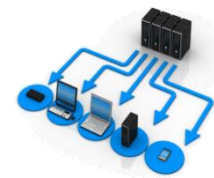


## 4.3.3 无分类编址 CIDR



划分子网在一定程度上缓解了互联网在发展中遇到的困难。互联网仍然面临三个必须尽早解决的问题：

- (1) B 类地址在 1992 年已分配了近一半；
- (2) 互联网主干网上的路由表中的项目数急剧增长（从几千个增长到几万个）。
- (3) 整个 IPv4 的地址空间最终将全部耗尽。
- 无分类域间编址**CIDR** (Classless Inter-Domain Routing)。

# CIDR 最主要的特点

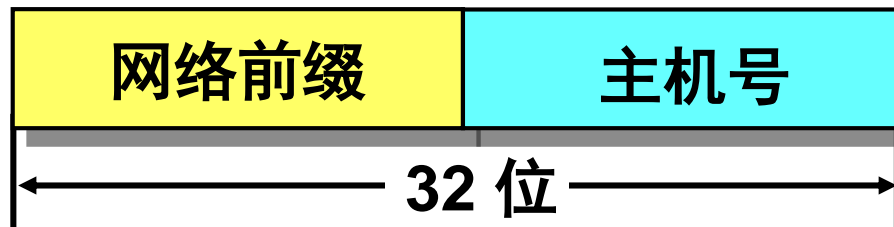


- CIDR 消除了传统的 A 类、B 类和 C 类地址以及划分子网的概念，因而可以更加有效地分配 IPv4 的地址空间。
- CIDR 使用各种长度的“**网络前缀**” (network-prefix)来代替分类地址中的网络号和子网号。
- IP 地址从三级编址（使用子网掩码）又回到了两级编址。

# 无分类的两级编址



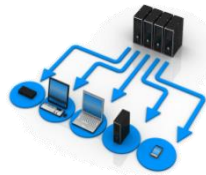
- 无分类的两级编址的记法是：



**IP地址 ::= {<网络前缀>, <主机号>} (4-3)**

- CIDR 使用“**斜线记法**” (slash notation), 它又称为 **CIDR 记法**, 即在 IP 地址面加上一个斜线“/”, 然后写上网络前缀所占的位数 (这个数值对应于三级编址中子网掩码中 1 的个数)。例如:  
**220.78.168.0/24**

# CIDR 地址块



- CIDR 把网络前缀都相同的连续的 IP 地址组成“**CIDR 地址块**”。
- 128.14.32.0/20 表示的地址块共有  $2^{12}$  个地址（因为斜线后面的 **20** 是网络前缀的位数，所以这个地址的主机号是 12 位）。
  - 这个地址块的起始地址是 128.14.32.0。
  - 在不需要指出地址块的起始地址时，也可将这样的地址块简称为“**/20 地址块**”。
  - 128.14.32.0/20 地址块的最小地址：128.14.32.0
  - 128.14.32.0/20 地址块的最大地址：128.14.47.255
  - 全 0 和全 1 的主机号地址一般不使用。

# 128.14.32.0/20 表示的地址 ( $2^{12}$ 个地址)

最小地址 →  
128.14.32.0

所有地址  
的 20 位  
前缀都是  
一样的

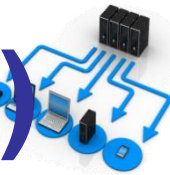
10000000	00001110	00100000	00000000
10000000	00001110	00100000	00000001
10000000	00001110	00100000	00000010
10000000	00001110	00100000	00000011
10000000	00001110	00100000	00000100
10000000	00001110	00100000	00000101
...			
10000000	00001110	00101111	11111011
10000000	00001110	00101111	11111100
10000000	00001110	00101111	11111101
10000000	00001110	00101111	11111110
10000000	00001110	00101111	11111111

最大地址 →  
128.14.47.255



- 地址的掩码是
- 11111111 11111111 11110000 00000000
- 255.255.240.0
- 最小地址 128.14.32.0
- 最大地址 128.14.47.255
- 分给主机的地址范围
- 128.14.32.1~128.14.47.254
- 可表示多少个C类地址块 ?
- $12-8=4$                    $2^4=16$ 个C类地址块

# 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。
- 路由聚合有利于**减少**路由表中的表项，从而提高了整个互联网的性能。
- 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）。
- 对于 **/20** 地址块，它的掩码是 20 个连续的 1。斜线记法中的数字就是掩码中 1 的个数。

# 常用的 CIDR 地址块



CIDR 前缀长度	点分十进制	包含的地址数	相当于包含分类的网络数
/13	255.248.0.0	512 K	8 个 B类或 2048 个 C 类
/14	255.252.0.0	256 K	4 个 B 类或1024 个 C 类
/15	255.254.0.0	128 K	2 个 B 类或512 个 C 类
/16	255.255.0.0	64 K	1 个 B 类或256 个 C 类
/17	255.255.128.0	32 K	128 个 C 类
/18	255.255.192.0	16 K	64 个 C 类
/19	255.255.224.0	8 K	32 个 C 类
/20	255.255.240.0	4 K	16 个 C 类
/21	255.255.248.0	2 K	8 个 C 类
/22	255.255.252.0	1 K	4 个 C 类
/23	255.255.254.0	512	2 个 C 类
/24	255.255.255.0	256	1 个 C 类
/25	255.255.255.128	128	1/4 个 C 类
/26	255.255.255.192	64	1/4 个 C 类
/27	255.255.255.224	32	1/8 个 C 类



# 构成超网



- 前缀长度不超过 23 位的 CIDR 地址块都包含了多个 C 类地址。
- 这些 C 类地址合起来就构成了超网。
- **CIDR 地址块中的地址数一定是 2 的整数次幂。**
- 网络前缀越短，其地址块所包含的地址数就越多。
- CIDR 的一个好处是：可以更加有效地分配 IPv4 的地址空间，可根据客户的需要分配适当大小的 CIDR 地址块。

# 练习（路由聚合）



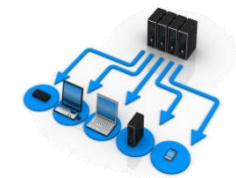
- 1. 设有两个子网202.118.133.0/24 和 202.118.130.0/24， 如果进行路由聚合， 得到的网络地址为 \_\_\_\_\_。
- 看第三个字节
- 133 二进制 10000101
- 130 二进制 10000010
- 10000000 00000000
- 202.118.128/21
- 这个地址包含了上面两个子网地址。

# 习题(分配地址)



- 某个大学分配到一个地址块**206.0.68.0/22**  
该大学有四个学院，理学院 需要500个地址，  
文学院需要200个地址， 工学院需要100个地址，  
农学院 需要100台地址，请给出每一个学院的分配的地址块（包含前缀），请写成每个学院的地址范围。

# 地址分配方案



A:地址 1024

B:地址数 512

C: 地址数 512

B块地址分给 理学院  
D块地址分给 文学院  
E块地址分给 工学院  
F分给 农学院

D:地址数 256	E :128	F:128
-----------	--------	-------

# 地址分配方案



- 206.0.68.0/22 主机有**10位**，展开后面两个字节
- 010001**00** 00000000
- 理学院 500  $2^8 < 500 < 2^9$  ,网络前缀为 $32-9=23$ ,
- 地址块 取 206.0. 010001**0**0 00000000  
(206.0.68.0/23)
- 地址范围 206.0.68.0/23~206.0.69.255/23
- ( 010001**0**1 11111111)
- 
- 
-



- 网络前缀每增加一位，地址数减半
- 文学院 200  $2^7 < 200 < 2^8$  ,网络前缀为32-8=24
- 地址块块 取 206.0. 01000110 00000000  
(206.0.70.0/24)
- 地址范围 206.0.70.0/24~206.0.70.255/24
- ( 01000110 11111111 )

# 地址分配方案



- 工学院需要100个地址， $2^6 < 100 < 2^7$ ，网络前缀为 $32-7=25$ ，
- 地址块块 取 206.0. 01000111 00000000 (206.0.71.0/25)
- 地址范围 206.0.71.0/25~206.0.71.127/25
- ( 01000111 01111111)
- 农学院的地址分配？（练习）
- 206.0.71.128/25 (( 01000111 10000000)
- ~206.0.71.255 ( 11111111)
- 总结：1) 按所需的地址 数由大到小分配较好。
- 2) 地址不能重叠

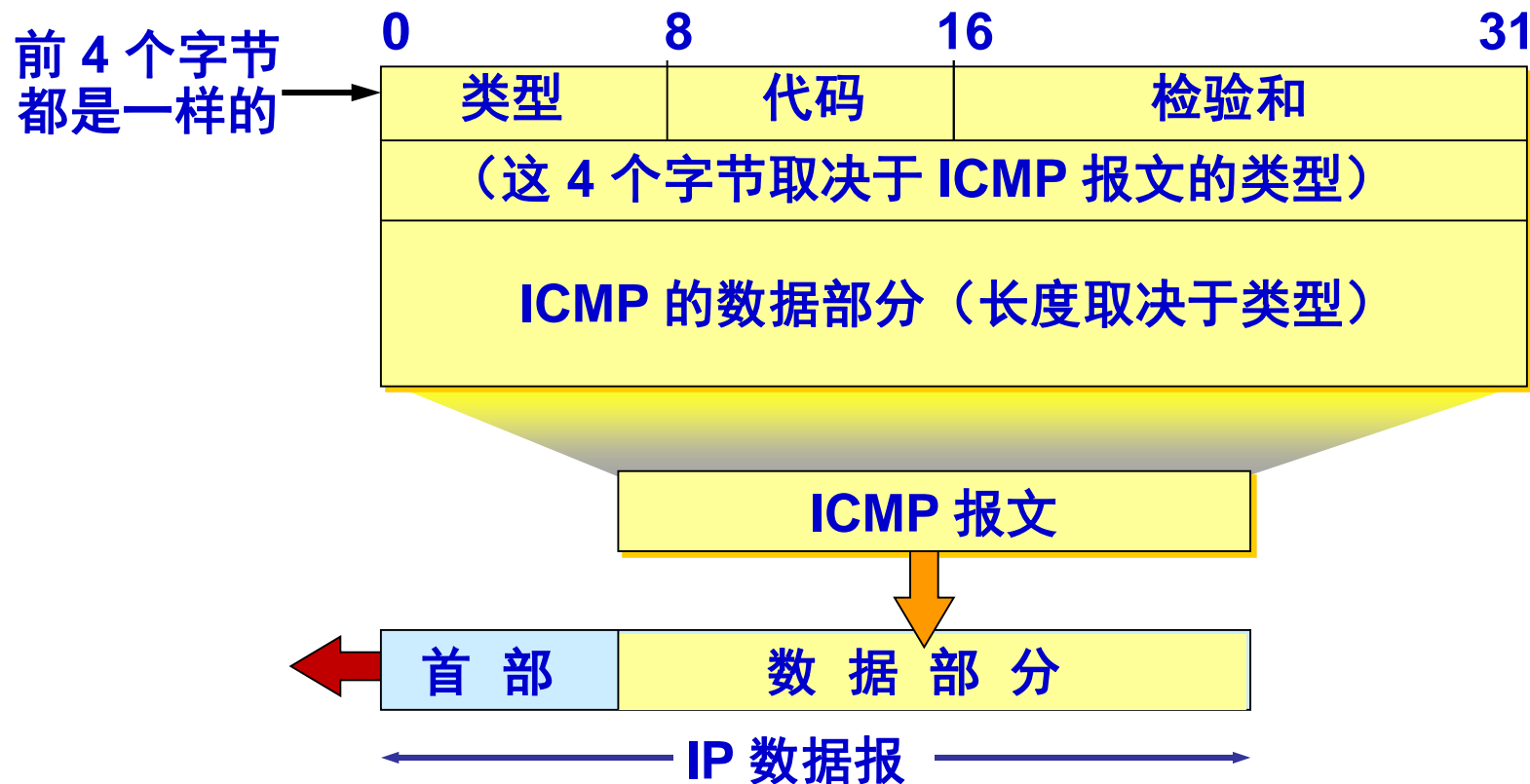
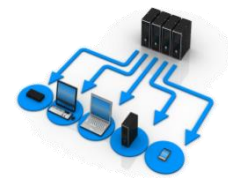
## 4.4 网际控制报文协议 ICMP



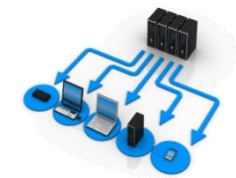
- 为了更有效地转发 IP 数据报和提高交付成功的机会，在网际层使用了网际控制报文协议 ICMP (Internet Control Message Protocol)。
- ICMP 是互联网的标准协议。
- ICMP 允许主机或路由器报告差错情况和提供有关异常情况的报告。
- 但 ICMP 不是高层协议（看起来好像是高层协议，因为 ICMP 报文是装在 IP 数据报中，作为其中的数据部分），而是 IP 层的协议。



# ICMP 报文的格式



## 4.4.1 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。

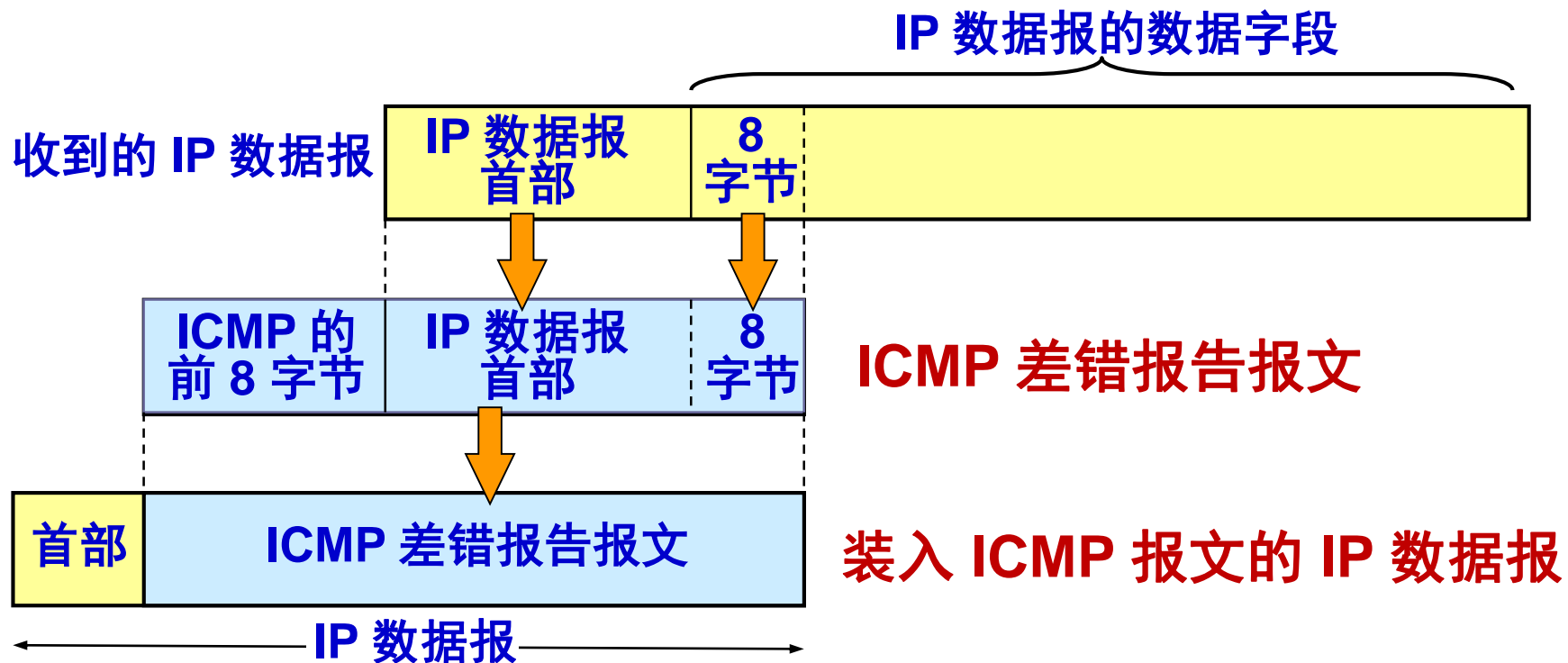
# ICMP 差错报告报文共有 4 种



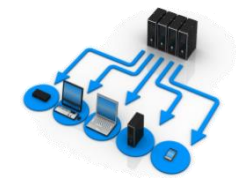
- 终点不可达，类型 值：3
- 时间超过            类型 值：11
- 参数问题            类型 值：12
- 改变路由（重定向）(Redirect) 类型值 5

类型	代码	含义
3	3	端口不可达（差错）
11	0	超时（差错）
8	0	回送请求（查询）
0	0	响应（查询）

# ICMP 差错报告报文的数据字段的内容



# ICMP 询问报文有两种



- 回送请求和回答报文 （类型值8 或0）
- 时间戳请求和回答报文 （类型值13 或14）

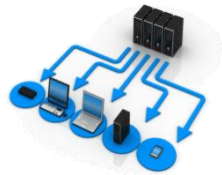
## 4.4.2 ICMP 的应用举例



### PING (Packet InterNet Groper)

- 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。

# 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
```

用 PING 测试主机的连通性

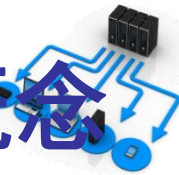
# 4.5 互联网的路由选择协议



- 4.5.1 有关路由选择协议的几个基本概念
- 4.5.2 内部网关协议 RIP
- 4.5.3 内部网关协议 OSPF
- 4.5.4 外部网关协议 BGP
- 4.5.5 路由器的构成



## 4.5.1 有关路由选择协议的几个基本概念



### 1. 理想的路由算法

- 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，这就是说，要有自适应性。
- 算法应具有稳定性。
- 算法应是公平的。
- 算法应是最佳的。

# 从路由算法的自适应性考虑



- **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。



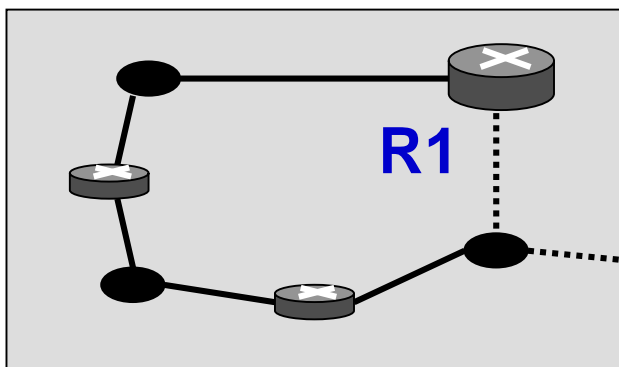
## 2.分层次的路由选择协议

- **互联网采用分层次的路由选择协议**
- **自治系统（AS）的定义：**在单一的技术管理下的一组路由器，而这些路由器使用一种 AS 内部的路由选择协议和**共同的度量**以确定分组在该 AS 内的路由，同时还使用一种 AS 之间的路由选择协议用以确定分组在 AS之间的路由。
- **重要的是一个 AS 对其他 AS 表现出的的是一个单一的和一致的路由选择策略。**

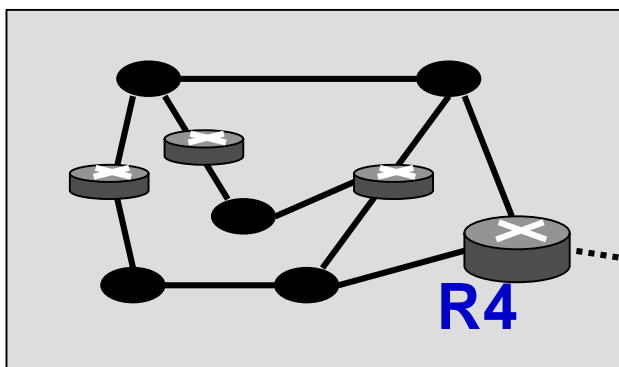
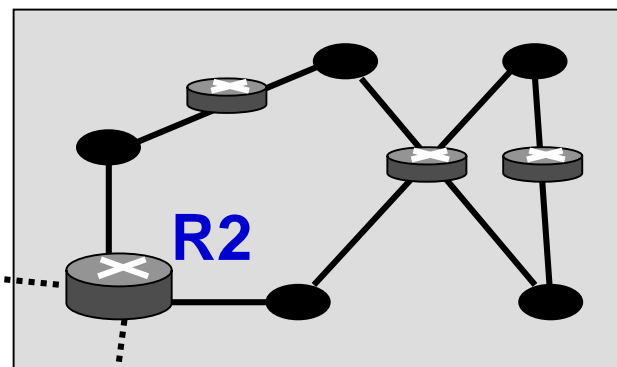
# 自治系统 AS



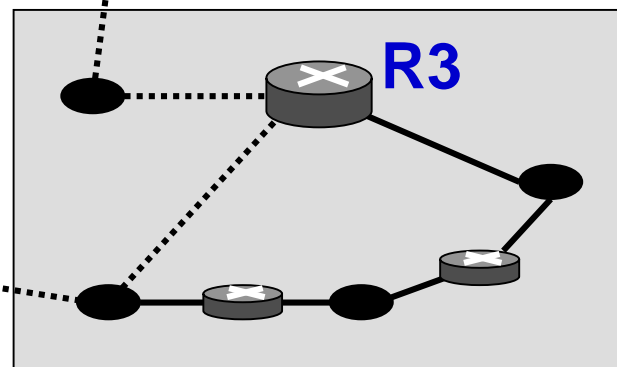
自治系统



自治系统



自治系统



自治系统

# 互联网有两大类路由选择协议



- **内部网关协议 IGP (Interior Gateway Protocol)**
  - 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。

# 自治系统和 内部网关协议、外部网关协议

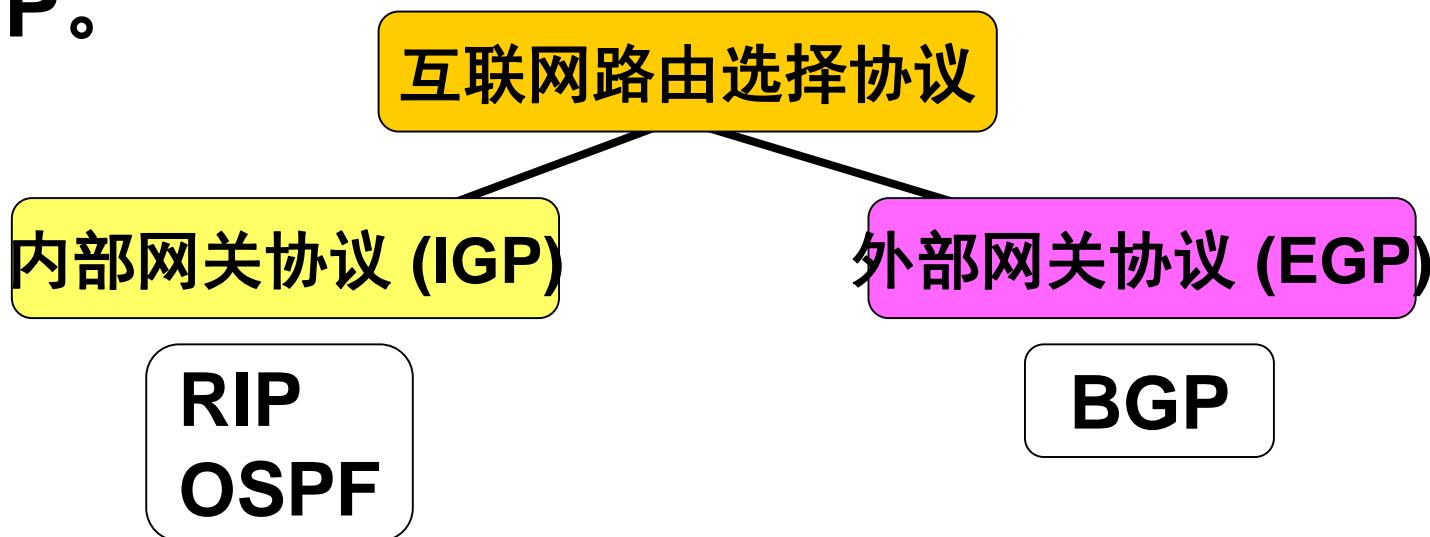


自治系统之间的路由选择也叫做**域间路由选择** (interdomain routing), 在自治系统内部的路由选择叫做**域内路由选择** (intradomain routing)。

# 互联网的路由选择协议



- **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。



## 4.5.2 内部网关协议 RIP



### 1. 工作原理

- 路由信息协议 RIP (Routing Information Protocol) 是内部网关协议 IGP 中最先得到广泛使用的协议。
- RIP 是一种**分布式的、基于距离向量的**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。



# “距离”的定义



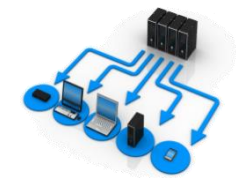
- 从一个路由器到**直接连接**的网络的距离定义为 1。
- 从一个路由器到非直接连接的网络的距离定义为所经过的路由器数加 1。
- RIP 协议中的“距离”也称为“**跳数**” (hop count), 因为每经过一个路由器, 跳数就加 1。
- 这里的“距离”实际上指的是“**最短距离**”。

# “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。

# RIP 协议的三个特点



- (1) 仅和**相邻路由器**交换信息。
- (2) 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**。
- (3) 按固定的时间间隔**交换路由信息**，例如，每隔 30 秒。当网络拓扑发生变化时，路由器也及时向相邻路由器通告拓扑变化后的路由信息。

# 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）。它的**路由表是空的**。
- 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。

## 2. 距离向量算法



路由器收到相邻路由器（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：把“下一跳”字段中的地址都改为 X，并把所有的“距离”字段的值加 1。

(2) 对修改后的 RIP 报文中的每一个项目，重复以下步骤：

若项目中的目的网络不在路由表中，则把该项目加到路由表中。(1)

否则

若下一跳字段给出的路由器地址是同样的，则把收到的项目替换原路由表中的项目。(2)

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；(3)

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，则把此相邻路由器记为不可达路由器，即将距离置为 16（表示不可达）。(5)

(4) 返回。

## 2. 距离向量算法



- 距离向量算法的基础就是 Bellman-Ford 算法（或 Ford-Fulkerson 算法）。
- 这种算法的要点是这样的：  
设 $X$ 是结点  $A$  到  $B$  的最短路径上的一个结点。  
若把路径  $A \rightarrow B$  拆成两段路径  $A \rightarrow X$  和  $X \rightarrow B$ ，则每一段路径  $A \rightarrow X$  和  $X \rightarrow B$  也都分别是结点  $A$  到  $X$  和结点  $X$  到  $B$  的最短路径。



【例4-5】已知路由器 R<sub>6</sub> 有表 4-9(a) 所示的路由表。现在收到相邻路由器 R<sub>4</sub> 发来的路由更新信息，如表 4-9(b) 所示。试更新路由器 R<sub>6</sub> 的路由表。

表 4-9(a) 路由器 R<sub>6</sub> 的路由表

目的网络	距离	下一跳路由器
Net2	3	R <sub>4</sub>
Net3	4	R <sub>5</sub>
...	...	...

表 4-9(b) R<sub>4</sub> 发来的路由更新信息

目的网络	距离	下一跳路由器
Net1	3	R <sub>1</sub>
Net2	4	R <sub>2</sub>
Net3	1	直接交付

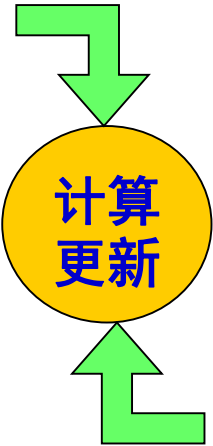


表 4-9(d) 路由器 R<sub>6</sub> 更新后的路由表

目的网络	距离	下一跳路由器
Net1	4	R <sub>4</sub>
Net2	5	R <sub>4</sub>
Net3	2	R <sub>4</sub>
...	...	...

距离加 1

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 <sub>4</sub>
Net2	5	R <sub>4</sub>
Net3	2	R <sub>4</sub>

# 【例2】路由表更新



从C来的RIP报文 增加跳数以后

Net2	4
Net3	8
Net6	4
Net8	3
Net9	5

从C来的RIP报文

Net2	5	C
Net3	9	C
Net6	5	C
Net8	4	C
Net9	6	C

B旧路由表

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

更新算法

B新路由表

???





Net1: 没有新信息, 不变

Net2: 相同的下一跳, 替换 (新消息)

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

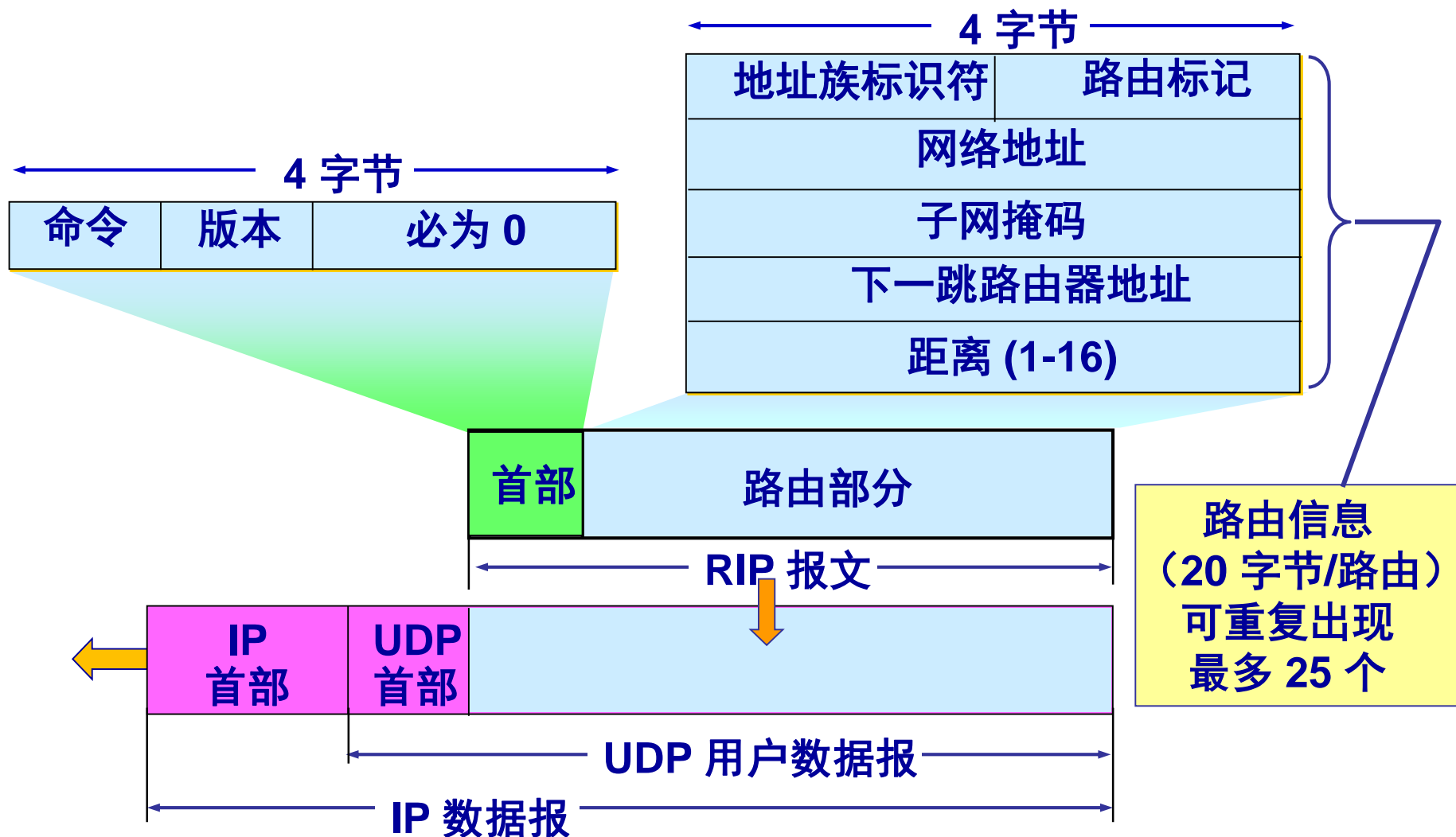
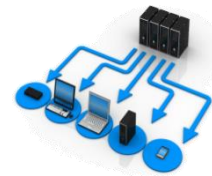
Net9: 不同的下一跳, 新跳数大, 不变

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

## 更新算法

Net1	7	A
<b>Net2</b>	<b>5</b>	<b>C</b>
Net3	9	C
Net6	5	C
Net8	4	E
Net9	4	F

# 3. RIP2 协议的报文格式



# RIP2 报文



- 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。
- 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。

# 好消息传播得快，坏消息传播得慢



- **RIP 协议特点：**好消息传播得快，坏消息传播得慢。
- **RIP 存在的一个问题：**当网络出现故障时，要经过比较长的时间 (例如数分钟) 才能将此信息传送到所有的路由器。

# RIP 协议的优缺点



## ■ 优点：

- 实现简单，开销较小。

## ■ 缺点：

- 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。
- “坏消息传播得慢”，使更新过程的收敛时间过长。

## 4.5.3 内部网关协议 OSPF



- 开放最短路径优先 OSPF (Open Shortest Path First)是为克服 RIP 的缺点在 1989 年开发出来的。
- OSPF 的原理很简单，但实现起来却较复杂。

# 1. 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
- 采用**分布式的链路状态协议** (link state protocol)。
- **注意：** 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”。

# OSPF 路由协议三个要点



- 1. 向本自治系统中所有路由器发送信息，这里使用的方法是**洪泛法**。
- 2. 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。
  - “链路状态”就是说明本路由器都和哪些路由器相邻，以及该链路的“**度量**”(metric)。
- 3. 只有当链路状态**发生变化**时，路由器才用洪泛法向所有路由器发送此信息。



# 链路状态数据库 (link-state database)



- 由于各路由器之间频繁地交换链路状态信息，因此所有的路由器最终都能建立一个链路状态数据库。
- 这个数据库实际上就是**全网的拓扑结构图**，**它在全网范围内是一致的**（这称为链路状态数据库的同步）。
- OSPF 的链路状态数据库能**较快地进行更新**，使各个路由器能及时更新其路由表。
- **OSPF 的更新过程收敛得快是其重要优点。**

# 路由选择算法常用的度量标准



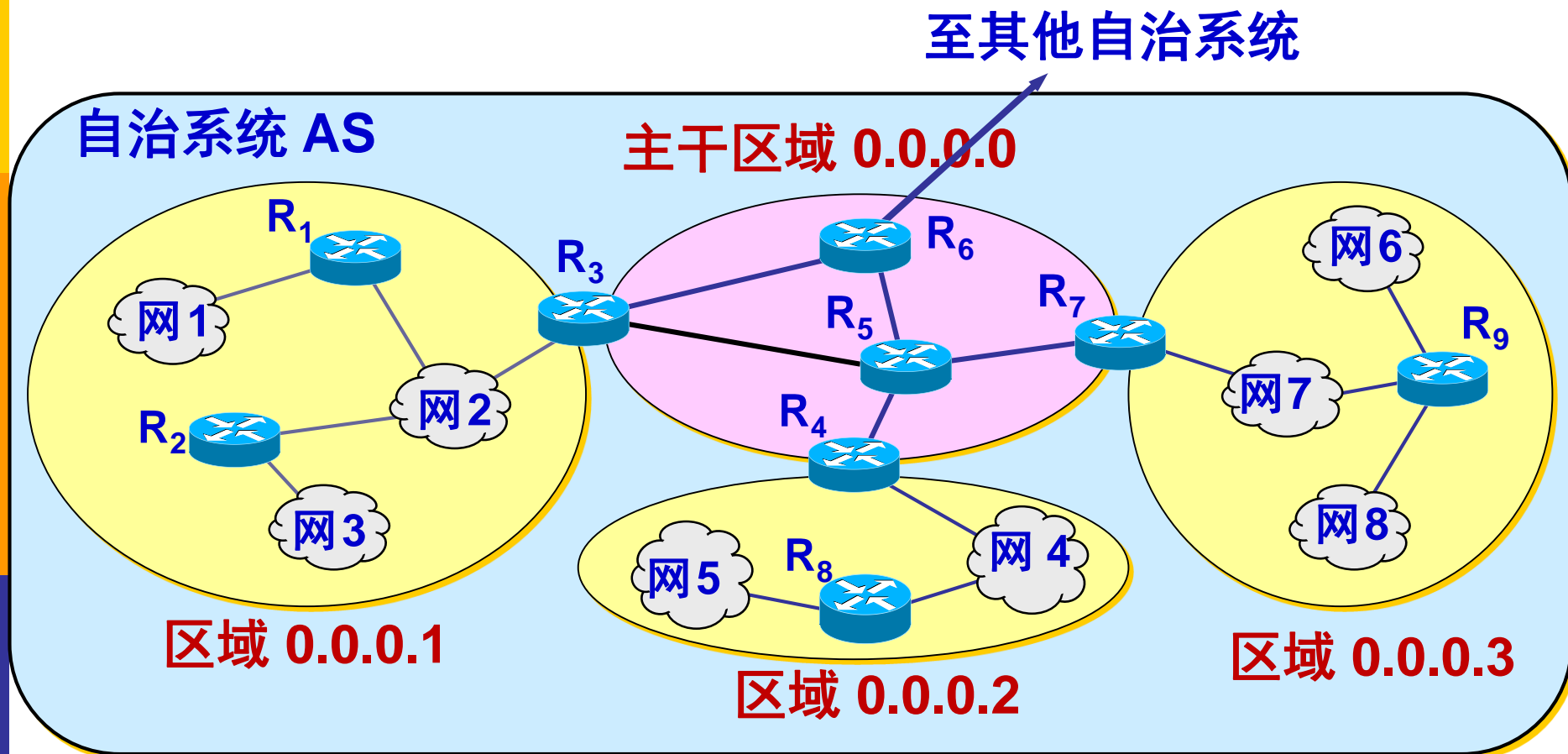
- (1) 带宽 (bandwidth) : 链路的数据容量。
- (2) 延迟 (delay) : 分组沿着从源到目的的每条链路移动所需的时间。
- (3) 负载 (load) : 网络资源的活动量。
- (4) 跳数 (hop count) : 分组到达目的地之前必须经过的路由器个数。
- (5) 代价 (cost) : 由管理员指派的基于带宽、花费等的度量值。

# OSPF 的区域 (area)

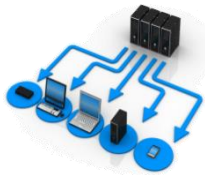


- 为了使 OSPF 能够用于规模很大的网络，OSPF 将一个自治系统再划分为若干个更小的范围，叫做**区域**。
- 每一个区域都有一个 32 位的区域标识符（用点分十进制表示）。
- 区域也不能太大，在一个区域内的路由器最好不超过 200 个。

# OSPF 划分为两种不同的区域

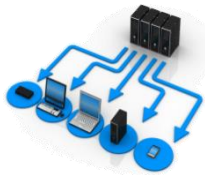


# 划分区域



- 划分区域的**好处**就是将利用洪泛法交换链路状态信息的范围局限于每一个区域而不是整个的自治系统，这就减少了整个网络上的通信量。
- 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的情况。**
- OSPF 使用**层次结构的区域划分**。在上层的区域叫做**主干区域 (backbone area)**。
- 主干区域的标识符规定为0.0.0.0。主干区域的**作用**是用来连通其他在下层的区域。

# OSPF 其他特点



- **OSPF 不用 UDP 而是直接用 IP 数据报传送。**
- **OSPF 构成的数据报很短。这样做可减少路由信息的通信量。**
- **OSPF 对不同的链路可根据 IP 分组的不同服务类型而设置成不同的度量值。因此，OSPF 对于不同类型的业务可计算出不同的路由。**
- **支持可变长度的子网划分和无分类编址 CIDR。**

# OSPF 分组



位	0	8	16	31
版 本		类 型	分 组 长 度	
路 由 器 标 识 符				
区 域 标 识 符				
检 验 和			鉴 别 类 型	
鉴			别	
鉴			别	

24 字节

OSPF 分组首部

类型 1 至类型 5 的 OSPF 分组



OSPF 分组用 IP 数据报传送

## 2. OSPF 的五种分组类型



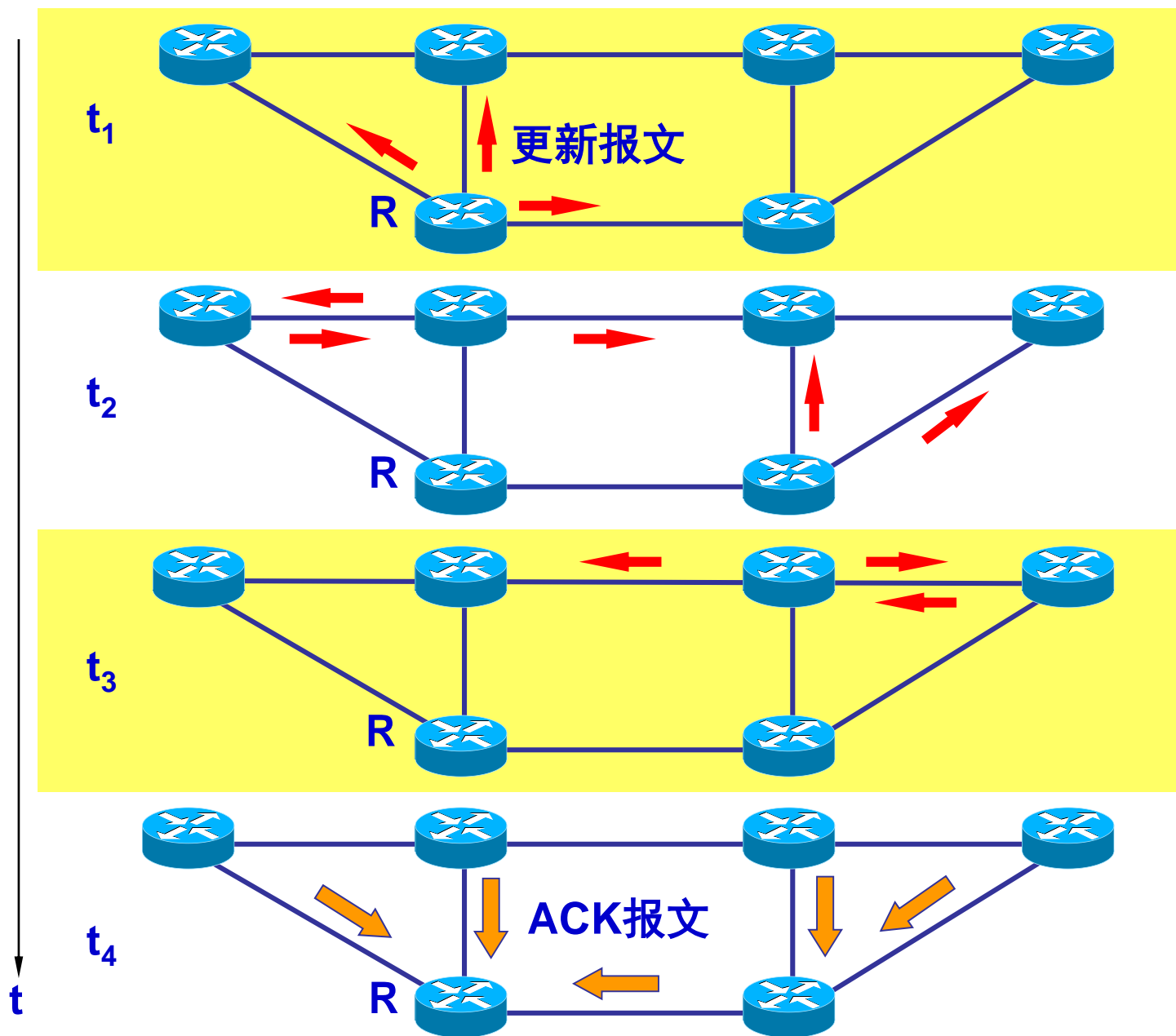
- **类型1**，问候 (Hello) 分组。
- **类型2**，数据库描述 (Database Description) 分组。
- **类型3**，链路状态请求 (Link State Request) 分组。
- **类型4**，链路状态更新 (Link State Update) 分组，  
用**洪泛法**对全网更新链路状态。
- **类型5**，链路状态确认 (Link State Acknowledgment) 分组。



# OSPF 的基本操作



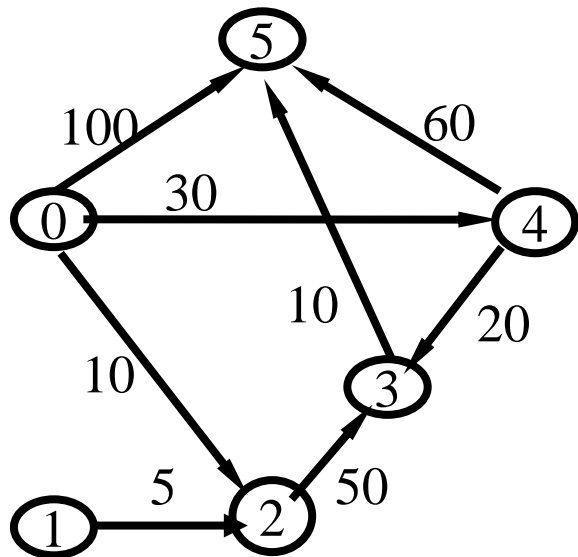
# 组



# 最短路径算法- Dijkstra算法



- 将路由器抽象为图中的结点，所经过的网络抽象为链路，形成网络图，度量按所提供的服务来定义；



从 R0到其他点的最短路径

R0-R2            10

R0-R4            30

R0-R3            R0-R4-R3    50

R0-R5            R0-R2-R3-R5   60

R0-R1                             $\infty$

迪杰斯特拉（Dijkstra）算法是按路径长度递增的次序产生最短路径

# OSPF 的其他特点



- OSPF 还规定每隔一段时间，如 30 分钟，要刷新一次数据库中的链路状态。
- 由于一个路由器的链路状态只涉及到与相邻路由器的连通状态。因此当互联网规模很大时，OSPF 协议要比距离向量协议 RIP 好得多。
- **OSPF 没有“坏消息传播得慢”的问题**，据统计，其响应网络变化的时间小于 100 ms。

# 4.6 IPv6

---



## ■ 4.6.2 IPv6 的地址

## 4.6 IPv6



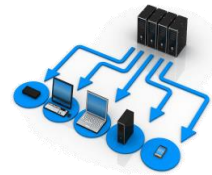
- IP 是互联网的核心协议。
- 互联网经过几十年的飞速发展，到 2011 年 2 月，IPv4 的 32 位地址已经耗尽。
- ISP 已经不能再申请到新的 IP 地址块了。
- 我国在 2014 – 2015 年也逐步停止了向新用户和应用分配 IPv4 地址。
- 解决 IP 地址耗尽的根本措施就是采用具有更大地址空间的新版本的 IP，即 IPv6。

## 4.6.2 IPv6 的地址



- IPv6 数据报的目的地址可以是以下三种基本类型地址之一：
  - (1) **单播** (unicast): 传统的点对点通信。
  - (2) **多播** (multicast): 一点对多点的通信。
  - (3) **任播** (anycast): 这是 IPv6 增加的一种类型。任播的目的站是一组计算机，但数据报在交付时只交付其中的一个，通常是距离最近的一个。

# 结点与接口



- IPv6 将实现 IPv6 的主机和路由器均称为**结点**。
- 一个结点就可能有多多个与链路相连的接口。
- IPv6 地址是分配给结点上面的接口的。
  - 一个接口可以有多个单播地址。
  - 其中的任何一个地址都可以当作到达该结点的目的地址。即一个结点接口的单播地址可用来唯一地标志该结点。



# 冒号十六进制记法



- 在 IPv6 中，每个地址占 128 位，地址空间大于  $3.4 \times 10^{38}$ 。
- 为了使地址再稍简洁些，IPv6 使用**冒号十六进制记法** (colon hexadecimal notation, 简称为 colon hex)。
- 每个 16 位的值用十六进制值表示，各值之间用冒号分隔。例如：

**68E6:8C64:FFFF:FFFF:0:1180:960A:FFFF**

- 在十六进制记法中，允许把数字前面的 0 省略。例如把 0000 中的前三个 0 省略，写成 1 个 0。

# 零压缩



- 冒号十六进制记法可以允许**零压缩** (zero compression), 即一连串连续的零可以为一对冒号所取代。

FF05:0:0:0:0:0:0:B3 可压缩为:

FF05::B3

- **注意：在任一地址中只能使用一次零压缩。**

# 点分十进制记法的后缀



- 冒号十六进制记法可结合使用点分十进制记法的后缀，这种结合在 IPv4 向 IPv6 的转换阶段特别有用。
- 例如： **0:0:0:0:0:0:128.10.2.1**  
再使用零压缩即可得出： **::128.10.2.1**
- **CIDR 的斜线表示法仍然可用。**
- 例如： 60 位的前缀 12AB00000000CD3 可记为：  
**12AB:0000:0000:CD30:0000:0000:0000:0000/60**  
或 **12AB::CD30:0:0:0:0/60** （零压缩）  
或 **12AB:0:0:CD30::/60** （零压缩）

# IPv6 地址分类



地址类型	二进制前缀
未指明地址	00...0（128位），可记为 ::/128。
环回地址	00...1（128位），可记为 ::1/128。
多播地址	11111111（8位），可记为 FF00::/8。
本地链路单播地址	1111111010（10位），可记为 FE80::/10。
全球单播地址	（除上述四种外，所有其他的二进制前缀）

# IPv6 地址分类



## ■ 未指明地址

- 这是 16 字节的全 0 地址，可缩写为两个冒号 “::”。
- 这个地址只能为还没有配置到一个标准的 IP 地址的主机当作源地址使用。
- 这类地址仅此一个。

## ■ 环回地址

- 即 0:0:0:0:0:0:0:1（记为 ::1）。
- 作用和 IPv4 的环回地址一样。
- 这类地址也是仅此一个。

# IPv6 地址分类



## ■ 多播地址

- 功能和 IPv4 的一样。
- 这类地址占 IPv6 地址总数的 1/256。

## ■ 本地链路单播地址 (Link-Local Unicast Address)

- 有些单位的网络使用 TCP/IP 协议，但并没有连接到互联网上。连接在这样的网络上的主机都可以使用这种本地地址进行通信，但不能和互联网上的其他主机通信。
- 这类地址占 IPv6 地址总数的 1/1024。

# IPv6 地址分类



## ■ 全球单播地址

- IPv6 的这一类单播地址是使用得最多的一类。
- 曾提出过多种方案来进一步划分这 128 位的单播地址。
- 根据 2006 年发布的草案标准 RFC 4291 的建议，IPv6 单播地址的划分方法非常灵活。

结 点 地 址 (128 bit)

子 网 前 缀 ( $n$  bit)

接 口 标 识 符 ( $128 - n$ ) bit

全球路由选择前缀 ( $n$  bit)

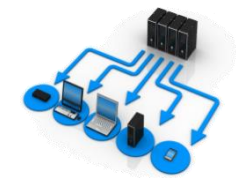
子网标识符 ( $m$  bit)

接口标识符 ( $128 - n - m$ ) bit

IPv6 单播地址的几种划分方法

# 第四章作业

---



- 4-20 (2)(3)
- 4-22, 4-26, 4-29, 4-37, 4-41, 4-54, 4-55





---

*THE END*