第4章 网络层

- 4.1 网络层提供的服务
- 4.2 网际协议IP
- 4.3 划分子网和构造超网
- 4.4 网际控制报文协议ICMP
- 4.5 因特网的路由选择协议
- 4.6 IP多播
- 4.7 其他网络举例

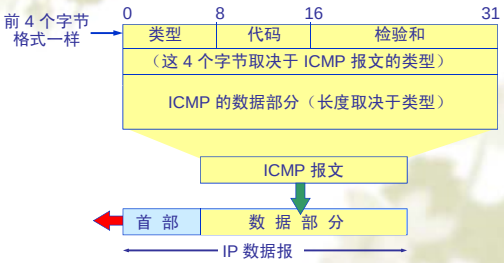
1

网际控制报文协议ICMP

- 为了提高 IP 数据报交付成功的机会，在网际层使用了网际控制报文协议 ICMP (Internet Control Message Protocol)。
- ICMP 允许主机或路由器报告差错情况和提供有关异常情况的报告。
- ICMP 不是高层协议，而是 IP 层的协议。
- ICMP 报文作为 IP 层数据报的数据，加上数据报的首部，组成 IP 数据报发送出去。

2

ICMP报文的格式



3

ICMP报文的种类

- ICMP 报文的种类有两种，即 ICMP 差错报告报文和 ICMP 询问报文。
- ICMP 报文的前 4 个字节是统一的格式，共有三个字段：即类型、代码和检验和。接着的 4 个字节的内容与 ICMP 的类型有关。
- 类型域用来指明消息的类型，有些消息还用代码域进一步定义说明。例如：类型为3的消息表示“目的不可达”的错误报告，每个消息的代码域进一步说明是“网络不可达”、“主机不可达”还是其他。

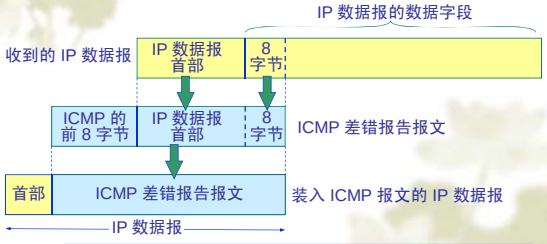
4

ICMP 差错报告报文共有 5 种

- 差错报告：向 不需要应答。
  - 网络不可达、主机不可达、协议不可达、端口不可达、需分片但DF为1、源路由失败等6种
- 差错报告的类
  - 终点不可达
  - 源点抑制(S
  - 时间超过
  - 参数问题
  - 改变路由（重定向） (Redirect)
  - 路由器收到TTL为0的数据 器或主机拥塞时，向源报告
  - 收到的数据报首部有字段不正确时
- 差错报告的方法
  - 在发送一个ICMP错误消息时，消息内容同时携带引起错误的IP分组的分组头及分组数据域中的前8个字节（正好含有端口号、报文序号等信息）。

5

ICMP差错报告报文的数据字段的内容



注：IP数据报数据部分的前8个字节，恰好对应着传输层的端口号（TCP/UDP）以及TCP的序号（sequence），这些信息对源站的高层协议是有用的

6

不应发送ICMP差错报告报文的几种情况

- 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。

7

ICMP询问报文有两种

- 询问报文：采用请求/回答方式用来请求一些消息。
- 询问报文的类型
  - 回送请求和回答报文
  - 时间戳请求和回答报文
- 下面的几种 ICMP 报文不再使用
  - 信息请求与回答报文
  - 掩码地址请求和回答报文
  - 路由器询问和通告报文

由主机或路由器向一个特定的目的主机发出询问，收到此报文的机器

请求某个主机或路由器回答当前的日期和时间。用于时钟同步或测量时间

8

ICMP的应用举例

PING (Packet InterNet Groper)

Internet数据包探测器

- PING 用来测试两个主机之间的连通性。
- PING 使用了 ICMP 回送请求与回送回答报文。
- PING 是应用层直接使用网络层 ICMP 的例子，它没有通过运输层的 TCP 或UDP。

9

PING的应用举例

```
C:\Documents and Settings\XXR>ping mail.sina.com.cn

Pinging mail.sina.com.cn [202.108.43.230] with 32 bytes of data:

Reply from 202.108.43.230: bytes=32 time=368ms TTL=242
Reply from 202.108.43.230: bytes=32 time=374ms TTL=242
Request timed out.
Reply from 202.108.43.230: bytes=32 time=374ms TTL=242

Ping statistics for 202.108.43.230:
    Packets: Sent = 4, Received = 3, Lost = 1 (25% loss),
Approximate round trip times in milli-seconds:
    Minimum = 368ms, Maximum = 374ms, Average = 372ms
```

10

Traceroute的应用举例

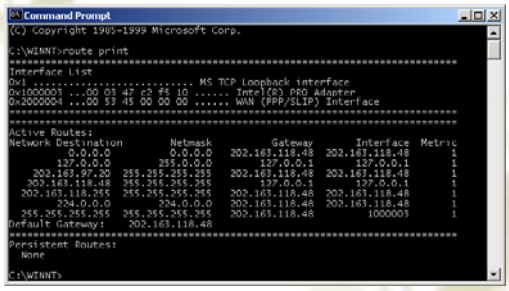
```
C:\Documents and Settings\XXR>tracert mail.sina.com.cn

Tracing route to mail.sina.com.cn [202.108.43.230]
over a maximum of 30 hops:

  1    24 ms    24 ms    23 ms  222.95.172.1
  2    23 ms    24 ms    22 ms  221.231.204.129
  3    23 ms    22 ms    23 ms  221.231.206.9
  4    24 ms    23 ms    24 ms  202.97.27.37
  5    22 ms    23 ms    24 ms  202.97.41.226
  6    28 ms    28 ms    28 ms  202.97.35.25
  7    50 ms    50 ms    51 ms  202.97.36.86
  8   308 ms   311 ms   310 ms  219.158.32.1
  9   307 ms   305 ms   305 ms  219.158.13.17
 10   164 ms   164 ms   165 ms  202.96.12.154
 11   322 ms   320 ms  2988 ms  61.135.148.50
 12   321 ms   322 ms   320 ms  freemail43-230.sina.com [202.108.43.230]

Trace complete.
```

第4章 网络层

- 4.1 网络层提供的服务
- 4.2 网际协议IP
- 4.3 划分子网和构造超网
- 4.4 网际控制报文协议ICMP
- 4.5 因特网的路由选择协议
- 4.6 IP多播
- 4.7 其他网络举例

12

路由表示例



路由选择协议的基本概念

- 理想的路由算法要求：
  - 算法必须是正确的和完整的。
  - 算法在计算上应简单。
  - 算法应能适应通信量和网络拓扑的变化，要有自适应性。
  - 算法应具有稳定性。即在网络通信量和网络拓扑相对稳定的情况下，路由算法应能收敛于一个可以接受的解，而不会使路由不停地变化。
  - 算法应是公平的。除了少数高优先级用户外，算法对用户平等。
  - 算法应是最佳的。以最低的“代价”实现路由算法。

关于“最佳路由”

- 不存在一种绝对的最佳路由算法。
- 所谓“最佳”只能是相对于某一种特定要求下得出的较为合理的选择而已。
- 实际的路由选择算法，应尽可能接近于理想的算法。
- 路由选择是个非常复杂的问题
  - 它是网络中的所有结点共同协调工作的结果。
  - 路由选择的环境往往是不断变化的，而这种变化有时无法事先知道。

从路由算法的自适应性考虑

- 从路由算法能否随着网络的通信量或拓扑结构的变化而自适应的调整，可将路由算法分为：
  - 静态路由选择策略：即非自适应路由选择，其特点是简单和开销较小，但不能及时适应网络状态的变化。
  - 动态路由选择策略：即自适应路由选择，其特点是能较好地适应网络状态的变化，但实现起来较为复杂，开销也比较大。

分层次的路由选择协议

- 因特网采用分层次的路由选择协议。
- 因特网的规模非常大。如果让所有的路由器知道所有的网络应怎样到达，则这种路由表将非常大，处理起来也太花时间。而所有这些路由器之间交换路由信息所需的带宽就会使因特网的通信链路饱和。
- 许多单位不愿意外界了解自己单位网络的布局细节和本部门所采用的路由选择协议（这属于本部门内部的事情），但同时还希望连接到因特网上。

自治系统 (Autonomous System)

- 大型网络如因特网，会被分解成为多个自治系统AS。一个自治系统内的所有网络都属于一个行政单位来管辖。
- 一个自治系统，其最重要的特点就是自治系统有权自主地决定在本系统内应采用何种路由选择协议。
- 一个自治系统的所有路由器在本自治系统内都必须是连通的。
- 尽管一个 AS 使用了多种内部路由选择协议和度量，但重要的是一个 AS 对其他 AS 表现出的是一个单一的和一致的路由选择策略。

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，当数据报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。

19

自治系统和
内部网关协议、外部网关协议

自治系统之间的路由选择也叫做
域间路由选择(interdomain routing)，
在自治系统内部的路由选择叫做
域内路由选择(intradomain routing)

20

这里要指出两点

- 因特网的早期 RFC 文档中未使用“路由器”而是使用“网关”这一名词。但是在新的 RFC 文档中又使用了“路由器”这一名词。应当把这两个属于当作同义词。
- IGP 和 EGP 是协议类别的名称。但 RFC 在使用 EGP 这个名词时出现了一点混乱，因为最早的一个外部网关协议的协议名字正好也是 EGP。因此在遇到名词 EGP 时，应弄清它是指旧的协议 EGP 还是指外部网关协议 EGP 这个类别。

21

因特网的路由选择协议

- 内部网关协议 IGP：具体的协议有多种，如 RIP 和 OSPF 等。
- 外部网关协议 EGP：目前使用的协议就是 BGP。

22

内部网关协议 RIP
(Routing Information Protocol)

- 路由[各节点通过相互交换路由信息，在本地独立的确定自己的路由表]最先得到[到达目的网络所需的费用]路径的[根据拓扑结构、]地址[信量的变化来]改变其路由选择
- 分布式的基于距离向量的动态路由选择协议。
- RIP 协议要求网络中的每一个路由器都要维护从它自己到其他每一个目的网络的距离记录。
- 通过与相邻路由器定期交换路由信息来更新自己的路由表

23

“距离”的定义

- 从一路由器到直接连接的网络的距离定义为 1。
- 从一个路由器到非直接连接的网络的距离定义为所经过的路由器数加 1。
- RIP 协议中的“距离”也称为“跳数”(hop count)，因为每经过一个路由器，跳数就加 1。
- 这里的“距离”实际上指的是“最短距离”。

24

“距离”的定义

- RIP 认为一个好的路由就是它通过的路由器的数目少，即“距离短”。
- RIP 允许一条路径最多只能包含 15 个路由器。
- “距离”的最大值为16 时即相当于不可达。可见 RIP 只适用于小型互联网。
- RIP 不能在两个网络之间同时使用多条路由。RIP 选择一个具有最少路由器的路由（即最短路由），哪怕还存在另一条高速(低时延)但路由器较多的路由。

25

RIP 协议的三个要点

- 和哪些路由器交换信息？
  - 仅和相邻路由器交换信息
- 交换什么信息？
  - 交换的信息是当前本路由器所知道的全部信息，即自己的路由表
- 在什么时候交换信息？
  - 按固定的时间间隔交换路由信息，例如每隔30秒

26

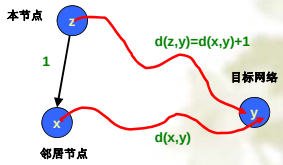
RIP路由表的信息

- 路由表的主要信息
  - 到某个网络的距离（即最短距离）
  - 应经过的下一跳地址
- 路由表的更新原则是找出到每个目的网络的最短距离
- RIP使用的更新算法称为距离向量算法

27

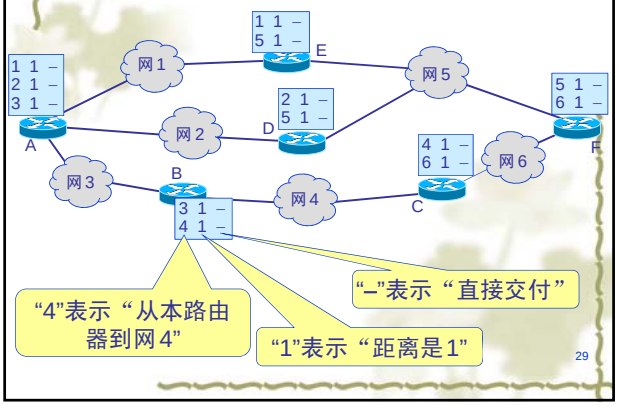
距离向量的更新过程

- 本节点为z，接收到来自邻居节点x的距离表，获知x到网络y的距离为d(x,y)，因为x与z相邻，则路由器z经过x到y的距离为d(x,y)+1
- 节点z根据其它邻居 x’发来的信息重复计算d(x’,y)+1
- 取z到y计算距离的最小值来更新本节点的路由表



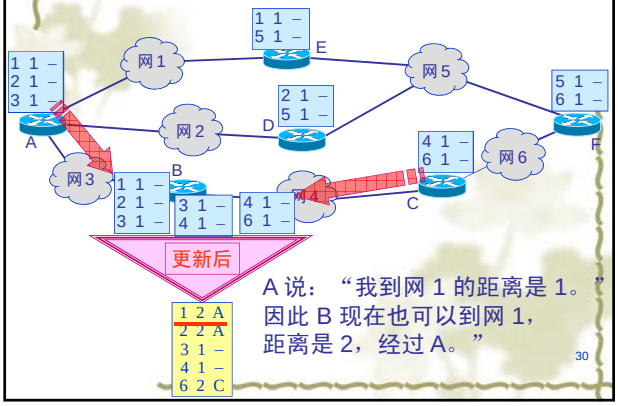
28

一开始，各路由器只有到直接连接的网络信息



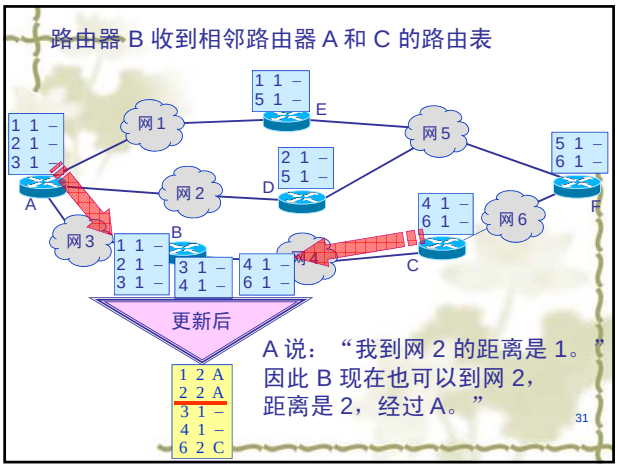
29

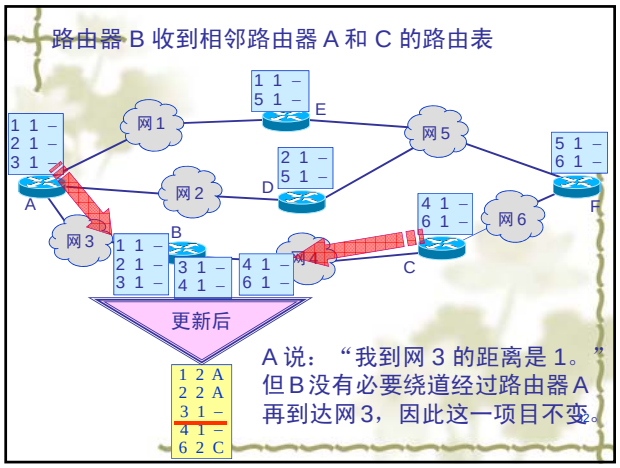
路由器 B 收到相邻路由器 A 和 C 的路由表

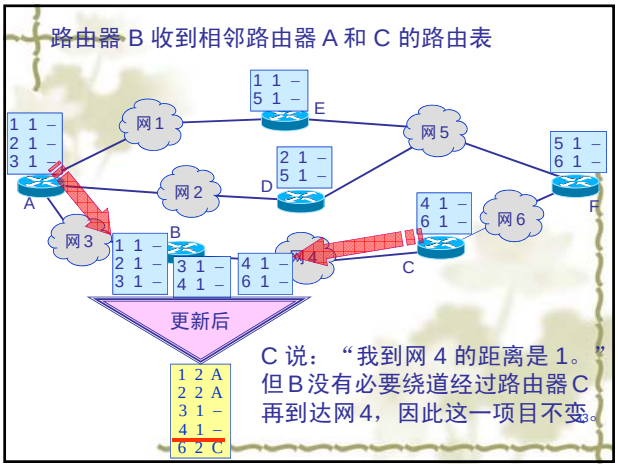


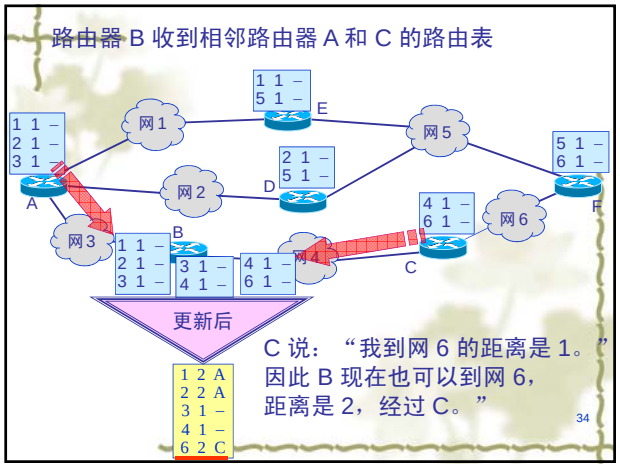
A 说：“我到网 1 的距离是 1。”因此 B 现在也可以到网 1，距离是 2，经过 A。”

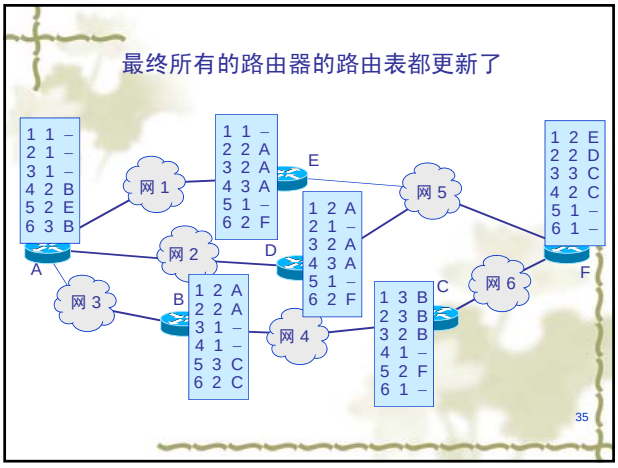
30











RIP路由表的建立

- 路由器在刚刚开始工作时，只知道到直接连接的网络的距离（此距离定义为1）。
- 以后，每一个路由器也只和数目非常有限的相邻路由器交换并更新路由信息。
- 经过若干次更新后，所有的路由器最终都会知道到达本自治系统中任何一个网络的最短距离和下一跳路由器的地址。
- RIP 协议的收敛(convergence)过程较快，即在自治系统中所有的结点都得到正确的路由选择信息的过程。

36

距离向量算法

收到相邻路由器（其地址为 X）的一个 RIP 报文：
(1) 先修改此 RIP 报文中的所有项目：将“下一跳”字段中的地址都改为 X，并将所有的“距离”字段的值加 1。 --1
(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：
若项目中的网络不在路由表中，则将该项目加到路由表中。--2
　否则
　　若下一跳字段给出的路由器地址是同样的，则将收到的项目替换原路由表中的项目。--3
　　否则
　　　若收到项目中的距离小于路由表中的距离，则进行更新，否则，什么也不做。--4
(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则将此相邻路由器记为不可达的路由器，即将距离置为16（距离为16 表示不可达）。
(4) 返回。

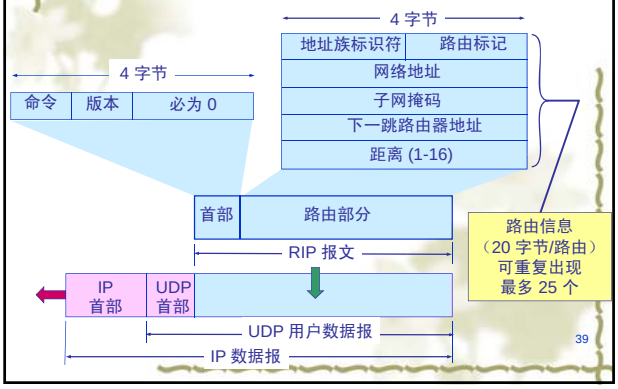
目的网络已在路由表中

下一跳地址不同

路由器之间交换信息

- RIP协议让互联网中的所有路由器都和自己的相邻路由器不断交换路由信息，并不断更新其路由表，使得从每一个路由器到每一个目的网络的路由都是最短的（即跳数最少）。
- 虽然所有的路由器最终都拥有了整个自治系统的全局路由信息，但由于每一个路由器的位置不同，它们的路由表当然也应当是不同的。

RIP2 协议的报文格式



RIP2的报文组成

- 首部(4字节)
  - 命令字段：表示报文的意义。“1”表示请求路由信息；“2”表示对请求路由信息的响应或未被请求而发出的路由更新报文。
  - 后面补“0”是为了补齐4个字节。

RIP2的报文组成

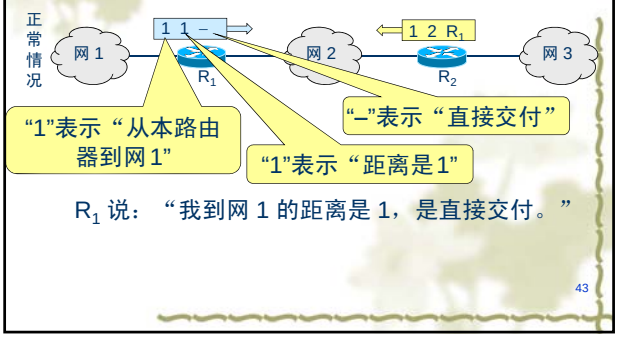
- 路由部分
  - 由若干个路由信息组成。每个路由信息需要用 20 个字节。
  - 地址族标识符（又称为地址类别）字段用来标志所使用的地址协议。
  - 路由标记填入自治系统的号码，这是考虑使RIP 有可能收到本自治系统以外的路由选择信息。
  - 再后面指出某个网络地址、该网络的子网掩码、下一跳路由器地址以及到此网络的距离。
- 最大RIP报文长度为：4＋20＊25＝504字节。

RIP 协议的优缺点

- RIP 协议最大的优点就是实现简单，开销较小。
- RIP [illegible]问题是当网络出现故障时，要经过比[illegible]能将此信息传送到所有的路由器。
- RIP 限制了网络的规模，它能使用的最大距离为15（16 表示不可达）。
- 路由器之间交换的路由信息是路由器中的完整路由表，因而随着网络规模的扩大，开销也就增加。

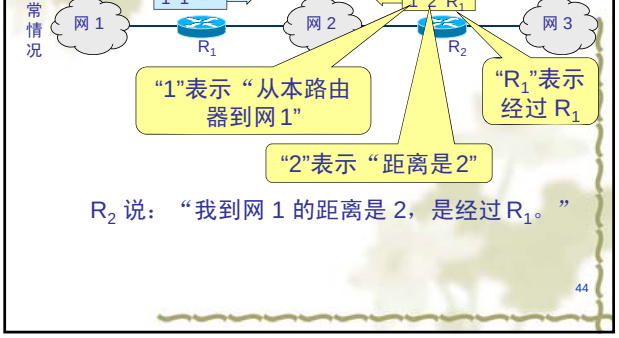
所谓好消息传得快，坏消息传得慢

RIP的“好消息传得快，坏消息传得慢”



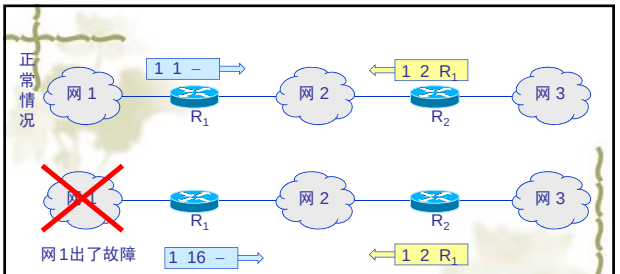
R_1 说：“我到网 1 的距离是 1，是直接交付。”

43



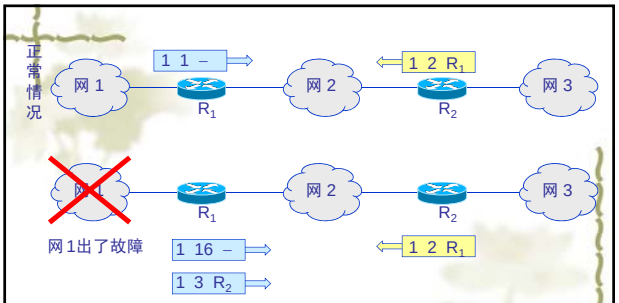
R_2 说：“我到网 1 的距离是 2，是经过 R_1。”

44



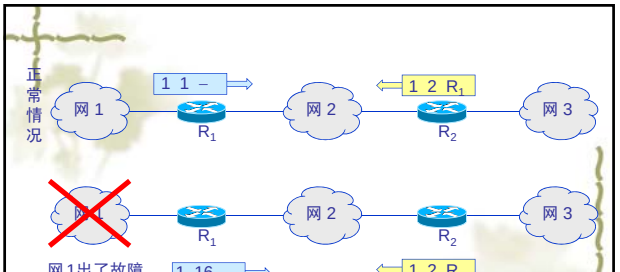
R_1 说：“我到网 1 的距离是 16（表示无法到达），是直接交付。”

但 R_2 在收到 R_1 的更新报文之前，还发送原来的报文，因为这时 R_2 并不知道 R_1 出了故障。



R_1 收到 R_2 的更新报文后，误认为可经过 R_2 到达网1，于是更新自己的路由表，说：“我到网 1 的距离是 3，下一跳经过 R_2”。然后将此更新信息发送给 R_2。

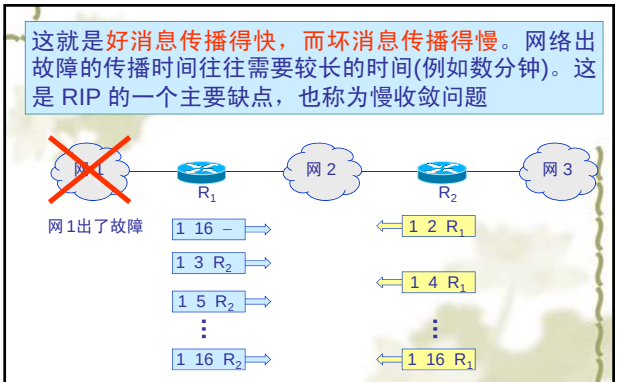
46



R_2 以后又更新自己的路由表为“1，4，R_1”，表明“我到网 1 距离是 4，下一跳经过 R_1”。

47

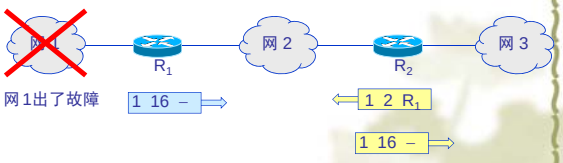
这就是好消息传播得快，而坏消息传播得慢。网络出故障的传播时间往往需要较长的时间(例如数分钟)。这是 RIP 的一个主要缺点，也称为慢收敛问题



这样不断更新下去，直到 R_1 和 R_2 到网 1 的距离都增大到 16 时，R_1 和 R_2 才知道网 1 是不可达的。

水平分割方法

- 主要思想：当一个节点把本地维护的距离向量信息发送给相邻节点时，它并不把从其相邻节点处学到的路由再回送到那些相邻节点



坏消息以每交换一次一个节点的速度传播

内部网关协议 OSPF
(Open Shortest Path First)

- OSPF 协议的基本特点
  - “开放”表明 OSPF 协议不是受某一家厂商控制，而是公开发表的。
  - “最短路径优先”是因为使用了 Dijkstra 提出的最短路径算法SPF
  - OSPF 只是一个协议的名字，它并不表示其他的路由选择协议不是“最短路径优先”。
  - 是分布式的链路状态协议。 RIP是距离向量协议。

最短路径问题

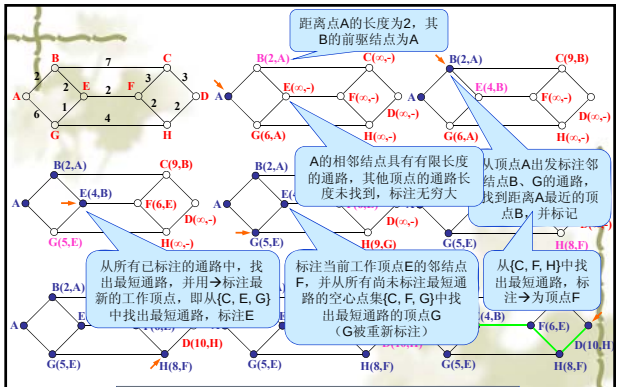
- 问题描述
  - 抽象为求图中一对顶点间的最短通路。
  - 图中的顶点代表网络节点；弧代表通信链路。
  - 权代表相邻顶点间的“代价”。代价可指物理上的距离、分组传输的时间、线路通信费用。
- 所谓路由最短是指：
  - 物理距离近
  - 分组传输延迟时间最小
  - 通信费用最低

迪杰斯特拉（Dijkstra）算法

- 算法思想
  - 按照通路长度递增的次序产生最短通路算法。
  - 详细的说：首先从起始点出发，找出距起始点最短的通路，然后在此基础上找出距起始点次短的通路，如此反复，每次都找出比前一次次短的通路，直到某个通路达到某个目的地。
- 实例分析
  - 带权的无向图，求顶点A到D的最短通路？

Dijkstra算法运行实例

- 顶点标记
  - 黑点：表示起点A到该顶点的最短通路已经找到；
  - 空心点：表示起点A到该顶点的最短通路尚未找到。
  - (x,y)：X表示距起点A通路长度；Y通路中该顶点的前趋顶点。
  - “→”：表示最近一次找到的次短通路的终点，也是下次试探的工作点。



然后用前驱结点一直回退到顶点A，即D→H→F→E→B→A

最短通路为：A-B-E-F-H-D，权值为10

Dijkstra算法的C语言实现

```
#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 {
        int predecessor;                /* the path being worked on */
        int length;                     /* previous node */
        enum {permanent, tentative} label; /* length from source to this node */
                                        /* 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 */
    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);
}
```

链路状态路由选择算法

- 基本思想
  - 通过各个节点之间的路由信息交换，每个节点可以获得全网的拓扑信息。
    - 全网的拓扑信息：包括网中所有的节点、各个节点间的链路连接以及各条链路的代价
  - 将上述拓扑信息组成一张带权无向图，然后利用最短通路路由选择算法计算到各个目的节点的最短通路。

56

OSPF协议的三个要点

- 向本自治系统中所有路由器发送信息，使用洪泛法。
  - **RIP仅向自己相邻的路由器发送信息。**
- 发送的信息就是与本路由器相邻的所有路由器的链路状态，但这只是路由器所知道的部分信息。
  - **RIP发送的是“到所有网络的距离和下一跳路由器”。**
- 只有当链路状态发生变化时，所有路由器发送此信息。
  - **RIP定期发送。**

本路由器和哪些路由器相邻，以及该链路的“度量”(metric)。如费用、距离、时延、带宽

所有路由器最终都能建立一个链路状态数据库，即全网的拓扑结构图

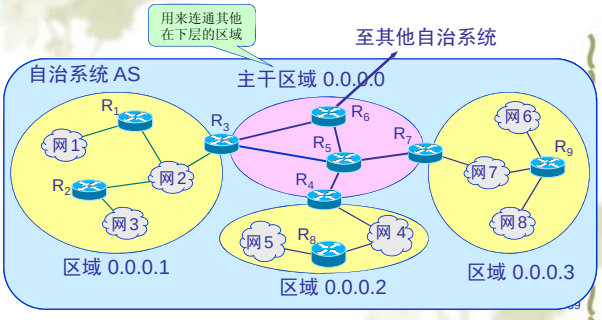
57

OSPF 的区域(area)

- 为了使 OSPF能够用于规模很大的网络，OSPF将一个自治系统再划分为若干个更小的范围，叫作区域
- 每一个区域都有一个32位的区域标识符（用点分十进制表示）
- 区域也不能太大，在一个区域内的路由器最好不超过200个
- 区域划分的好处是什么？
  - 将洪泛的范围缩小，减少网络通信量
  - 在一个区域内部的路由器只知道本区域的完整网络拓扑，而不知道其他区域的网络拓扑情况

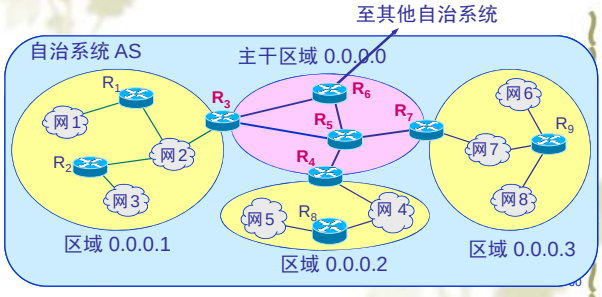
58

OSPF 划分为两种不同的区域



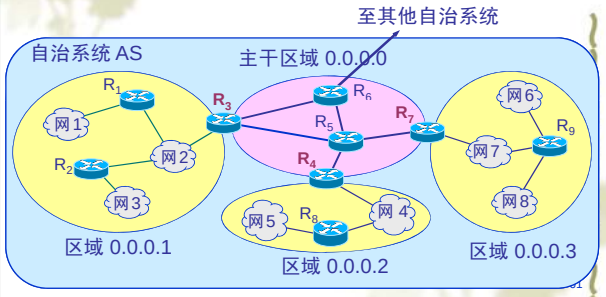
59

主干路由器

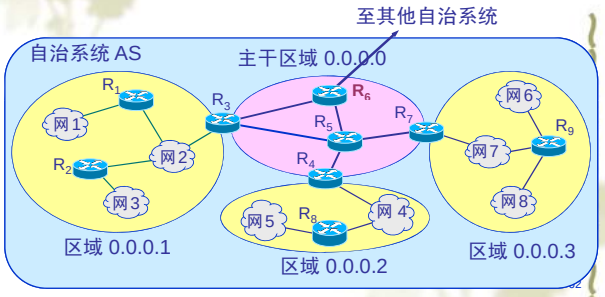


60

区域边界路由器



自治系统边界路由器

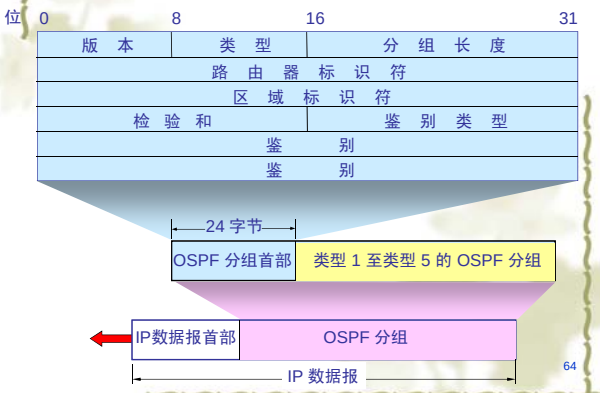


OSPF 的特点

- OSPF不用UDP而是直接用IP数据报传送。
  - IP数据报首部的协议字段值为89。
- OSPF 构成的数据报很短。
  - 可减少路由信息的通信量。
  - 可以不必将长的数据报分片传送。分片传送的数据报只要丢失一个，就无法组装成原来的数据报，而整个数据报就必须重传。
- 支持可变长度的子网划分和无分类编址 CIDR。
- 每一个链路状态都带上一个 32 位的序号，序号越大状态越新。

63

OSPF 分组



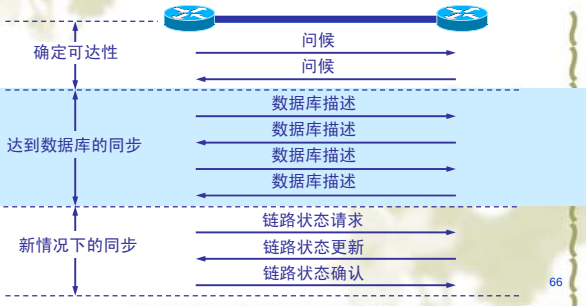
64

OSPF 的五种分组类型

- 类型1，问候(Hello)分组。
- 类型2，数据库描述(Database Description)分组。
- 类型3，链路状态请求(Link State Request)分组。
- 类型4，链路状态更新(Link State Update)分组，用洪泛法对全网更新链路状态。
- 类型5，链路状态确认(Link State Acknowledgment)分组。

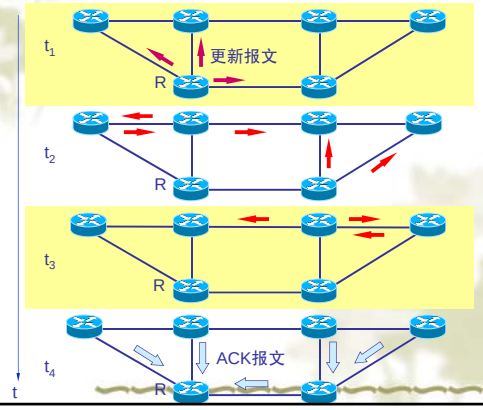
65

OSPF的基本操作



66

OSPF 使用的是可靠的洪泛法



67

OSPF vs. RIP

- RIP
  - 节点告诉相邻节点它所知道的*所有*路由信息
  - 节点根据来自相邻节点的路由信息更新自己的路由表
  - 定期交换信息
  - 可扩展性差
- OSPF
  - 节点告诉所有节点它的*相邻节点*的状态信息
  - 每个节点都有一个全局的拓扑结构，并以此计算路由表
  - 链路状态变化时才交换信息
  - 可扩展性好，可靠
  - 与整个互联网的规模无直接联系。没有“坏消息传播得慢”的问题

68

外部网关协议 BGP

- 不同自治系统的路由器之间交换路由信息的协议。
- 因特网的规模太大，使得自治系统之间路由选择非常困难。要在自治系统之间寻找最佳路由是很不现实的。
- 自治系统之间的路由选择必须考虑有关策略。
- 因此，边界网关协议 BGP 只能是力求寻找一条能够到达目的网络且比较好的路由（不能兜圈子），而并非要寻找一条最佳路由。
- 采用路径向量（path vector）路由选择协议。

69

BGP 发言人(BGP speaker)

- 每一个自治系统的管理员要选择至少一个路由器作为该自治系统的“ BGP 发言人” 。
- 一般说来，两个 BGP 发言人都是通过一个共享网络连接在一起的，而 BGP 发言人往往就是 BGP 边界路由器，但也可以不是 BGP 边界路由器。

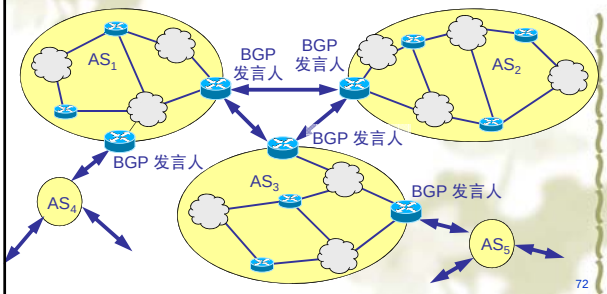
70

BGP 交换路由信息

- 一个 BGP 发言人与其他自治系统中的 BGP 发言人要交换路由信息，就要先建立 TCP 连接，然后在此连接上交换 BGP 报文以建立 BGP 会话(session)，利用 BGP 会话交换路由信息。
- 使用 TCP 连接能提供可靠的服务，也简化了路由选择协议。
- 使用 TCP 连接交换路由信息的两个 BGP 发言人，彼此成为对方的邻站或对等站。

71

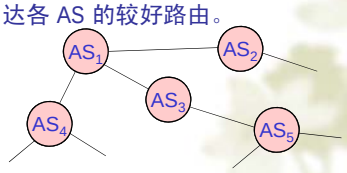
BGP 发言人和自治系统 AS 的关系



72

AS 的连通图举例

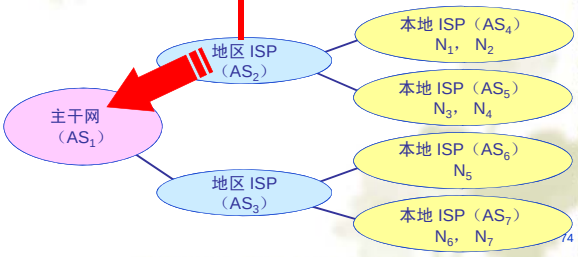
- BGP 所交换的网络可达性的信息就是要到达某个网络所要经过的一系列 AS。
- 当 BGP 发言人互相交换了网络可达性的信息后，各 BGP 发言人就根据所采用的策略从收到的路由信息中找出到达各 AS 的较好路由。



73

BGP 发言人交换路径向量

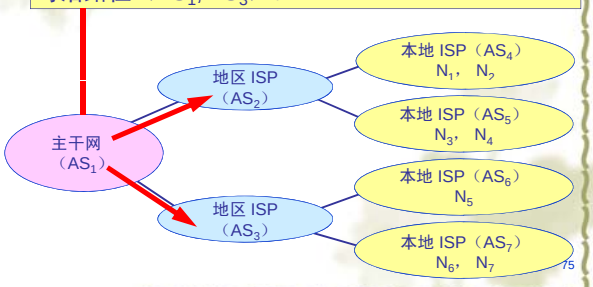
自治系统 AS_2 的 BGP 发言人通知主干网的 BGP 发言人：“要到达网络 N_1, N_2, N_3 和 N_4 可经过 AS_2。”



74

BGP 发言人交换路径向量

主干网还可发出通知：“要到达网络 N_5, N_6 和 N_7 可沿路径（AS_1, AS_3）。”



75

BGP 协议的特点

- 交换路由信息时就是自治系统数的量级。
- 每一个自治系统的边界路由器（BGP 发言人）的数目是很少的。
- BGP 支持 CIDR，因此 BGP 的路由表也就应当包括目的网络前缀、下一跳路由器，以及到达网络所要经过的各个自治系统序列。
- 在BGP 刚刚运行时，BGP 的邻站是交换整个的 BGP 路由表。但以后只需要在发生变化时更新有变化的部分。

比这些自治系统中的网络数少很多

使得自治系统之间的路由选择不致过分复杂

由于使用了路径向量信息，可避免产生兜圈子的路由

节省网络带宽，减少路由器的处理开销

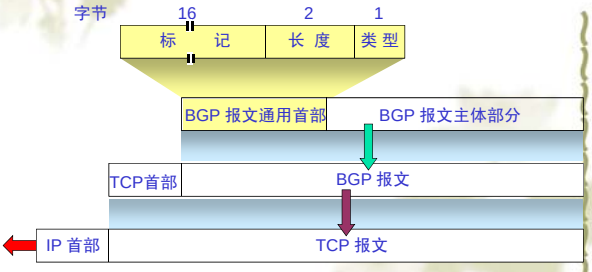
76

BGP-4 共使用四种报文

- 打开(OPEN)报文，用来与相邻的另一个BGP发言人建立关系。
- 更新(UPDATE)报文，用来发送某一路由的信息，以及列出要撤消的多条路由。
- 保活(KEEPALIVE)报文，用来确认打开报文和周期性地证实邻站关系。
- 通知(NOTIFICATION)报文，用来发送检测到的差错。

77

BGP 报文具有通用的首部



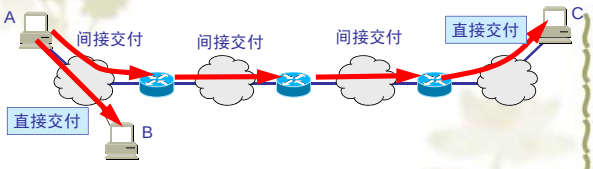
78

路由器在网际互连中的作用

- 当主机 A 要向另一个主机 B 发送数据报时，先要检查目的主机 B 是否与源主机 A 连接在同一个网络上。
- 如果是，就将数据报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，则应将数据报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数据报转发给下一个路由器。这就叫作间接交付。

79

直接交付和间接交付



直接交付不需要使用路由器
但间接交付就必须使用路由器

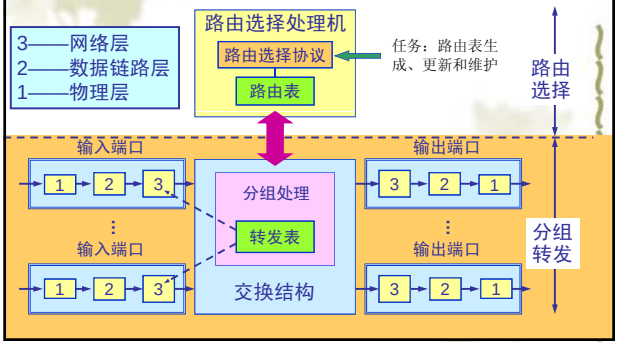
80

路由器的结构

- 路由器是一种具有多个输入端口和多个输出端口的专用计算机，其任务是转发分组。也就是说，将路由器某个输入端口收到的分组，按照分组要去的目的地（即目的网络），把该分组从路由器的某个合适的输出端口转发给下一跳路由器。
- 下一跳路由器也按照这种方法处理分组，直到该分组到达终点为止。

81

典型的路由器的结构



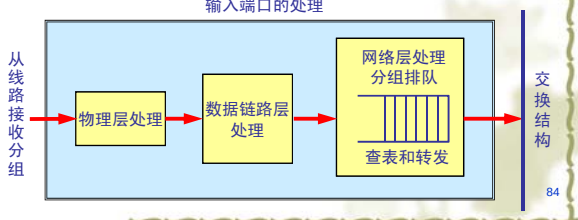
“转发”和“路由选择”的区别

- “转发”(forwarding)就是路由器根据转发表将用户的 IP 数据报从合适的端口转发出去。
- “路由选择”(routing)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。

83

输入端口对线路上收到的分组的处理

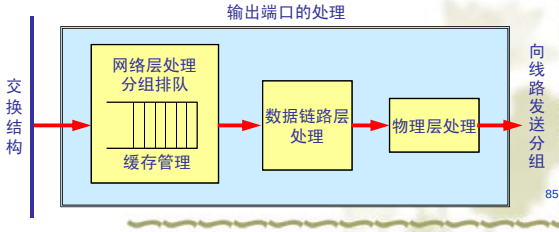
- 数据链路层剥去帧首部和尾部后，将分组送到网络层的队列中排队等待处理。这会产生一定的时延。



84

输出端口将交换结构传送来的分组发送到线路

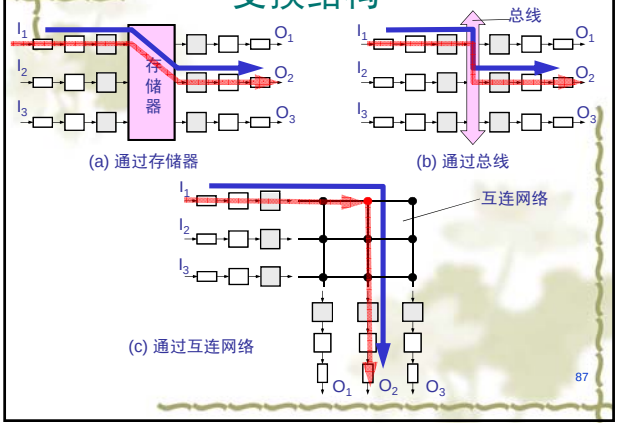
- 当交换结构传送过来的分组先进行缓存。数据链路层处理模块将分组加上链路层的首部和尾部，交给物理层后发送到外部线路。



分组丢弃

- 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。

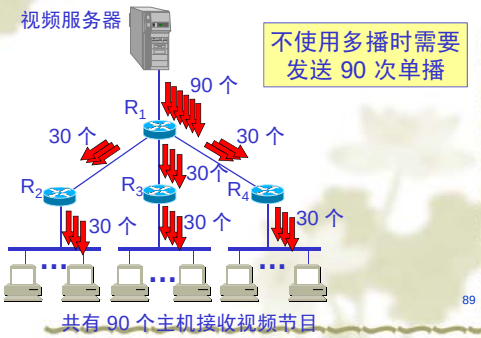
交换结构



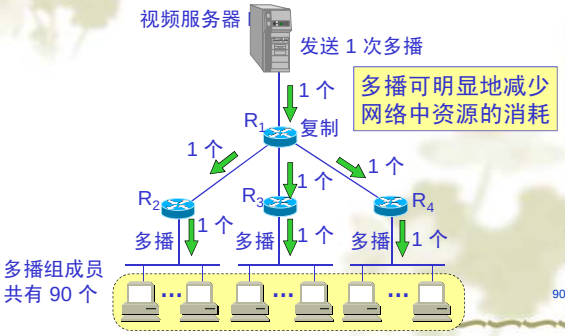
第4章 网络层

- 4.1 网络层提供的服务
- 4.2 网际协议IP
- 4.3 划分子网和构造超网
- 4.4 网际控制报文协议ICMP
- 4.5 因特网的路由选择协议
- 4.6 IP多播
- 4.7 其他网络举例

IP多播



IP多播的基本概念



IP 多播的一些特点

- 多播使用组地址—— IP 使用 D 类地址支持多播。多播地址只能用于目的地址，而不能用于源地址。
- 永久组地址——由因特网号码指派管理局 IANA 负责指派。
- 动态的组成员
- 使用硬件进行多播

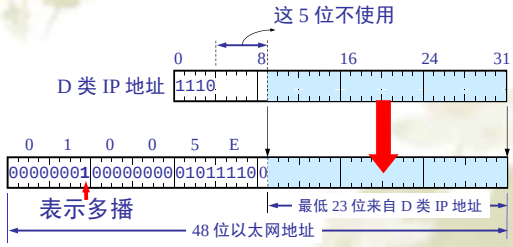
91

在局域网上进行硬件多播

- 因特网号码指派管理局 IANA 拥有的以太网地址块的高 24 位为 00-00-5E。
- 因此 TCP/IP 协议使用的以太网多播地址块的范围是：从 00-00-5E-00-00-00
  到 00-00-5E-FF-FF-FF
- D 类 IP 地址可供分配的有 28 位，在这 28 位中的前 5 位不能用来构成以太网硬件地址。

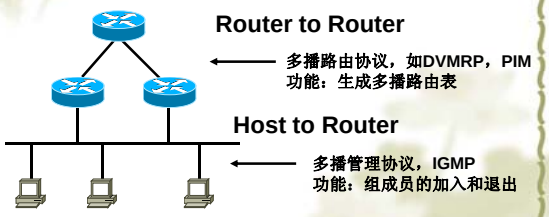
92

D 类 IP 地址
与以太网多播地址的映射关系



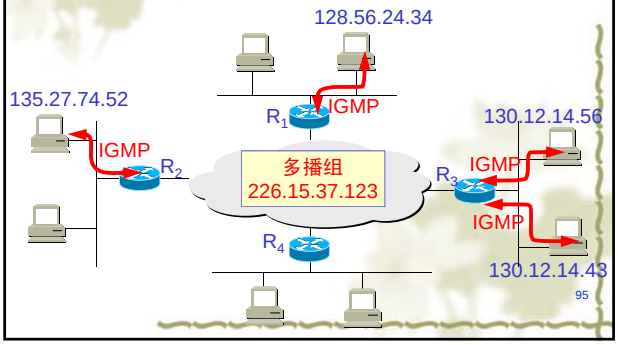
93

多播相关协议



94

网际组管理协议 IGMP



95

多播路由协议

- 基于源的多播协议，以发送方为多播树的根，并且包含了所有的组成员
  - DVMRP（ Distance Vector Multicast Routing Protocol ）
  - MOSPF (Multicast Extension to OSPF)
- 基于共享树的协议，对于每个组使用同一颗树
  - CBT（Core Based Trees）
- 混合以上两种方法的协议
  - PIM (Protocol Independent Multicast)
  - PIM-DM (Dense Mode), PIM-SM (Sparse Mode)

96

第4章 网络层

- 4.1 网络层提供的服务
- 4.2 网际协议IP
- 4.3 划分子网和构造超网
- 4.4 网际控制报文协议ICMP
- 4.5 因特网的路由选择协议
- 4.6 IP多播
- 4.7 其他网络举例

97

其他网络举例

- 4.7.1 X.25网
- 4.7.2 帧中继FR
- 4.7.3 ATM网络

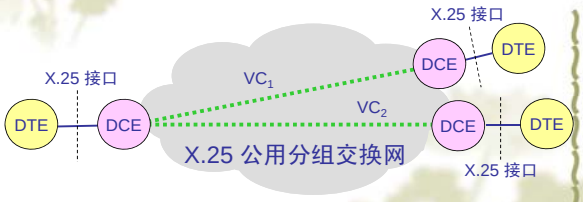
98

4.7.1 X.25网

- X.25 网就是 X.25 分组交换网，它是在二十多年前根据 CCITT（即现在的 ITU-T）的 X.25 建议书实现的计算机网络。
- X.25 是以面向连接的虚电路服务为基础。
- X.25 规定了DTE和DCE之间的接口标准。

99

X.25规定了DTE-DCE 的接口



100

X.25 网与 IP 网

- 基于IP协议的因特网是无连接的，只提供尽最大努力交付的数据报服务，无服务质量可言。
- X.25 网是面向连接的，能够提供可靠交付的虚电路服务，能保证服务质量。
- 正因为 X.25 网能保证服务质量，在二十多年前它曾经是颇受欢迎的一种计算机网络。
- 20 世纪 90 年代，X.25 网退出了历史舞台。

101

4.7.2 帧中继FR

- 在 20 世纪 80 年代后期，许多应用都迫切要求增加分组交换服务的速率。
- 帧中继 FR (Frame Relay)就是一种支持高速交换的网络体系结构。
- 帧中继在许多方面非常类似于 X.25，被称为第二代的 X.25。
- 也叫快速分组交换网，它与X.25分组交换网不同，在链路层实现复用和转接，故名帧中继。

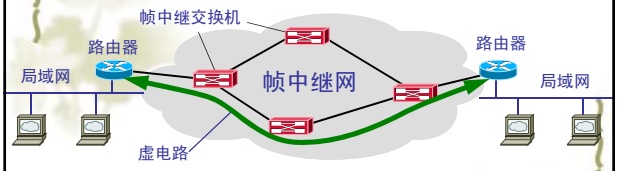
102

帧中继的特点

- 帧中继减少结点处理时间
  - 当帧中继交换机收到一个帧的首部时，只要一查出帧的目的地址就立即进行转发。
- 帧中继对差错的处理
  - 当检测到有误码时，结点要立即中止这次传输。
- 帧中继网络向上提供面向连接的虚电路服务
  - 交换虚电路 SVC和永久虚电路 PVC
- 帧中继的控制信令
  - 在与用户数据分开的另一个逻辑连接上传送（带外信令）

103

帧中继提供虚电路服务



虚电路像一条专用电路
用户看不见帧中继网络内的帧中继交换机

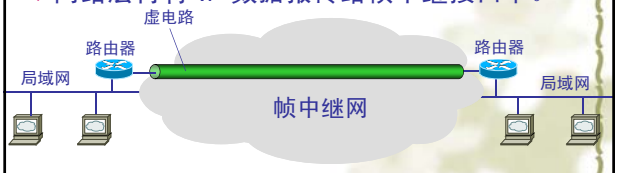
帧中继网络的工作过程

- 用户在局域网上传送的 MAC 帧传到与帧中继网络相连接的路由器。

105

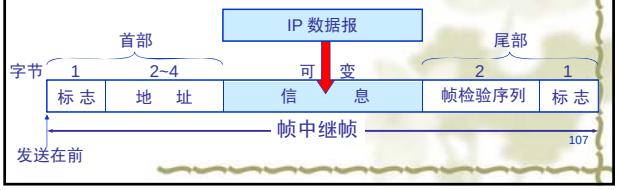
帧中继网络的工作过程

- 路由器就剥去 MAC 帧的首部，将 IP 数据报交给路由器的网络层。
- 网络层再将 IP 数据报传给帧中继接口卡。



106

- 帧中继接口卡把IP数据报封装到帧中继帧的信息字段。
- 加上帧中继帧的首部（包括帧中继的标志字段和地址字段，帧中继帧的标志字段和PPP帧的一样），进行CRC检验后，加上帧中继帧的尾部（包含帧检验序列字段和标志字段），就构成了帧中继帧。



107

帧中继网络的工作过程

- 为了区分开不同的永久虚电路 PVC，每一条 PVC 的两个端点都各有一个数据链路连接标识符 DLCI。
- DCLI 是 Data Link Connection Identifier。

108

帧中继网络的工作过程

- 帧中继接口卡将封装好的帧通过向电信公司租来的专线发送给帧中继网络中的帧中继交换机。
- 帧中继交换机收到帧中继帧就按地址字段中的虚电路号转发帧（若检查出有差错则丢弃）。

109

帧中继网络的工作过程

- 当帧中继帧被转发到虚电路的终点路由器时，终点路由器就剥去帧中继帧的首部和尾部，加上局域网的首部和尾部，交付给连接在此局域网上的目的主机。

110

帧中继网络的工作过程

- 目的主机若发现有差错，则报告上层的 TCP 协议处理。
- 即使 TCP 协议对有错误的数据进行了重传，帧中继网也仍然当作是新的帧中继帧来传送，而并不知道这是重传的数据。

111

4.7.3 异步传输模式ATM

- Asynchronous Transfer Mode 异步传输模式
- 异步传输模式ATM是结合了电路交换和分组交换的优点而产生的一种新的交换技术。
- ATM的目的：是使宽带综合业务数字网B-ISDN能通过公用网络传输声音、数据和图象等信息。
- ATM方式有别于传统的电路交换和分组交换技术。

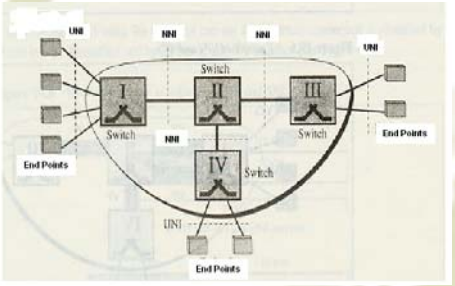
112

ATM的基本原理：UNI与NNI

- 两种接口UNI与NNI
- 用户-网络接口UNI
  - User-to-Network Interface
  - 指的是端设备和网络内节点（交换机）的接口。
- 网络-网络接口NNI
  - Network-to-Network Interface
  - 指的是网络内节点（交换机）之间的接口。

113

ATM中的UNI与NNI



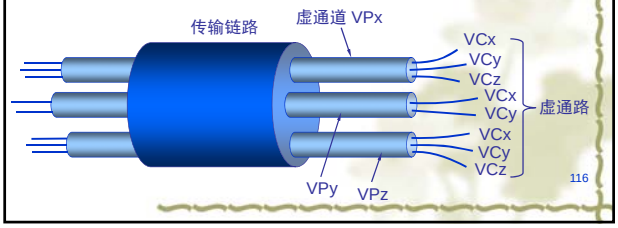
114

ATM基本原理：两级虚电路

- VP与VC
  - 虚通道VP（Virtual Path）指的是两个交换机之间的一组逻辑连接，而其中用于传送信元的逻辑连接就称为虚通路VC（Virtual Channel）。
- VC链路
  - VC链路：相邻两点间信元的传输通道，用VCI标识。
  - 信源、信宿间若干条VC链路，级连成一条VC连接(虚电路)，可为某对通信服务。
- VP链路
  - 由一束具有相同端点的VC链路组成，用VPI标识。

VPI与VCI

- ATM 连接用信元首部中的两级标号来识别。
- 虚通路标识 VCI (Virtual Channel Identifier)
- 虚通道标识符 VPI (Virtual Path Identifier)



信元（Cell）

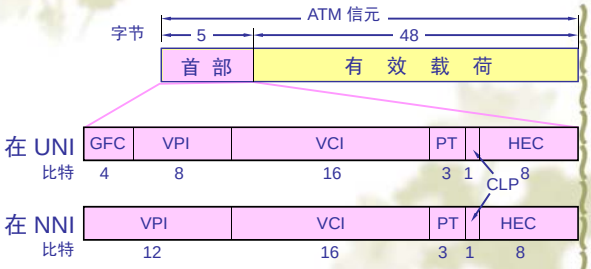
- ATM 采用定长分组作为传输和交换的单位。这种定长分组叫做信元(cell)。
- ATM的信元由5个字节的头部和48个字节的信息字段组成。

ATM信元的两种不同首部

网络到网络接口 NNI (Network-to-Network Interface)

用户到网络接口 UNI (User-to-Network Interface)

ATM信元的两种不同首部



本章小结

- 了解网络层的基本功能，掌握IP地址的编址方法，子网划分，无分类编址，熟悉IP数据报的格式及分片操作
- 了解ARP协议和ICMP协议的基本概念，掌握RIP、OSPF、BGP路由算法，路由器转发分组的流程，IP多播的概念

代价（Cost）

- 在研究路由选择时，需要给每一条链路指明一定的代价(cost)。
- “代价”是由一个或几个因素综合决定的一种度量(metric)，如链路长度、数据率、链路容量、是否要保密、传播时延等，甚至还可以是一天中某一个小时内的通信量、结点的缓存被占用的程度、链路差错率等。

121

解释1

- 目的：便于进行本路由表的更新
- 设从地址为X的相邻路由器发来的RIP报文的某一个项目为“Net2，3，Y”，表示“我到网络Net2的距离为3，要经过的下一跳路由器为Y”
- 本路由器可以推断：若我将下一跳路由器的地址选为X，则到网络Net2的距离为3＋1＝4，于是将收到的RIP报文的这一条目改为“Net2，4，X”，作为下一步比较使用，以确定是否需要更新路由表
- 返回

122

解释2

- 表明这是新的目的网络，应该加入到路由表中
- 例如：本路由表中没有到目的网络Net2的路由，那么需要将新的条目“Net2,4,X”加入到路由表中
- 返回

123

解释3

- 为什么要替换呢？因为这是最新的路由消息，路由表要以最新的消息为准
- 到目的网络的距离有可能增大、减小或不变。因此，不管原来的路由表条目是“Net2,3,X”还是“Net2,5,X”，都要替换成“Net2,4,X”
- 返回

124

解释4

- 因为更新后的路由到网络的距离更短
- 例如：若路由表中已有条目“Net2,5,P”，需要更新为“Net2,4,X”，因为更新后的距离从5变为4，更短了
- 返回

125