# 第4章 网络层



## 本章学习目标

- 理解网络层服务的原理:
  - 网络层服务模型
  - 转发和路由选择
  - 路由器如何工作
  - 路由选择（路径选择）
  - 广播、多播
- 实例化，在因特网上的实现



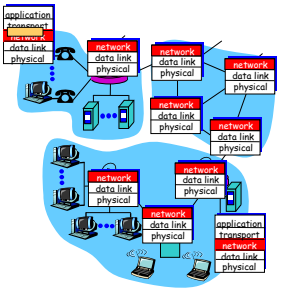
## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 选路概念
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## 网络层

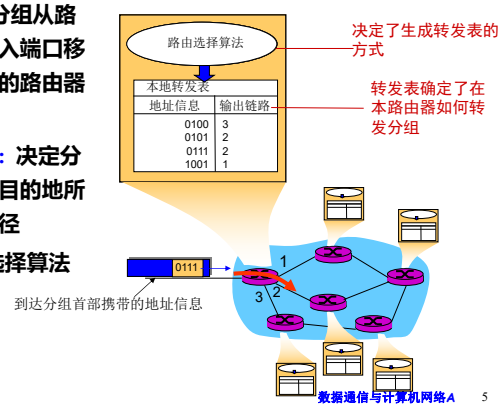
- 从发送主机到接收主机传输数据段
- 发送端：将数据段封装进数据报
- 接收端：向传输层交付数据段
- 网络层协议运行于每台主机、路由器中
- 路由器检查所有通过它的IP数据报首部字段






## 网络层的核心功能——转发和路由

- 转发: 将分组从路由器的输入端口移动到适当的路由器输出端口
- 路由选择: 决定分组从源到目的地所采用的路径
  - 路由选择算法






## 连接建立

- 在某些网络体系结构中第三个核心功能:
  - ATM, 帧中继, X.25
- 在数据分组传输之前，两台主机之间需要创建虚拟连接
  - 路由器参与连接建立
- 网络层和运输层的连接服务的对比:
  - 网络层: 在两台主机之间
  - 传输层: 在两个进程之间

## 网络层服务模型:

问题：网络层应该为传输层提供什么类型的服务?

对单个分组:
- 确保交付
- 具有时延上界的确保交付

对分组流:
- 有序分组交付
- 确保最小带宽
- 确保最大时延抖动
- 安全性服务

表4-1 因特网、ATM CBR 和ATM ABR服务模型

| 网络体系结构 | 服务模型 | 带宽保证 | 无丢包保证 | 有序 | 定时 | 拥塞指示 |
|---|---|---|---|---|---|---|
| 因特网 | 尽力而为 | 无 | 无 | 任何可能的顺序 | 不维护 | 无 |
| ATM | CBR | 保证恒定速率 | 是 | 有序 | 维护 | 拥塞不出现 |
| ATM | ABR | 保证最小速率 | 无 | 有序 | 不维护 | 提供拥塞指示 |



## 网络层服务模型:

- 在迄今为止的所有主要计算机网络体系结构中，网络层提供的服务类型可以归纳为两类:
  - 无连接服务
    - 数据报网络
  - 有连接服务
    - 虚电路网
- 与运输层服务的区别:
  - 网络层：提供主机到主机的服务，在端系统和网络核心实现
  - 传输层：进程到进程的服务，仅在端系统实现



## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 选路概念
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## 网络层连接和无连接服务——虚电路

- 虚电路网络提供网络层连接服务
- 数据报网络提供网络层无连接服务
- 与运输层服务的类比:
  - 服务: 主机到主机
  - 实现: 在网络核心实现



## 虚电路网络

- 从源端到目的端，建立的一条类似于电路交换的路径，路径上的每一个路由器，维护虚电路的连接状态
- 数据传输采用分组交换形式，每个分组携带虚电路标识符 VCID
- 链路、路由器资源(带宽、缓存等)可以面向虚电路进行资源预留



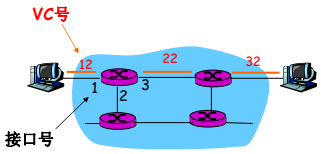
## 虚电路的组成

一条虚电路包括:

1. 从源到目的地的一条路径(逻辑连接)
2. 虚电路号：沿着该路径的每段链路的编号
3. 沿着该路径的每个路由器中转发表，记录了经过该路由器的每一条虚电路

- 属于一条虚电路的分组携带一个虚电路号
- 每到达一段链路，分组携带的虚电路号必须改变
  - 新的虚电路号从连接链路的路由器的转发表获得

## 虚电路转发表




路由器中的VC转发表：

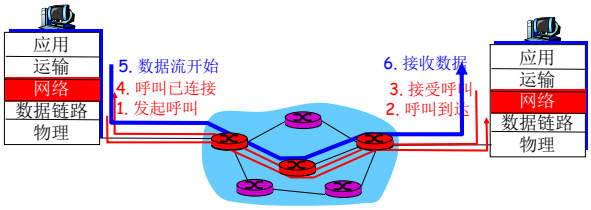
| 输入接口 | 输入VC # | 输出接口 | 输出VC # |
|---|---|---|---|
| 1 | 12 | 3 | 22 |
| 2 | 63 | 1 | 18 |
| 3 | 7 | 2 | 17 |
| 1 | 97 | 3 | 87 |
| ... | ... | ... | ... |

每一个路由器都需要维护虚电路的连接状态信息!



## 虚电路信令协议

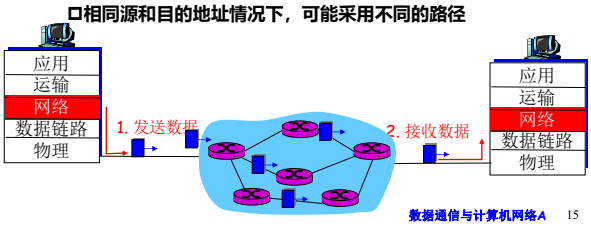
- 用于建立、维护和拆除VC
- 应用于典型的虚电路网络：ATM、帧中继、X.25中
- 没有用于今天的因特网中





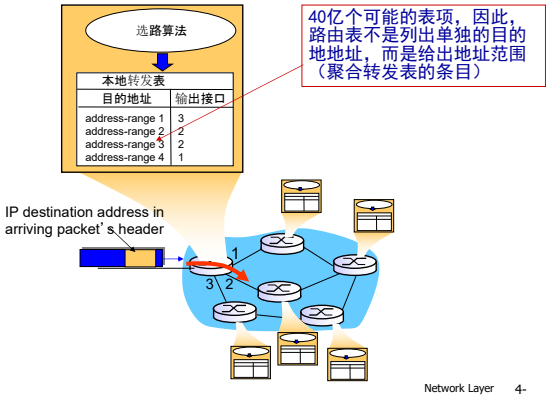

## 数据报网络

- 在网络层无连接
- 分组携带目的主机地址
- 路由器根据分组目的地址转发分组
  - 基于路由选择协议构造转发表
  - 每个分组独立选路
    - 相同源和目的地址情况下，可能采用不同的路径






## 转发表






## 转发表

| 目标地址范围 | 接口 |
|---|---|
| 11001000 00010111 00010000 00000000<br>到<br>11001000 00010111 00010111 11111111 | 0 |
| 11001000 00010111 00011000 00000000<br>到<br>11001000 00010111 00011000 11111111 | 1 |
| 11001000 00010111 00011100 00000000<br>到<br>11001000 00010111 00011111 11111111 | 2 |
| 其他 | 3 |

## 最长前缀匹配优先

**最长前缀匹配原则**

在查找给定目的地址的转发表项时，使用与目标地址匹配的最长地址前缀.

| 目的地址范围 | 接口 |
|---|---|
| 11001000 00010111 00010*** ********* | 0 |
| 11001000 00010111 00011000 ********* | 1 |
| 11001000 00010111 00011*** ********* | 2 |
| otherwise | 3 |

例子:

DA: 11001000 00010111 00010110 10100001 哪个接口?

DA: 11001000 00010111 00011000 10101010 哪个接口?

## 数据报或虚电路网络: why?

因特网（数据报网络）

- 在计算机间交换数据
  - “弹性”服务，无严格的定时要求
- “智能” 端系统 (计算机)
  - 能够适应，执行控制，差错控制
  - 网络内部简单，“边缘”复杂
- 许多链路类型
  - 不同特点、难以提供统一服务

ATM(虚电路网络)

- 从电话网络演化而来
- 人类交谈:
  - 严格定时，可靠性要求高
  - 需要有保障的服务
  - “哑” 端系统
  - 电话，网络边缘简单
  - 网络内部复杂



## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成（略）
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 选路概念
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



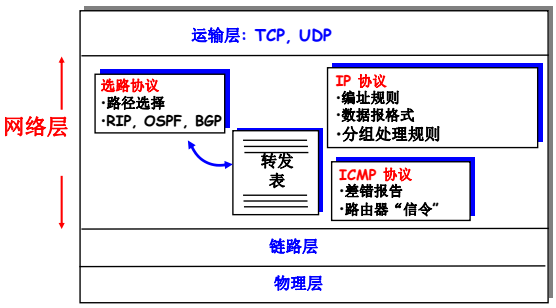
## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 选路概念
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## The Internet 网络层

主机，路由器网络层功能:



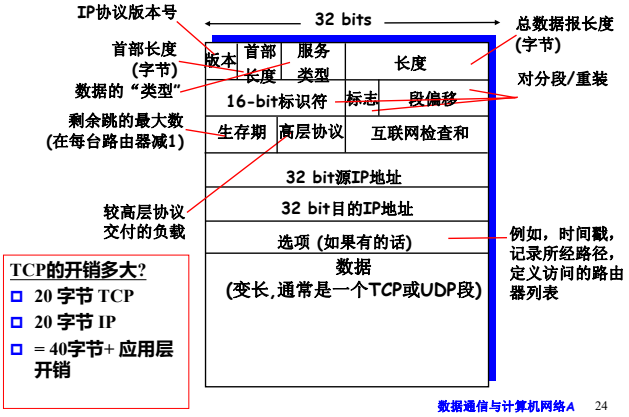



## IP:无连接交付系统

- 互联网服务被定义成不可靠的、尽力而为、无连接分组交付系统。
  - 服务是不可靠的，因为分组可能丢失、重复、延迟或不按序交付等，但服务不检测这些情况，也不提醒发送方和接收方。
  - 服务是尽力而为的，互联网并不随意地丢弃分组；只有资源用完或底层网络出现故障时才可能出现不可靠性。
  - 服务是无连接的，因为每个分组都是独立对待的。分组序列可能经过不同的传输路径或者有的丢失有的到达。



## IP 数据报格式

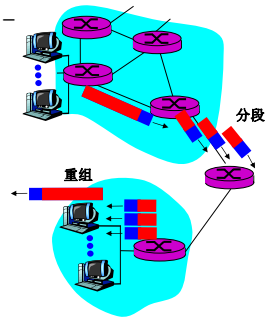



TCP的开销多大?

- 20 字节 TCP
- 20 字节 IP
- = 40字节+ 应用层开销

## IP分片和重新组装

- 网络链路有MTU (最大传输长度) – 最大可能的链路级帧
  - 不同的链路类型，不同的
- 在网络中，大IP 数据报被分割（“分段”）
  - 一个数据报分割为几个数据报
  - “重新装配”仅在目的主机
  - IP首部某些字段用于标识、排序相关分片






## IP分片和重新组装



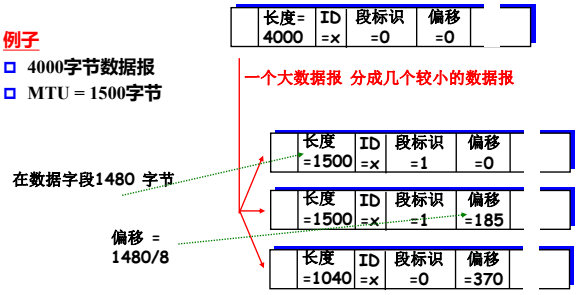

## IP分片和重新组装

- Fragmentation Identifier-分片标识符
  - 16bits，用于标识来自于同一个IP数据报的分片数据
- Fragmentation Flag-分片标志位
  - 3bits，从左到右分别为：
    - 预留，为0
    - DF，为1表示：禁止分片；为0表示：允许分片
    - MF，为1表示：之后还有更多分片；为0表示：是最后一个分片的数据报
- Fragmentation Offset-段偏移
  - 19bits，标识一个分片在整个数据报中的位置
  - 其值×8B为实际的偏移值



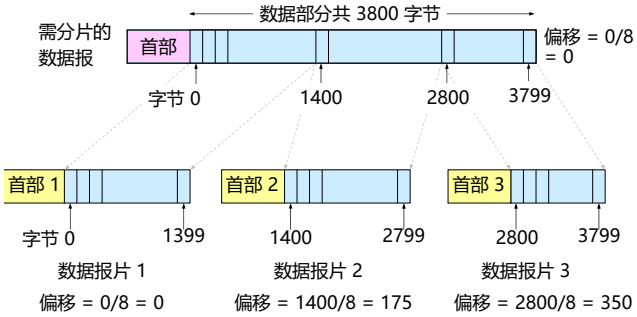
## IP分片和重新组装

例子
- 4000字节数据报
- MTU = 1500字节






## IP分片和重新组装



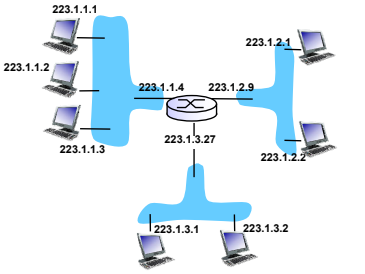



## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP

## IP编址:概述

- IP地址: 对主机、路由器接口的32-bit 标识符
- 接口: 在主机/路由器和物理链路之间的连接
  - 路由器通常具有多个接口
  - 主机可能具有多个接口
  - 每个接口都需要进行IP编址，即对应一个IP地址




223.1.1.1 = 11011111 00000001 00000001 00000001

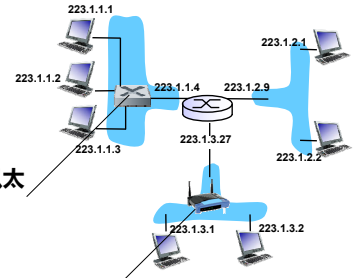
223 1 1 1



## IP addressing: introduction

Q: 接口如何连接?

A: 有线以太网接口与以太网交换机连接

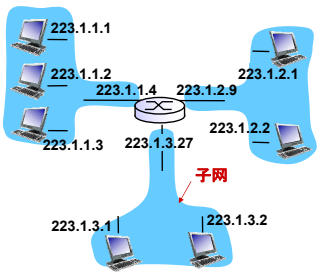



A: 无线以太网接口与WiFi基站连接



## 子网

- IP地址:
  - 子网部分(高阶比特)
  - 主机部分(低阶比特)
- 什么是子网 ?
  - IP地址具有相同子网部分的设备接口
  - 能够物理上互相到达而没有中间路由器



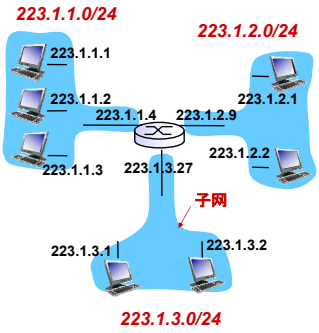

网络由3个子网组成



## 子网

判断方法

- 为了确定子网，从其主机或路由器分离每个接口，生成网络孤岛。每个网络孤岛被称为一个子网

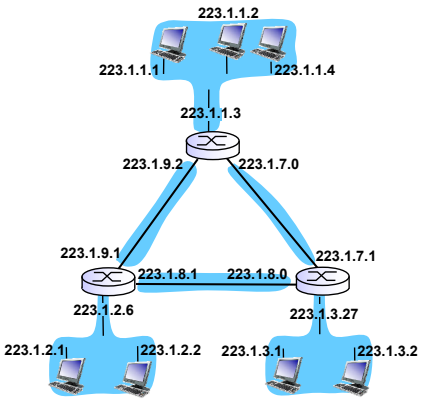



子网掩码：/24



## 子网

多少个子网?





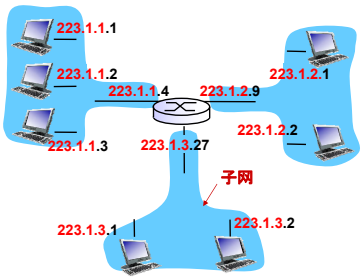

## IP编址：分类的IP地址

- 采用两级的 IP 地址描述互联网上的设备接口:

| 网络号 | 主机号 |
|---|---|

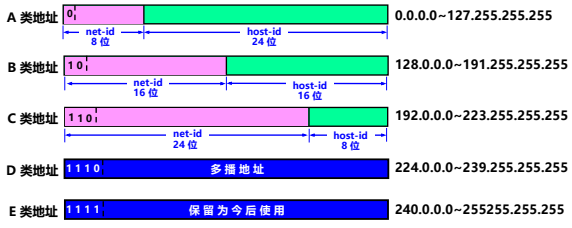
32 位

- 网络号( net-id)标志主机（或路由器）所在的特定网络
- 主机号 (host-id)标志该主机（或路由器）的接口
- 主机号在网络号所指明的网络范围内必须是唯一的



## IP编址：分类的IP地址

- 全球所有IPV4地址可分为以下5类：




五类IP 地址的指派范围

| 网络类别 | 最大可指派的网络数 | 第一个可指派的网络号 | 最后一个可指派的网络号 | 每个网络中最大主机数 |
|---|---|---|---|---|
| A | 126 ($2^7 - 2$) | 1 | 126 | 16777214 |
| B | 16383 ($2^{14} - 1$) | 128.1 | 191.255 | 65534 |
| C | 2097151 ($2^{21} - 1$) | 192.0.1 | 223.255.255 | 254 |

## IP编址：分类的IP地址

- 一般不使用的特殊的 IP 地址：

| 网络号 | 主机号 | 源地址使用 | 目的地址使用 | 代表的意思 |
|---|---|---|---|---|
| 0 | 0 | 可以 | 不可以 | 在本网络范围内表示本机；在路由表中用于表示默认路由 |
| 0 | host-id | 可以 | 不可以 | 本网络上主机号=host-id的特定主机 |
| 全 1 | 全 1 | 不可以 | 可以 | 只在本网络上进行广播（各路由器均不转发） |
| net-id | 全 0 | 不可以 | 不可以 | 做为网络地址，表示一个网络 |
| net-id | 全 1 | 不可以 | 可以 | 对 net-id指向的网络上所有主机进行广播 |
| 127 | 非全 0 或全 1 的任何数 | 可以 | 可以 | 用于本地软件环回测试 |

- 私有地址：

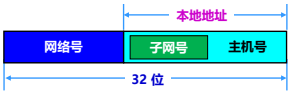
| 类 | net-id | 地址块 |
|---|---|---|
| A | 10 | 1 |
| B | 172.16~172.31 | 16 |
| C | 192.167.0.*~192.168.255.* | 256 |

## IP编址：分类的IP地址

- 采用分级IP 地址结构的优点：
  - IP 地址管理机构在分配 IP 地址时只分配网络号，而剩下的主机号则由得到该网络号的单位自行分配。方便对 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（而不考虑目的主机号）。
- 采用分级IP 地址结构的缺点：
  - IP 地址空间的利用率有时很低。
  - 给每一个物理网络分配一个网络号会使路由表变得太大因而使网络性能变坏。
  - 两级的 IP 地址不够灵活。

## IP编址：子网划分

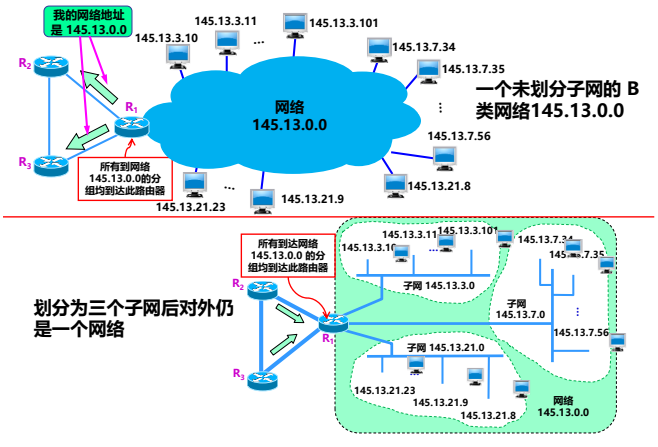
- 从 1985 年起，从IP 地址的主机号 host-id借用若干个位作为子网号 subnet-id， 增加了一个“子网号字段”，使两级的 IP 地址变成为三级的 IP 地址。
- 这种做法叫做划分子网 (subnetting) 。




## IP编址：子网划分

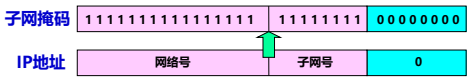
- 划分子网纯属一个单位内部的事务。单位对外仍然表现为没有划分子网的网络。
- 凡是从其他网络发送给此单位某个主机的 IP 数据报，仍然是根据 IP 数据报的目的网络号 net-id，先找到连接在本单位网络上的路由器。
- 然后此路由器在收到 IP 数据报后，再按目的网络号 net-id 和子网号 subnet-id 找到目的子网。
- 最后将 IP 数据报直接交付目的主机。

## IP编址：子网划分



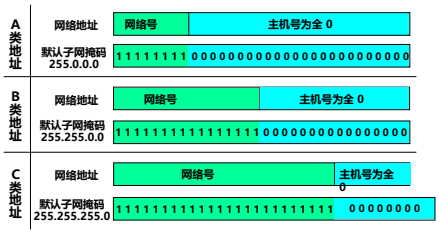
## IP编址：子网划分

- 从一个 IP 数据报的首部并无法判断源主机或目的主机所连接的网络是否进行了子网划分。
- 使用子网掩码 (subnet mask) 可以找出 IP 地址中的子网部分。
- 规则：
  - 子网掩码左边部分的一连串 1，对应于网络号和子网号
  - 子网掩码右边部分的一连串 0，对应于主机号



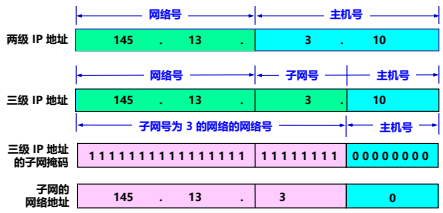

## IP编址：子网划分

A、B、C三类地址默认的子网掩码




## IP编址：子网划分

(IP 地址) AND (子网掩码) =网络地址




## IP编址：子网划分

【例4-2】已知 IP 地址是 141.14.72.24，子网掩码是 255.255.192.0。试求网络地址。


【例4-3】上例中，若子网掩码改为 255.255.224.0，试求网络地址，讨论所得结果。

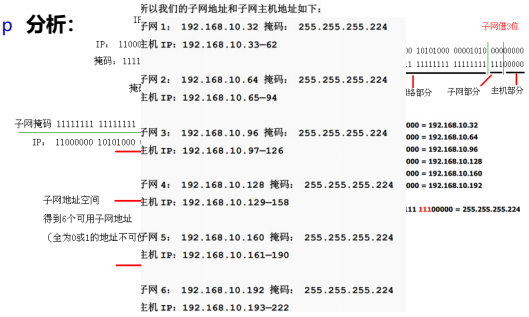

不同的子网掩码得出相同的网络地址。
但不同的掩码的效果是不同的。

## IP编址：子网划分

- 总结：子网掩码是一个网络的重要属性。
  - 路由器在和相邻路由器交换路由信息时，必须把自己所在网络（或子网）的子网掩码告诉相邻路由器。
  - 路由器的路由表中的每一个项目，除了要给出目的网络地址外，还必须同时给出该网络的子网掩码。

## IP编址：子网划分

- 有固定长度子网和变长子网两种子网划分方法
- 例如：学院新建 4 个机房，每个房间有 25 台机器，给定一个网络地址块：192.168.10.0，现在需要将其划分为4 个子网。
- 分析：



## IP编址：子网划分

- 有固定长度子网和变长子网两种子网划分方法
- 例如：一家公司目前有 5个部门 A 至 E，其中：A 部门有 50 台 PC，B 部门 20 台，C 部门 30 台，D 部门 15 台，E 部门 20 台，企业信息经理分配了一个总的网络地址 192．168．2．0 / 24 给你，作为网络管理员，请问如何为每个部门划分单独的子网段?

## IP编址：子网划分

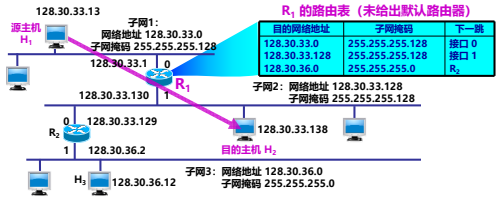
- 答：192.168.2.0/24 该范围为：192.168.2.0—192.168.2.255
- A 部门
  - IP 范围：192.168.2.0—192.168.2.63
  - 子网掩码：255.255.255.192
  - 其中 192.168.2.0 为网络地址，192.168.2.63 为广播地址
- B 部门
  - IP 范围：192.168.2.64—192.168.2.95
  - 子网掩码：255.255.255.224
  - 其中 192.168.2.64 为网络地址，192.168.2.95 为广播地址
- C 部门
  - IP 范围：192.168.2.96—192.168.2.127
  - 子网掩码：255.255.255.224
  - 其中 192.168.2.96 为网络地址，192.168.2.127 为广播地址
- D 部门
  - IP 范围：192.168.2.128—192.168.2.159
  - 子网掩码：255.255.255.224
  - 其中 192.168.2.128 为网络地址，192.168.2.159 为广播地址
- E 部门
  - IP 范围：192.168.2.160—192.168.2.191
  - 子网掩码：255.255.255.224
  - 其中 192.168.2.160 为网络地址，192.168.2.191 为广播地址
- 冗余 IP 范围为 192.168.2.192-192.168.2.255/255.255.255.192

## IP编址：子网划分的分组转发算法

- 从收到的分组的首部提取目的 IP 地址 *D*。
- 先用各网络的子网掩码和 *D* 逐位相“与”，看是否和相应的网络地址匹配。若匹配，则将分组直接交付。否则就是间接交付，执行(3)。
- 若路由表中有目的地址为 *D* 的特定主机路由，则将分组传送给指明的下一跳路由器；否则，执行 (4)。
- 对路由表中的每一行，将子网掩码和 *D* 逐位相“与”。若结果与该行的目的网络地址匹配，则将分组传送给该行指明的下一跳路由器；否则，执行 (5)。
- 若路由表中有一个默认路由，则将分组传送给路由表中所指明的默认路由器；否则，执行 (6)。
- 报告转发分组出错。

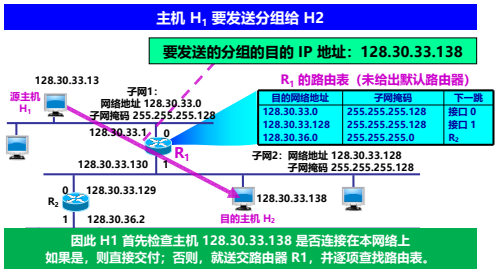
## IP编址：子网划分的分组转发算法

【例】已知互联网和路由器 $R_1$ 中的路由表。主机 $H_1$ 向 $H_2$ 发送分组。试讨论 $R_1$ 收到 $H_1$ 向 $H_2$ 发送的分组后查找路由表的过程。




## IP编址：子网划分的分组转发算法

【例】已知互联网和路由器 $R_1$ 中的路由表。主机 $H_1$ 向 $H_2$ 发送分组。试讨论 $R_1$ 收到 $H_1$ 向 $H_2$ 发送的分组后查找路由表的过程。




## IP编址：子网划分的分组转发算法

【例】已知互联网和路由器 $R_1$ 中的路由表。主机 $H_1$ 向 $H_2$ 发送分组。试讨论 $R_1$ 收到 $H_1$ 向 $H_2$ 发送的分组后查找路由表的过程。

主机 $H_1$ 首先将本子网的子网掩码 255.255.255.128
与分组的 IP 地址 128.30.33.138 逐比特相与”(AND 操作)

255.255.255.128 AND 128.30.33.138 的计算

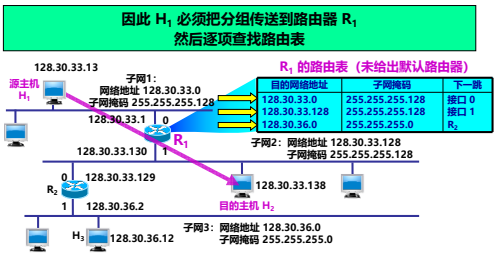
255 就是二进制的全 1，因此 255 AND xyz = xyz，
这里只需计算最后的 128 AND 138 即可。

128 → 1 0000000
138 → 1 0001010
逐比特 AND 操作后 1 0000000 → 128

| | |
|---|---|
| 逐比特 AND 操作 | 255.255.255. 128 |
| | 128. 30. 33. 138 |
| | 128. 30. 33. 128 ≠ $H_1$ 的网络地址 |

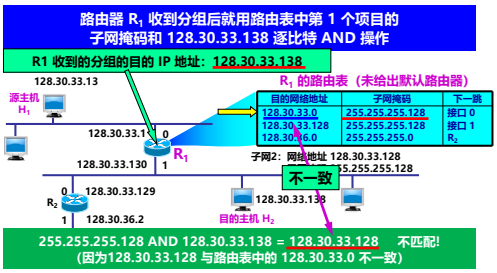
## IP编址：子网划分的分组转发算法

【例】已知互联网和路由器 $R_1$ 中的路由表。主机 $H_1$ 向 $H_2$ 发送分组。试讨论 $R_1$ 收到 $H_1$ 向 $H_2$ 发送的分组后查找路由表的过程。



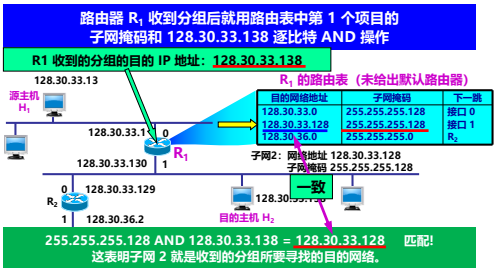

## IP编址：子网划分的分组转发算法

【例】已知互联网和路由器 $R_1$ 中的路由表。主机 $H_1$ 向 $H_2$ 发送分组。试讨论 $R_1$ 收到 $H_1$ 向 $H_2$ 发送的分组后查找路由表的过程。



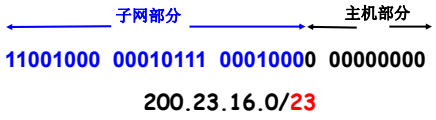

## IP编址：子网划分的分组转发算法

【例】已知互联网和路由器 $R_1$ 中的路由表。主机 $H_1$ 向 $H_2$ 发送分组。试讨论 $R_1$ 收到 $H_1$ 向 $H_2$ 发送的分组后查找路由表的过程。




## IP编址: CIDR

- 无类型域间路由(Classless InterDomain Routing,CIDR)
  - 任意长度的IP地址子网部分
  - 消除传统的有类地址划分（A、B、C、D、E、F）
  - 地址格式: a.b.c.d/x, 其中x是子网部分的比特长度






## IP编址: CIDR

128.14.32.0/20 表示的地址（$2^{12}$ 个地址）


## IP编址: CIDR

- 无类型域间路由(Classless InterDomain Routing,CIDR)
  - 提高IPV4地址空间分配效率
  - 提高路由效率，使得路由信息通告更高效
    - 构造超网（supernetting）/路由聚合
    - 路由聚合有利于减少路由器之间的路由选择信息的交换，从而提高了整个互联网的性能

# IP编址: CIDR实现路由聚合

- 路由聚合：将几个相邻子网的连续网络地址进行汇总，以单个CIDR网络地址形式表示聚合之后的网络。
- 利用CIDR实现路由聚合至少要满足两个基本条件
  - 待汇总地址的网络号拥有相同的高位
  - 待汇总地址是连续的
  - 待汇总网络地址的数目必须是2的整数幂次



例：某公司分配到4个C类网络：192.168.1.0；192.168.2.0；192.168.3.0；192.168.4.0，如何把4个C类网络组合成一个超网？超网的网络ID和子网掩码是多少？

- 首先将4个C类网络的网络ID转换为二进制数表示，其中192.168对应二进制形式为11000000.10101000:
  192.168.1.0: 192.168.00000001.00000000
  192.168.2.0: 192.168.00000010.00000000
  192.168.3.0: 192.168.00000011.00000000
  192.168.4.0: 192.168.00000100.00000000
- 根据二进制的网络ID，选择具有相同高位的部分作为超网的网络ID
  11000000.10101000.00000xxx.xxxxxxxx
- 根据网络ID计算掩码为
  11111111.11111111.11111000.00000000，对应的点分十进制表达为：255.255.248.0
- 所以汇聚后的超网是192.168.0.0/21，其中网络ID为：192.168.0.0，掩码为：255.255.248.0



例：有两组连续的C类网络地址块如下，能否汇聚成一个超网？

A：201.66.32.0 - 201.66.47.0

B：201.66.71.0 - 201.66.86.0

答：A和B都是连续地址块，且网络地址数满足第二个条件

对于A：

202.66.32.0 = 202.66.00100000.0

202.66.47.0 = 202.66.00101111.0

所以，可以聚合成超网202.66.32.0/20，掩码为255.255.240.0

对于B：

202.66.71.0 = 202.66.01000111.0

202.66.86.0 = 202.66.01010110.0

取网络相同的高位部分作为网络ID，如果聚合成超网，则表示为202.66.64.0/19：

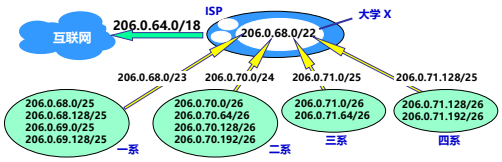
但是：这样的网络ID所表示的超网，包含了实际中并不存在的子网（路由黑洞），所以不能聚合。



# IP编址:如何得到IP地址?

问题：组织或机构如何得到一个IP地址块?

回答: 向组织或机构的ISP申请，从ISP的地址空间分配得到

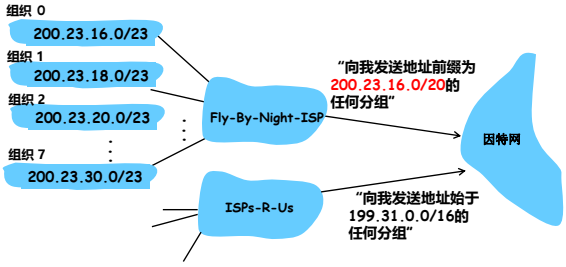



| 单位 | 地址块 | 二进制表示 | 地址数 |
|---|---|---|---|
| ISP | 206.0.64.0/18 | 11001110.00000000.01* | 16384 |
| 大学 | 206.0.68.0/22 | 11001110.00000000.010001* | 1024 |
| 一系 | 206.0.68.0/23 | 11001110.00000000.0100010* | 512 |
| 二系 | 206.0.70.0/24 | 11001110.00000000.01000110.* | 256 |
| 三系 | 206.0.71.0/25 | 11001110.00000000.01000111.0* | 128 |
| 四系 | 206.0.71.128/25 | 11001110.00000000.01000111.1* | 128 |



# IP编址: CIDR实现路由聚合

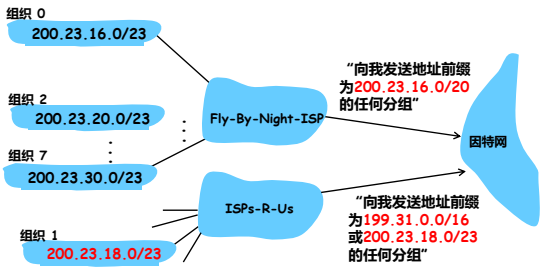
路由聚合允许有效的通告选路信息:






# IP编址: 更为特定的路由

根据最长前缀匹配优先原则，ISPs-R-Us具有到组织 1的更具体的路由



## IP编址: 其他问题

问题：ISP怎样得到地址块?

回答: 因特网名字与号码分配团体( Internet Corporation for Assigned Names and Numbers,ICANN )

- 分配地址
- 管理DNS
- 分配域名，调解争议



## IP编址: 如何得到一个地址?

问题：主机怎样获得IP地址?

- 由系统管理员在文件中硬编码（静态配置）
  - Wintel: 控制面板->网络->配置->TCP/IP->性质
  - UNIX: /etc/rc.config
- 动态主机配置协议(Dynamic Host Configuration Protocol DHCP): 动态地从服务器得到地址
  - “即插即用”



## IP编址: 如何得到一个地址?

问题：主机怎样获得IP地址?

- 由系统管理员在文件中硬编码（静态配置）
  - Wintel: 控制面板->网络->配置->TCP/IP->性质
  - UNIX: /etc/rc.config



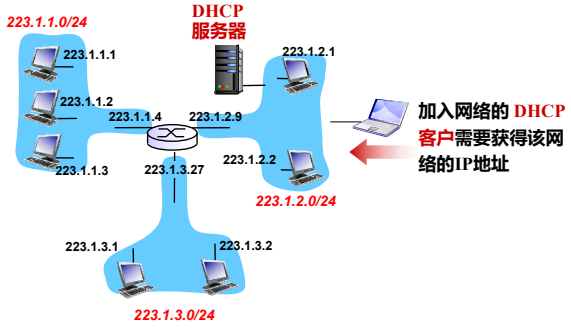
## IP编址: 如何得到一个地址?

问题：主机怎样获得IP地址?

- 由系统管理员在文件中硬编码（静态配置）
  - Wintel: 控制面板->网络->配置->TCP/IP->性质
  - UNIX: /etc/rc.config
- 动态主机配置协议(Dynamic Host Configuration Protocol DHCP): 动态地从服务器得到：IP地址、子网掩码、默认网关地址、DNS服务器名称与IP地址
  - “即插即用”、地址续订、重用 、支持移动IP

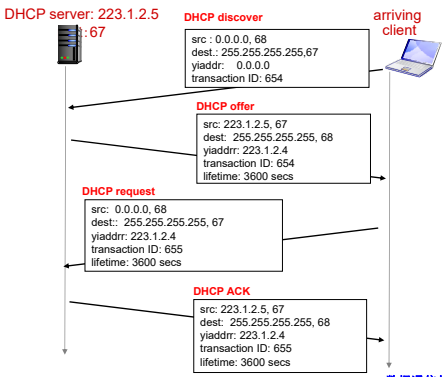


## DHCP 客户端-服务器场景






## DHCP 客户端-服务器场景

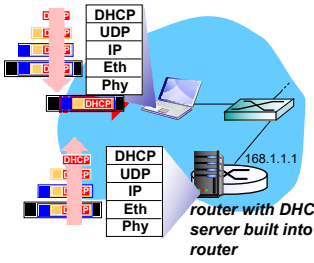


## DHCP: 获取除IP地址外更多的信息

- DHCP流程总结:
  - 主机广播“DHCP 发现”报文
  - DHCP服务器以“DHCP 提供”报文向客户做出响应
  - 主机发送“DHCP 请求”报文请求IP地址
  - DHCP服务器用“DHCP ACK”报文进行相应，发送IP地址
- DHCP除了分配主机IP地址，还能使主机获取更多其他信息:
  - 客户机的第一跳路由器IP地址
  - 本地DNS服务器主机名和IP地址
  - 子网掩码(用于指示网络号与主机地址部分)



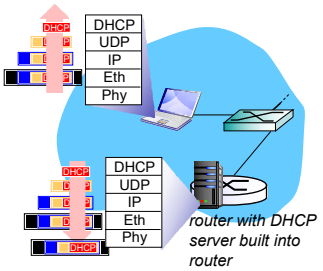
## DHCP: 例子




- 连接中的笔记本电脑需要获得IP地址、第一跳路由器地址、DNS服务器地址：使用DHCP协议
- DHCP 请求报文按照如下顺序被被封装在：UDP分组-IP数据报-802.1以太网帧
- 以太网帧以 (dest: FFFFFFFFFFFF)在局域网内广播，被内嵌运行于路由器的DHCP 服务器接收
- 再依序从以太网帧-IP数据报-UDP分组中，解封装出 DHCP请求报文



## DHCP: 例子




- DHCP服务器应答DHCP ACK消息，包含：客户机IP地址列表、客户机第一跳路由器IP地址、DNS服务器主机名和IP地址
- DHCP服务器将DHCP ACK报文封装成帧，发送给客户机；客户机解封装数据帧，得到DHCP ACK
- 现在客户机可以得知：服务器可以分配的IP地址列表、客户机第一跳路由器IP地址、DNS服务器主机名和IP地址



## DHCP: Wireshark 输出 (家庭网络)

请求

```
Message type: Boot Request (1)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 0.0.0.0 (0.0.0.0)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 0.0.0.0 (0.0.0.0)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP Request
Option: (61) Client identifier
    Length: 7; Value: 010016D323688A;
    Hardware type: Ethernet
    Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Option: (t=50,l=4) Requested IP Address = 192.168.1.101
Option: (t=12,l=5) Host Name = "nomad"
Option: (55) Parameter Request List
    Length: 11; Value: 010F03062C2E2F1F21F92B
    1 = Subnet Mask; 15 = Domain Name
    3 = Router; 6 = Domain Name Server
    44 = NetBIOS over TCP/IP Name Server
    .....
```

应答

```
Message type: Boot Reply (2)
Hardware type: Ethernet
Hardware address length: 6
Hops: 0
Transaction ID: 0x6b3a11b7
Seconds elapsed: 0
Bootp flags: 0x0000 (Unicast)
Client IP address: 192.168.1.101 (192.168.1.101)
Your (client) IP address: 0.0.0.0 (0.0.0.0)
Next server IP address: 192.168.1.1 (192.168.1.1)
Relay agent IP address: 0.0.0.0 (0.0.0.0)
Client MAC address: Wistron_23:68:8a (00:16:d3:23:68:8a)
Server host name not given
Boot file name not given
Magic cookie: (OK)
Option: (t=53,l=1) DHCP Message Type = DHCP ACK
Option: (t=54,l=4) Server Identifier = 192.168.1.1
Option: (t=1,l=4) Subnet Mask = 255.255.255.0
Option: (t=3,l=4) Router = 192.168.1.1
Option: (6) Domain Name Server
    Length: 12; Value: 445747E2445749F244574092;
    IP Address: 68.87.71.226;
    IP Address: 68.87.73.242;
    IP Address: 68.87.64.146
Option: (t=15,l=20) Domain Name = "hsd1.ma.comcast.net."
```

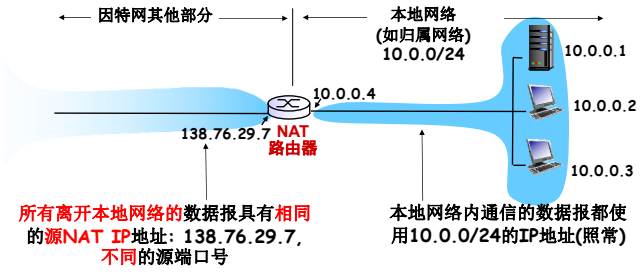


## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 选路概念
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## NAT: 网络地址转换



## NAT: 网络地址转换

- 动机: 本地网络对外只使用一个IP地址，外部仅关注该地址：
  - 对ISP无需分配地址范围：对所有设备只用一个IP地址
  - 能够改变本地网络中的设备地址，而不必通知外部
  - 能够改变ISP而不需更改本地网络的设备地址
  - 本地网络内部结构外部不可见，增强安全性



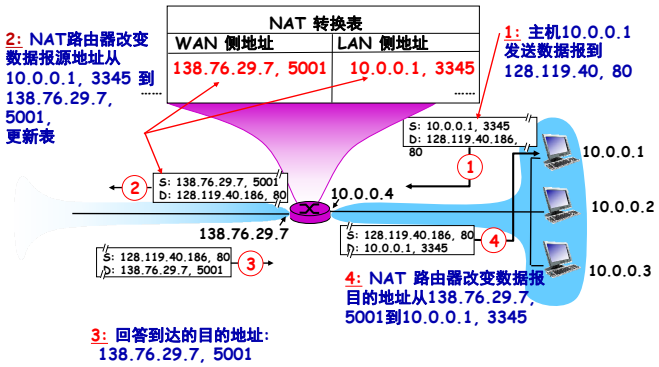
## NAT: 网络地址转换

实现: NAT 路由器必须:

- 转发数据报: 每个外出的数据报用(NAT IP地址, 新port #)代替(源IP地址, port #)
  - . . . 远程的客户机/路由器的响应，将用(NAT IP地址, new port #)作为目的地址
- 记住(在NAT转换表中)每个 (源IP地址, port #)到(NAT IP地址, 新port #) 转换对
- 入数据报: 对每个入数据报的地址字段用存储在NAT表中的(源IP地址, port #)替代对应的 (NAT IP地址, 新port #)



## NAT: 网络地址转换





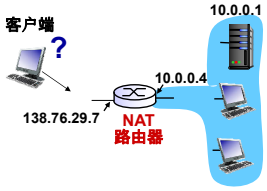

## NAT: 网络地址转换

- 16-bit 端口号字段:
  - 用一个LAN侧地址，同时支持60,000 并行连接!
- NAT 引起争议:
  - 应当用于进程编址而非主机编址
  - 路由器的处理上升为第三层
  - 违反了端到端通信原则
  - NAT妨碍了 P2P应用程序
  - 地址短缺应当由IPv6来解决而非NAT



## NAT 穿越问题

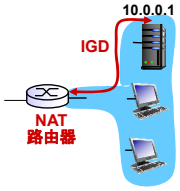
- 问题描述：外部客户端希望连接内网服务器
  - 服务器本地局域网地址为10.0.0.1 (客户端无法使用该地址为目的地址)
  - 外部仅可见NAT转换后的地址: 138.76.29.7
- 解决方案1:在NAT路由器静态配置NAT转换表，将特定端口号传入的连接请求转发到服务器。例如(138.76.29.7, port 2500) 总是转发到(10.0.0.1, port 80)





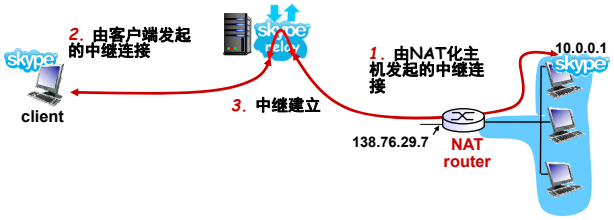

## NAT 穿越问题

- 解决方案2: 通用即插即用(UPnP) 因特网网关设备 (IGD) 协议。
- 允许内部网络NAT化的主机:
  - 获取公共 IP地址 (138.76.29.7)
  - 自动增加/移除（公共IP地址，公共端口号）和（私有IP地址，私有端口号）之间的映射
    - 自动化静态NAT端口映射配置
- 使得外部主机能使用TCP/UDP向NAT化的主机发起通信会话



## NAT 穿越问题

- 解决方案 3: 中继代理(例如Skype)
  - NAT化的客户端建立到中继代理服务器的连接
  - 外部网络的客户端也连接到中继代理服务器
  - 中继代理服务器在两个连接之间转发数据报






## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## 因特网控制报文协议 ICMP

- Internet可能发生的差错，导致数据报传输失败
  - 通信线路、处理器、目的主机关机
  - 数据报生命期变为0
  - Router拥塞 等等
- 必须报告错误
  - 对于同一个网络
    - 可以利用特殊硬件来报告错误
  - 对于互联网络
    - 发送方很难判断传送失败的原因，如错误发生在哪里
    - IP协议本身没有处理错误的机制



## 因特网控制报文协议 ICMP

- 为了提高IP数据报交付成功的机会，网际层使用 ICMP协议让主机或路由器向源端du报告差错情况和提供有关异常情况的报告。
- 但ICMP协议并没有指定对错误所应采取的措施，而是把差错交给应用程序或其他协议处理
  - 大部分由传输协议(TCP、UDP)处理



## 因特网控制报文协议 ICMP

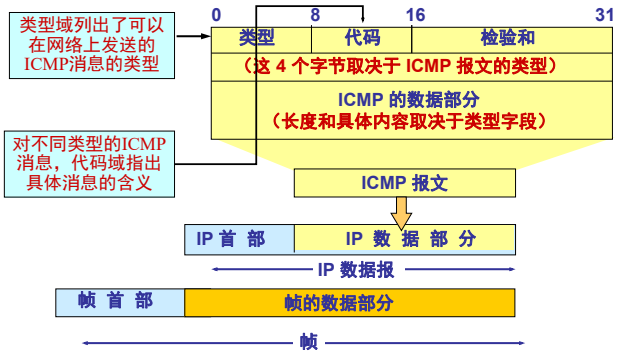
- 有几种情况不会发送ICMP报文:
  - 对 ICMP 差错报告报文不再发送 ICMP 差错报告报文。
  - 对第一个分片的数据报片的所有后续数据报片都不发送 ICMP 差错报告报文。
  - 对具有多播地址的数据报都不发送 ICMP 差错报告报文。
  - 对具有特殊地址（如127.0.0.0 或 0.0.0.0）的数据报不发送 ICMP 差错报告报文。



## 常用的ICMP 报文类型

- ICMP差错报告报文(划线表示不再使用)
  - 目的不可达（Destination Unreachable-type3）
  - ~~源抑制（Source Quench-type4）~~
  - 超时（Time Exceeded-Type11)
  - 参数问题（Parameter Problem-Type12）
  - 重定向（Redirect-Type5）
- ICMP询问报文(划线表示不再使用)
  - 回送请求和应答报文（Echo Request/Reply-Type0/8）
  - 时间戳请求和应答报文
    （TimeStamp Request/Reply-Type13/14）
  - ~~掩码地址请求和应答报文~~
    （Address Mask Request/Relpy-Type17/18）
  - ~~路由器询问和通告报文~~
    （Router Advertisement/Solicitation-Type9/10）

# ICMP 报文的格式及两级封装

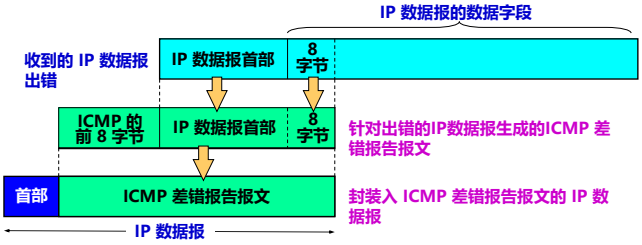





# 常用的ICMP 报文类型和编码

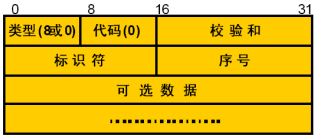
| 类型 | 编码 | 描述 |
|---|---|---|
| 0 | 0 | 回送请求 |
| 3 | 0 | 目的网络不可达 |
| 3 | 1 | 目的主机不可达 |
| 3 | 2 | 目的协议不可达 |
| 3 | 3 | 目的端口不可达 |
| 3 | 6 | 目的网络未知 |
| 3 | 7 | 目的主机未知 |
| 5 | 0 | 重定向 |
| 8 | 0 | 回送应答 |
| 11 | 0 | TTL超期 |
| 13 | 0 | 时间戳请求 |
| 14 | 0 | 时间戳应答 |

# ICMP 差错报告报文的数据字段的内容




# Echo Request/Reply查询消息

- 是两个最常用的ICMP消息类型。
- 主机或路由器可以向指定目的主机或路由器发送一个Echo Request报文，收到请求的目的机器向发送端返回一个Echo Reply报文，应答报文包含了请求报文中的数据拷贝。
- 可用于检测目的站的可达性与状态
- ICMP Echo Request/Reply 报文的格式





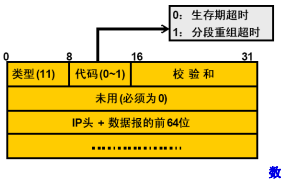

# 用Ping进行连通性测试

- Ping允许用户向目标主机/路由器发送一个或多个ICMP Echo Request消息，计算从探测开始到收到ICMP Echo Reply消息所花费的时间，来测试两个设备之间的网络连接是否正常。
- Ping能正确获得响应的四点保证
  - 源主机与目的主机之间的Router工作正常
  - 源主机为IP数据报选路的软件工作正常
  - 目的主机正在运行,IP和ICMP正常工作
  - 返回路径上Router有正确路由
  - 目标主机没有设置防火墙过滤ICMP Echo/Request消息



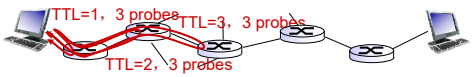
# Time Exceeded(超时)错误消息

- Time-To-Live Exceeded in Transit
  - 当TTL为0，而数据报还没有到达最终目的地
  - Router只能抛弃该数据报，并发送这条ICMP消息
  - TTL缺省值为30或更多，除非出现路由循环一般够用
- Fragment Reassembly Time Exceeded
  - 没有在收到数据报第一个分片的TTL时间终止前收到所有的分片，则发送该消息(Unix,60秒)，该消息将导致源发送端重发整个数据报
- 报文格式：



## 用Traceroute进行路径发现

- ICMP的Time Exceeded错误消息可用于检测循环或过长的路由，亦可与UDP报文配合，用于Traceroute路由发现
  - 发送端向目的主机发送UDP数据包(TTL为1)，第一跳路由器对TTL进行减量运算后TTL=0，数据报会被丢弃，第一跳路由器向发送端返回ICMP Time Exceeded错误消息 -> 获得第一跳路由器地址
  - 依此类推发送TTL=2，TTL=3，...，的UDP数据包，直到某个UDP数据包到达目的主机，目的主机返回ICMP Port Unreachable消息
  - 发送端主机记录下到达目的主机所用的往返时间






## 用Traceroute进行路径发现

- 几点说明：
  - 当UDP数据包到达目的主机后，即使TTL=1，目的主机也不会丢弃该数据报，因而不会返回ICMP Time Exceeded错误消息
  - 为了能通知源主机数据报已经到达目的主机，Traceroute选择一个很大的（大于30000）、目的主机上任何一个进程都不会使用的UDP目的端口号，设置为UDP数据包的端口号。
  - 这样，当数据报到达目的主机后，目的端将返回一条ICMP Port Unreachable错误消息。
  - 大部分UNIX缺省版本的Traceroute一次性发送3个UDP数据报，每个数据报的目的端口号都在增加
  - 其他操作系统的Traceroute发送TTL增量为1的ICMP Echo Request查询消息
  - 存在被防火墙阻断的可能



## 第4章 网络层

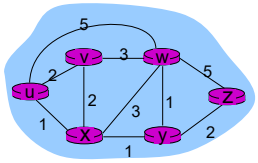
- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## 图的概念及选路

选路 协议

目的：决定从源到目的地通过网络的“好的路径”(路由器序列)




选路算法的图论抽象:

- 图中的结点是路由器
- 图中的边是物理链路
  - 链路代价: 时延，费用或拥塞等级

- 图: G = (N,E)
- 路由器集合 = { u, v, w, x, y, z }
- 链路的集合 ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }
- “好的”路径:
  通常意味着最小费用的路径
  其他定义也是可能的



## 选路算法分类

使用全局的？或分散的信息?

- 全局的:所有路由器具有完全的拓扑、链路费用信息
  - “链路状态”算法
- 分散的: 路由器知道到达其直接物理相连的邻居的链路费用
  - 通过迭代计算，与邻居交换信息
  - “距离矢量”算法

静态的或动态的?

- 静态: 路由随时间缓慢变化，手工配置，优先级高
- 动态: 路由更快地变化
  - 周期的自动更新
  - 适应链路费用变化



## 一类链路状态选路算法

Dijkstra算法

- 所有结点知道网络拓扑、链路费用
  - 经“链路状态广播” (洪泛)完成
  - 所有结点具有相同信息
- 从一个结点(源)到所有其他结点计算最低费用路径
  - 给出对这些结点的转发表
- 迭代: k次迭代后，得知到k个目的地的最低费用路径

概念:

- c(x,y): 从结点x到y的链路费用; 如果不是直接邻居= ∞
  D(v):从源到目的地v路径费用的当前值
- p(v): 从源到v沿路径的，v的前序结点
- N': 已知在最小费用路径中的结点集合

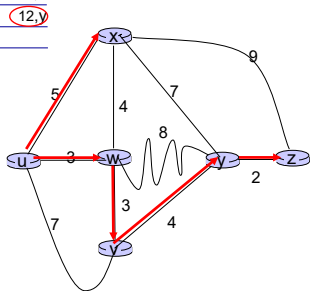
# 链路状态路由算法——Dijsktra算法

```
1 初始化:
2   N' = {u}
3   对所有结点v
4    if v 临近 u
5       then D(v) = c(u,v)
6    else D(v) = ∞
7
8  Loop
9   找出w不在N'中使得D(w)最小
10  将w加入N'
11  对于所有v临近w并不在N'中，更新D(v):
12     D(v) = min( D(v), D(w) + c(w,v) )
13  /* 到v的新费用或是到v的老费用或到w加上从w到v的已知
最短路费用*/
15 until 所有结点在 N'中
```



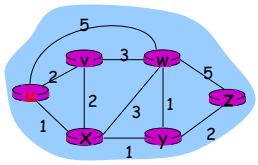
# Dijkstra 算法: 例子

| Step | N' | D(v) p(v) | D(w) p(w) | D(x) p(x) | D(y) p(y) | D(z) p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 7,u | 3,u | 5,u | ∞ | ∞ |
| 1 | uw | 6,w | | 5,u | 11,w | ∞ |
| 2 | uwx | 6,w | | | 11,w | 14,x |
| 3 | uwxv | | | | 10,v | 14,x |
| 4 | uwxvy | | | | | 12,y |
| 5 | uwxvyz | | | | | |




# Dijkstra 算法另一个: 例子

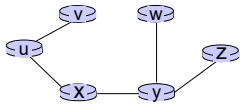
| 步骤 | N' | D(v),p(v) | D(w),p(w) | D(x),p(x) | D(y),p(y) | D(z),p(z) |
|---|---|---|---|---|---|---|
| 0 | u | 2,u | 5,u | 1,u | ∞ | ∞ |
| 1 | ux | 2,u | 4,x | | 2,x | ∞ |
| 2 | uxy | 2,u | 3,y | | | 4,y |
| 3 | uxyv | | 3,y | | | 4,y |
| 4 | uxyvw | | | | | 4,y |
| 5 | uxyvwz | | | | | |






# Dijkstra 算法另一个: 例子

以u为根结点的最短路径树:




u的转发表:

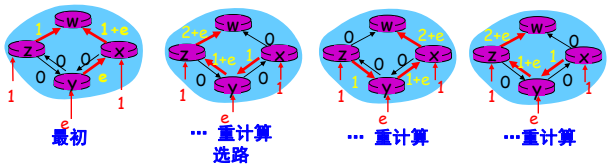
| destination | link |
|---|---|
| v | (u,v) |
| x | (u,x) |
| y | (u,x) |
| w | (u,x) |
| z | (u,x) |

# Dijkstra算法, 讨论

算法复杂性: n个结点

- 每次迭代: 需要检查所有不在集合N'中的结点w
- n(n+1)/2 对比: $O(n^2)$
- 更有效的实现是可能的: O(nlogn)

可能存在振荡问题:假设链路费用是该链路承载的流量






# 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议
    BGP

## 距离矢量算法(1)

**Bellman-Ford方程 (动态规划)**

**定义 $d_x(y)$ := 从x到y最低费用（距离）路径的费用（距离）**

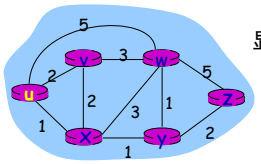
**则** $d_x(y) = \min \{c(x,v) + d_v(y) | 对所有邻居 \}$

x到邻居v的费用

从邻居v到达目的y的费用



## Bellman-Ford 例子




显然，$d_v(z) = 5$, $d_x(z) = 3$, $d_w(z) = 3$

根据B-F 方程:

$$d_u(z) = \min \{ c(u,v) + d_v(z), \; c(u,x) + d_x(z), \; c(u,w) + d_w(z) \}$$
$$= \min \{2 + 5, \; 1 + 3, \; 5 + 3\} = 4$$

**结点u得到了到达结点z的最短路径的下一跳是x→ 转发表**



## 距离矢量算法(2)

**结点x需要维护下列信息:**

- 对于每个直接相连的邻居$v$，$x$到$v$的费用c(x,y)
- 结点$x$的距离向量，即 $D_x=[D_x(y)$，for each node y ∈ N]，表示$x$到结点集合$N$中所有目的地为结点$y$的费用的估计值
- 结点$x$的每个邻居的距离向量$D_v=[D_v(y)$，for each node y ∈ N]

**算法基本思路:**

- 每个结点周期性的向邻居发送自己的距离向量D$x$
- 当结点$x$接收到任何一个邻居$v$新的距离向量，使用B-F方程更新自己的距离向量： $D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\}$ for each node y ∈ N
- 如果有D$v$导致D$x$更新，则结点$x$向所有邻居发送更新后的D$x$



## 距离矢量算法(4)

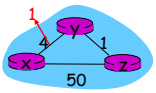
**迭代、异步:** 每次本地迭代由下列引起:

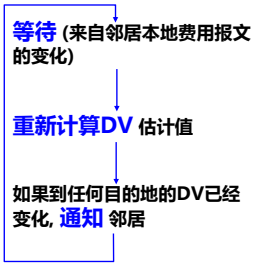
- 本地链路费用改变
- DV从邻居更新报文

**分布式:**

- 每个结点仅当其DV改变时通知邻居
  - 如果必要，邻居则通知它们的邻居

**每个结点:**





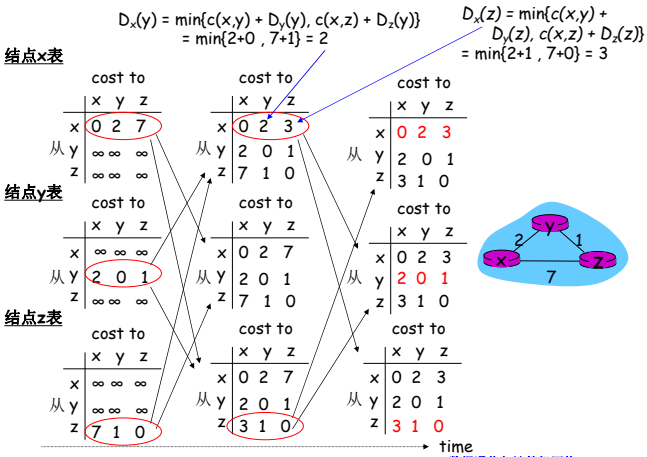

## 距离矢量算法(3)

距离向量（DV）算法

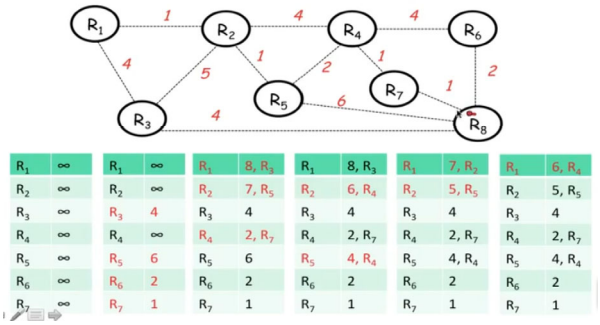
在每个结点 $x$:

```
1  Initialization:
2     for all destinations y in N:
3        Dx(y) = c(x,y)    /* if y is not a neighbor then c(x,y) = ∞ */
4     for each neighbor w
5        Dw(y) = ? for all destinations y in N
6     for each neighbor w
7        send distance vector Dx = [Dx(y): y in N] to w
8
9  loop
10    wait (until I see a link cost change to some neighbor w or
11          until I receive a distance vector from some neighbor w)
12
13    for each y in N:
14       Dx(y) = minv{c(x,v) + Dv(y)}
15
16    if Dx(y) changed for any destination y
17       send distance vector Dx = [Dx(y): y in N] to all neighbors
18
19 forever
```





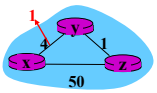
## An example




| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| $R_1$ | ∞ | $R_1$ | ∞ | $R_1$ | 8, $R_3$ | $R_1$ | 8, $R_3$ | $R_1$ | 7, $R_2$ | $R_1$ | 6, $R_4$ |
| $R_2$ | ∞ | $R_2$ | ∞ | $R_2$ | 7, $R_5$ | $R_2$ | 6, $R_4$ | $R_2$ | 5, $R_5$ | $R_2$ | 5, $R_5$ |
| $R_3$ | ∞ | $R_3$ | 4 | $R_3$ | 4 | $R_3$ | 4 | $R_3$ | 4 | $R_3$ | 4 |
| $R_4$ | ∞ | $R_4$ | ∞ | $R_4$ | 2, $R_7$ | $R_4$ | 2, $R_7$ | $R_4$ | 2, $R_7$ | $R_4$ | 2, $R_7$ |
| $R_5$ | ∞ | $R_5$ | 6 | $R_5$ | 6 | $R_5$ | 4, $R_4$ | $R_5$ | 4, $R_4$ | $R_5$ | 4, $R_4$ |
| $R_6$ | ∞ | $R_6$ | 2 | $R_6$ | 2 | $R_6$ | 2 | $R_6$ | 2 | $R_6$ | 2 |
| $R_7$ | ∞ | $R_7$ | 1 | $R_7$ | 1 | $R_7$ | 1 | $R_7$ | 1 | $R_7$ | 1 |

## 距离矢量算法: 链路费用改变

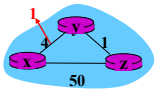
- 如果结点检测到链路费用改变
  - 更新路由信息，重新计算距离向量
  - 如果距离向量改变，通告邻居
  - 好消息传播得快
  - 坏消息传播得慢—“计数到无穷”问题!
    - 在算法稳定前，迭代44 次: 参见课文






## 距离矢量算法: 链路费用改变

- 好消息传播得快




在时刻$t_0$：$y$ 检测到链路费用变化，更新其$DV$，并将此变化通告它的邻居

在时刻$t_1$：$z$ 接收到来自 $y$ 的更新报文，并据此更新了 $z$ 自己的距离表.
$z$ 计算出到 $x$ 的新的最低费用，向其邻居发送了它新的$DV$

在时刻 $t_2$：$y$ 接收到$z$ 的更新消息，更新它的距离表，$y$ 的最低费用没有改变，因此 $y$ 不发送任何报文给 $z$ ，算法进入静止状态



## 距离矢量算法: 链路费用改变

- 坏消息传播得慢—“计数到无穷”问题!

### A Problem with Bellman-Ford

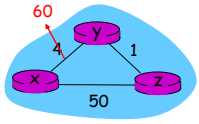
“Bad news travels slowly”


Consider the calculation of distances to $R_4$:

| Time | $R_1$ | $R_2$ | $R_3$ |
|---|---|---|---|
| 0 | 3,$R_2$ | 2,$R_3$ | 1, $R_4$ |
| 1 | 3,$R_2$ | 2,$R_3$ | 3,$R_2$ |
| 2 | 3,$R_2$ | 4,$R_3$ | 3,$R_2$ |
| 3 | 5,$R_2$ | 4,$R_3$ | 5,$R_2$ |
| … | “Counting to infinity” | | … |

Link $R_3$ → $R_4$ fails

## 距离矢量: 链路费用变化




解决无穷计数问题

1、设置迭代次数限制（=16次结束）

2、毒性逆转:

- 如果Z路由通过Y得到 X :
  - Z通告Y，自己到X的距离是∞ ，因此Y不能经Z路由到X
- 这将完全解决计数到无穷问题? 涉及到3个以上结点的环路不能被毒性逆转技术检测到



## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议
    - BGP

## 等级选路

之前的选路研究至此，是基于如下假设：

- 所有路由器是等同的
- 网络“扁平”

… 实践中并不真实，必须引入等级选路，因为：

真正的互联网: 具有几亿个目的地:

- 在路由表中不能存储所有的目的地!
- 路由信息交换和路由选择将成为链路带宽杀手!

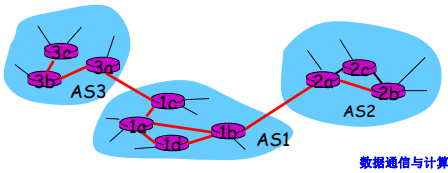
自治管理

- 互联网 = 网络的网络
- 每个网络管理员希望能控制自己网络中的选路，保护内部网络细节



## 等级选路

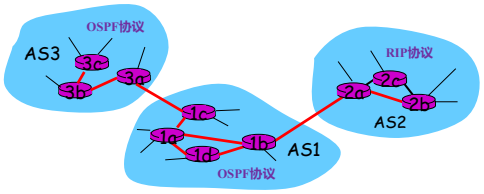
- 将一组处在相同管理控制下的路由器聚合成为 “自治系统” (Autonomous System，AS)
- 一个AS由全局唯一的AS号（ASN)标识，ASN由ICANN区域管理注册局分配
- 同一个AS中的路由器运行相同的路由选择算法
  - “intra-AS” 选路协议






## 等级选路

- AS之间的通信必须运行相同的自治系统之间路由选择协议
  - “Inter-AS”路由选择协议
- 网关路由器：位于AS边缘，通过链路连接其他AS的网关路由器






## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP

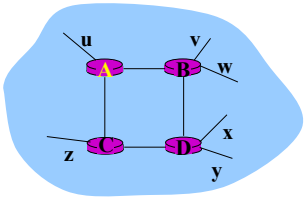


## AS内部路由协议

- Internet采用层次路由
- AS内部路由协议也称为内部网关协议IGP
- 最常见的AS内部路由协议
  - 路由信息协议RIP(Routing Information Protocol)
  - 开放最短路径优先OSPF(Open Shortest Path First)

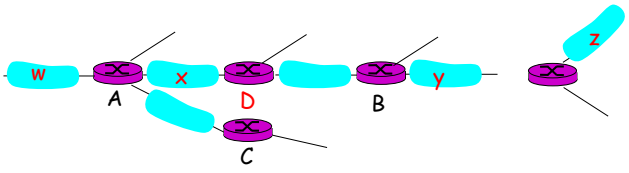
## RIP ( 分布式、基于距离向量算法)

- 距离度量为跳的数量(最大 = 15跳，16表示不可达，仅适用于小型互联网);
- 距离矢量每30秒邻居之间经通告报文交换各自的路由表
- 结点检测到网络拓扑变化后，向邻居通告拓扑变化后的信息
- 在180 sec后，如果没有收到邻居通告 -->该邻居/链路被标记为不可达路由，距离设置为16




| 距离 | 跳 |
|---|---|
| u | 1 |
| v | 2 |
| w | 2 |
| x | 3 |
| y | 3 |
| z | 2 |

# RIP: 例子




D的路由表

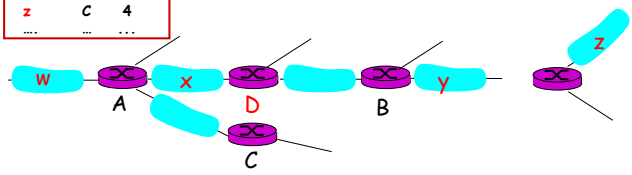
| 目的网络 | 下一个路由器 | 到目的地的跳数 |
|---|---|---|
| w | A | 2 |
| y | B | 2 |
| z | B | 7 |
| x | -- | 1 |
| .... | .... | .... |



# RIP: 例子

| 目的地 | 下一个 | 跳 |
|---|---|---|
| w | - | - |
| x | - | - |
| z | C | 4 |
| .... | ... | ... |

从A到D通告




D的路由表

| 目的网络 | 下一个路由器 | 到目的地的跳数 |
|---|---|---|
| w | A | 2 |
| y | B | 2 |
| z | ~~B~~ A | ~~7~~ 5 |
| x | -- | 1 |
| .... | .... | |



# RIP: 例子

- 【谢版例4-5】已知路由器 R6 有表 4-9(a) 所示的路由表。现在收到相邻路由器 R4 发来的路由更新信息，如表 4-9(b) 所示。试更新路由器 R6 的路由表。

表 4-9(a) 路由器 $R_6$ 的路由表

| 目的网络 | 距离 | 下一跳路由器 |
|---|---|---|
| Net2 | 3 | $R_4$ |
| Net3 | 4 | $R_5$ |
| ... | ... | ... |

表 4-9(b) R4 发来的路由更新信息

| 目的网络 | 距离 | 下一跳路由器 |
|---|---|---|
| Net1 | 3 | $R_1$ |
| Net2 | 4 | $R_2$ |
| Net3 | 1 | 直接交付 |

计算更新

表 4-9(c) 修改后的表 4-9(b)

| 目的网络 | 距离 | 下一跳路由器 |
|---|---|---|
| Net1 | 4 | $R_4$ |
| Net2 | 5 | $R_4$ |
| Net3 | 2 | $R_4$ |

表 4-9(d) 路由器 $R_6$ 更新后的路由表

| 目的网络 | 距离 | 下一跳路由器 |
|---|---|---|
| Net1 | 4 | $R_4$ |
| Net2 | 5 | $R_4$ |
| Net3 | 2 | $R_4$ |
| ... | ... | ... |

# RIP: 例子

- 【谢版例子】路由表更新

从C来的RIP报文

| | |
|---|---|
| Net2 | 4 |
| Net3 | 8 |
| Net6 | 4 |
| Net8 | 3 |
| Net9 | 5 |

增加跳数以后从C来的RIP报文

| | |
|---|---|
| Net2 | 5 |
| Net3 | 9 |
| Net6 | 5 |
| Net8 | 4 |
| Net9 | 6 |

Net1: 没有新信息，不变
Net2: 相同的下一跳，替换
Net3: 一条新路由，增加
Net6: 不同的下一跳，新跳数小，替换
Net8: 不同的下一跳，跳数相同，不变
Net9: 不同的下一跳，新跳数大，不变

旧路由表

| | | |
|---|---|---|
| Net1 | 7 | A |
| Net2 | 2 | C |
| Net6 | 8 | F |
| Net8 | 4 | E |
| Net9 | 4 | F |

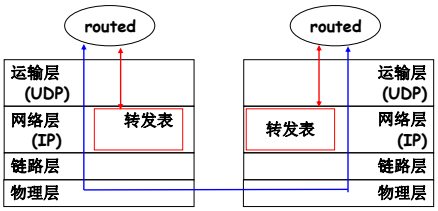
更新算法

新路由表

| | | |
|---|---|---|
| Net1 | 7 | A |
| Net2 | 5 | C |
| Net3 | 9 | C |
| Net6 | 5 | C |
| Net8 | 4 | E |
| Net9 | 4 | F |

# RIP路由表处理及报文格式

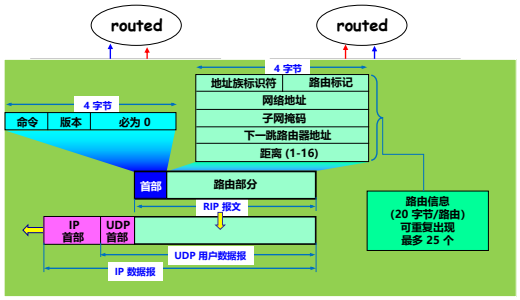
- RIP路由表由称为route-d (守护进程)的应用级进程管理，封装在UDP分组中发送，周期地重复






# RIP路由表处理及报文格式

- RIP路由表由称为route-d (守护进程)的应用级进程管理，封装在UDP分组中发送，周期地重复



## RIP报文格式详解

- RIP2 报文由首部和路由部分组成。
- RIP2 报文中的路由部分由若干个路由信息组成。每个路由信息需要用 20 个字节。地址族标识符（又称为地址类别）字段用来标志所使用的地址协议。
- 路由标记填入自治系统的号码，这是考虑使 RIP 有可能收到本自治系统以外的路由选择信息。
- 再后面指出某个网络地址、该网络的子网掩码、下一跳路由器地址以及到此网络的距离。

## RIP报文格式详解

- 一个 RIP 报文最多可包括 25 个路由，因而 RIP 报文的最大长度是 4+20 x25=504 字节。如超过，必须再用一个 RIP 报文来传送。
- RIP2 具有简单的鉴别功能。
  - 若使用鉴别功能，则将原来写入第一个路由信息（20 个字节）的位置用作鉴别。
  - 在鉴别数据之后才写入路由信息，但这时最多只能再放入 24 个路由信息。

## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## OSPF (开放最短路优先)

- “开放”：公共可用
- 使用分布式链路状态算法构建AS的完整拓扑
  - 路由器使用Dijkstra确定以自身为根到所有子网的最短路径树
  - 路由器周期性（每30秒）向整个AS广播（“洪泛”）链路状态信息
    - 通告中每一个入口对应一个邻居
    - 信息携带在OSPF报文中，直接封装在IP数据报(而不是TCP或UDP)



## OSPF “先进的”特色 (RIP所不具有的)

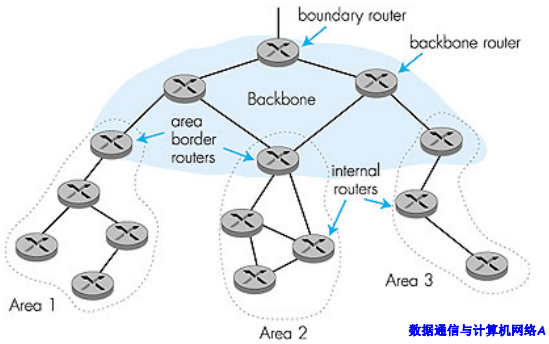
- 安全性: 只有受信任的路由器能参与AS内部的路由信息交换
- 运行多条费用相同的路径(在RIP中仅一条路径)
- 对每条链路，对不同的TOS(服务类型)，设置多种费用度量(如卫星链路费用置为用于尽力而服务为“低”，高为实时服务)
- 集成了单播和多播支持:
  - 多播OSPF (MOSPF)使用与OSPF相同的拓扑数据库
- OSPF支持对大规模的AS分层



## 层次OSPF

两级层次: 本地(Area), 主干(backbone)

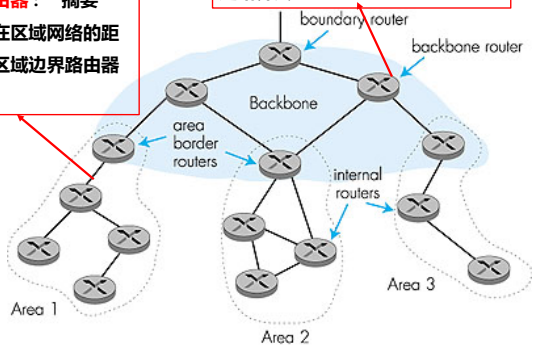
- 链路状态通告仅在本地
- 每个结点（路由器）掌握详细的区域拓扑；但仅知道到其他区域网络的方向(最短路)



## 层次OSPF

区域边界路由器：“摘要”到达自己所在区域网络的距离，向其他区域边界路由器通告

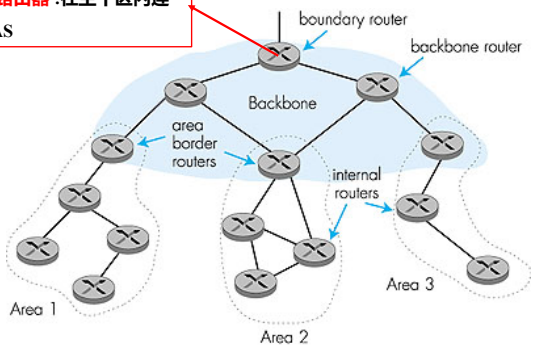
主干路由器:在主干区内 运行OSPF选路算法






## 层次OSPF

AS边界路由器:在主干区内连接其他AS





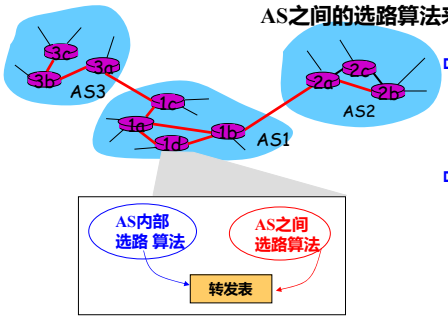

## 第4章 网络层

- 4. 1 概述
- 4.2 虚电路和数据报网络
- 4.3 路由器的构成
- 4.4 IP: 网际协议
  - 数据报格式
  - IPv4编址
  - NAT
- 4.5 ICMP协议
- 4.6 选路算法
  - 链路状态
  - 距离矢量
- 4.7 互联网中的选路
  - 等级选路
  - 内部网关路由选择协议
    - RIP
    - OSPF
  - 边界网关路由选择协议 BGP



## 互联的AS

- AS内部路由器1d的转发表由AS内部、AS之间的选路算法来共同配置：
  - AS内部选路算法设置AS内部目的网络表项
  - AS之间选路算法和AS内部选路算法，共同设置对AS外部的目的网络的表项





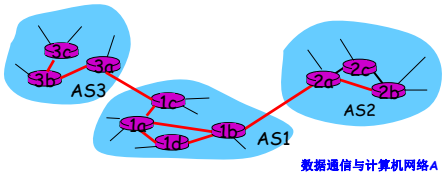

## AS间的路由任务

- 假定在AS1中的路由器接收到的数据报，其目的地址在AS1外部
  - 路由器应当将分组朝着哪个网关路由器转发

AS1需要知道:

1. 通过AS2、AS3可分别到达哪些目的网络
2. 传播这些网络可达性信息到AS1中所有路由器

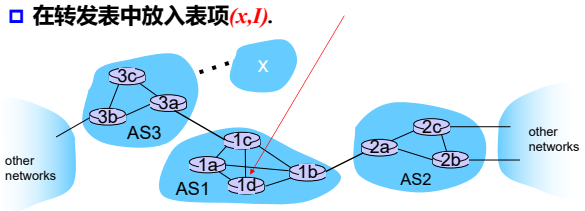
由AS间选路协议完成!





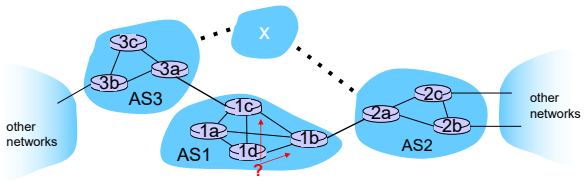

## 例子: 设置路由器1d的转发表

- 假定AS1从AS间协议得知子网 *x* 从AS3(网关1c)可达，而AS2不可达
  - AS间协议传播可达性信息到所有内部路由器
- 路由器1d从AS内部路由信息决定，确定到1c的最低费用路径的接口是它的接口*I*
- 在转发表中放入表项*(x,I)*.



## 例子: 设置路由器1d的转发表

- 假定AS1从AS间协议得知子网 *x* 从AS3(网关1c)和AS2（网关1b）可达
- 为了配置转发表, 路由器1d必须决定对目的地x ，它应当将分组转发向哪个网关
- 这也是AS间选路协议的工作!.






## 例子: 在多个AS之间选择

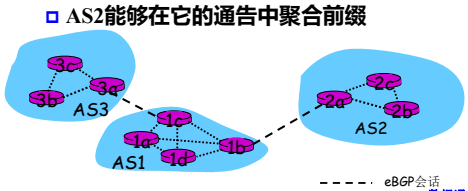
- 热土豆选路: 将分组奉送给最近的网关路由器

从AS间协议得知，子网x经多个网关可达 → 使用来自AS内部协议选路信息，计算到每个网关的最低费用路径 → 热土豆选路: 选择具有最小费用的网关 → 从转发表找到通向最低费用网关的接口I，将表项 (x,I) 添加在转发表中



## BGP基础

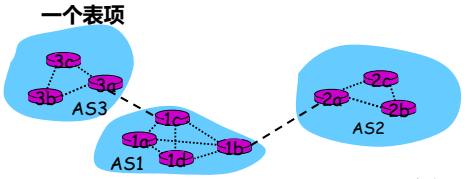
- 路由器的对(BGP对等方)交换选路信息通过半永久的TCP连接: BGP会话
- BGP会话不对应着物理链路(覆盖网络)
- 当AS2向AS1通告一个前缀, AS2承诺它将转发任何指向该前缀的数据报
  - AS2能够在它的通告中聚合前缀






## 分发可达性信息

- 在3a和1c之间有eBGP 会话，AS3向AS1发送前缀可达性信息
- 1c则能使用iBGP来向AS1中的所有路由器分发这种新前缀可达信息
- 1b则能经1b到2a的eBGP会话向AS2重新通告新的可达信息
- 当路由器知道了一个新前缀，它将在其转发表中为该前缀创建一个表项






# 本章结束