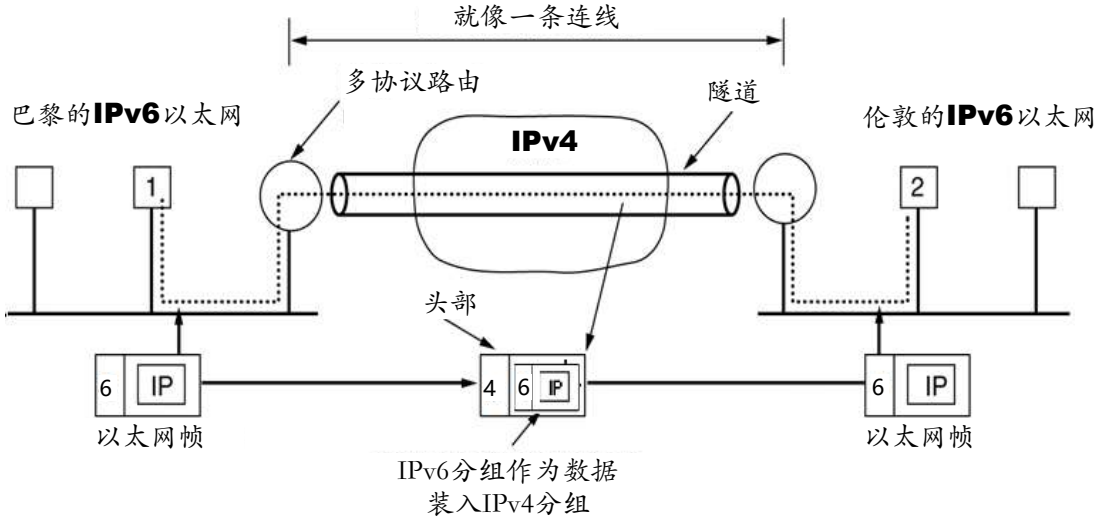
网络互联比较

- 级联虚电路
  - 级联虚电路的优点
    - 路由器预留缓冲区等资源，保证服务质量
    - 包按序号传输
    - 短分组头
  - 级联虚电路的缺点
    - 路由器需要大量内存，存储虚电路信息
    - 一旦发生拥塞，没有其它路由
    - 健壮性差
    - 如果网络中有一个不可靠的数据报子网，级连虚电路很难实现
- 无连接网络互连
  - 无连接网络互连的优点
    - 能够容忍拥塞，并能适应拥塞
    - 健壮性好
    - 可用于多种网络互连
  - 无连接网络互连的缺点
    - 长分组头
    - 分组不能保证按序号到达
    - 不能保证服务质量

跨异构网络的连接

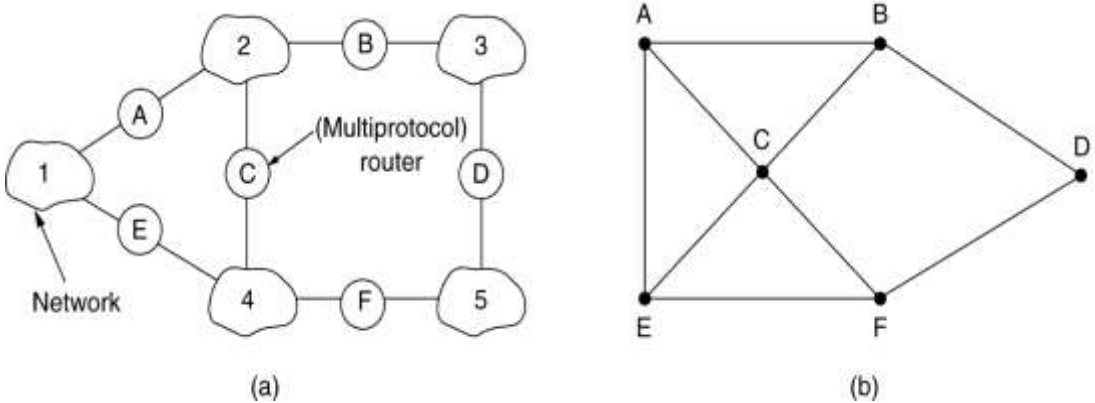
- 隧道技术
  - 源和目的主机所在网络类型相同，连接它们的是一个不同类型的网络，这种情况下可以采隧道技术





互联网路由

- 互连网与图示



- 域内：内部网关协议RIP、OSPF
- 域间：外部网关协议，因特网中称为边界网关协议BGP
- 自治系统AS

支持不同分组长度：分段

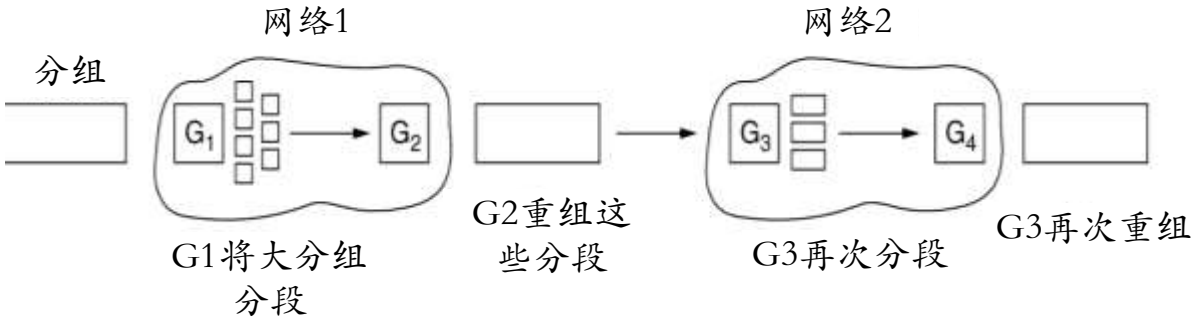
- 为何分段（Fragmentation）
  - 解决网络最大长度限制
- 如何分段
  - 透明分段
    - ATM网采用的策略
    - 网关将大分组分段后，每段都要经过同一出口网关，并在那里重组
  - 不透明分段
    - IP网采用的策略
    - 中间网关不做重组，而由目的主机做
- 分段方法与示例

为何分段

- 每种网络都对最大分组长度有限制，有以下原因
  - 硬件，例如 TDM 的时槽限制
  - 操作系统
  - 协议，例如分组长度域的比特个数
  - 与标准的兼容性
  - 希望减少传输出错的概率
  - 避免一个分组占用信道时间过长
- 大分组如何经过小分组网络
  - 方法一：源端确定路径最大传输单元MTU，不发送大于MTU的大分组
  - 方法二：网关要将大分组分成若干段（fragment），每段作为独立的分组传输

透明分段

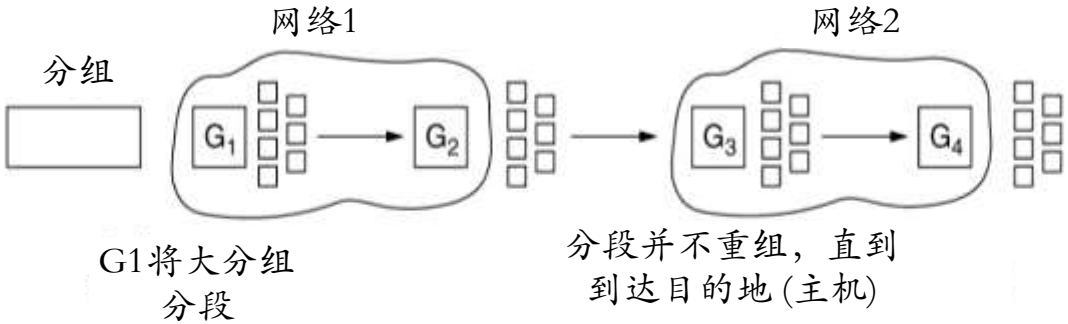
- 分段重组过程对其它网络透明



- 带来的问题
  - 出口网关需要知道何时所有分组都到齐
  - 所有分组必须从同一出口网关离开
  - 大分组经过一系列小分组网络时，需要反复地分段重组，开销大

不透明分段

- 分段重组过程对其它网络不透明



- 带来的问题
  - 对主机要求高，能够重组
  - 每个段都要有一个分组头，网络开销增大

分段方法

- 标记段
  - 树型标记法
    - 例，分组0分成三段，分别标记为 0.0、0.1、0.2

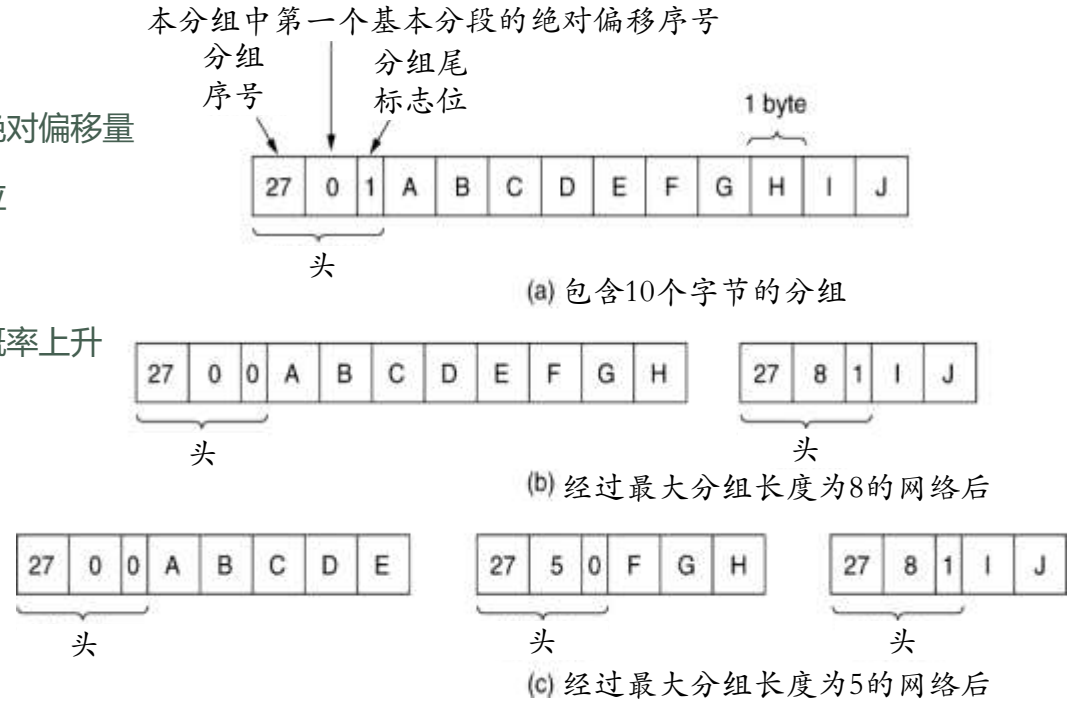
      段0.0构成的分组再被分成三段，分别标记为 0.0.0、0.0.1、0.0.2
    - 存在的问题

      段标记域要足够长；分段长度前后要一致
  - 偏移量法
    - 定义一个基本段长度，使得基本段能够通过所有网络
    - 分组分段时，除最后一个段小于等于基本段长度外，所有段长度都等于基本段长度
    - 一个分组可以包括几个段，分组头中包括：

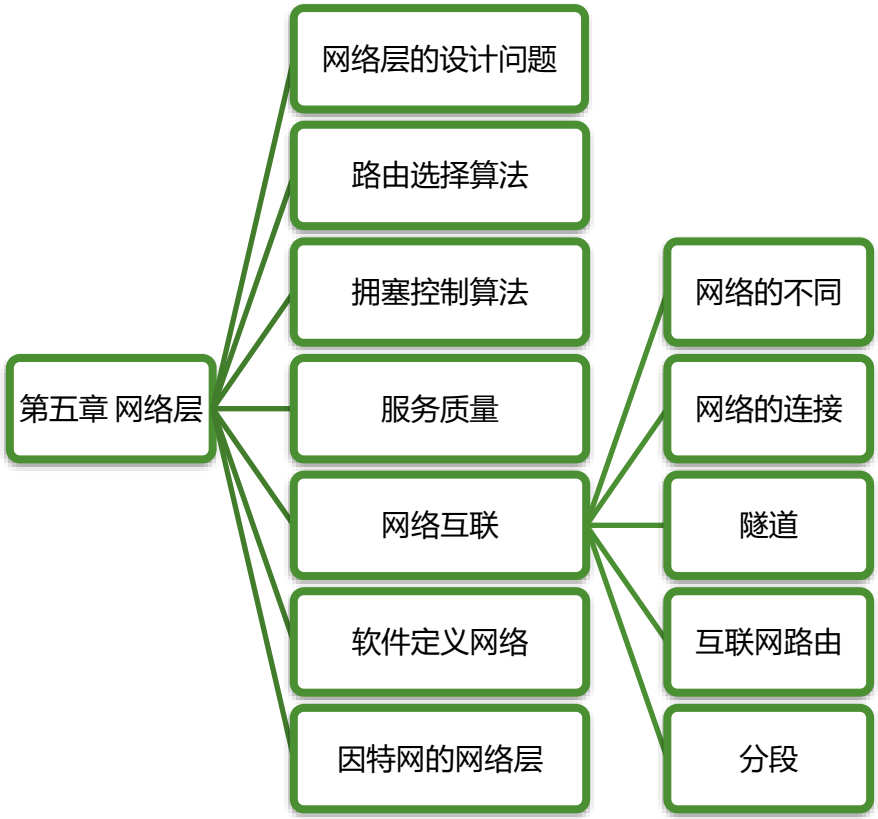
      原始分组序号，本分组中第一个基本段的偏移量，最后段指示位

分段示例

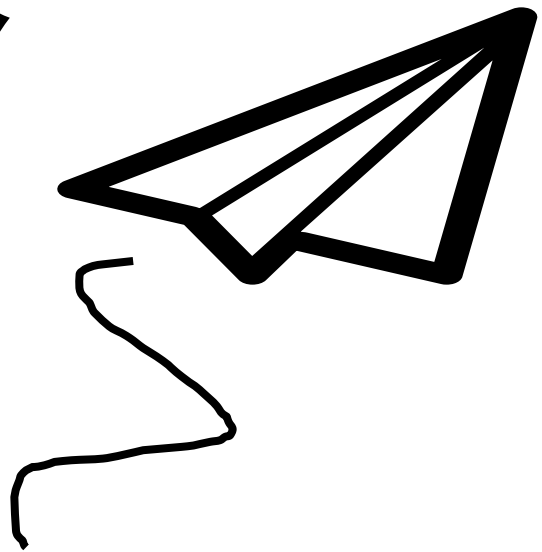
- IP网采用的设计思想
  - 每个字段保存
    - 分组序号；分组内字段绝对偏移量
    - 是否到达分组尾的标志位
  - 问题
    - 增加头开销，分组丢失概率上升
  - 回到路径MTU
    - 增加不允许分段标志
    - 不断试错，或高层告知
- 例子
  - 分组经过两个小分组网络



本章导航与要点


第五章 网络层
网络层的设计问题
路由选择算法
拥塞控制算法
服务质量
网络互联
软件定义网络
因特网的网络层
网络的不同
网络的连接
隧道
互联网路由
分段

本节课程结束



5.6 软件定义网络

- 概述
- SDN控制平面：逻辑中心化的软件控制
- SDN数据平面：可编程硬件
- 可编程网络测量

概述

- SDN的主要思想

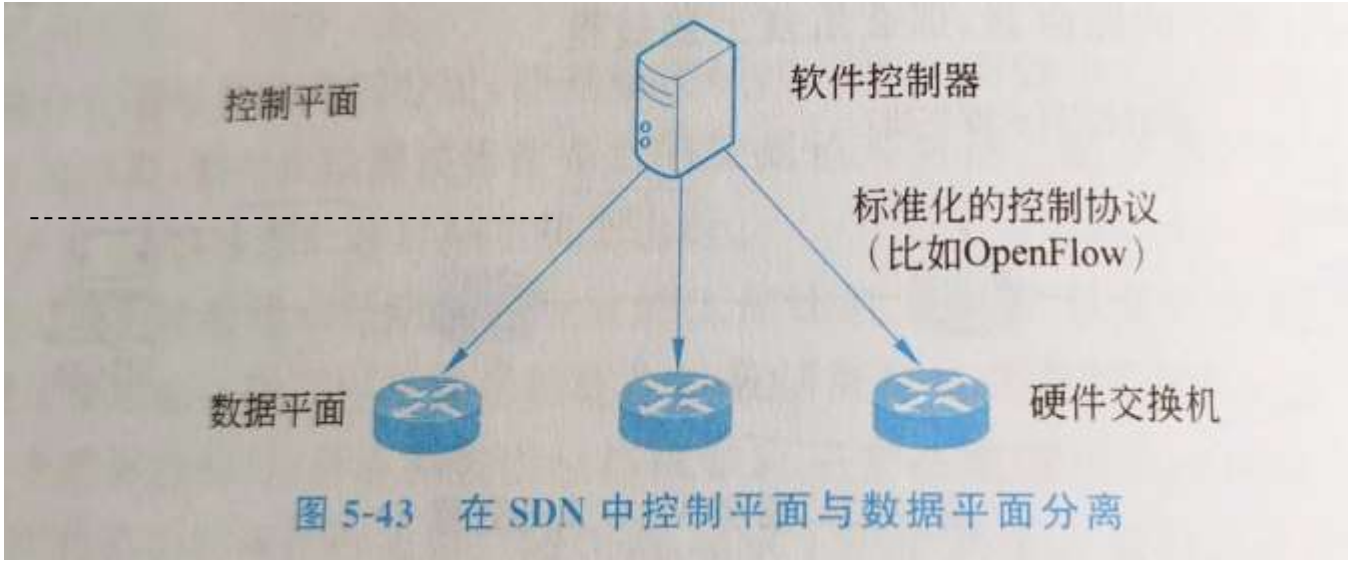


图 5-43　在 SDN 中控制平面与数据平面分离

SDN控制平面：逻辑中心化的软件控制

- SDN控制器
  - 将网络中的控制逻辑集中到中心
  - 系统的控制中心，负责网络的内部交换路径和边界业务路由的生成，并负责处理网络状态变化事件
- OpenFlow协议
  - 匹配-动作表
  - 匹配数据包头部字段（MAC地址、IP地址）
  - 转发到特定端口

SDN数据平面：可编程硬件

- 转发交换机
  - 主要由转发器和连接器的线路构成基础转发网络，这一层负责执行用户数据的转发，转发过程中所需要的转发表项是由控制层生成的
- 协议独立的交换机体系结构

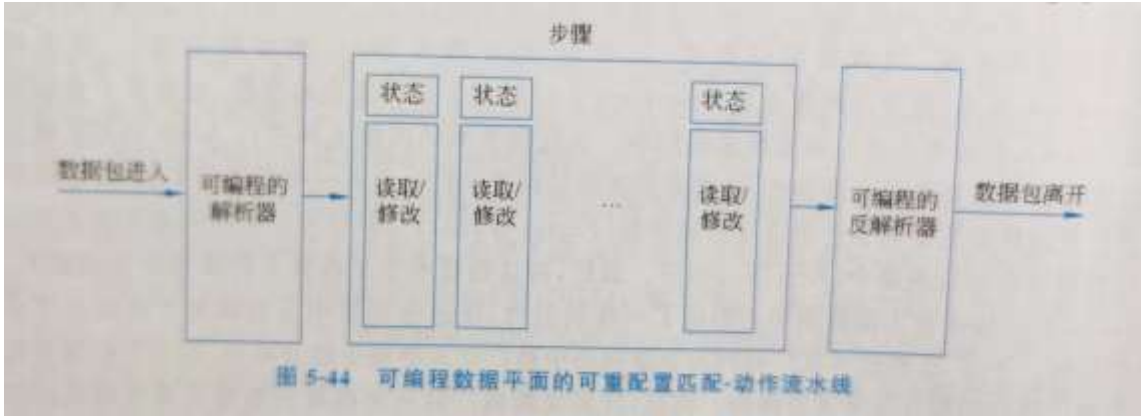


图 5-44 可编程数据平面的可重配置匹配-动作流水线

可编程网络测量

- 网络测量：包括网络状态参数、网络性能参数和网络流量参数
  - 网络状态参数包括网络的链路信息和拓扑结构，即显示网络的基本信息
  - 网络性能参数包括吞吐量、链路时延、丢包率等，即反映网络的瞬时状态
  - 网络流量参数是对一个测量周期内的网络流量进行采集和分析
- SDN 转控分离
  - 控制器具有全局视野，能对网络进行调度、制定策略，实时提供网络信息
  - 可编程交换机可以支持更灵活的网络测量

5.7 因特网上的网络层

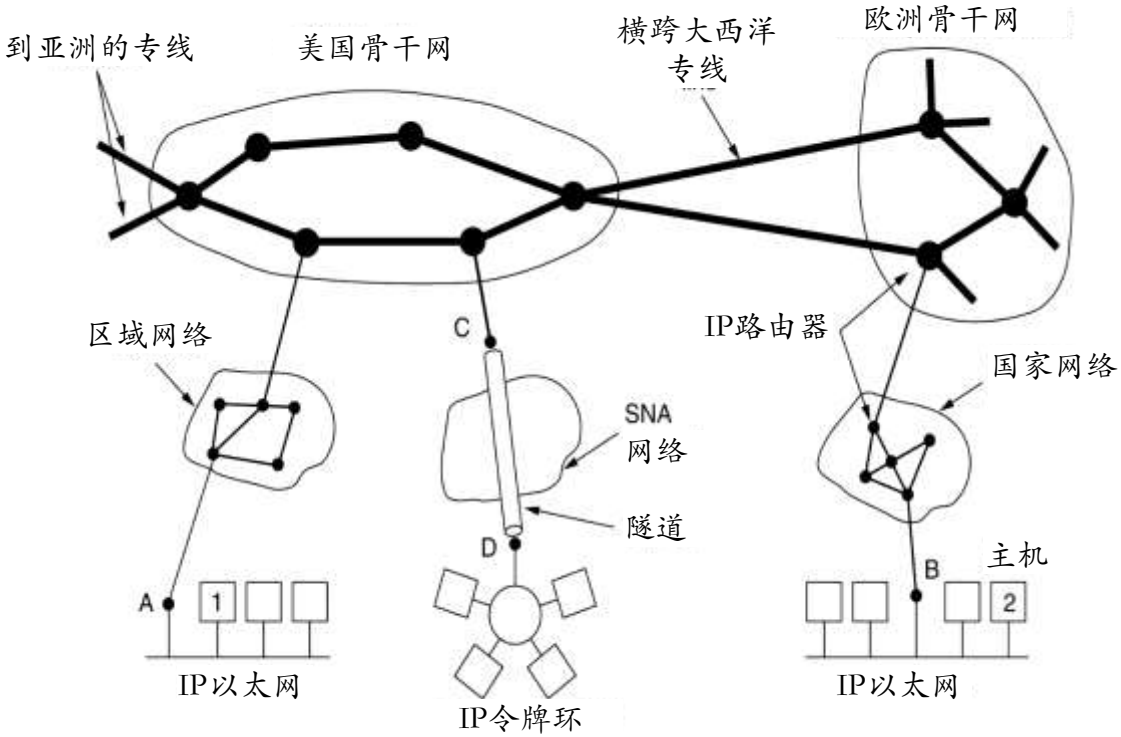
- IPv4协议
- IP地址
- IPv6协议
- 因特网控制协议
- 标签交换与MPLS
- OSPF——内部网关路由协议
- BGP——外部网关路由协议
- 因特网组播

Internet设计原则

1.	Make sure it works.	确保工作
2.	Keep it simple.	保持简单
3.	Make clear choices.	明确选择
4.	Exploit modularity.	模块开发
5.	Expect heterogeneity.	期望异构性
6.	Avoid static options and parameters.	避免静态选项和参数
7.	Look for a good design; it need not be perfect.	寻找好的而不是完美的设计
8.	Be strict when sending and tolerant when receiving.	严于发送，宽于接收
9.	Think about scalability.	考虑扩展性
10.	Consider performance and cost.	考虑性能和成本

因特网结构

- 互联网络的集合
  - 一级骨干网
  - 自治域骨干网
  - ISP
  - 局域网



协议传送顺序

- 不同的主机存放和传送数据的顺序不同
  - 大端点主机次序：从左到右，如0A0D （高位在前，参见第四章PPT-p67，802.5、802.6）
  - 小端点主机次序：从低到高，如0D0A
    - 地址表示 0000：0A0D；实际存放 00:0D，01:0A

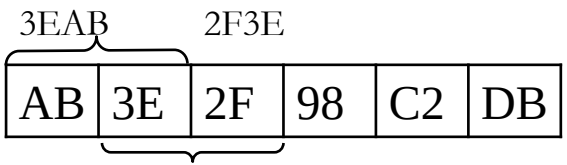
8位cpu按地址读取结果：0D

0000：0D 0A

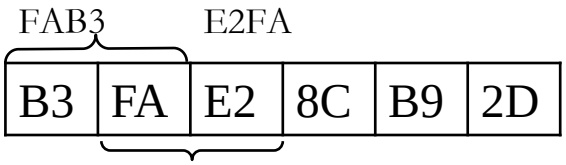
16位cpu按地址读取结果：0A0D

- 实例：FAT12数据的存放
  - 文件簇号索引数据：AB3，E2F，98C，2DB，……

直接拼接存放

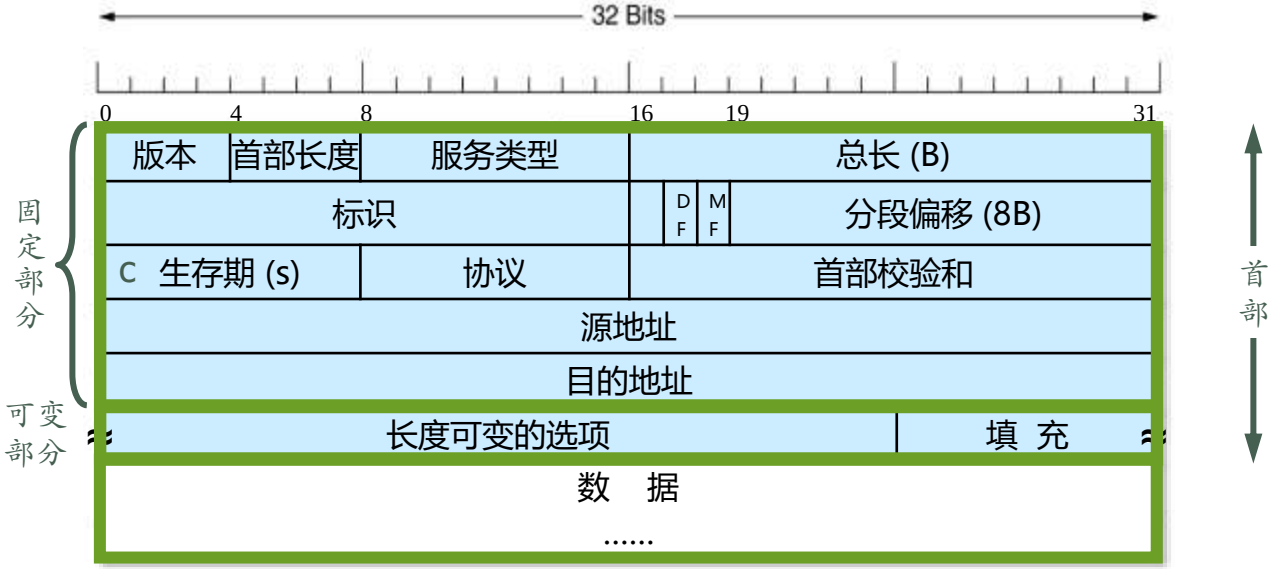


实际设计存放

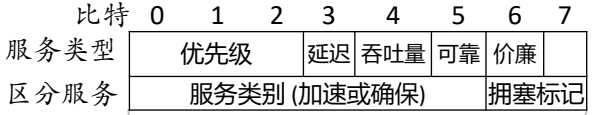


IPv4协议

- IP分组由首部和数据两部分组成
  - 首部的前一部分是固定长度，共 20 字节，是所有 IP分组必须具有的
  - 在首部的固定部分的后面是一些可选字段，其长度是可变的



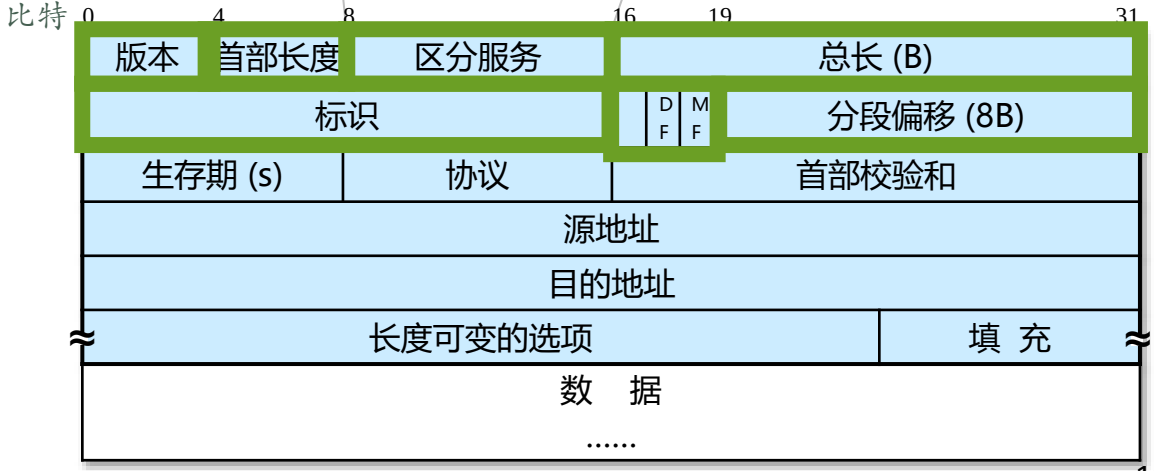
分组格式



- 版本：占 4bit，指IP协议的版本，目前的 IP 协议版本号为 4 (即 IPv4)
- 首部长度：4bit，值为5~15个单位(4 字节)，因此 IP分组首部长度最大值为60字节
- 区分服务：8bit，最初为服务类型(没用)；现在前6位标记服务类别，后2位携带显式拥塞信息
- 总长：16bit，指首部和数据之和，单位为字节，因此分组的最大长度为64K字节。注意不要超过最大传送单元 MTU
- 标识：16bit，分组序号
- 标志位：3bit，第1位未用，

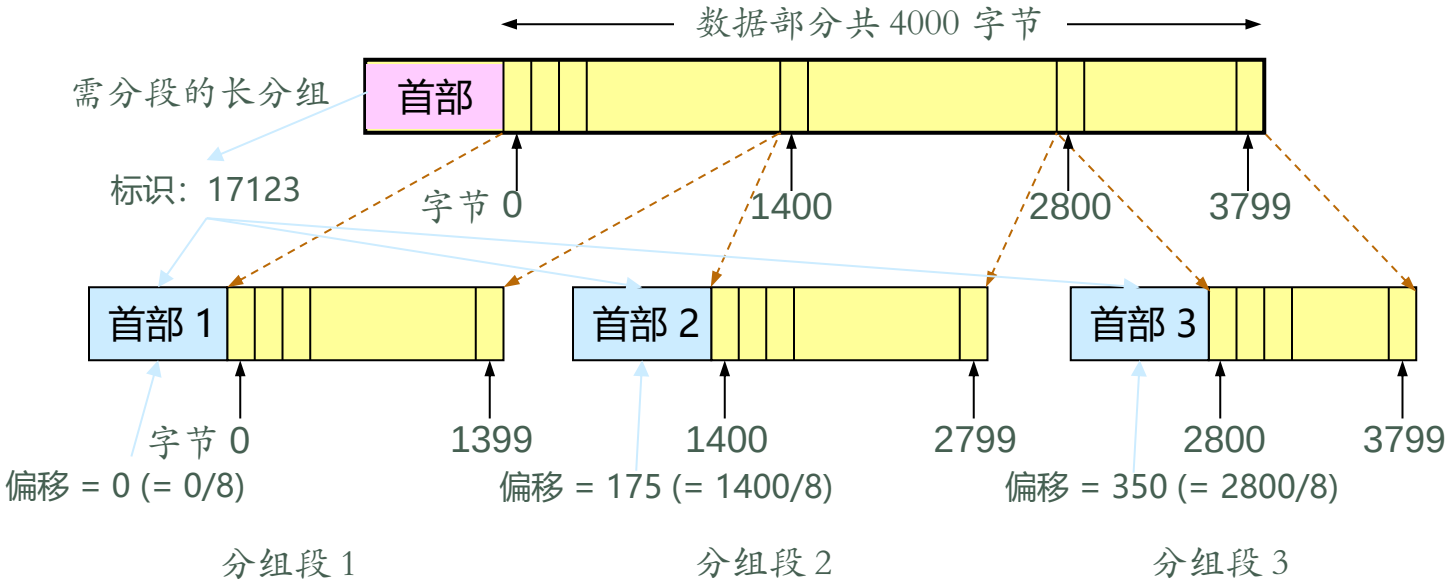
  第2位DF代表不要分段，

  第3位MF代表还有分段



分组格式 (2)

- 分段偏移：占 13bit，较长分组在分段后，本分段在原分组中的相对位置；分段偏移以 8 个字节为偏移单位，最后段之外的段长度必须为8的倍数，最多8192段
- IP 分组分段的例子 (分组长=4020，MTU = 1460B)



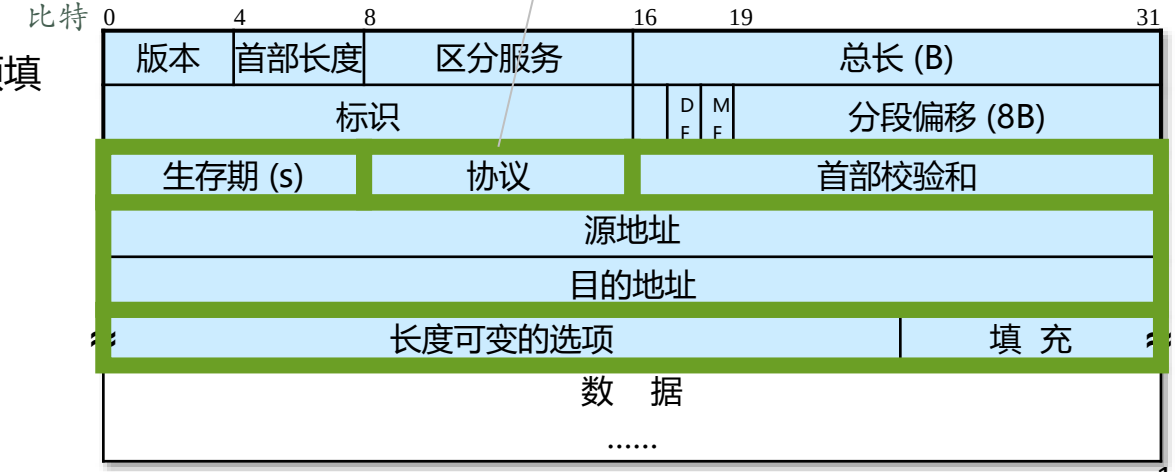
分组格式 (3)

TCP	UDP	ICMP	GGP	EGP	IGP	OSPF
6	17	1	3	8	9	89

- 生存期：8bit，记为 TTL (Time To Live)，分组在网络中的寿命计数器，每跳减1，单位为秒
- 协议：8bit，字段指出分组的数据使用何种协议，以便目的主机 IP 层将数据上交给哪个处理过程
- 首部校验和：16bit，只检验数据报的首部，不采用 CRC 而采用简单的计算方法
- 源地址、目的地址：各 4 字节，IP地址
- 选项：提供扩展途经；必须填

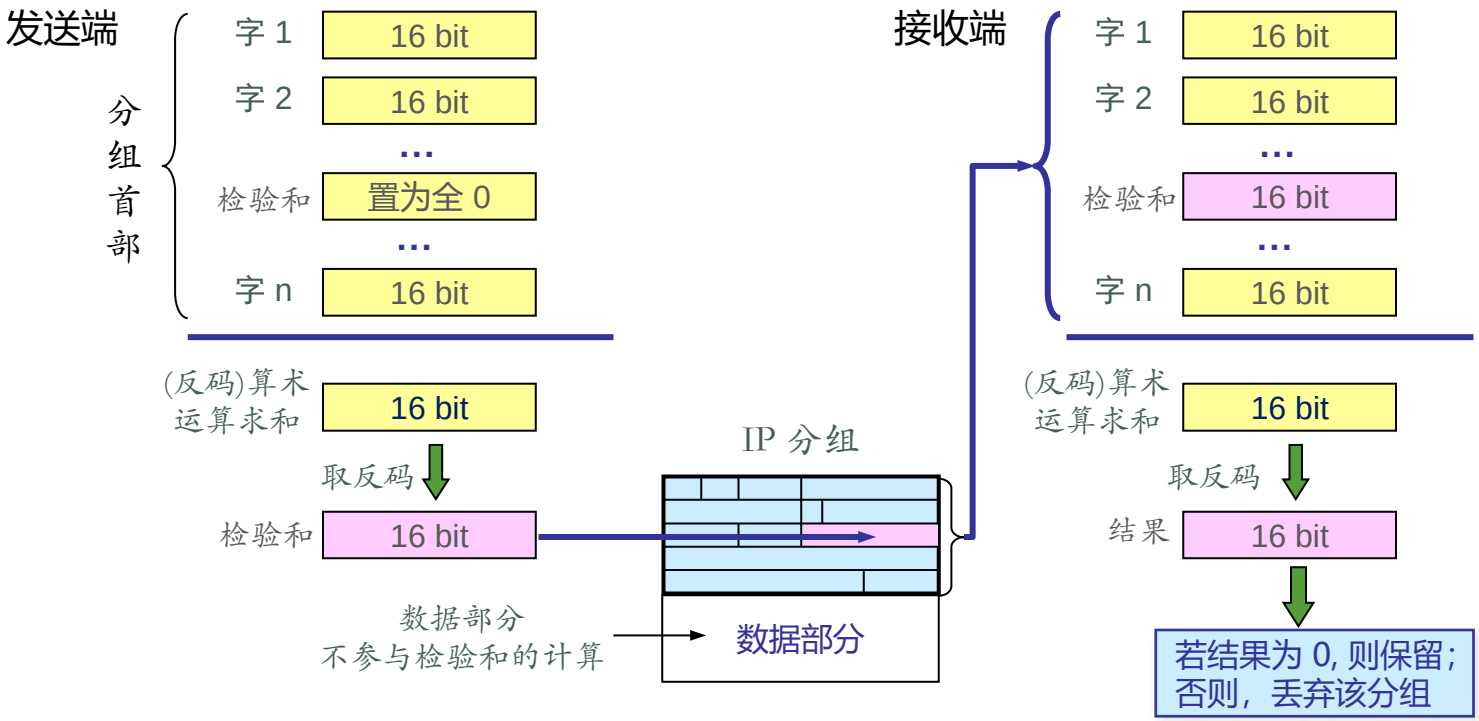
  充到4字节的倍数；IP选项：

  - 安全性
  - 严格源路由
  - 松散源路由
  - 记录路由
  - 时间戳



分组格式 (4)

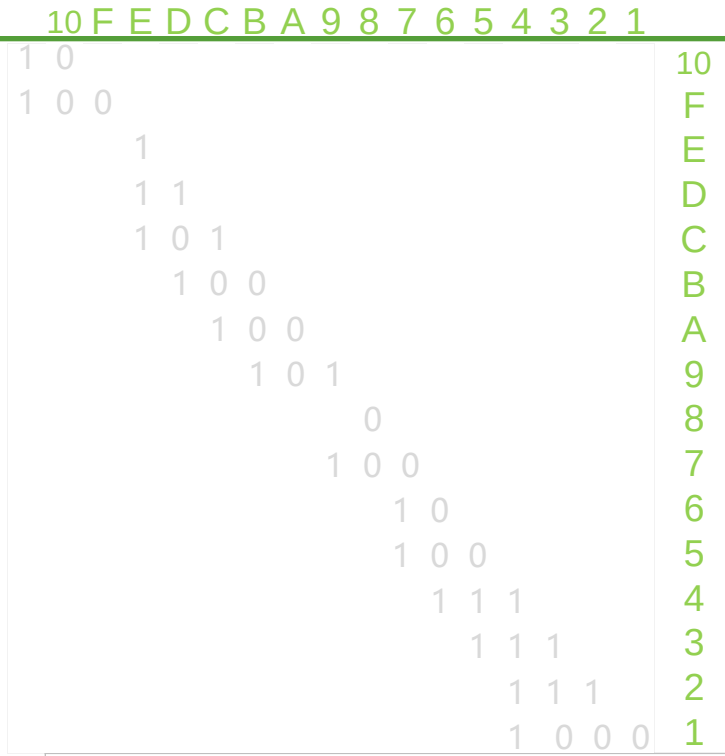
- 首部校验和的计算和校验



反码算术运算求和

10	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1
1	0	0	1	1	0	0	1	0	0	0	1	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	1
1	0	1	0	1	0	1	1	0	0	0	0	0	0	1	0
0	0	0	0	1	1	1	0	0	0	0	0	1	0	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	0	0	0	1	0	0	0	0	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	0	1	0	1	0	0	0	1	0	0	0	1	0	1
0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	0
0	1	0	0	1	0	0	1	0	1	0	0	1	1	1	1
0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0

10	F	E	D	C	B	A	9	8	7	6	5	4	3	2	1



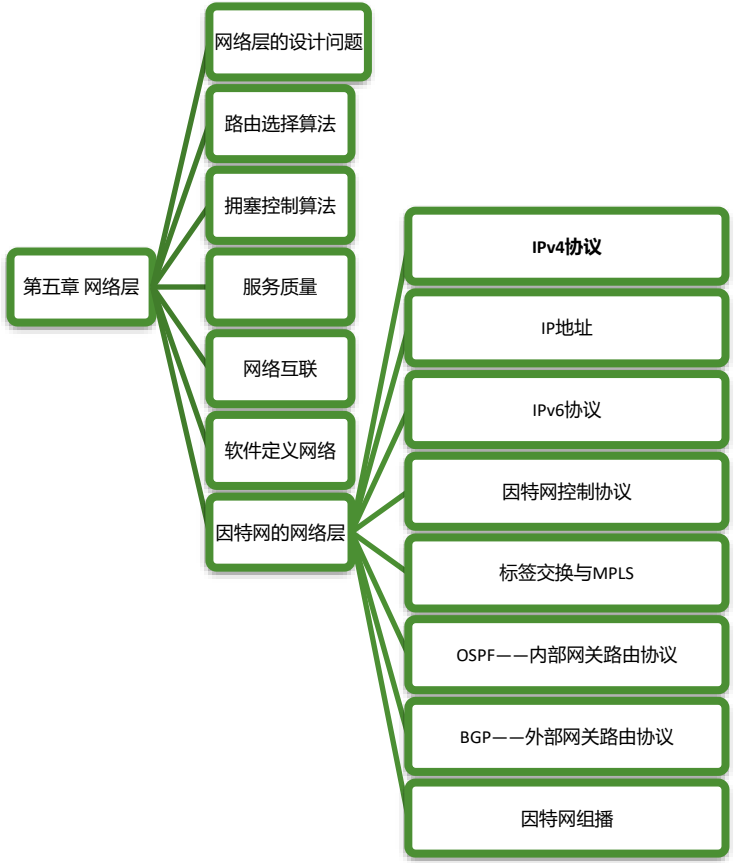
1	0	0	1	0	1	1	0	1	1	1	0	1	0	1	0	求和
1	0	0	1	0	1	1	0	1	1	1	0	1	1	0	0	进位
0	1	1	0	1	0	0	1	0	0	0	1	0	0	1	1	校验

IP协议选项

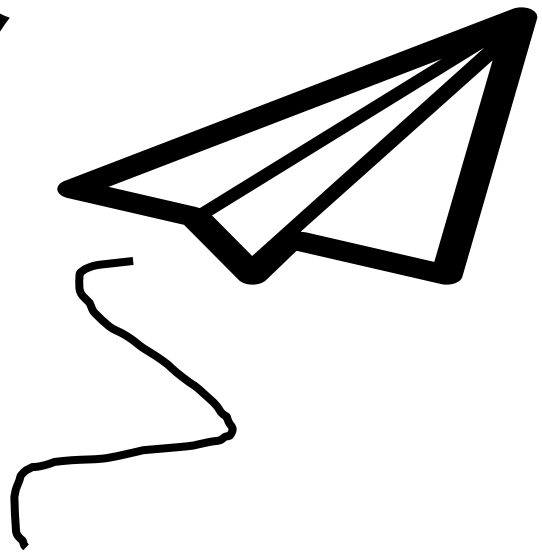
- IP选项

选项	描述
安全性	标明数据报的安全级别
严格源路由	给出必须遵循的完整路径
松散源路由	给出不能错过的路由器列表
记录路由	要求每个路由器附加上自己的IP地址
时间戳	要求每个路由器附加上自己的IP地址和时间戳

本章导航与要点


第五章 网络层
网络层的设计问题
路由选择算法
拥塞控制算法
服务质量
网络互联
软件定义网络
因特网的网络层
IPv4协议
IP地址
IPv6协议
因特网控制协议
标签交换与MPLS
OSPF——内部网关路由协议
BGP——外部网关路由协议
因特网组播

本节课程结束



IP地址

- 前缀
- 子网
- CIDR——无类域间路由选择
- 分类和特殊寻址
- NAT——网络地址转换

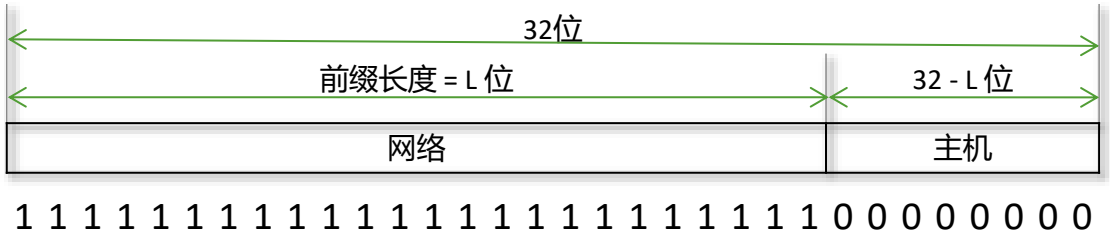
前缀

- IP 地址及其表示方法
  - 我们将整个因特网看成为一个单一的、抽象的网络。IP 地址就是给每个连接在因特网上的主机（或路由器）分配一个在全世界范围内唯一的 32 bit 的标识符
- IP地址的前缀和子网掩码
  - IP地址有层次性，由可变长网络和主机两部分构成，同一网络部分构成连续的IP地址空间，称为地址的前缀

IP地址 ::= {<网络号>，<主机号>}

::= 代表“定义为”

  - 子网掩码：用 /L 表示
    - 如右图：用 /24 或 255.255.255.0表示

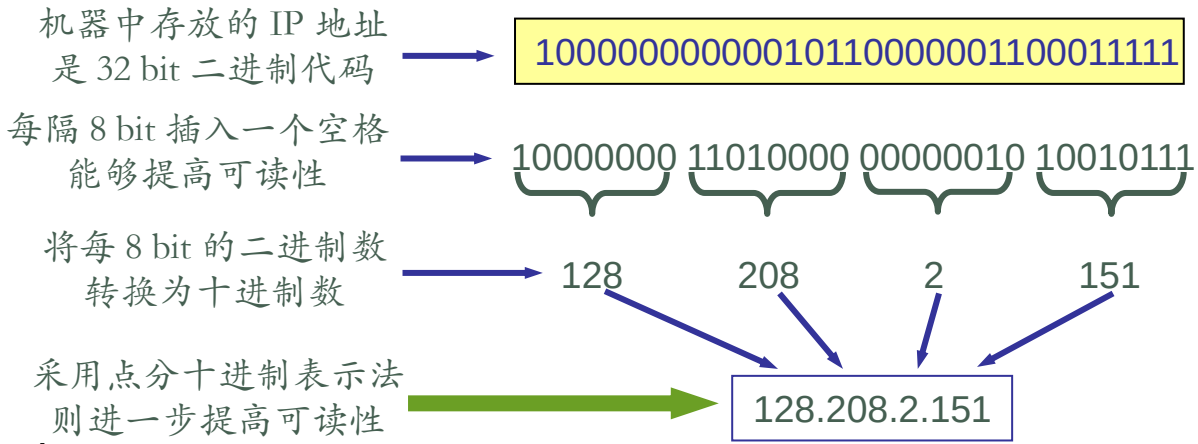


点分十进制表示法

- IPv4地址与书写方式
  - 二进制表示
  - 点分十进制表示
  - 十六进制表示

    80D00297**H**

    0x80D00297



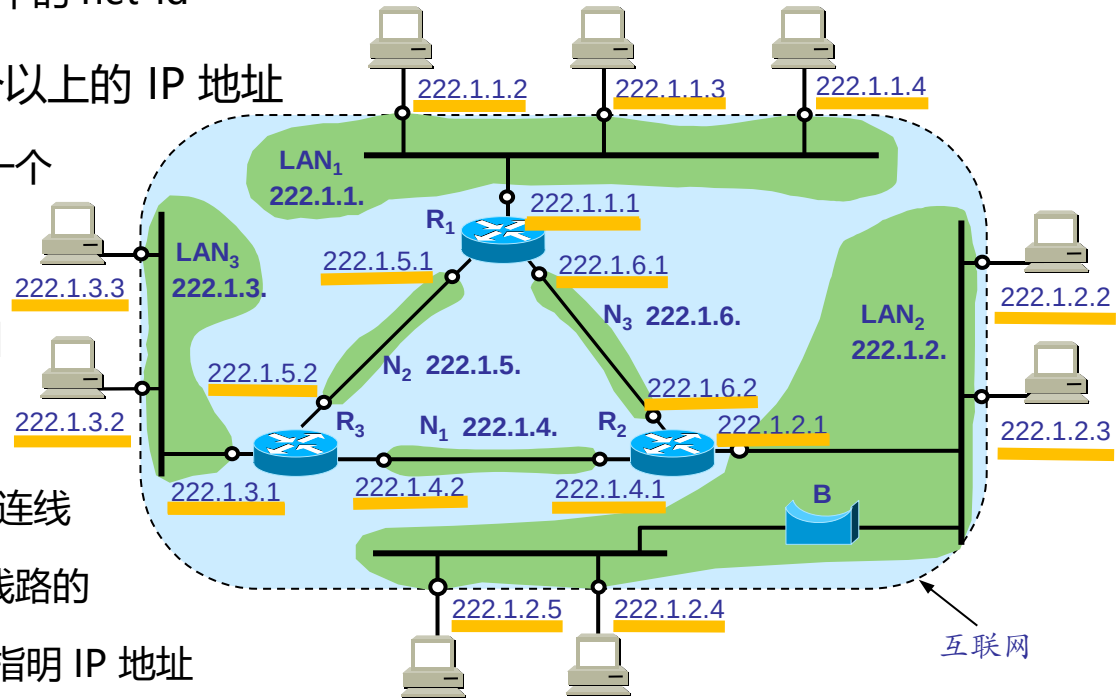
- 子网掩码 (subnet mask)
  - 可用于提取IP地址中的前缀，表达网络大小
  - 128.208.2.151/24 (255.255.255.0)，拥有 2^8 个地址，网络号可表示为128.208.2.0/24
  - 128. 208.2.151/16 (255.255.0.0) ，拥有 2^{16} 个地址，网络号可表示为128.208.0.0/16

IP 地址的一些重要特点

1. IP 地址是一种分等级的地址结构，分级的好处：
   - IP 地址管理机构只分配IP 地址网络号，主机号则由得到该网络号的单位自行分配。这样就方便了 IP 地址的管理
   - 路由器仅根据目的主机所连接的网络号来转发分组，这样就可以使路由表中的项目数大幅度减少，从而减小了路由表所占的存储空间
2. IP 地址是标志一个主机（或路由器）和一条链路的接口
   - 当一个主机同时连接到两个网络上时，该主机就必须同时具有两个相应的 IP 地址，其网络号 net-id 必须是不同的。这种主机称为多接口主机
   - 由于一个路由器至少应当连接到两个网络（这样它才能将 IP 分组从一个网络转发到另一个网络），因此一个路由器至少应当有两个不同的 IP 地址
3. 用集线器或网桥连接起来的若干局域网仍为一个网络，因此这些局域网都具有同样的网络号 net-id
4. 所有分配到网络号 net-id 的网络，无论范围很小的局域网，还是可能覆盖很大地理范围的广域网，都是平等的

互联网中的 IP 地址

- 在同一个局域网上的主机或路由器的 IP 地址中的网络号必须是一样的
  - 图中的网络号就是 IP 地址中的 net-id
- 路由器总是具有两个或两个以上的 IP 地址
  - 路由器的每一个接口都有一个不同网络号的 IP 地址
- 两个路由器直接相连的接口处，指不指明 IP 地址均可
  - 如指明 IP 地址，则这一段连线就构成了一种只包含一段线路的特殊“网络”。现在常不指明 IP 地址



IP 地址的编址方法

- 分类的 IP 地址
  - 这是最基本的编址方法，在 1981 年就通过了相应的标准协议
- 子网的划分
  - 这是对最基本的编址方法的改进，其标准[RFC 950]在 1985 年通过
- 无类域间路由
  - 这是比较新的无类编址方法。1993 年提出后很快就得到推广应用

路由器转发分组的步骤

- 先按所要找的 IP 地址中的网络号 net-id 把目的网络找到
- 当分组到达目的网络后，再利用主机号host-id 将分组直接交付给目的主机
- 按照整数字节划分 net-id 字段和 host-id 字段（分类）
  - 可以使路由器在收到一个分组时能够更快地将地址中的网络号提取出来

子网

- 子网由来
  - 因特网域名和地址分配机构 (ICANN)
    - IP 地址由 ICANN (Internet Corporation for Assigned Names and Numbers) 进行分配
  - 分配按区域分级进行
    - ICANN一次将部分地址空间授权给区域机构
    - 区域机构再将这些IP地址分配给ISP和公司
  - 网络的再次分级
    - 例如：为某大学分配了“/16”大小的网络

      大学扩大规模，增加了院系，校园网需要重新划分
    - 网络中划分子网

划分子网的基本思路

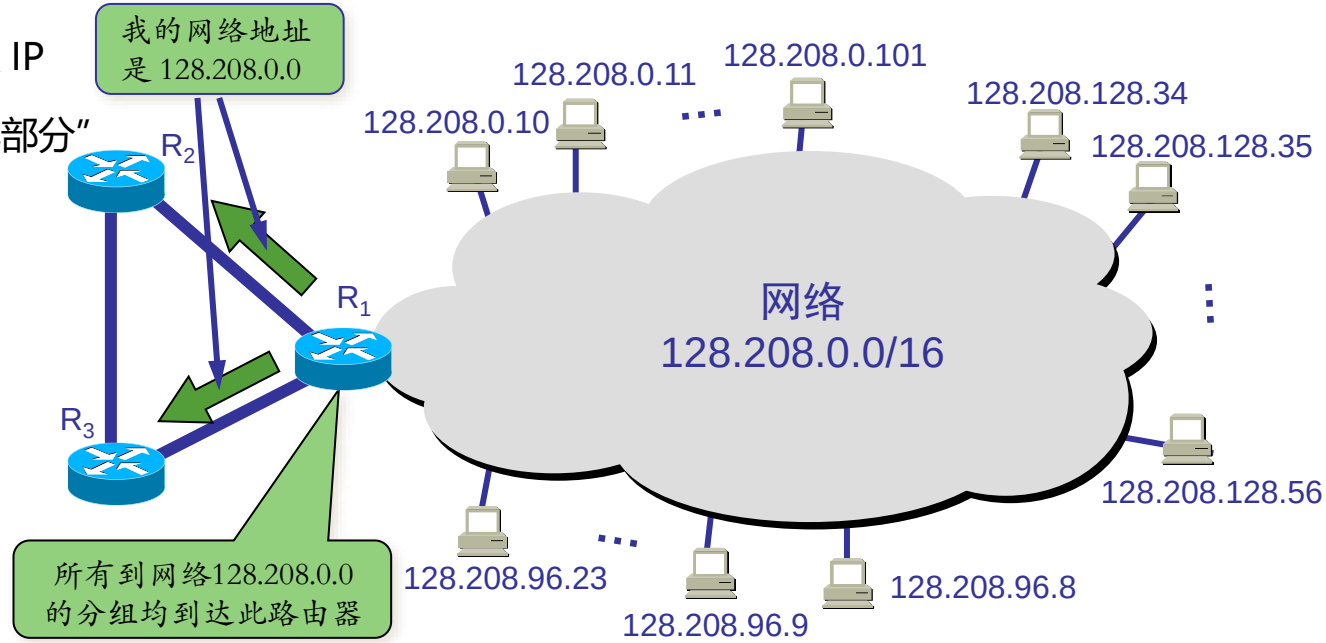
- 对外网络号不变，内部主机号减少
  - 划分子网纯属一个单位内部的事情。单位对外仍然表现为没有划分子网的网络
  - 从主机号借用若干个比特作为子网号 subnet-id，而主机号 host-id 也就相应减少了若干个比特

IP地址 ::= {<网络号>, <子网号>, <主机号>}

- 路由查找分两级
  - 凡是从其他网络发送给本单位某个主机的 IP 分组，仍然是根据 IP 分组的目的网络号 net-id，先找到连接在本单位网络上的路由器
  - 然后此路由器在收到 IP 分组后，再按目的网络号 net-id 和子网号 subnet-id 找到目的子网
  - 最后就将 IP 分组直接交付给目的主机

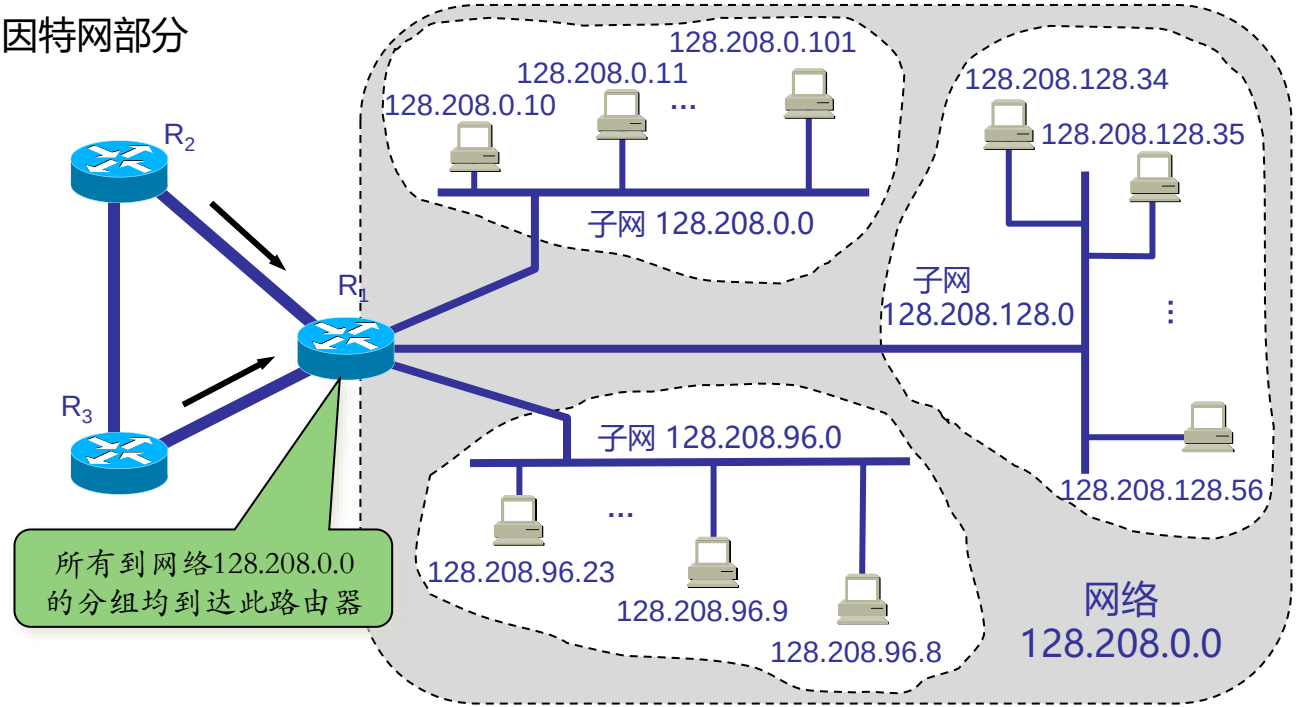
划分子网示例

- 一个未划分子网的网络 128.208.0.0/16
  - 当没有划分子网时，IP 地址是两级结构，地址的网络号字段也就是 IP 地址的“因特网部分”，主机号字段是 IP 地址的“本地部分”



划分子网示例 (2)

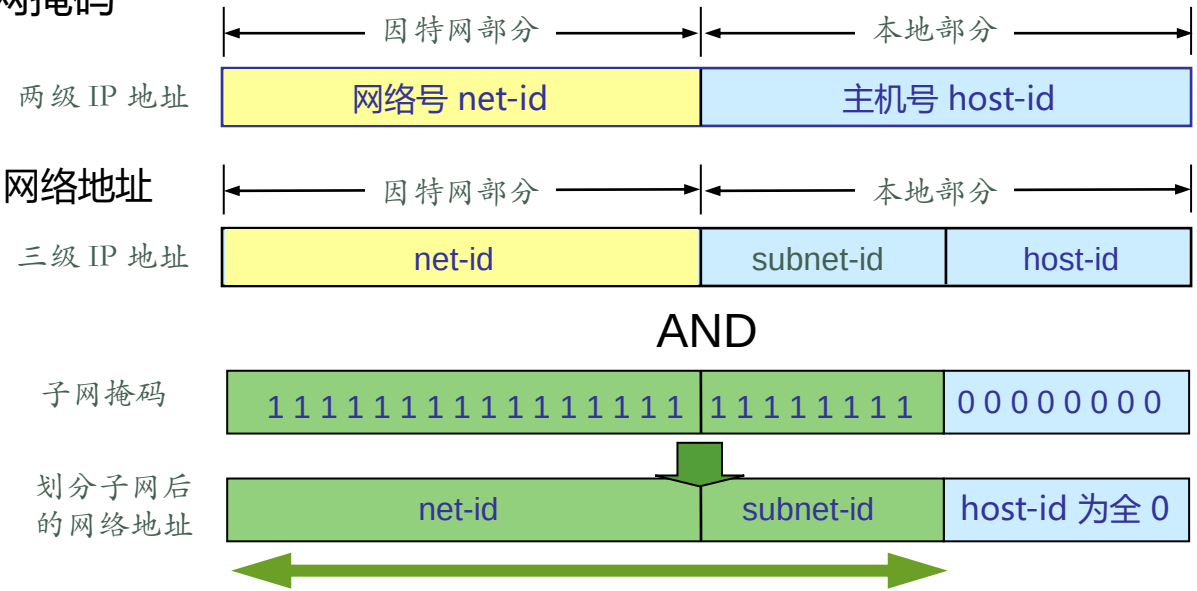
- 划分为三个子网后对外仍是一个网络
  - 划分子网后 IP 地址就变成了三级结构。划分子网只是将 IP 地址的本地部分进行再划分，而不改变 IP 地址的因特网部分



子网掩码

- 仅从一个IP分组的首部，无法判断源主机或目的主机所连接的网络是否划分了子网
  - 使用子网掩码可以找出 IP 地址中的子网部分
- IP 地址的前缀和子网掩码
- 网络地址的计算

(IP地址) AND (子网掩码) = 网络地址



使用子网掩码的分组转发过程

- 分类的IP地址，不划分子网
  - 分类IP地址中隐含子网掩码信息，在不划分子网的两级 IP 地址下，由 IP 地址很容易得出网络地址
- 划分子网
  - 在划分子网的情况下，网络地址取决于那个网络所采用的子网掩码，但分组的首部并没有提供子网掩码信息
  - 故分组转发的算法也必须做相应的改动
    - 路由器的路由表中必须提供子网掩码

- 路由器分组转发的算法
  1. 从收到分组首部提取目的IP地址D
  2. D先和直连网络的子网掩码相“与”。若匹配，则将分组直接交付；否则间接交付，转3
  3. D和间接网络的路由表项匹配。相符则将分组传送给表项中的下一跳路由器；否则转4
  4. 若路由表中有一个默认路由，则将分组传送给默认路由；否则转5
  5. 报告转发分组出错

划分子网后分组的转发举例

- 已知R1的路由表，主机H1发送分组，目的IP地址为：128.30.33.138
  - H1并不知道H2处于哪个网络，因此 H1 首先检查目的主机 128.30.33.138 是否在本网络上

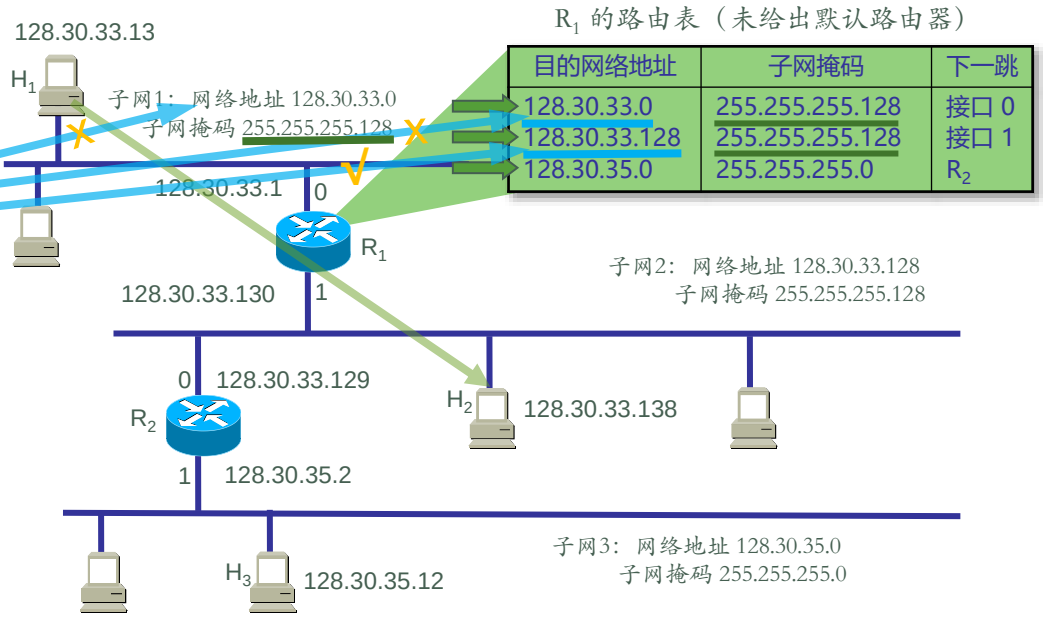
	128. 30. 33.1	0001010
AND	255.255.255.1	0000000
	128. 30. 33.1	0000000
=	128.30.33.128	

  - 如果是，直接交付；

    否则送交路由器 R1
  - R1收到分组，目的IP地址为：

    128.30.33.138

    逐项查找路由表并匹配计算

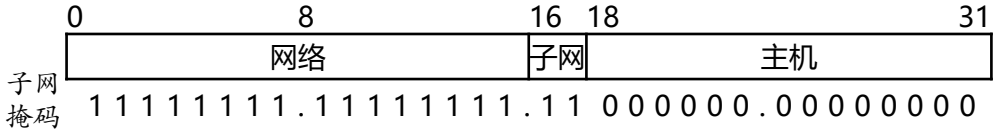


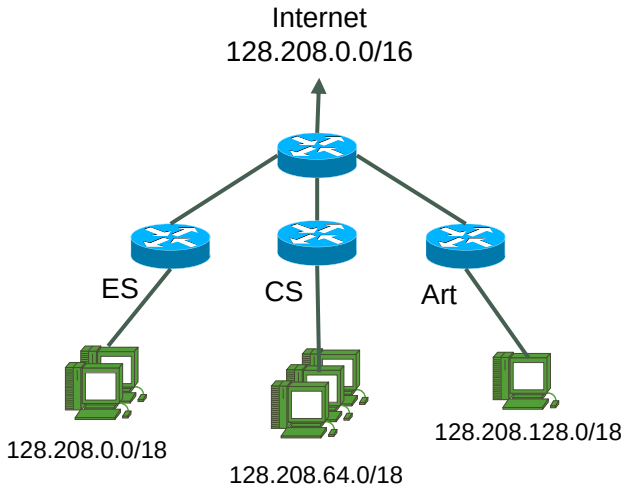
R_1 的路由表（未给出默认路由器）

目的网络地址	子网掩码	下一跳
128.30.33.0	255.255.255.128	接口 0
128.30.33.128	255.255.255.128	接口 1
128.30.35.0	255.255.255.0	R_2

子网划分

- 例1：128.208.0.0/16 的地址空间划分为三片
  - 简单划分：均匀分为四份
  - 原来 16 bit 前缀不变
  - 从16 bit 主机号中分出 2 bit 作为子网号
  - 对外网络号：128.208.0.0/16
  - 内部子网号与子网掩码
    - 电子工程系：1000000 0.11010000. 00 xxxxxx.xxxxxxxx
    - 计算机系：1000000 0.11010000. 01 xxxxxx.xxxxxxxx
    - 艺术系：1000000 0.11010000. 10 xxxxxx.xxxxxxxx
    - 保留：128.208.192.0，255.255.192.0





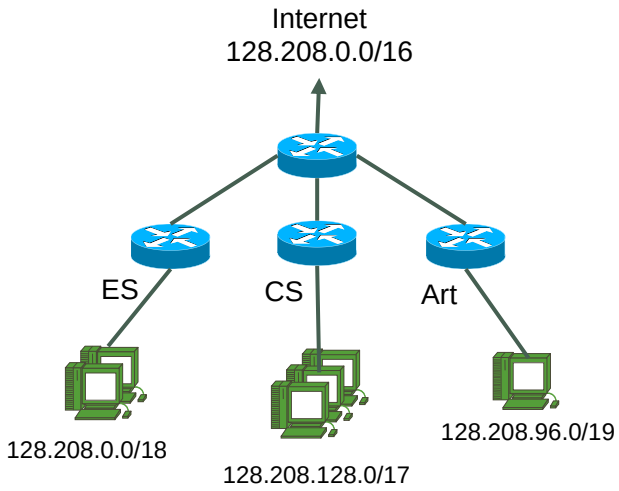
子网划分 (2)

- 例2：按实际需要划分 128.208.0.0/16 的地址空间
  - 三个系需要的地址空间分别约为：10000、20000、5000
  - 计算主机号需要的二进制位数为： 14、15、13 (2^{12}=4096、2^{13}=8192 、2^{14}=16384、2^{15}=32768)
  - /16 地址空间的四分之一 (/18)分给电子工程、一半(/17)分给计算机、八分之一(/19)分给艺术
  - 内部子网号与子网掩码
    - 电子工程系：10000000.11010000.00|xxxxxx.xxxxxxxx
    - 计算机系： 10000000.11010000.1|xxxxxxx.xxxxxxxx
    - 艺术系： 10000000.11010000.011|xxxxx.xxxxxxxx
    - 保留： 10000000.11010000.010|xxxxxx.xxxxxxxx
    - 地址范围与子网掩码

ES：128.208.0.0~128.208.63.255， 子网掩码255.255.192.0

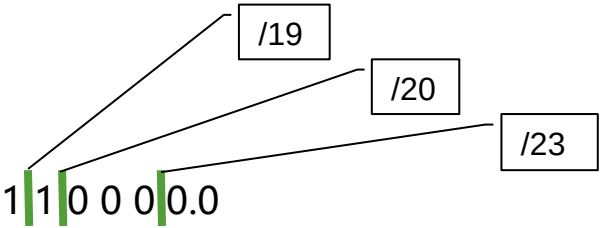
CS：128.208.128.0~128.208.255.255， 子网掩码255.255.128.0

Art：128.208.96.0~128.208.127.255， 子网掩码255.255.224.0



子网划分 (3)

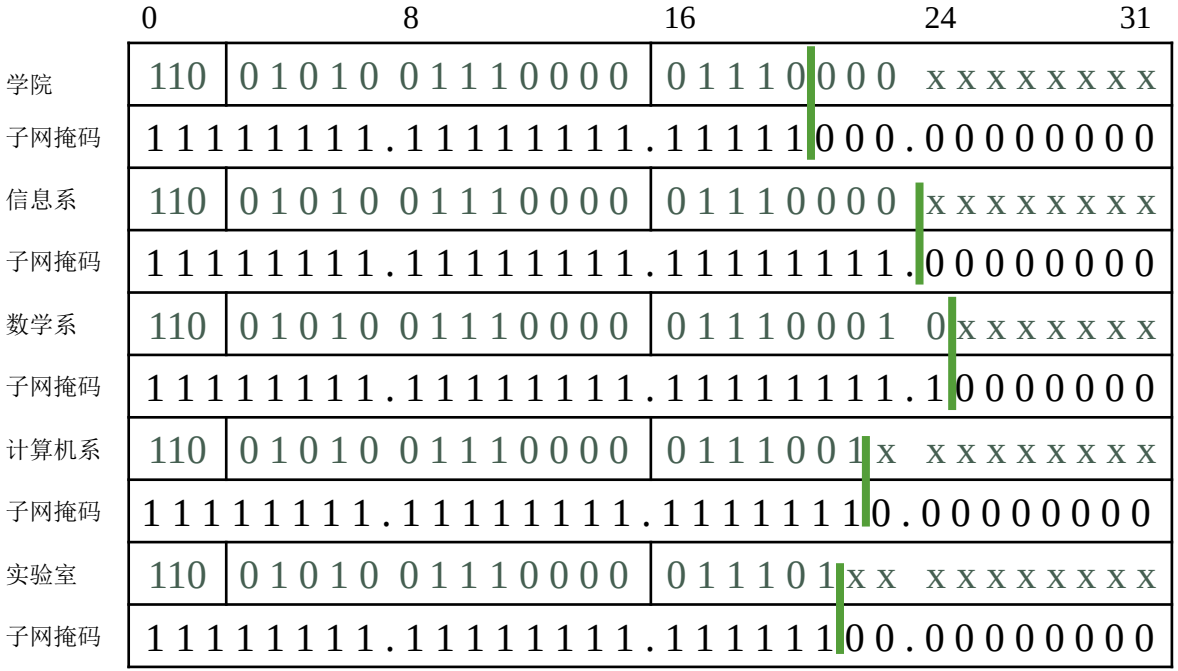
- 例3：中国人民大学早期分到的地址空间为202.112.112.0 /x
  - 最大整块空间可为多少?
  - 共8个学院，如何分配?
- 解答：
  - 202.112.112.0/x，/x最大为多少?
  - 8个学院



	0	8	16	24	31	
	110	01010 01110000	0111 0000	00000000		
子网掩码	11111111.11111111.1111 0000.00000000					255.255.240.0
8个学院	110	01010 01110000	0111nnnx	xxxxxxxx		
子网掩码	11111111.11111111.11111110.00000000					255.255.254.0

子网划分 (4)

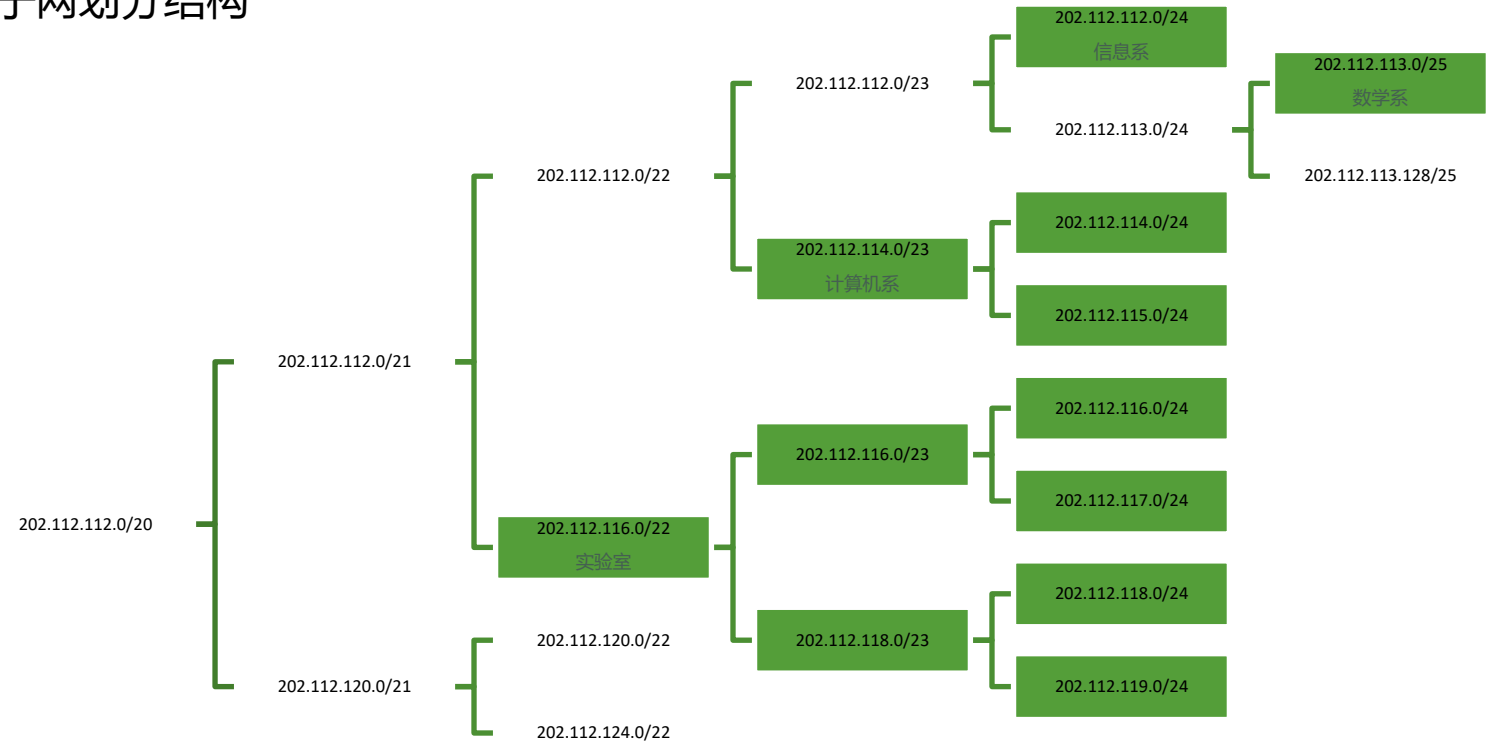
- 例4：信息学院分到的地址空间为202.112.112.0 /21
  - 信息系：200；数学系：100；计算机：500；实验室：1000



202.112.112.1~202.112.112.254
255.255.255.0，/24
202.112.113.1~202.112.113.126
255.255.255.128，/25
202.112.114.1~202.112.115.254
255.255.254.0，/23
202.112.116.1~202.112.119.254
255.255.252.0，/22

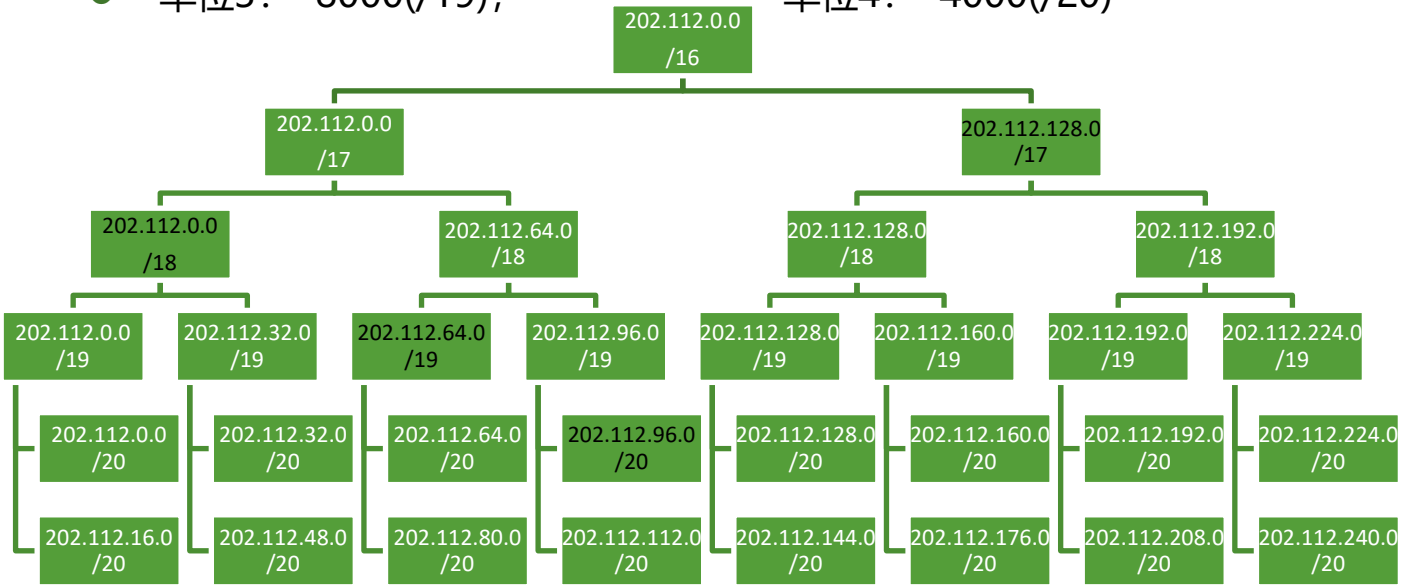
子网划分 (5)

- 子网划分结构



子网划分 (6)

- 子网划分结构：202.112.0.0/16
  - 单位1：16000(/18)；
  - 单位2：32000(/17)；
  - 单位3： 8000(/19)；
  - 单位4： 4000(/20)



CIDR——无类域间路由

- 网络数
  - 一个组织：子网路由表项，外网缺省路由
  - ISP与骨干网：网络数目100万以上，路由表爆炸
- CIDR（Classless Inter-Domain Routing）
  - 路由聚合：一个IP地址，可以对应于（/22）网络路由，也可以聚合为（/20）网络路由
  - 地址分块
    - 194.0.0.0~195.255.255.255：欧洲 (33,554,432个地址)
      - 194.0.0.xx~195.255.255.xx : 131,072
    - 198.0.0.0~199.255.255.255：北美
    - 200.0.0.0~201.255.255.255：中美和南美
    - 202.0.0.0~203.255.255.255：亚洲和太平洋地区
    - 204.0.0.0~223.255.255.255：保留 (335,544,320个地址)

路由聚合(route aggregation)

- 一个 CIDR 地址块可以表示很多地址，这种地址的聚合常称为路由聚合，它使得路由表中的一个项目可以表示很多个（例如上千个）原来传统分类地址的路由
- CIDR 虽然不使用子网了，但仍然使用“掩码”这一名词（但不叫子网掩码）

无类域间路由实例

- 例1：用IP地址按可变大小块方式分配，194.24.0.0/19，8192个地址
  - Cambridge: 2048，/21，255.255.248.0
  - Oxford: 4096，/20，255.255.240.0
  - Edinburgh: 1024，/22，255.255.252.0

①
194.24.00000 xxx.x/21

194.24.00001 000.0 ③ 194.24.000010 xx.x/22

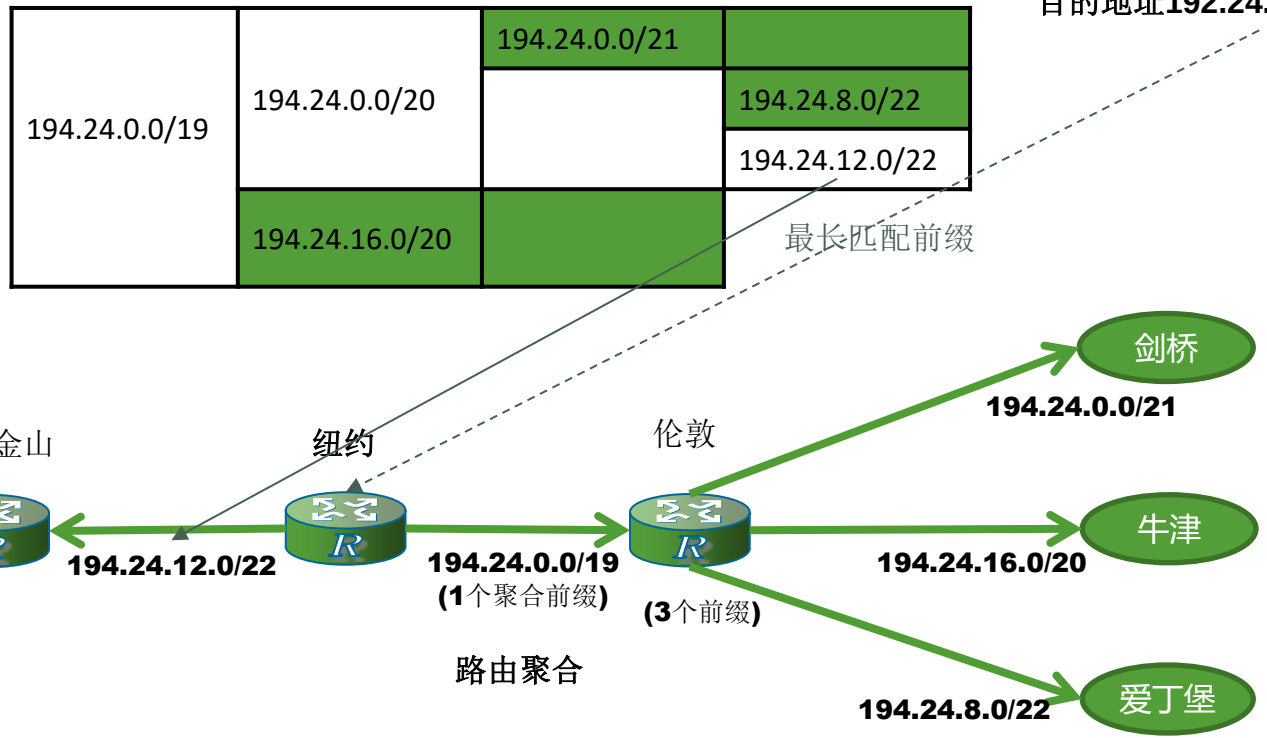
② 194.24.0001 xxxx.x/20

	University	First address	Last address	How many	Written as
①	Cambridge	194.24.0.0	194.24.7.255	2048	194.24.0.0/21
③	Edinburgh	194.24.8.0	194.24.11.255	1024	194.24.8.0/22
	(Available)	194.24.12.0	194.24.15.255	1024	194.24.12.0/22
②	Oxford	194.24.16.0	194.24.31.255	4096	194.24.16.0/20

无类域间路由实例(2)

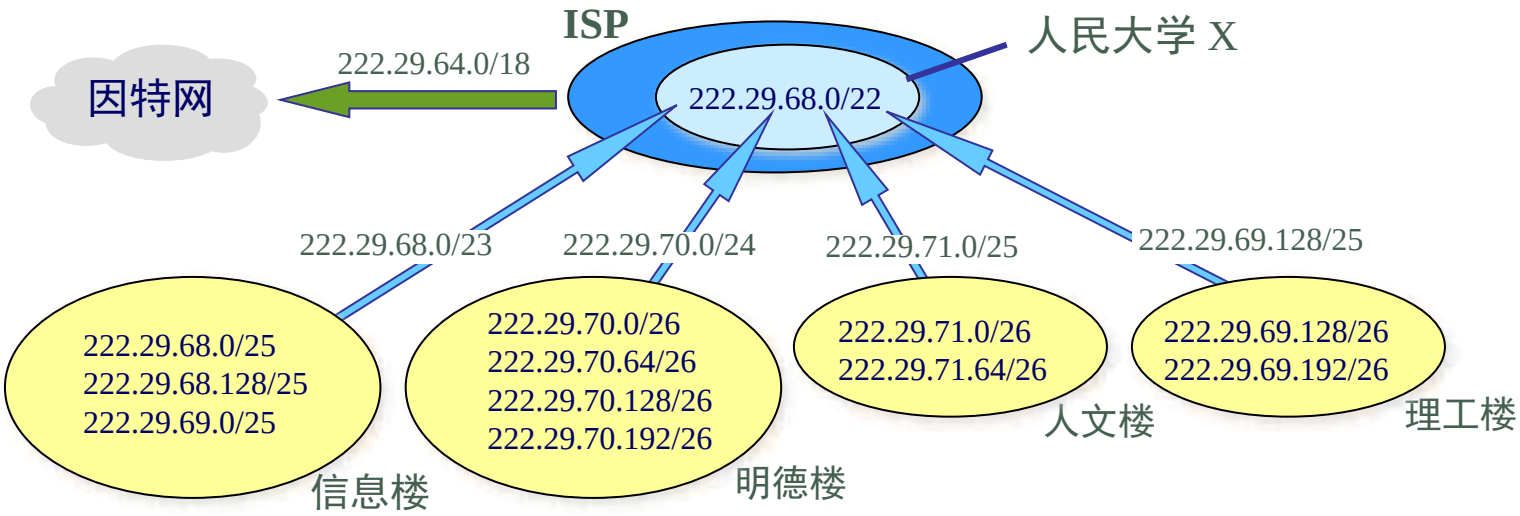
- 194.24.0.0 ~ 194.24.31.255/19

思考题：
目的地址192.24.15.19的路由？



无类域间路由实例(3)

- 例2



这个 ISP 共有 64 个 C 类网络。
如果不采用 CIDR 技术，则在与该 ISP 的路由器交换路由信息的每一个路由器的路由表中，就需要有 64 个项目。
但采用地址聚合后，只需用路由聚合后的 1 个项目 222.29.64.0/18 就能找到该 ISP。

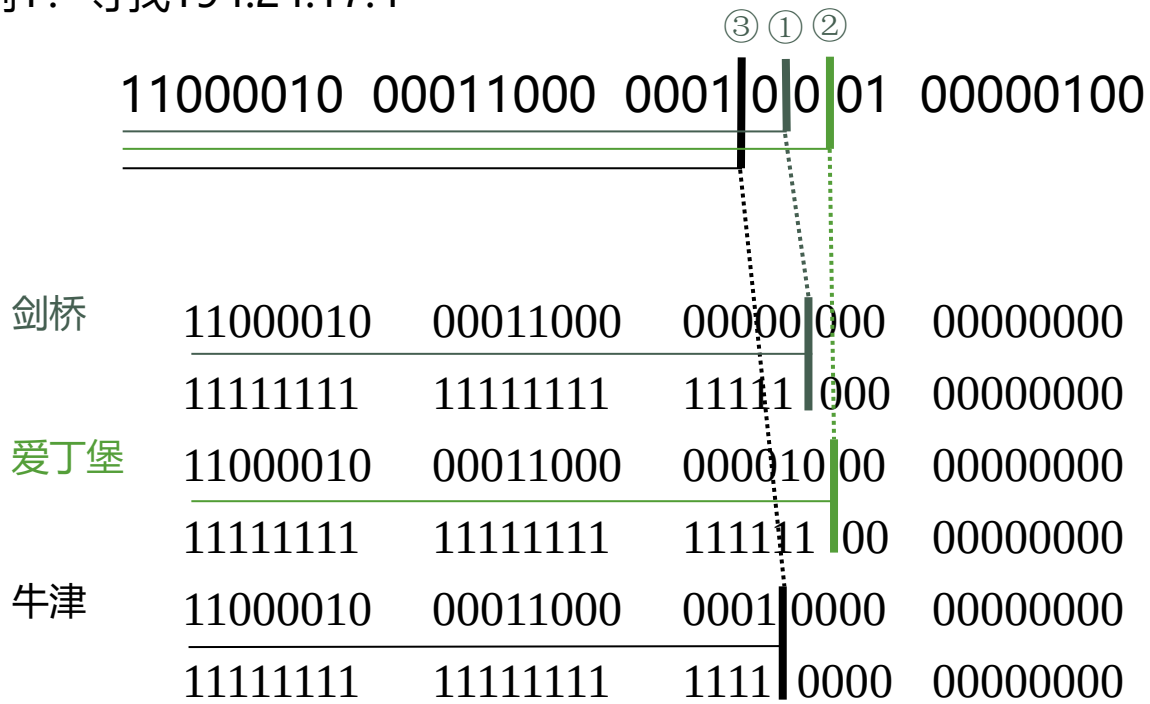
单位	地址块	二进制表示	地址数
ISP	222.29.64.0/18	11001010.00011101.01*	16384
人民大学	222.29.68.0/22	11001010.00011101.010001*	1024
信息楼	222.29.68.0/23	11001010.00011101.0100010*	512
明德楼	222.29.70.0/24	11001010.00011101.01000110.*	256
人文楼	222.29.71.0/25	11001010.00011101.01000111.0*	128
理工楼	222.29.69.128/25	11001010.00011101.01000101.1*	128

最长前缀匹配

- 使用 CIDR 时，路由表中的每个项目由“网络前缀”和“下一跳地址”组成。在查找路由表时可能会得到不止一个匹配结果
- 应当从匹配结果中选择具有最长网络前缀的路由：最长前缀匹配 (longest-prefix matching)
- 网络前缀越长，其地址块就越小，因而路由就越具体
- 最长前缀匹配又称为最长匹配或最佳匹配

最长前缀匹配

- 例1：寻找194.24.17.4



最长前缀匹配(2)

- 例2：收到的分组的目的地址 D = 222.29.69.129
  - 路由表中的项目：222.29.68.0/23　　(信息楼)

    222.29.69.128/25　　(理工楼)
  - 第1项

		222.	29.	01000100	.0
	M =	11111111	11111111	11111110	00000000
AND	*D* =	222.	29.	01000101	.129
		222.	29.	010001	匹配

  - 第2项

		222.	29.	69.10000000	
	M =	11111111	11111111	11111111	10000000
AND	*D* =	222.	29.	69.10000001	
		222.	29.	69.1	匹配

最长前缀匹配(3)

- 路由选择：

  D AND (11111111 11111111 11111110 00000000)

  = 222.29.69.0/23　　匹配

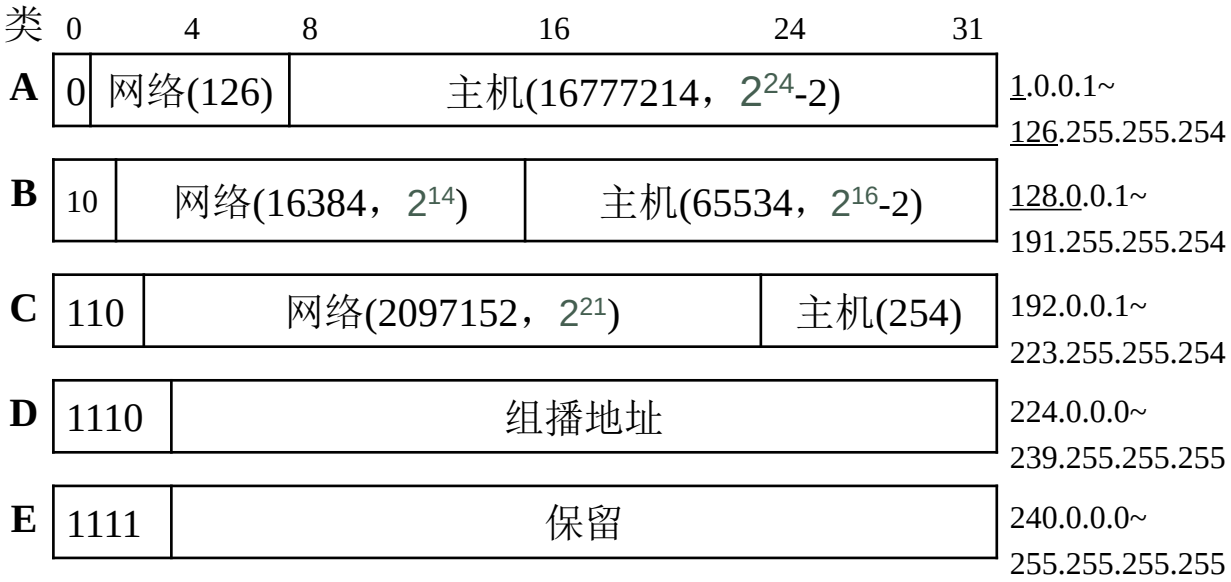
  D AND (11111111 11111111 11111111 10000000)

  = 222.29.69.128/25　　匹配

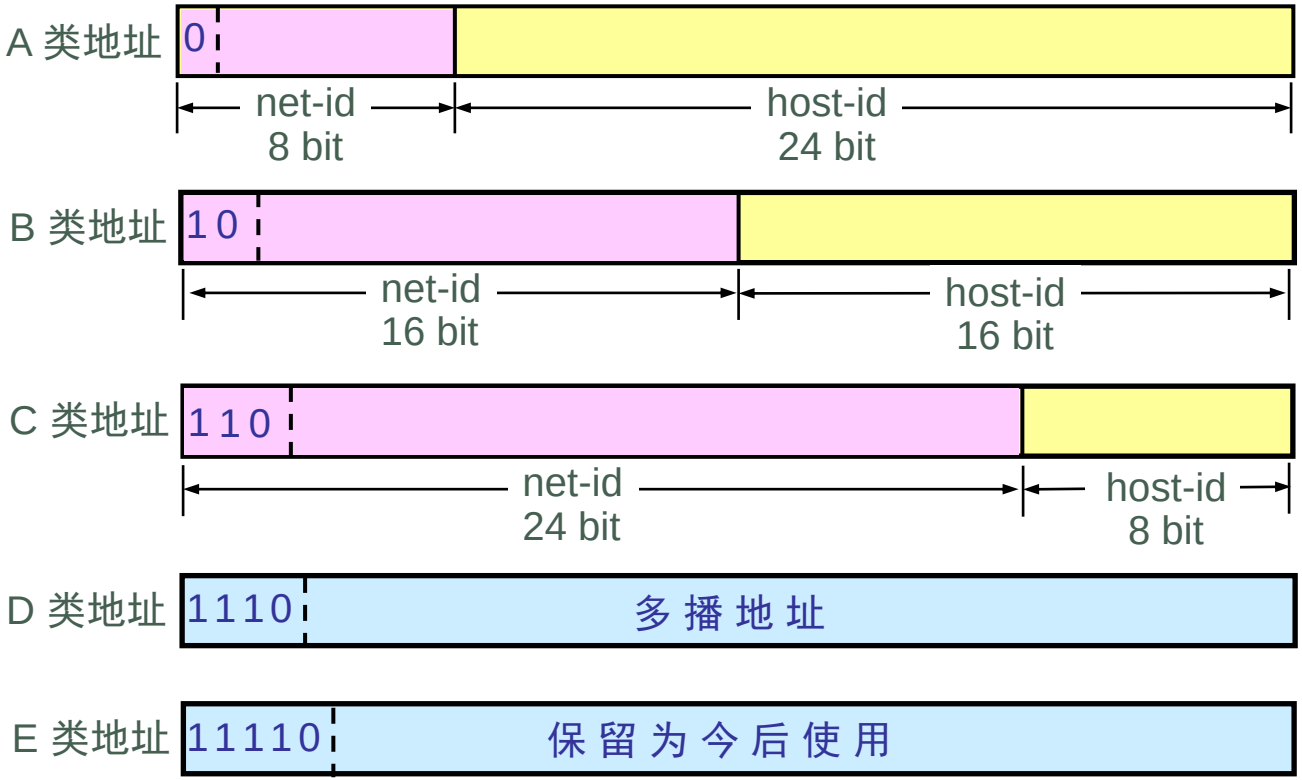
  - 选择两个匹配的地址中更具体的一个，即选择最长前缀的地址

分类和特殊寻址

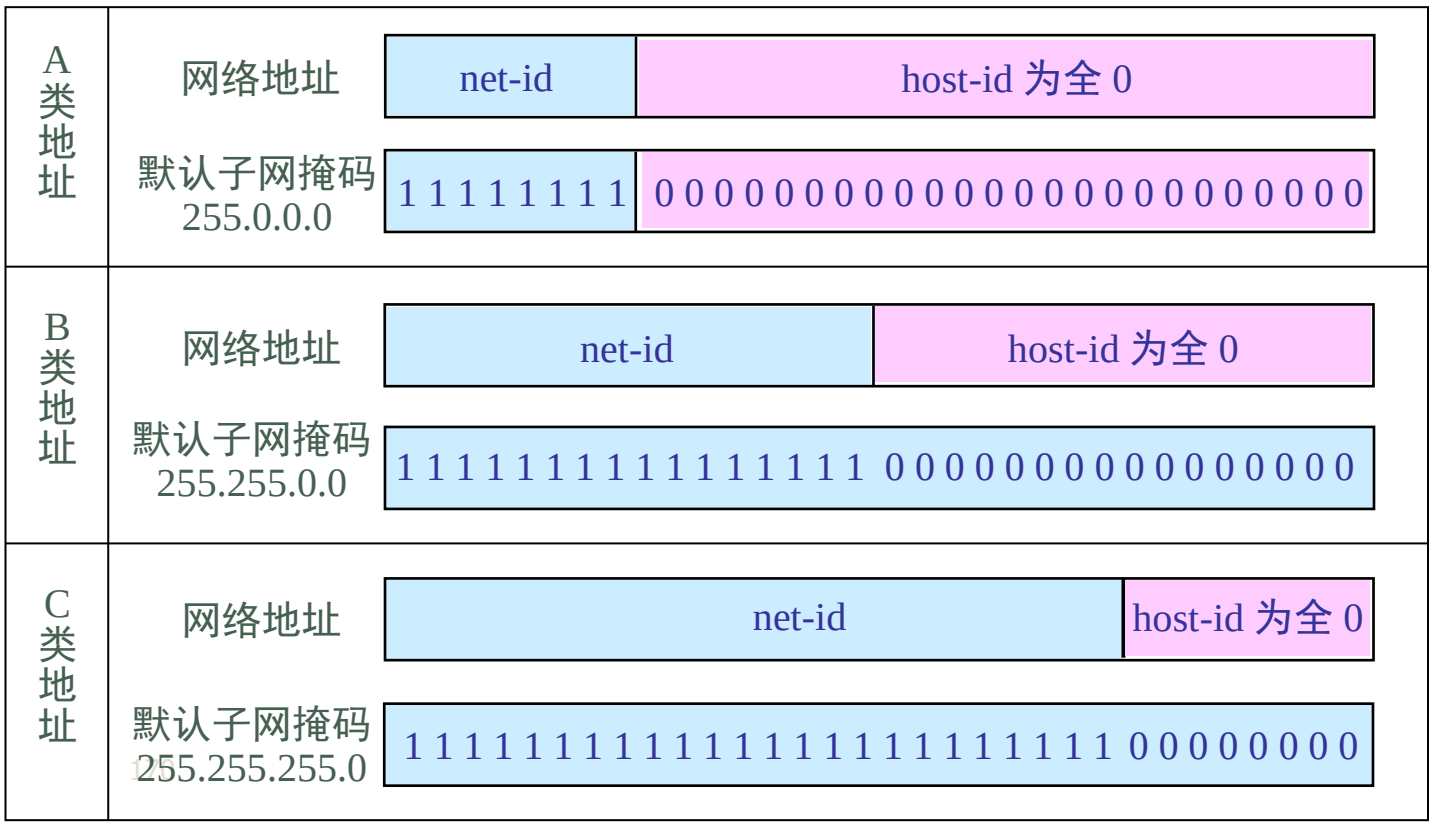
- 1993年以前，IP地址分为5类（**分类寻址)**



IP 地址中的网络号字段和主机号字段

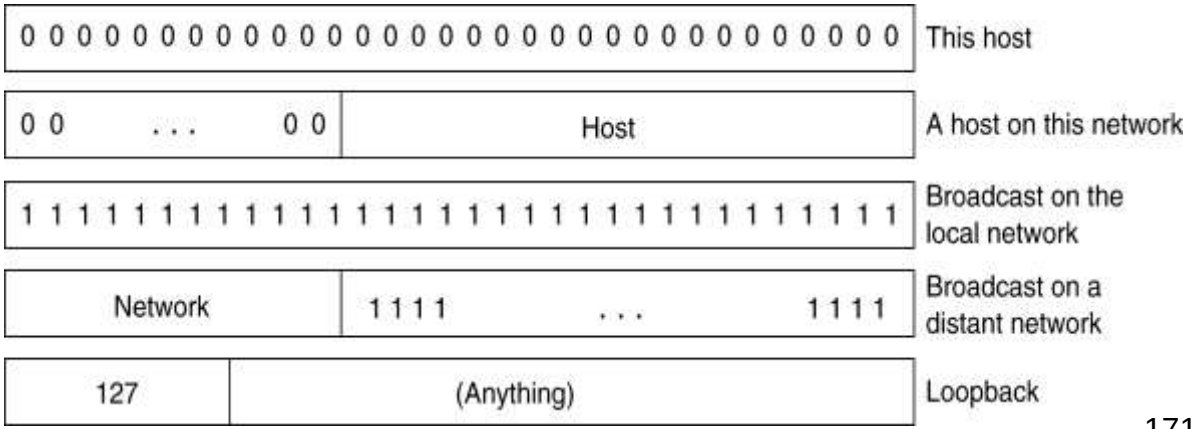


A 类、B 类和 C 类 IP 地址的默认子网掩码



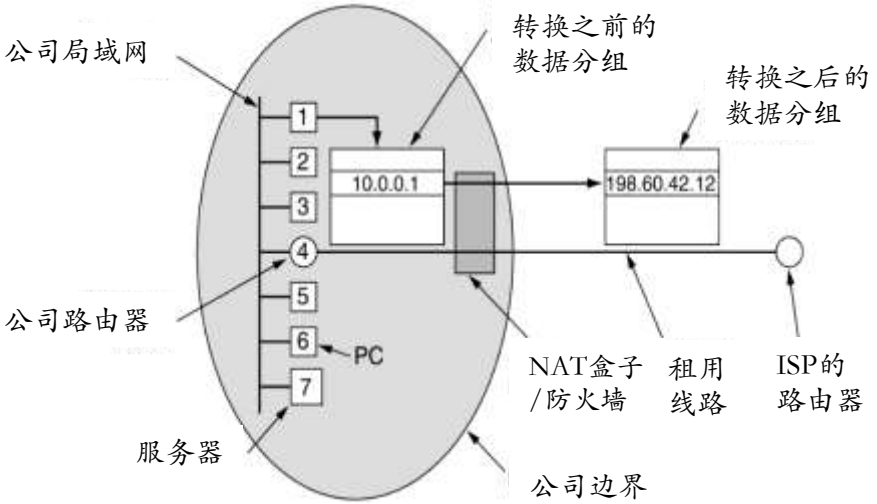
特殊地址

- 特殊IP地址
  - 0.0.0.0：　　本机
  - 00…00 主机：　　本网中的主机
  - 255.255.255.255：　　局域网中的广播
  - 网络 11…11：　　对一个远程网的广播
  - 127.x.y.z：回路
- 私有地址：
  - 10.x.x.x
  - 172.16~31.x.x
  - 192.168.x.x

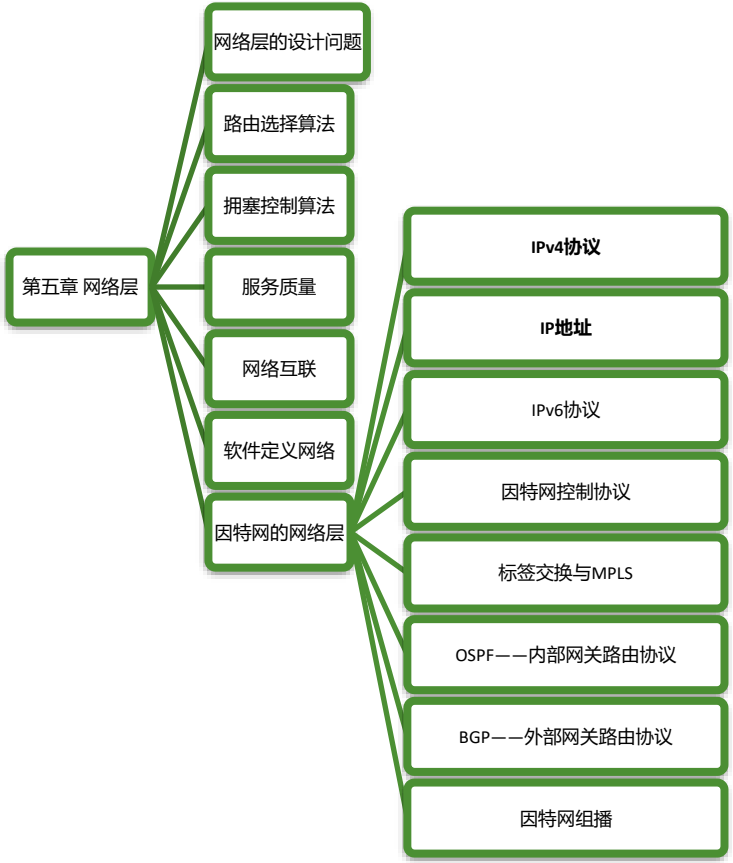


NAT——网络地址转换

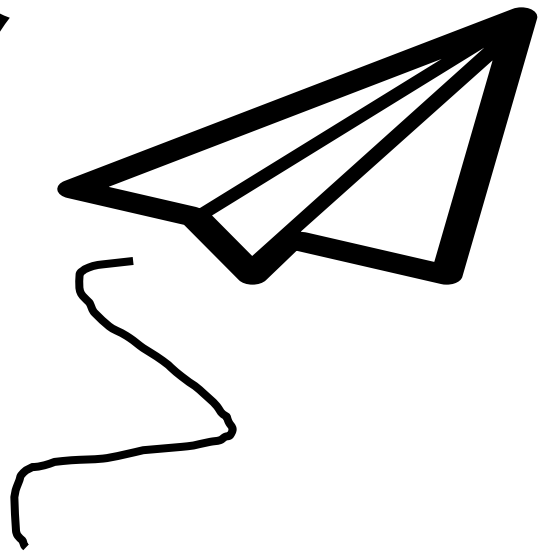
- 私有地址
  - 10.0.0.0~10.255.255.255/8 (16,777,214，2^{24}-2)
  - 172.16.0.0~172.31.255.255/12 (1,048,574，2^{20}-2)
  - 192.168.0.0~192.168.255.255/16 (65,534，2^{16}-2)



本章导航与要点


第五章 网络层
网络层的设计问题
路由选择算法
拥塞控制算法
服务质量
网络互联
软件定义网络
因特网的网络层
IPv4协议
IP地址
IPv6协议
因特网控制协议
标签交换与MPLS
OSPF——内部网关路由协议
BGP——外部网关路由协议
因特网组播

本节课程结束

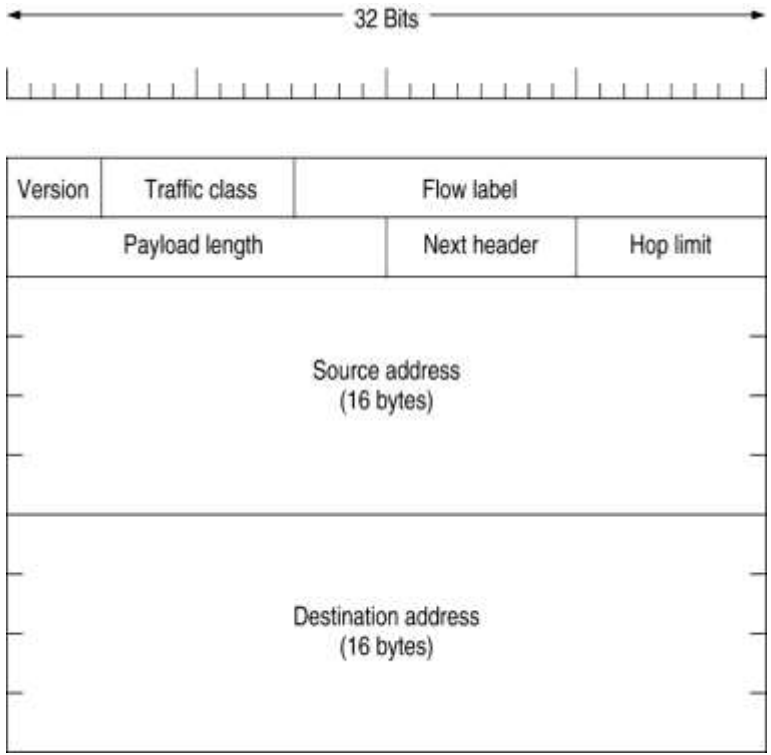


IPv6协议

- IP新版本的目标
  - 远超几十亿台主机(IPv4地址4,294,967,296)
  - 降低路由表大小
  - 简化协议，加快路由器处理速度
  - 安全性：身份认证与隐私权
  - 关注服务类型，特别是实时数据
  - 组播、漫游(无须改变地址)
  - 协议演进、新老协议共存

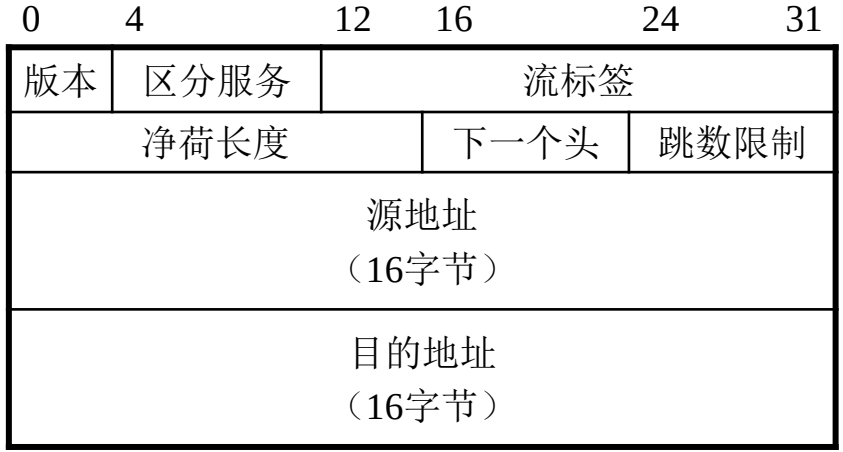
IPv6头部

- IPv5
  - 用于试验性实时数据流协议
- IPv6的改进
  - 16字节地址
  - 头部简化：由IPv4的13个字段变为7个
  - 选项处理：必需字段变为选项提高效率
  - 安全性：身份认证与隐私权



主头部格式

- 区分服务：最初为流量类型，使用方式同IPv4，最低两位用于拥塞
- 流标签：流由源、目的地址和流编号制定
- 净荷长度：头部40字节除外
- 下一个头：可选扩展头
- 跳数限制：与TimeToLive相同
- 地址表示
  - 8000:0000:0000:0000:0123:4567:89AB:CDEF
  - 8000::123:4567:89AB:CDEF
  - IPv4 ::192.31.20.46
  - 地址数目：2^128，3*10^38，每平方米7*10^25



扩展头部

- IPv6扩展头

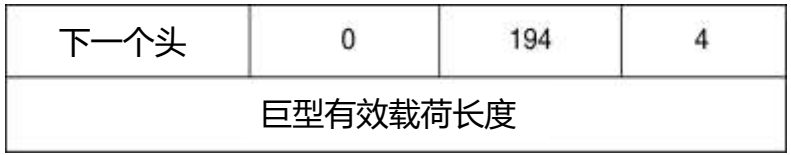
扩展头	描述
逐跳选项	路由器的混杂信息
目标选项	给目的地的额外信息
路由	必须访问的松散路由器列表
分段	管理分组分段
认证	验证发送方身份
加密安全净荷	有关加密内容信息

- 三元组：
  - <Type，Length，Value>
    - 类型(1B)：什么选项；长度(1B)：值字段长度；值(<=255B)：扩展头所需信息
  - 逐跳头
  - 路由扩展头

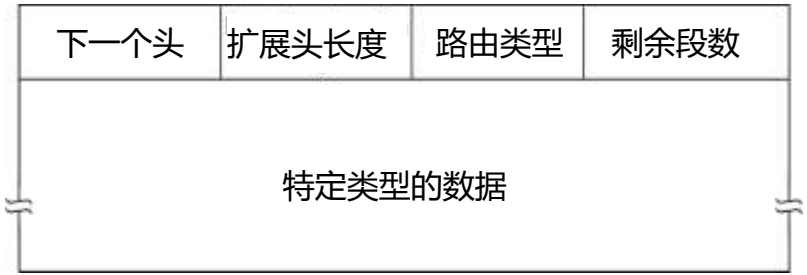
下一个头	0	194	
	扩展头长度	路由类型	剩余段数
分组长度（4字节为单位）			

扩展头部(2)

- 逐跳头



- 路由扩展头



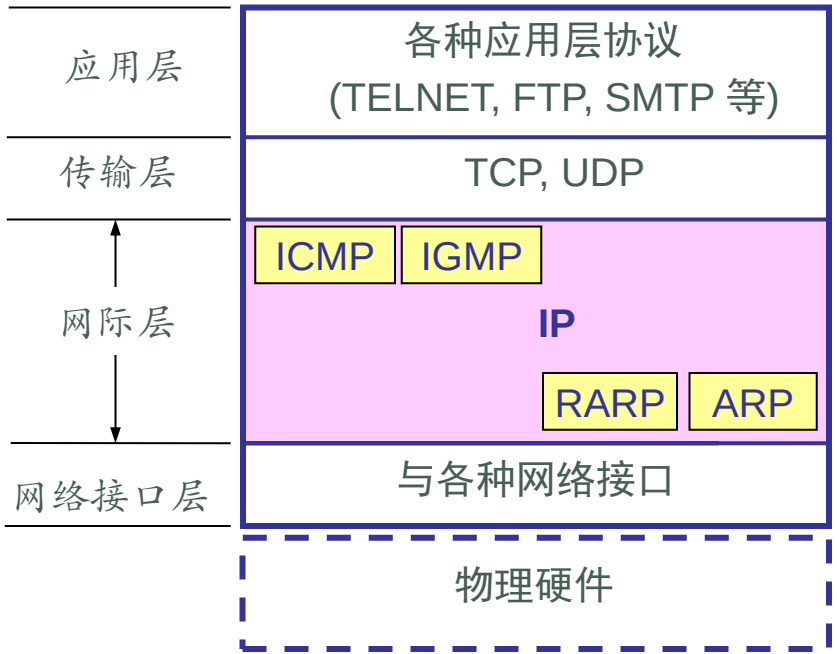
因特网控制协议

- ICMP因特网控制报文协议
- ARP地址解析协议
- DHCP动态主机配置协议

与 IP 协议配套使用的四个协议

- 地址解析协议 ARP
  (Address Resolution Protocol)
- 逆地址解析协议 RARP
  (Reverse Address Resolution Protocol)
- 因特网控制报文协议 ICMP
  (Internet Control Message Protocol)
- 因特网组管理协议 IGMP
  (Internet Group Management Protocol)

网际协议 IP 及其配套协议

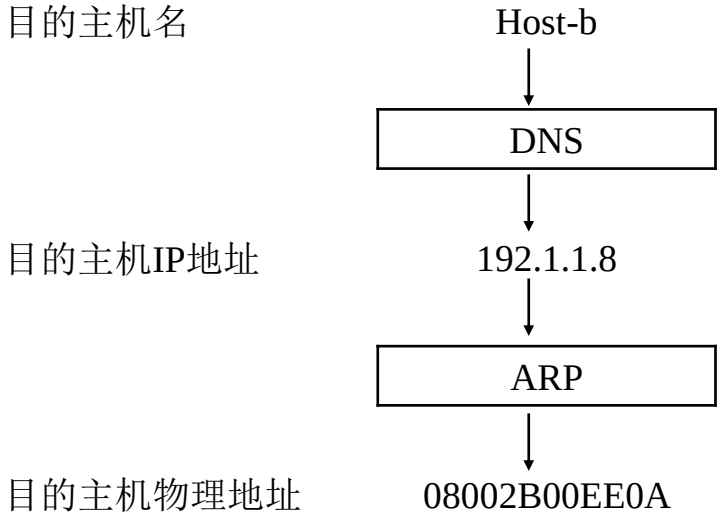


ICMP因特网控制报文协议

- ICMP (The Internet Control Message Protocol)
  - 可将路由器意外等事件报告给源端
    - 主要用来报告出错和测试，ICMP报文封装在IP包中
  - 重要消息类型

消息类型	描述
目的地不可达	数据分组无法传递
超时	TTL字段减为0
参数问题	无效的头字段
源抑制	抑制分组
重定向	告知路由器有关位置信息
回显与回显应答	检查机器是否活着
请求/应答时间戳	与回显相同，再加时间戳
路由器通告/恳求	发现邻居路由器

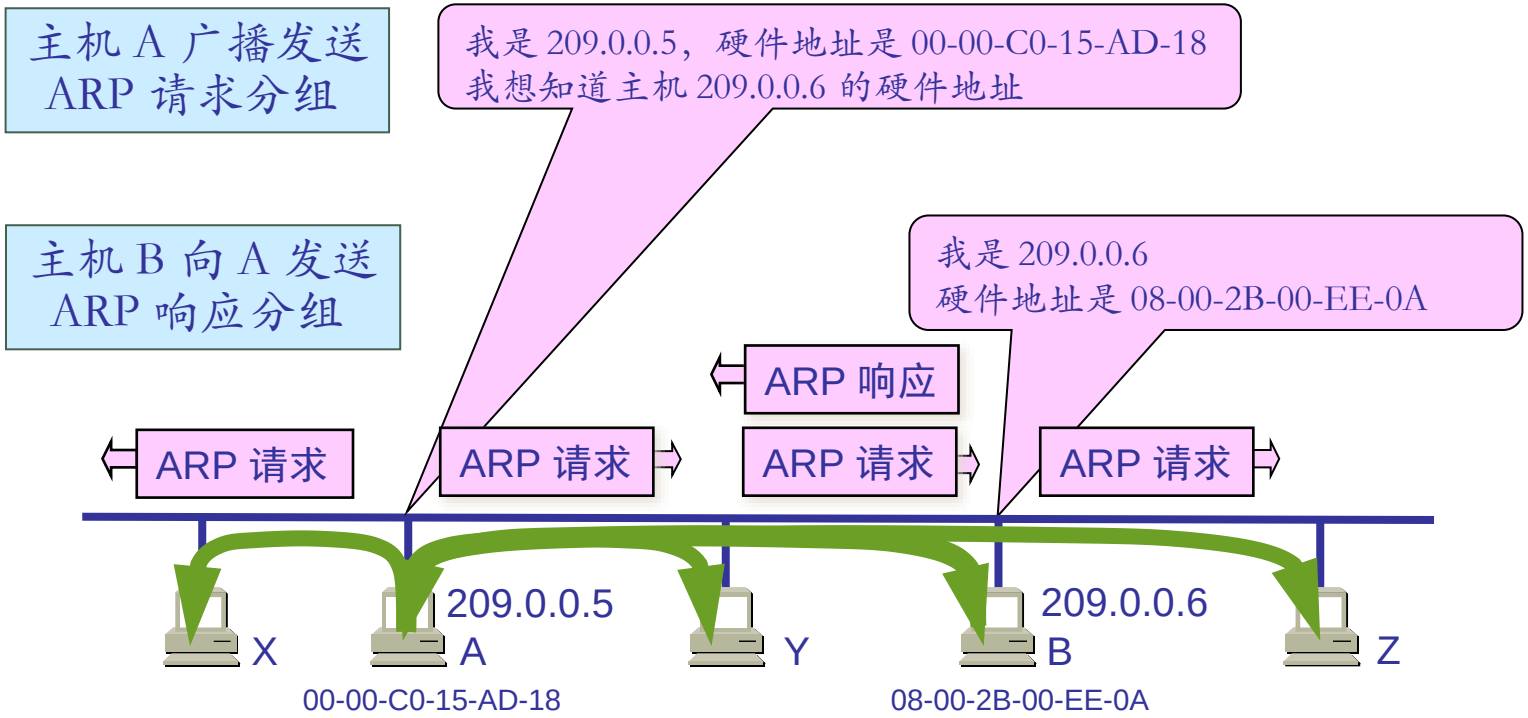
ARP地址解析协议


目的主机名
Host-b
DNS
目的主机IP地址
192.1.1.8
ARP
目的主机物理地址
08002B00EE0A

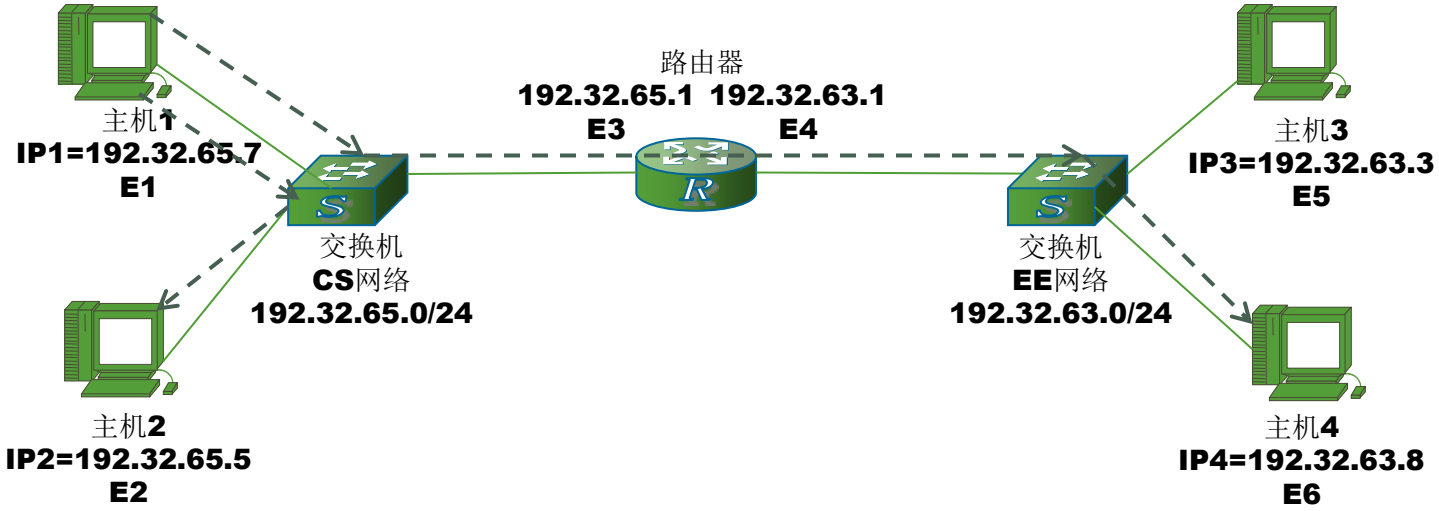
ARP 和 RARP

- 不管网络层使用的是什么协议，在实际网络的链路上传送数据帧时，最终还是必须使用硬件地址
- 每一个主机都设有一个 ARP 高速缓存(ARP cache)，里面有所在的局域网上的各主机和路由器的 IP 地址到硬件地址的映射表
- 当主机 A 欲向本局域网上的某个主机 B 发送 IP 分组时，就先在其 ARP 高速缓存中查看有无主机 B 的 IP 地址。如有，就可查出其对应的硬件地址，再将此硬件地址写入 MAC 帧，然后通过局域网将该 MAC 帧发往此硬件地址

ARP示例

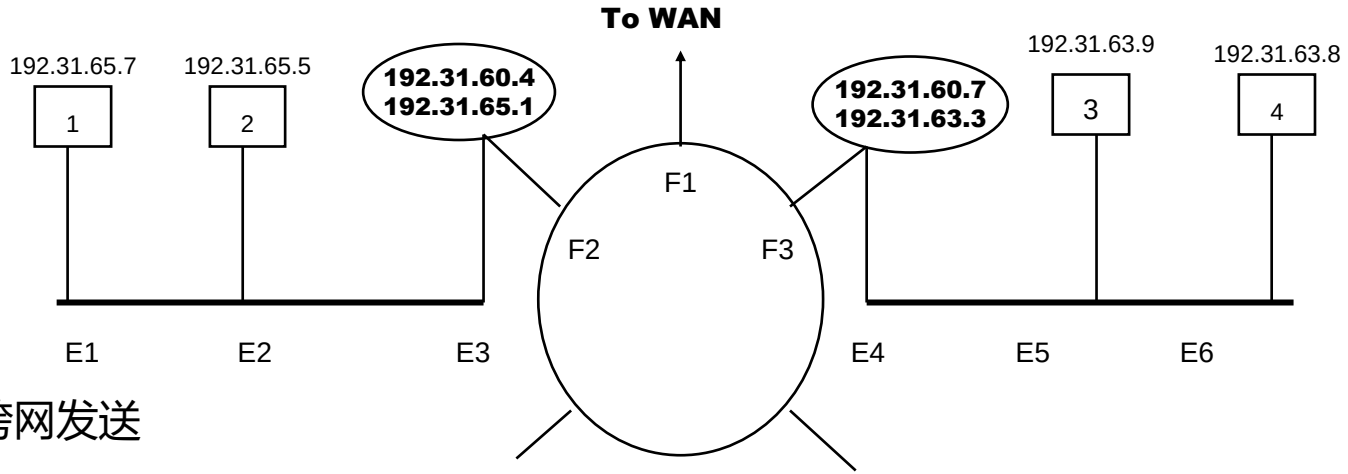

主机 A 广播发送 ARP 请求分组
我是 209.0.0.5，硬件地址是 00-00-C0-15-AD-18
我想知道主机 209.0.0.6 的硬件地址
主机 B 向 A 发送 ARP 响应分组
我是 209.0.0.6
硬件地址是 08-00-2B-00-EE-0A
ARP 响应
ARP 请求
ARP 请求
ARP 请求
ARP 请求
209.0.0.5
209.0.0.6
X
A
Y
B
Z
00-00-C0-15-AD-18
08-00-2B-00-EE-0A

ARP示例(2)



帧	源IP	源MAC	目的IP	目的MAC
主机1到主机2，CS网络	IP1	E1	IP2	E2
主机1到主机4，CS网络	IP1	E1	IP4	E3
主机1到主机4，EE网络	IP1	E4	IP4	E6

ARP示例(3)



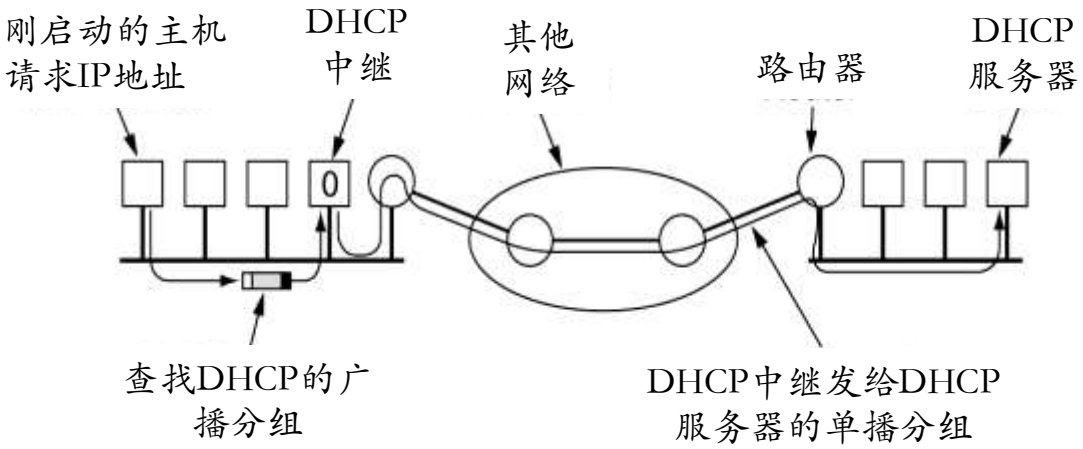
- 跨网发送

  1. 配置路由器识别全部其他网络
  2. 主机识别发往网关

DATA1								DATA1
192.31.63.8 (DATA1)		192.31.63.8 (DATA1)	192.31.63.8 (DATA1)		192.31.63.8 (DATA1)	192.31.63.8 (DATA1)		192.31.63.8 (DATA1)
E3[192.31.63.8 (DATA1)]		E3[192.31.63.8 (DATA1)]	F3[192.31.63.8 (DATA1)]		F3[192.31.63.8 (DATA1)]	E6[192.31.63.8 (DATA1)]		E6[192.31.63.8 (DATA1)]

DHCP动态主机配置协议

- DHCP的操作过程
  - 发送DHCP Discover
  - 中继转发
  - 回复DHCP Offer

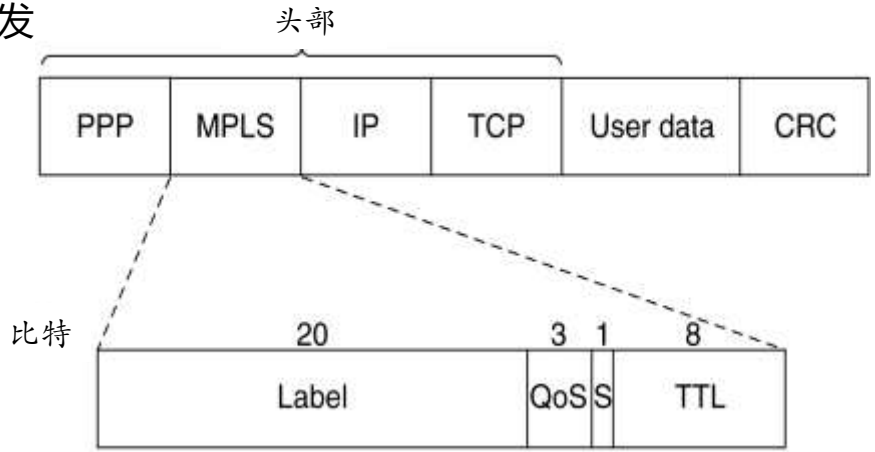


标签交换与MPLS

- MPLS(Multi-Protocol Label Switching)

  由目标地址改为根据标签实施转发

  - 标签位置

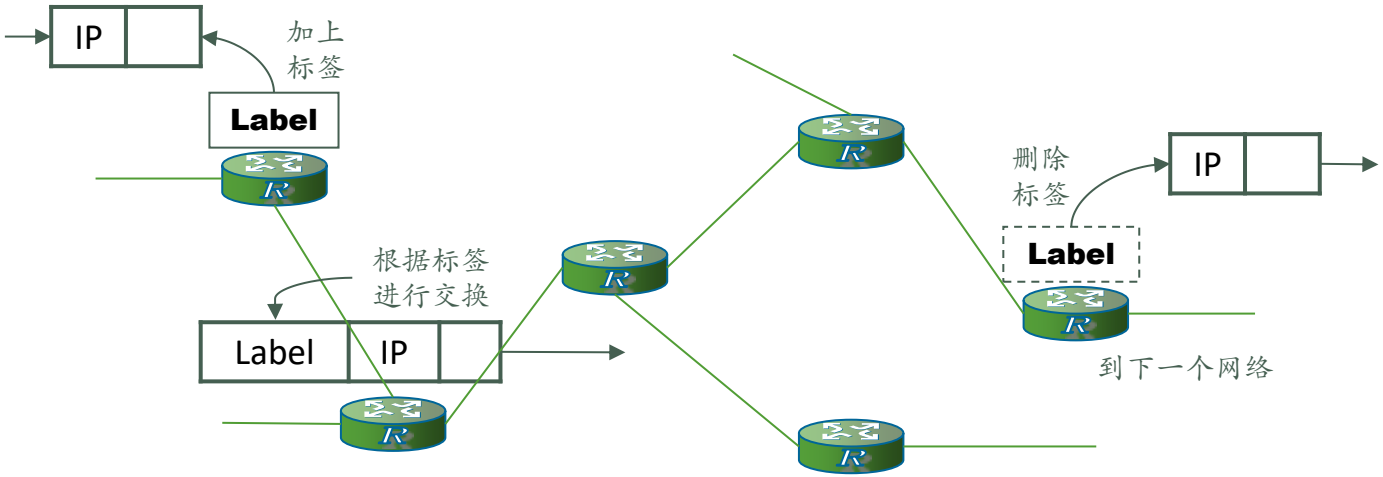


  - 标签Lable：索引
  - QoS：服务类型
  - S：堆栈字段，涉及多标签，为1表示最底部标签
  - TTL：生存期

MPLS

- MPLS
  - 标签交换路由器(LSR,Label Switched Router)
    - 转发与交换的区别
      - IP转发使用最长前缀匹配算法
      - 交换则使用标签作为索引，查询转发表，更为简单快速
  - 标签边缘路由器(LER,Label Edge Router)
    - 附加和删除标签

MPLS(2)



- 转发等价类(FEC,Forwarding Equivalence Class)
  - 将终止于某特定路由器或LAN的多个流合并成一组，使用同一标签，这些流称为FEC

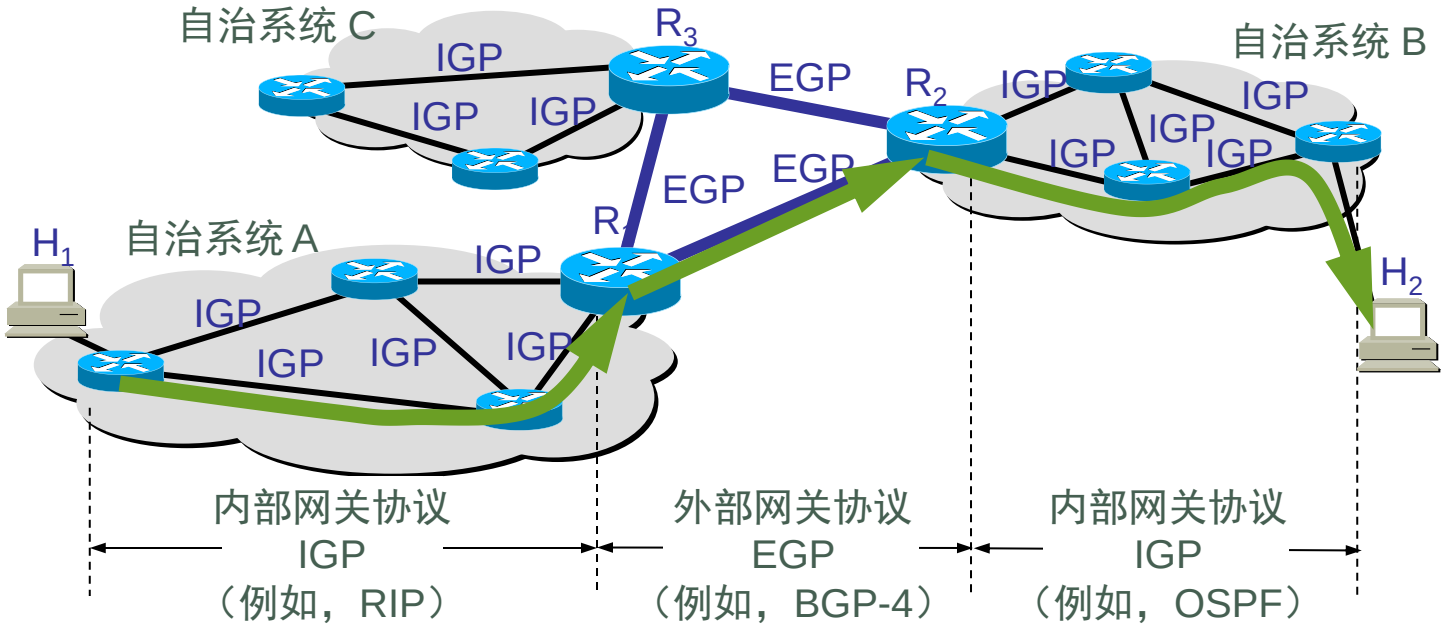
OSPF—内部网关路由协议

- 自治系统内部
  - 域内路由算法
- 内部网关协议
  - 距离矢量
    - RIP路由信息
  - 链路状态
    - OSPF(Open Shortest Path First)，开放最短路径优先

因特网有两大类路由选择协议

- 内部网关协议 IGP (Interior Gateway Protocol)
  - 即在一个自治系统内部使用的路由选择协议
  - 目前这类路由选择协议使用得最多，如 RIP 和 OSPF 协议
- 外部网关协议EGP (External Gateway Protocol)
  - 若源站和目的站处在不同的自治系统中，当分组传到一个自治系统的边界时，就需要使用一种协议将路由选择信息传递到另一个自治系统中
  - 这样的协议就是外部网关协议 EGP
  - 在外部网关协议中目前使用最多的是 BGP-4

自治系统和内部网关协议、外部网关协议



OSPF需求

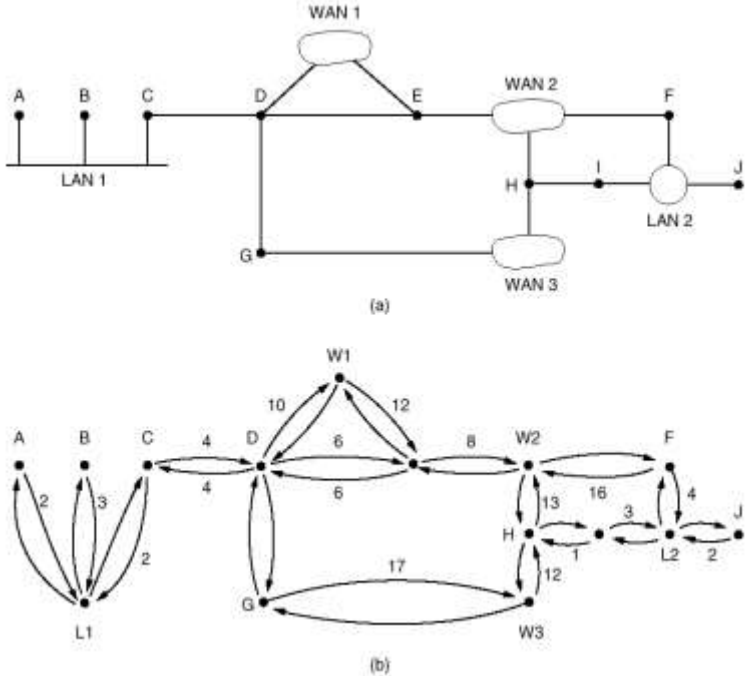
- 开放最短路径优先OSPF
  - 开放，公开发表
  - 支持多种距离衡量尺度，例如，物理距离、延迟等
  - 动态算法
  - 支持基于服务类型的路由
  - 负载平衡
  - 支持分层系统
  - 适量的安全措施
  - 支持隧道技术

有向拓扑图

- 构造有向拓扑图
  - 根据实际的网络、路由器和线路构造有向图
  - 每个弧赋一个开销值
  - 两个路由器之间的线路用一对弧来表示，弧权可以不同
  - 多路访问（multiaccess）网络，网络用一个结点表示，每个路由器用一个结点表示，网络结点与路由器结点的弧权为0

有向拓扑图(2)

- (a) 一个自治系统示例
- (b) 针对图(a)的图形表示

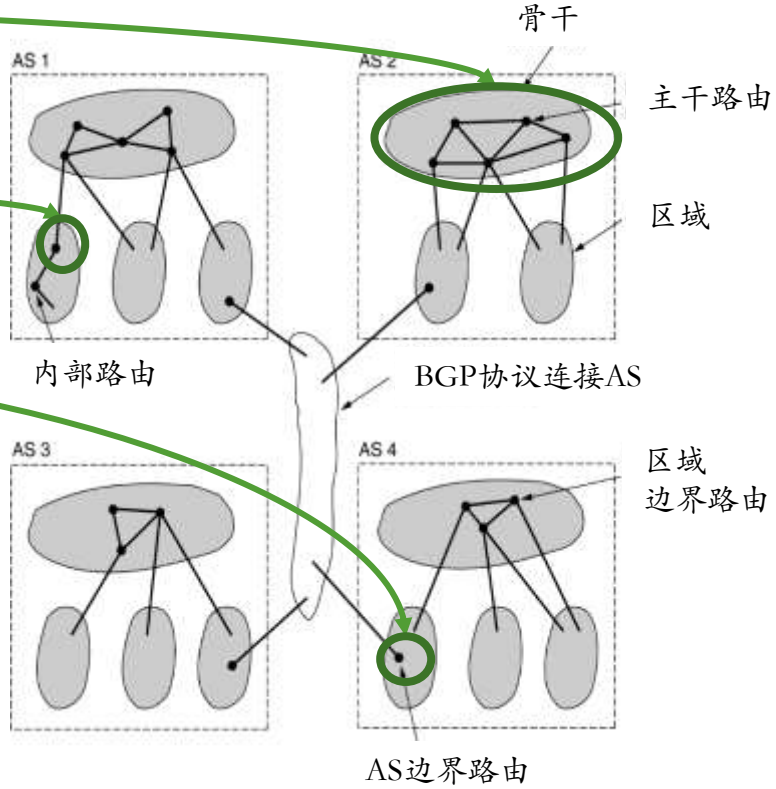


分层路由

- 分层路由
  - 自治系统AS可以划分区域（areas）；
  - 每个AS有一个主干（backbone）区域，称为区域0，所有区域与主干区域相连；
  - 一般情况下，有三种路由
    - 区域内
    - 区域间
      - 从源路由器到主干区域
      - 穿越主干区域到达目的区域
      - 到达目的路由器
    - 自治系统间

OSPF

- 骨干区域
  - 主干路由器
- 区域边界路由器
  - 连接多个区域的区域边界路由器
- AS边界路由器
  - 自治系统边界路由器
- 内部路由器
  - 完全在一个区域内的内部路由器



OSPF(2)

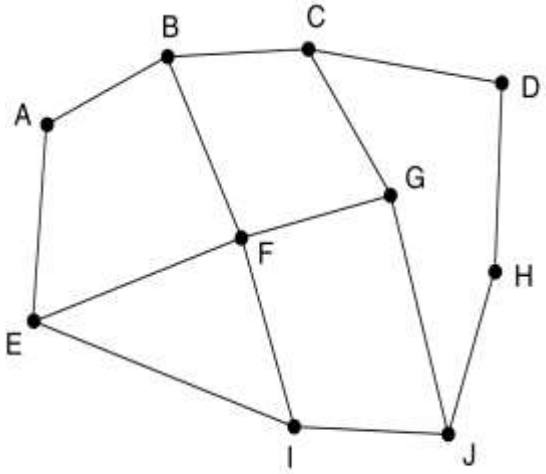
- OSPF的5类消息

消息类型	描述
Hello	用来发现所有邻居
Link state update	链路状态更新
Link state ack	链路状态更新确认
Database description	链路状态描述数据库
Link state request	链路状态请求

BGP——外部网关路由协议

- 外部网关协议与内部网关协议的不同
  - 路由策略涉及三方面
    - 政治：微软的流量不经过谷歌
    - 安全：五角大楼的流量不经过俄罗斯
    - 经济：教育网路部承载商业流量
- 边界网关协议BGP（Border Gateway Protocol）
  - 通过TCP连接传送路由信息
  - 采用路径矢量算法，路由信息中记录路径的轨迹

BGP



(a)

F从邻居收到有关D的信息

From B: "I use BCD"
From G: "I use GCD"
From I: "I use IFGCD"
From E: "I use EFGCD"

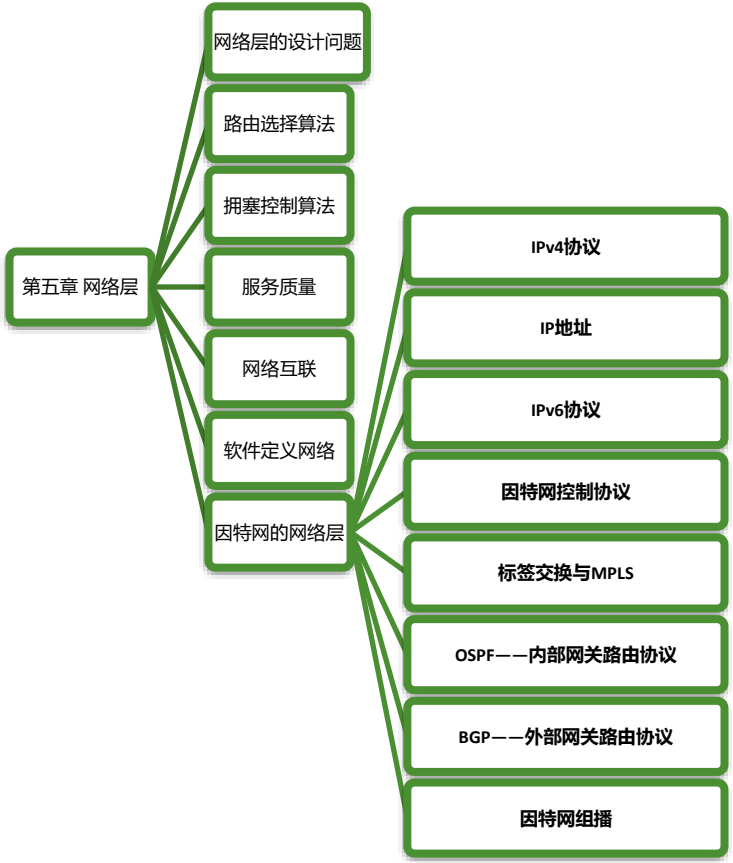
(b)

- BGP
  - 距离矢量协议，用政策选择路由（排除经过自己的路径，剩余线路逐一评价）
  - 路径矢量协议，BGP跟踪所使用的路径

因特网多播

- 本地多播地址(永久组地址)
  - 224.0.0.1　　LAN上的所有系统
  - 224.0.0.2　　LAN上的所有路由器
  - 224.0.0.5　　LAN上的所有OSPF路由器
  - 224.0.0.251　　LAN上的所有DNS服务器

本章导航与要点


第五章 网络层
网络层的设计问题
路由选择算法
拥塞控制算法
服务质量
网络互联
软件定义网络
因特网的网络层
IPv4协议
IP地址
IPv6协议
因特网控制协议
标签交换与MPLS
OSPF——内部网关路由协议
BGP——外部网关路由协议
因特网组播

本章课程结束