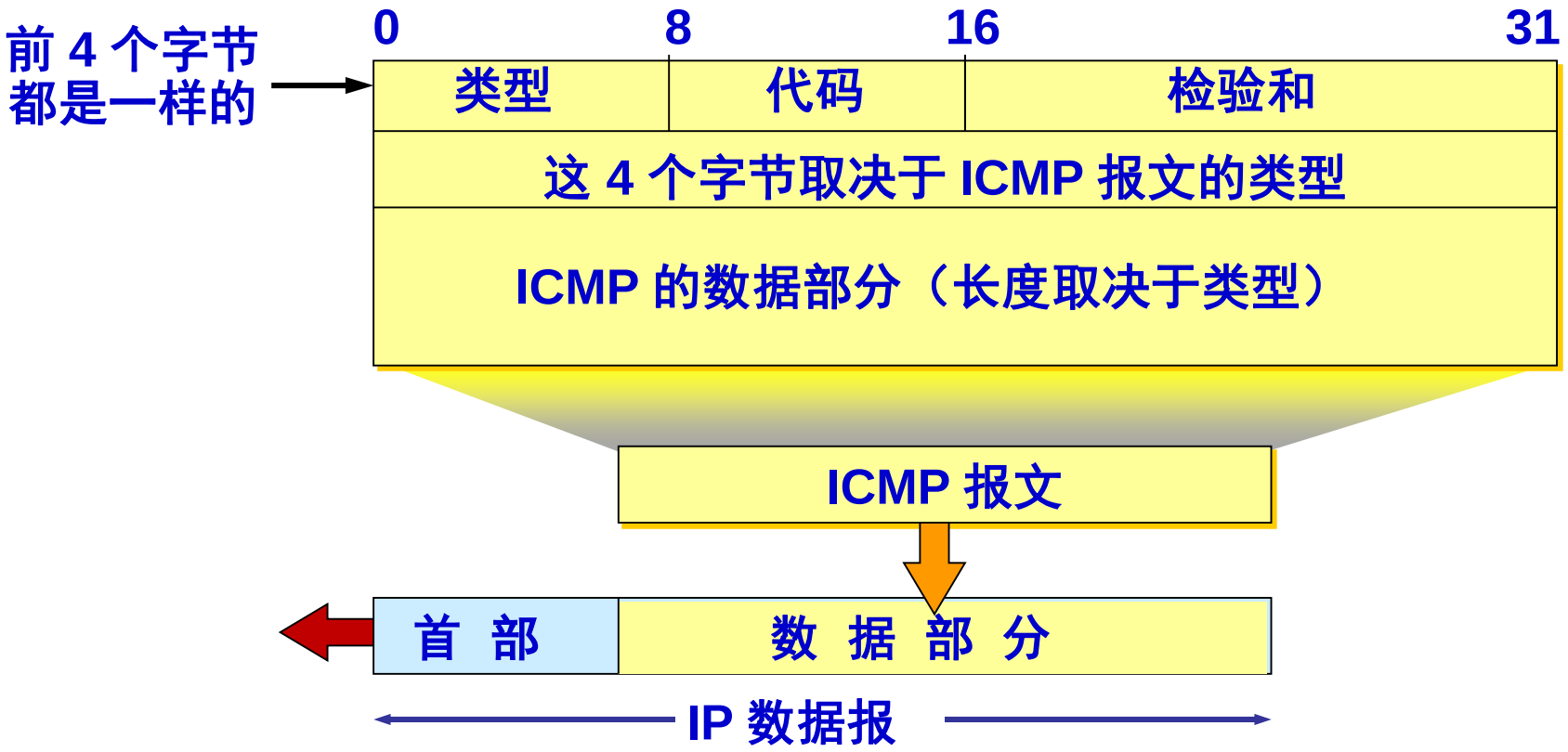
4.4 网际控制报文协议 ICMP

- 4.4.1 ICMP 报文的种类
- 4.4.2 ICMP 的应用举例

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 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


前 4 个字节
都是一样的
0
8
16
31
类型
代码
检验和
这 4 个字节取决于 ICMP 报文的类型
ICMP 的数据部分（长度取决于类型）
ICMP 报文
首 部
数 据 部 分
IP 数据报

4.4.1 ICMP 报文的种类

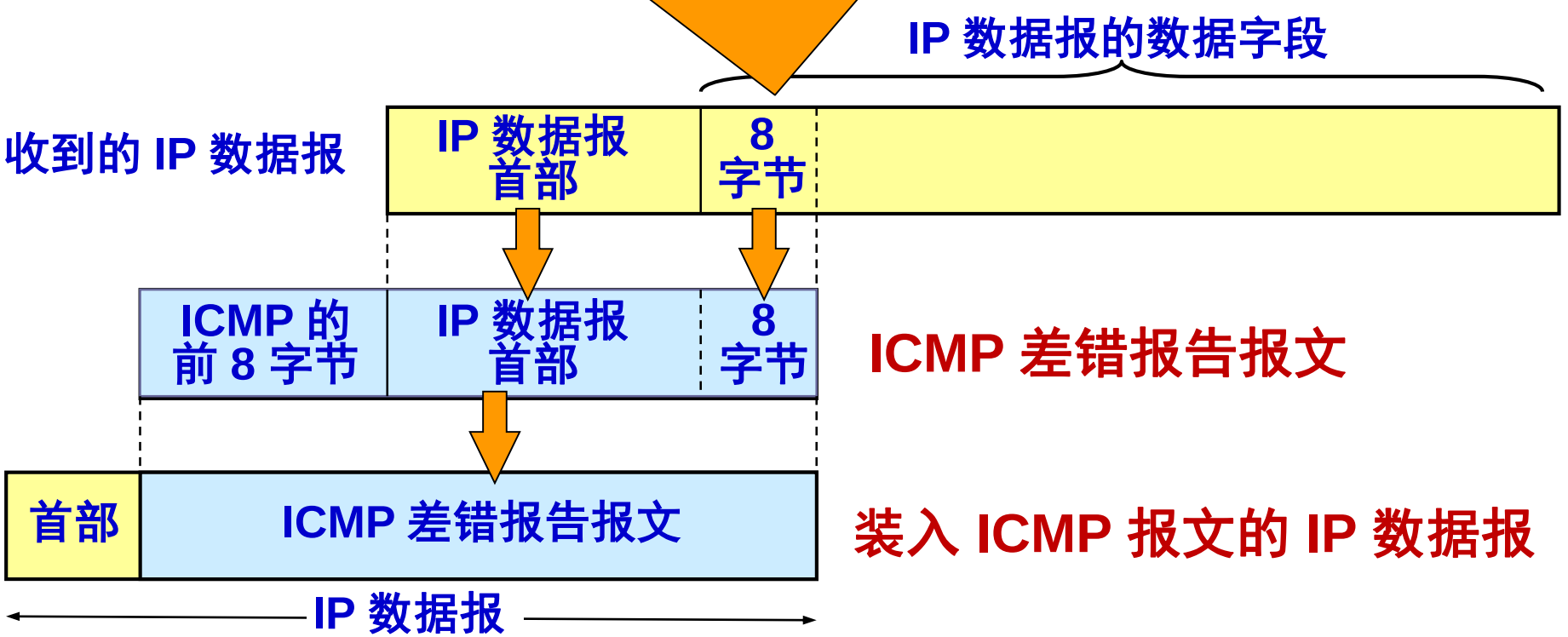
- ICMP 报文的种类有两种，即 ICMP 差错报告报文和 ICMP 询问报文。
- ICMP 报文的前 4 个字节是统一的格式，共有三个字段：即类型、代码和检验和。接着的 4 个字节的内容与 ICMP 的类型有关。
- 类型域用来指明消息的类型，有些消息还用代码域进一步定义说明。例如：类型为3的消息表示“目的不可达”的错误报告，每个消息的代码域进一步说明是“网络不可达”、“主机不可达”还是其他

ICMP 差错报告报文共有 4 种

- **差错报告**
  - 向源站点报告错误信息，不需要应答。
- **差错报告的类型**
  - ① **终点不可达**：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。
  - ② **时间超过**：路由器收到TTL为0的数据报时，丢弃且报告。
  - ③ **参数问题**：收到的数据报首部有字段不正确时。
  - ④ **改变路由**：路由器发给主机让其知道下次应将数据报发送给另外的路由器。
- **差错报告的方法**
  - 报文的数据部分包含ICMP差错报告报文的前8个字节，再加上引起错误的IP数据报的首部和数据字段的前8个字节。

ICMP 差错报告报文的数据字段的内容

为了得到运输层的端口号（TCP和UDP）以及运输层报文的发送序号，这些信息对源站的高层协议是有用的



不应发送 ICMP 差错报告报文的几种情况

① 对 ICMP 差错报告报文不再发送 ICMP 差错报告报文。

② 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。

③ 对具有多播地址的数据报都不发送 ICMP 差错报告报文。

④ 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。

ICMP 询问报文有两种

- 询问报文：采用请求/应答方式进行交互，用来请求一些消息。
- 询问报文的类型
  - 回送请求和回答报文
    - 由主机或者路由器向一个特定的目的主机发出询问，收到此报文的机器给源主机应答。一般用来测试目的机器是否可达。如：PING
  - 时间戳请求和回答报文
    - 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。

4.4.2 ICMP 的应用举例

PING (Packet InterNet Groper)

分组网间探测

- 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

4.4.2 ICMP 的应用举例

Traceroute 的应用举例

- 在 Windows 操作系统中这个命令是 tracert。

- 用来跟踪一个分组从源点到终点的路径。
- 它利用 IP 数据报中的 TTL 字段和 ICMP 时间超过差错报告报文、终点不可达差错报告报文 实现对从源点到终点的路径的跟踪。

4.4.2 ICMP 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.
```

用 tracert 命令获得到目的主机的路由信息

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

4.5.1 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，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。

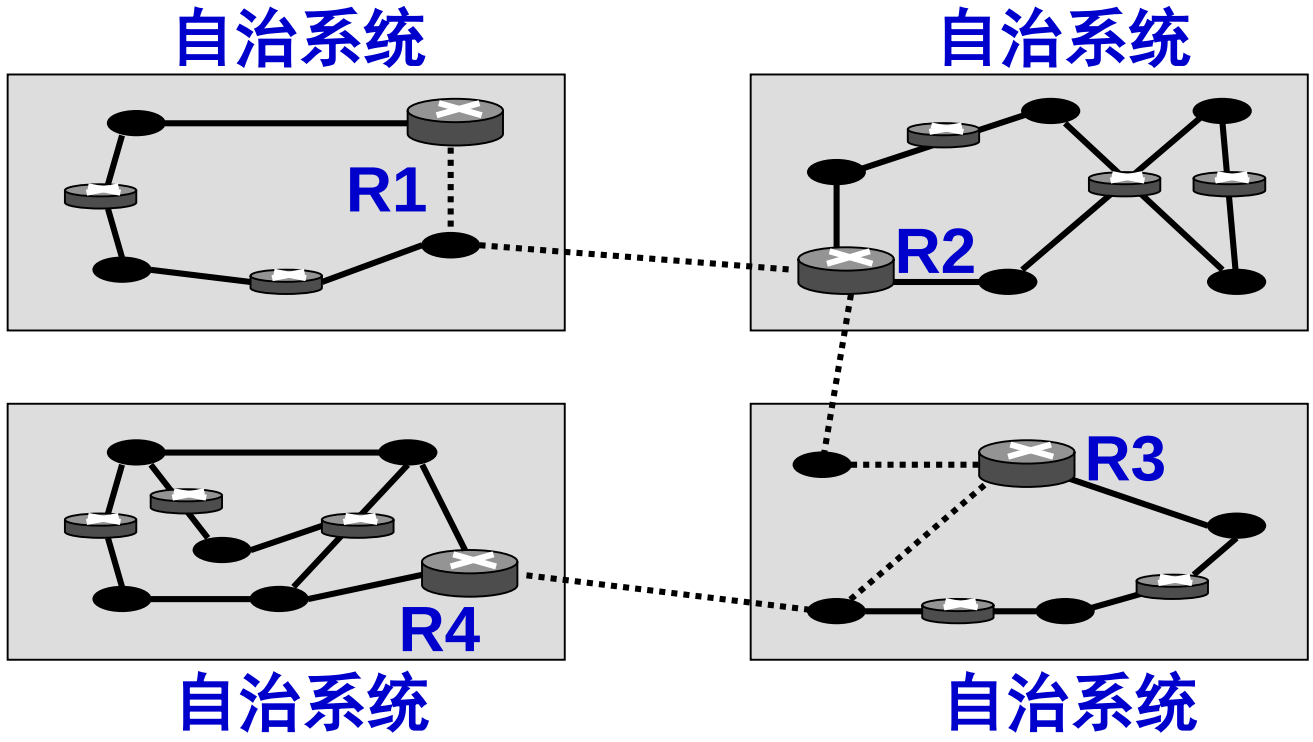
- **动态路由选择策略**：即**自适应路由选择**，其特点是能较好地适应网络状态的变化，但实现起来较为复杂，开销也比较大。

2. 分层次的路由选择协议

- 互联网的路由选择协议主要是自适应的（即动态的）、分布式的路由选择协议。
- 互联网采用分层次的路由选择协议。这是因为：
  - (1) 互联网的规模非常大。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互联网的通信链路饱和。
  - (2) 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互联网上。

自治系统 AS(Autonomous System)

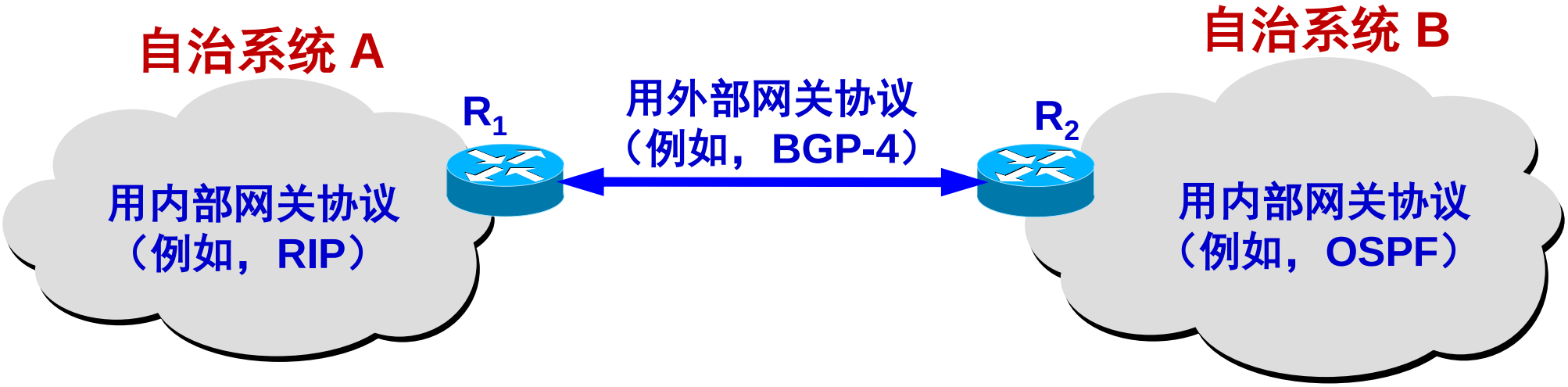
- **自治系统 AS 的定义：**在单一的技术管理下的一组路由器，而这些路由器使用一种 AS 内部的路由选择协议和共同的度量。
- **一个 AS 对其他 AS 表现出的是一个单一的和一致的路由选择策略。**



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



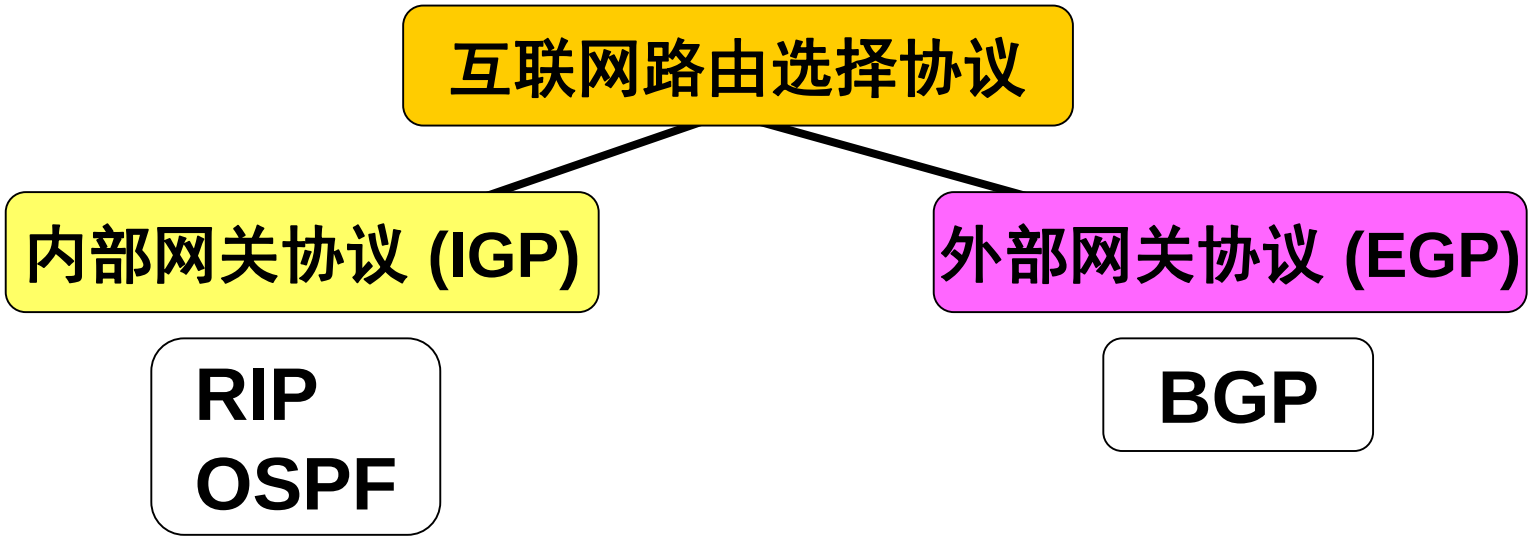
自治系统之间的路由选择也叫做**域间路由选择** (interdomain routing)，在自治系统内部的路由选择叫做**域内路由选择** (intradomain routing) 。

这里要指出两点

- 互联网的早期 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。应当把这两个术语当作同义词。
- 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。

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

- RIP (Routing Information Protocol) (路由信息协议)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、基于距离向量的动态路由选择协议。

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

到达目的网络所需的费用

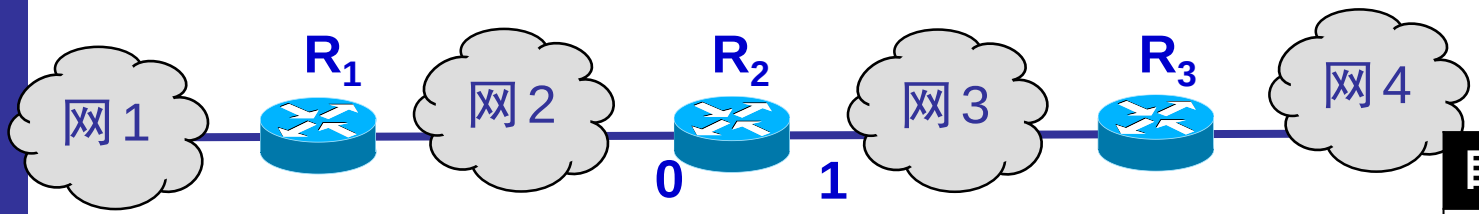
一组距离的记录

根据拓扑结构、通信量的变化来改变其路由选择

- 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唯一最佳距离记录。

“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**”。



交换

R_2的路由表

目的网络	距离	下一跳
网2	1	直接交付
网1	2	R1
网4	2	R3

“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

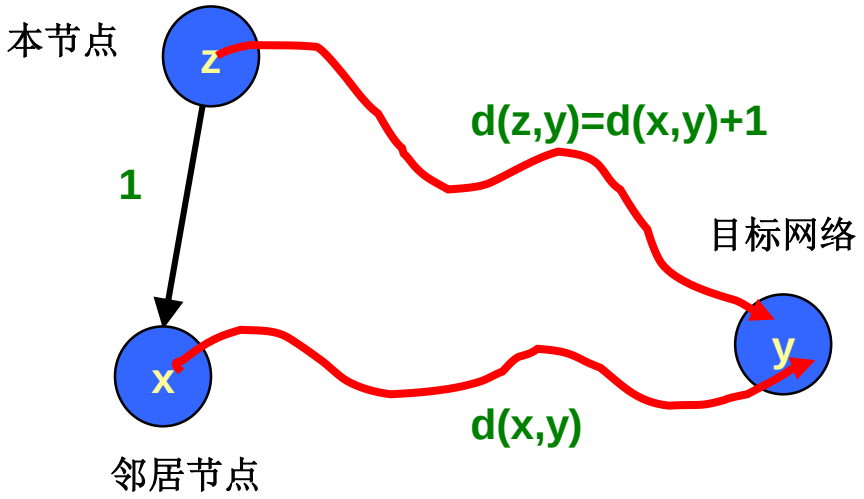
- (1) 和哪些路由器交换信息？
  - 仅和相邻路由器交换信息。
- (2) 交换什么信息？
  - 交换的信息是当前本路由器所知道的全部信息，即自己的路由表。
- (3)在什么时候交换信息？
  - 按固定的时间间隔交换路由信息，例如，每隔 30 秒。当网络拓扑发生变化时，路由器也及时向相邻路由器通告拓扑变化后的路由信息。若超过180s没收到邻居通告，则判定邻居没了并更新路由表。

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

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



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。

否则（目的网络已在路由表中）

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。

否则（下一跳地址不同）

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。

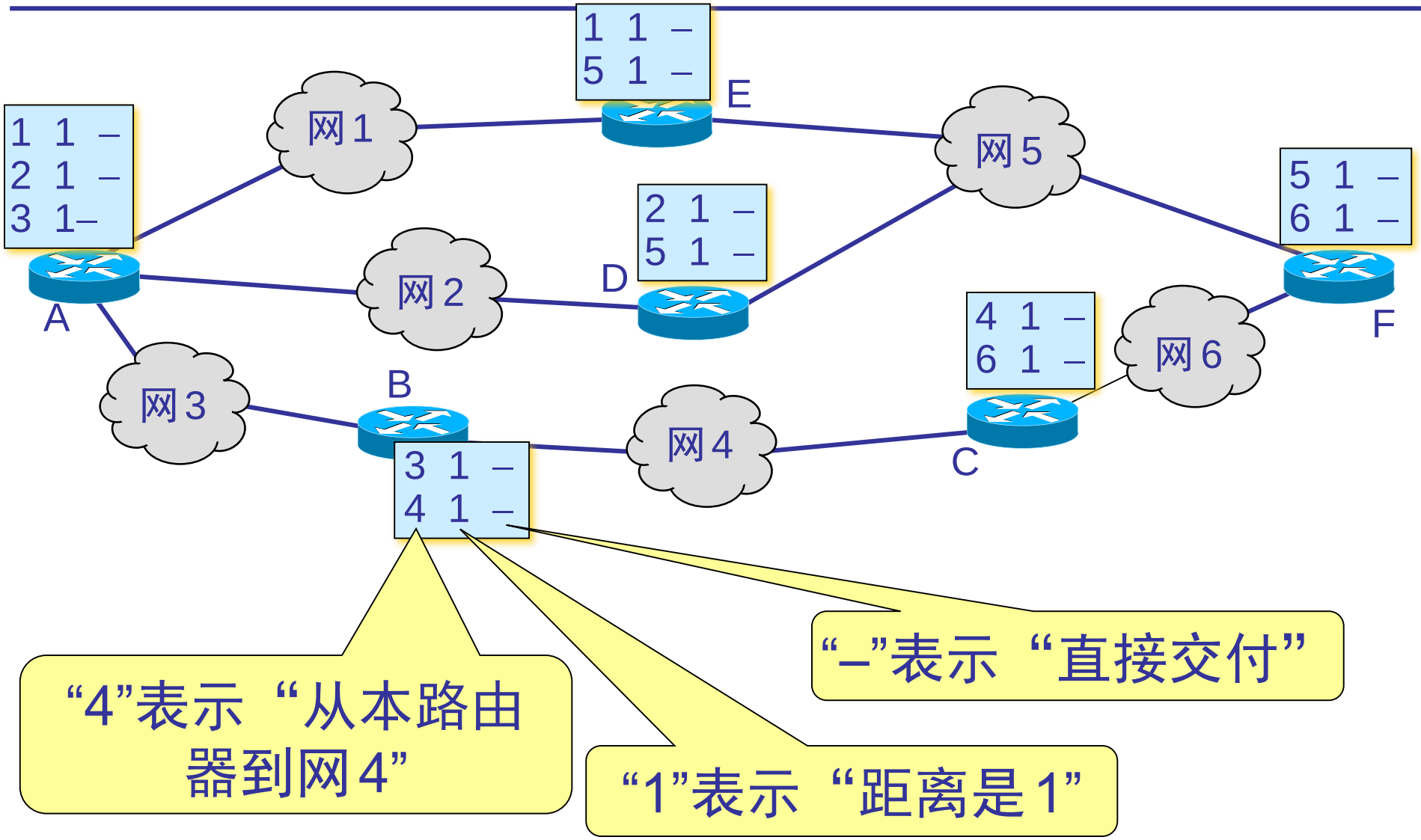
(3) 若 3 分钟还没有收到相邻路由器的更新路由表，则把此相邻路由器记为不可达路由器，即将距离置为 16（表示不可达）。

(4) 返回。

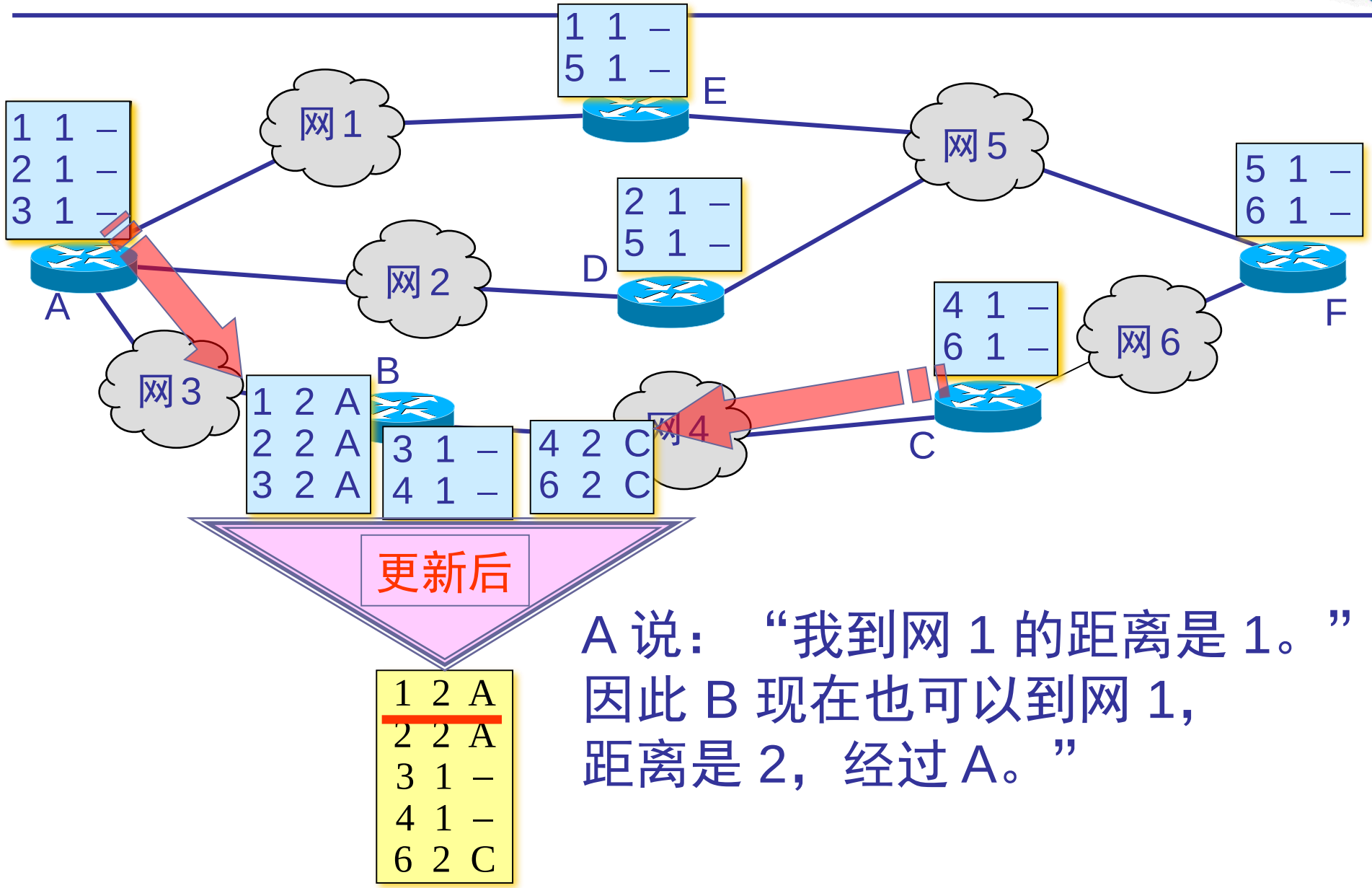
路由表的建立

- 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。
- RIP 协议的收敛 (convergence) 过程较快。“收敛”就是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。

一开始，各路由器只有到直接连接的网络信息

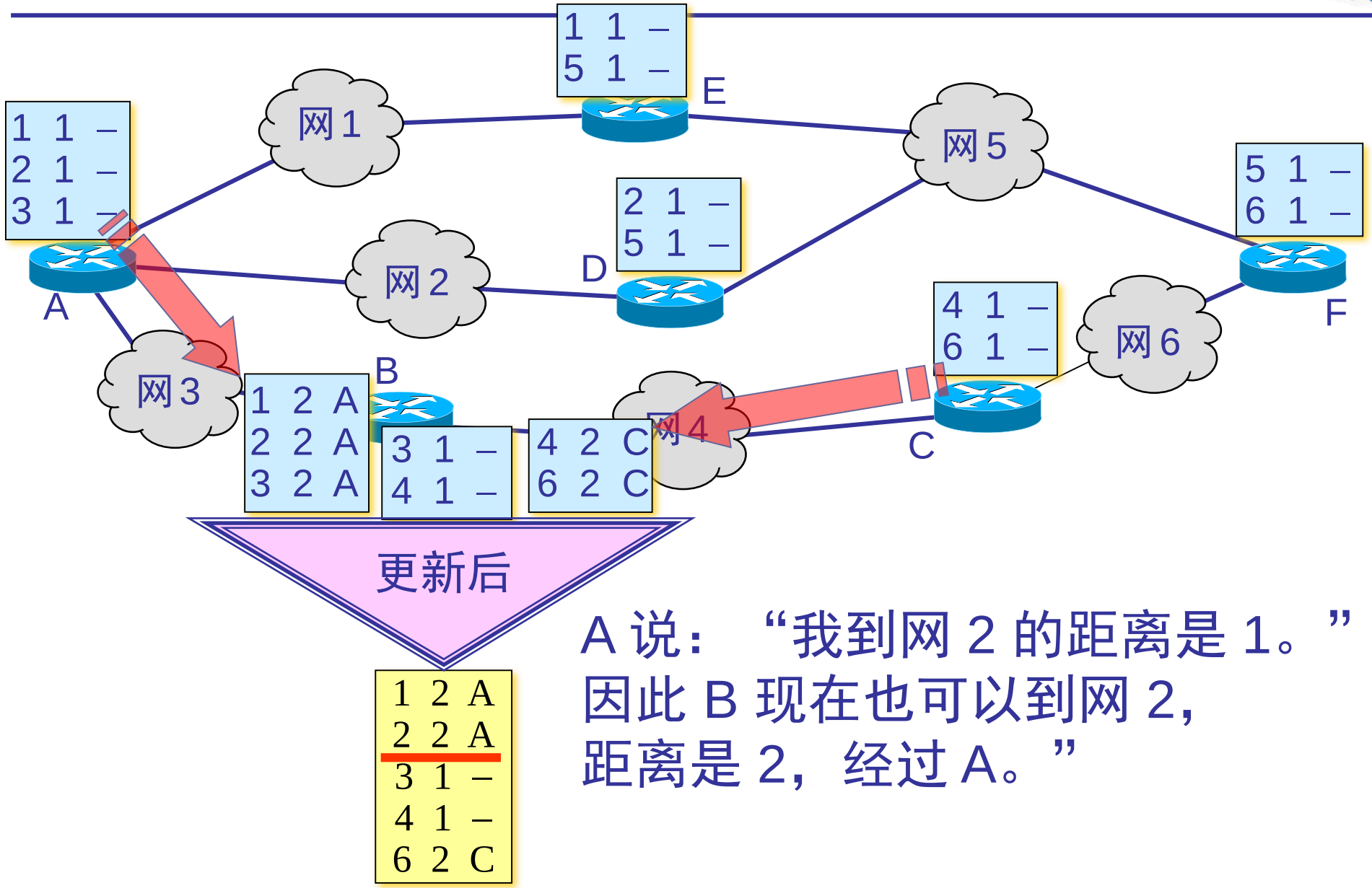


路由器 B 收到相邻路由器 A 和 C 的路由表



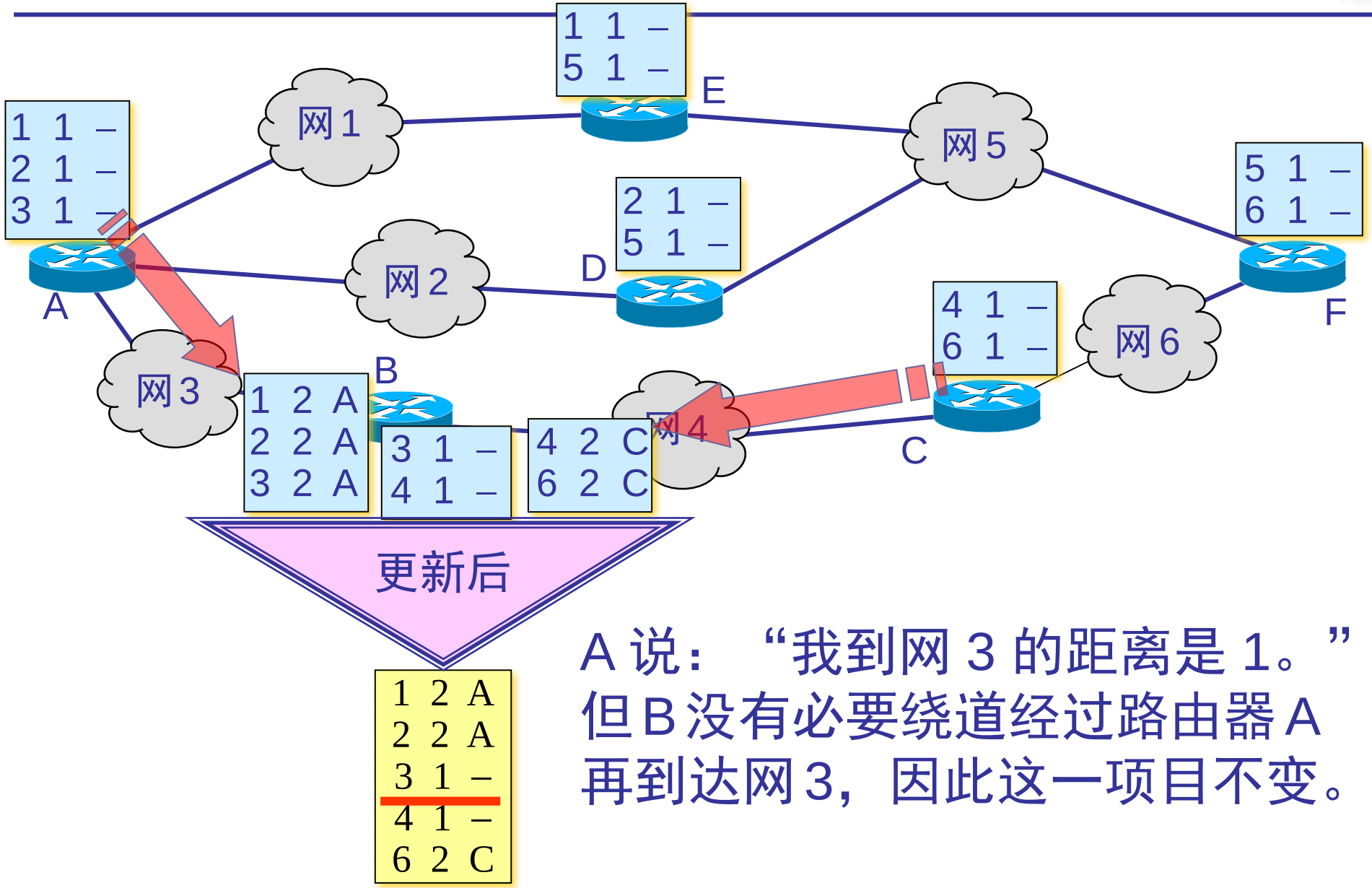
A 说：“我到网 1 的距离是 1。”
因此 B 现在也可以到网 1，
距离是 2，经过 A。”

路由器 B 收到相邻路由器 A 和 C 的路由表



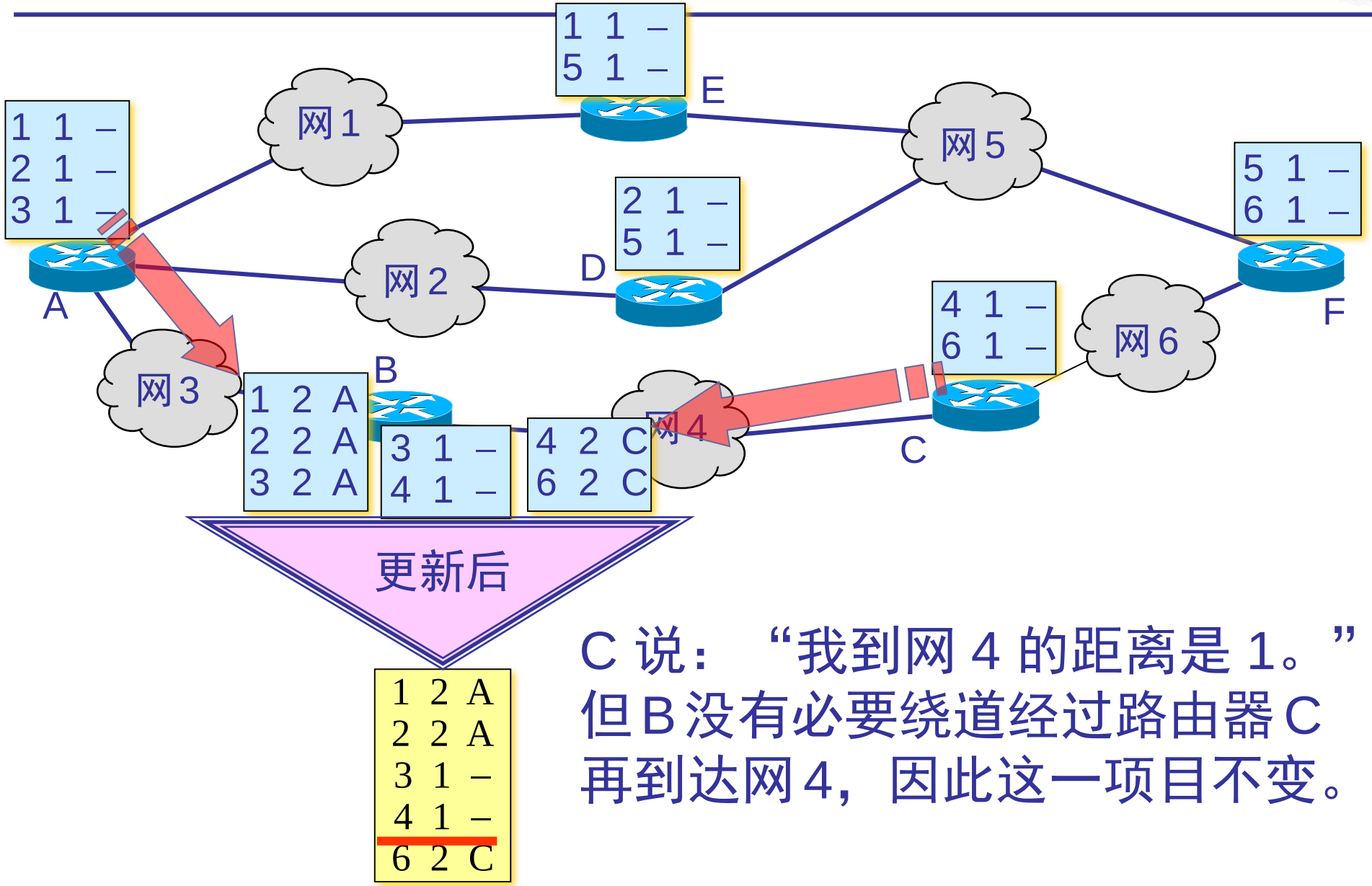
A 说：“我到网 2 的距离是 1。”
因此 B 现在也可以到网 2，
距离是 2，经过 A。”

路由器 B 收到相邻路由器 A 和 C 的路由表



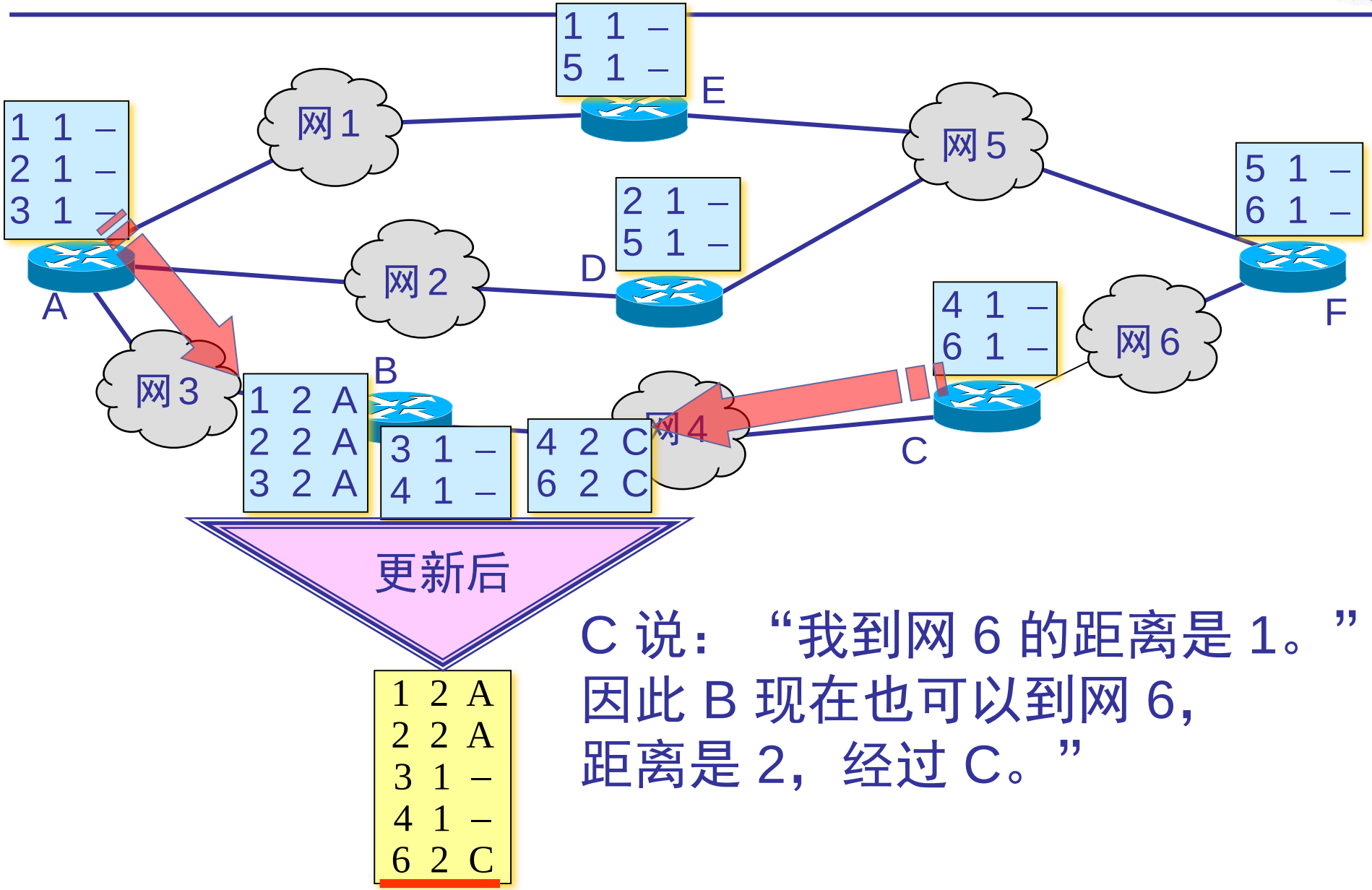
A 说：“我到网 3 的距离是 1。”
但 B 没有必要绕道经过路由器 A 再到达网 3，因此这一项目不变。

路由器 B 收到相邻路由器 A 和 C 的路由表



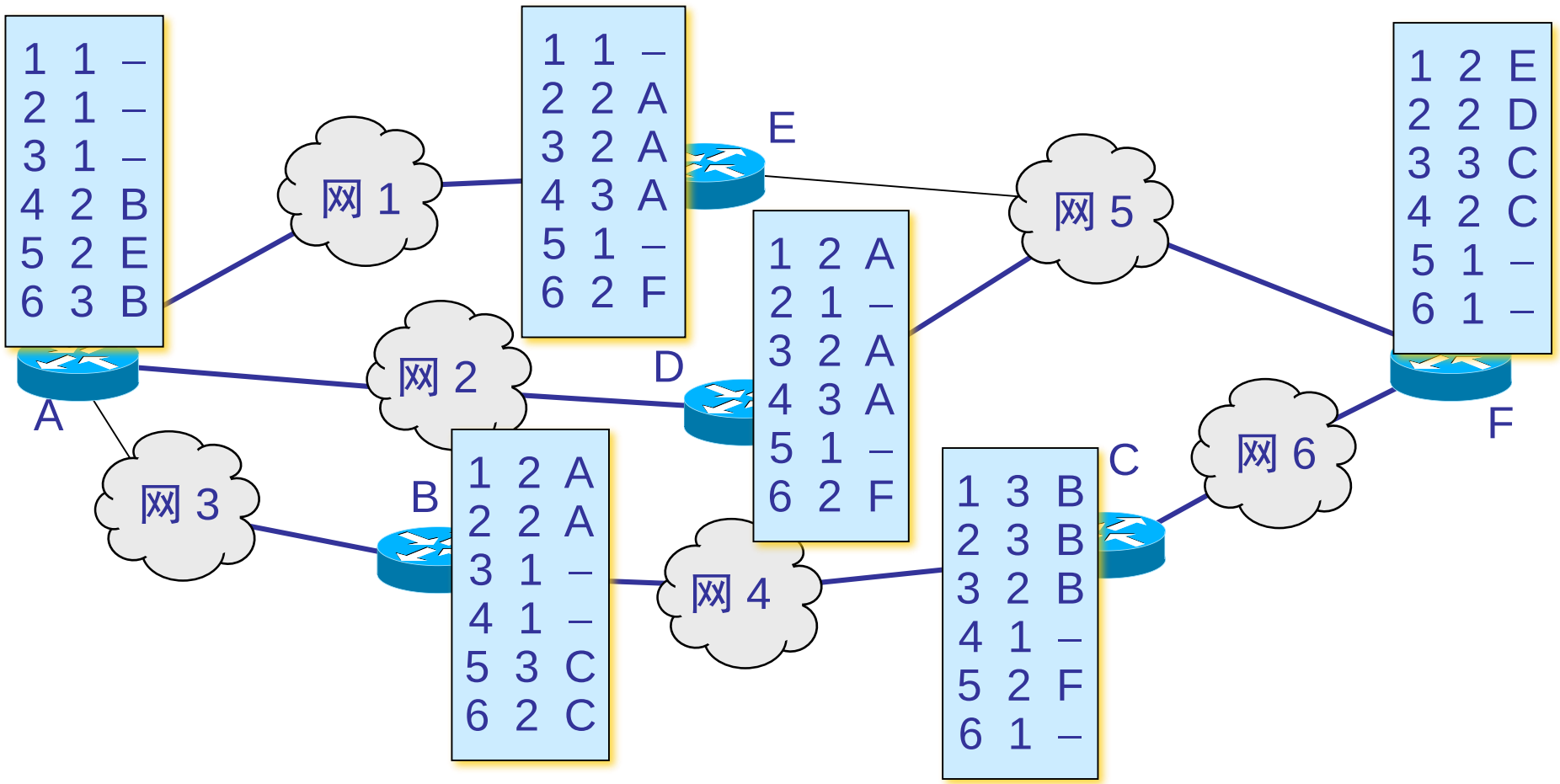
C 说：“我到网 4 的距离是 1。”
但 B 没有必要绕道经过路由器 C
再到达网 4，因此这一项目不变。

路由器 B 收到相邻路由器 A 和 C 的路由表



C 说：“我到网 6 的距离是 1。”
因此 B 现在也可以到网 6，
距离是 2，经过 C。”

最终所有的路由器的路由表都更新了



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→B 拆成两段路径 A→X 和 X→B，则每一段路径 A→X 和 X→B 也都分别是结点 A 到 X 和结点 X 到 B 的最短路径。

路由表更新

- RIP 协议让一个自治系统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。

【例1】已知路由器 R_6 有表 4-9(a) 所示的路由表。现在收到相邻路由器 R_4 发来的路由更新信息，如表 4-9(b) 所示。试更新路由器 R_6 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) R_4 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

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_4
Net2	5	R_4
Net3	2	R_4

计算更新

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
…	…	…

【例2】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	C
Net3	9	C
Net6	5	C
Net8	4	C
Net9	6	C

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

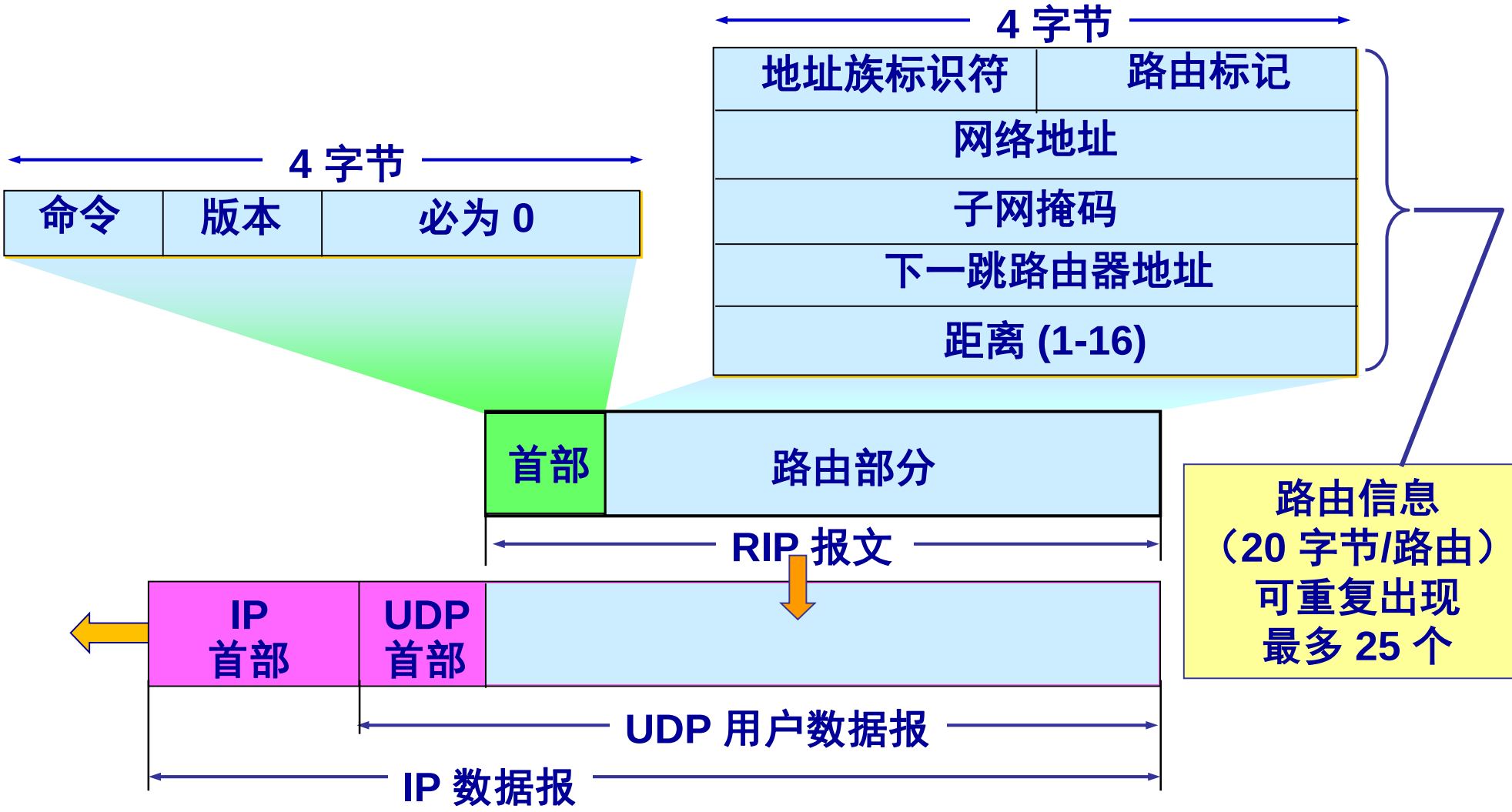
Net1	7	A
Net2	2	C
Net6	8	F
Net8	4	E
Net9	4	F

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

3. RIP2 协议的报文格式



RIP2 报文

- RIP2 是1998年公布的较新的RIP版本。
- RIP2 报文由首部和路由部分组成。
- 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 报文

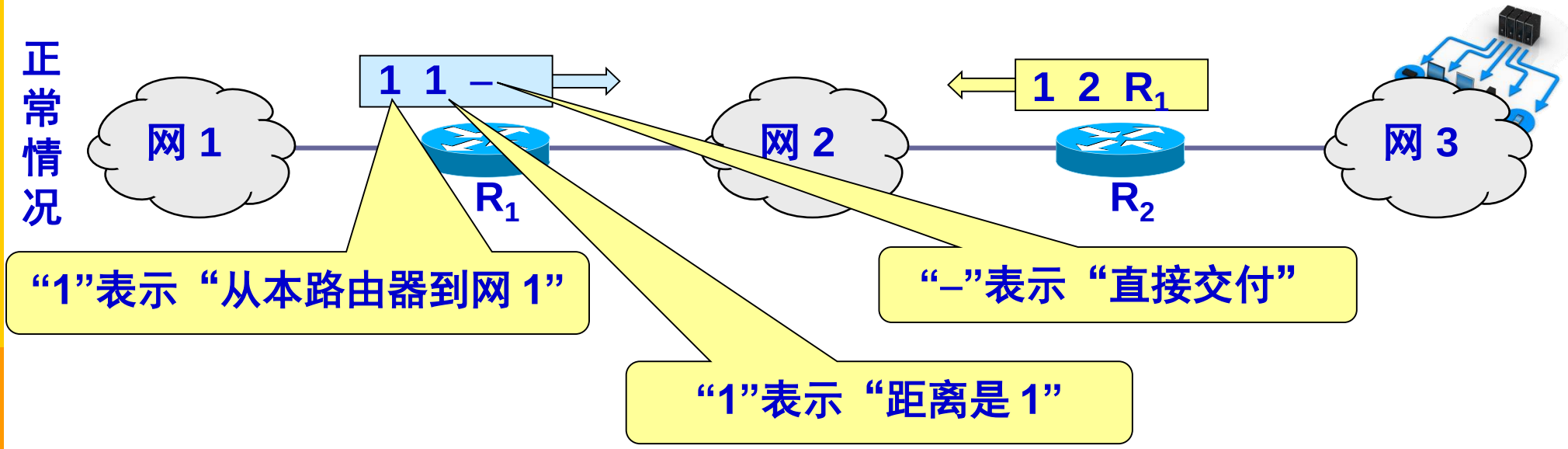
- 路由部分
  - 由若干个路由信息组成。每个路由信息需要用 20 个字节。
  - 地址族标识符（又称为地址类别）字段用来标志所使用的地址协议。（IP地址该字段为2）
  - 路由标记填入自治系统号ASN(Autonomous System Number)，这是考虑使 RIP 有可能收到本自治系统以外的路由选择信息。
  - 再后面指出某个网络地址、该网络的子网掩码、下一跳路由器地址以及到此网络的距离。
- 最大RIP报文长度为：4＋20*25＝504字节（最多包含25条路由）。如超过，必须再用一个 RIP 报文来传送。

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。
  - 当网络出现故障时，要经过比较长的时间才能将此信息传送到所有的路由器。

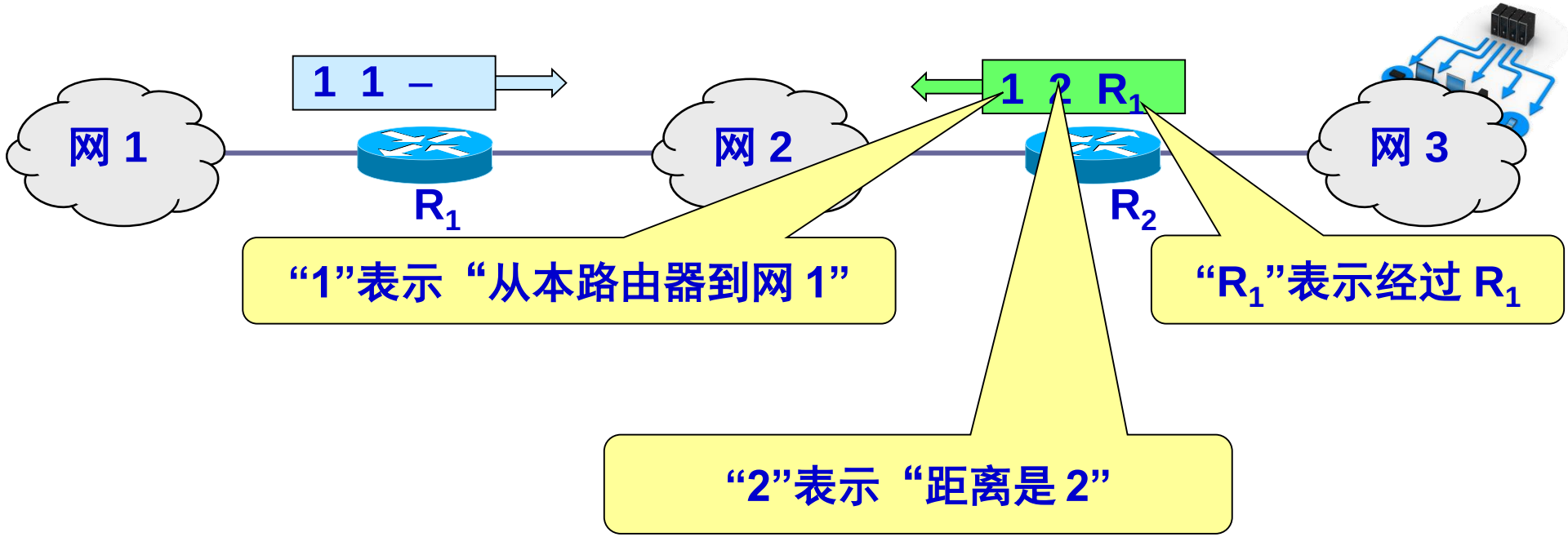
好消息传播得快，坏消息传播得慢。

正常情况



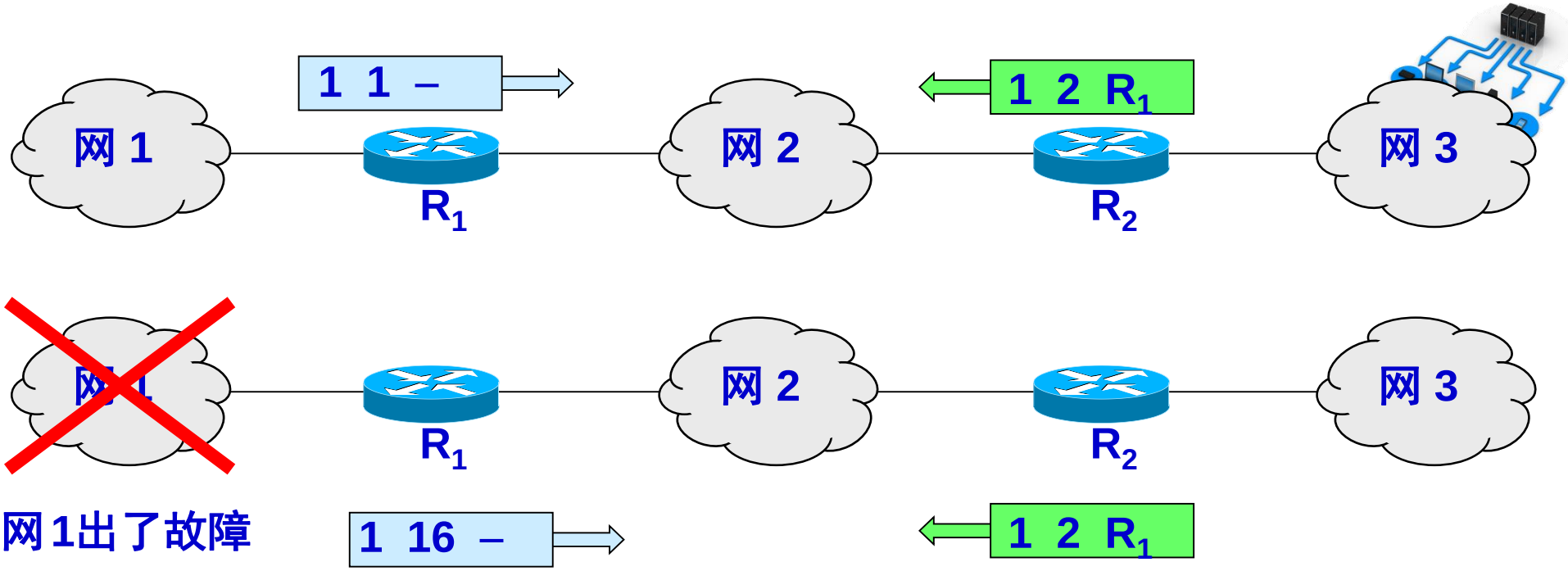
R_1 说：“我到网 1 的距离是 1，是直接交付。”

正常情况



R_2 说：“我到网 1 的距离是 2，是经过 R_1。”

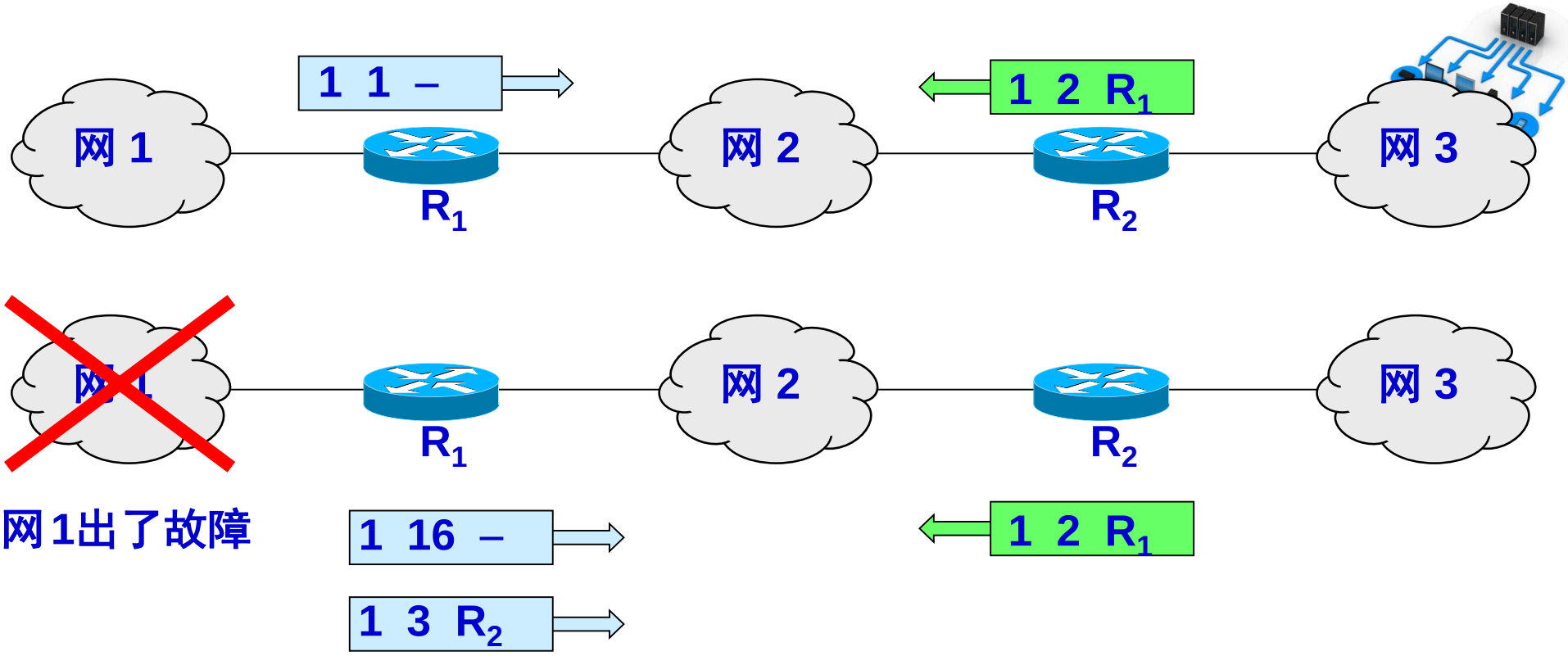
正常情况



R_1 说：“我到网 1 的距离是 16 （表示无法到达），直接交付”

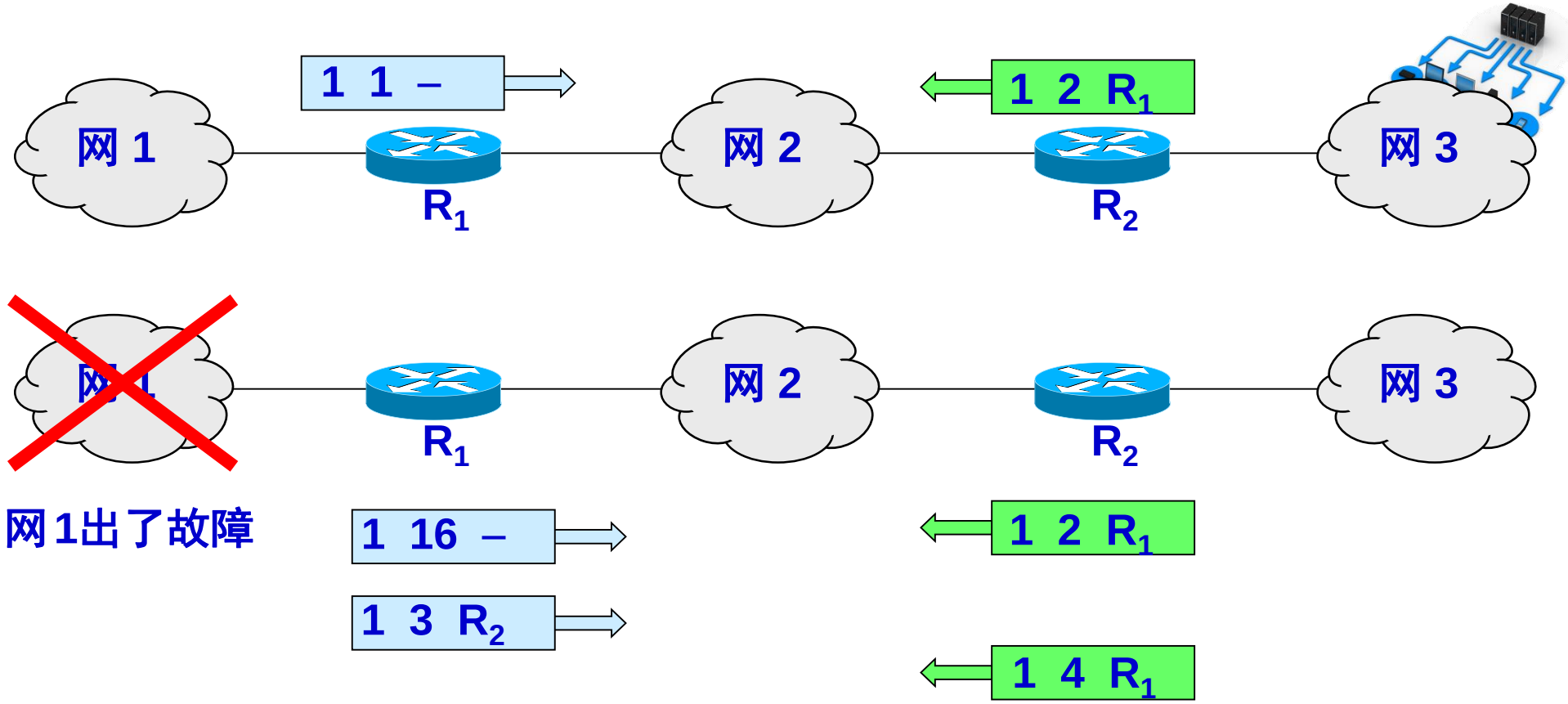
但 R_2 在收到 R_1 的更新报文之前，还发送原来的报文，因为这时 R_2 并不知道 R_1 出了故障。

正常情况



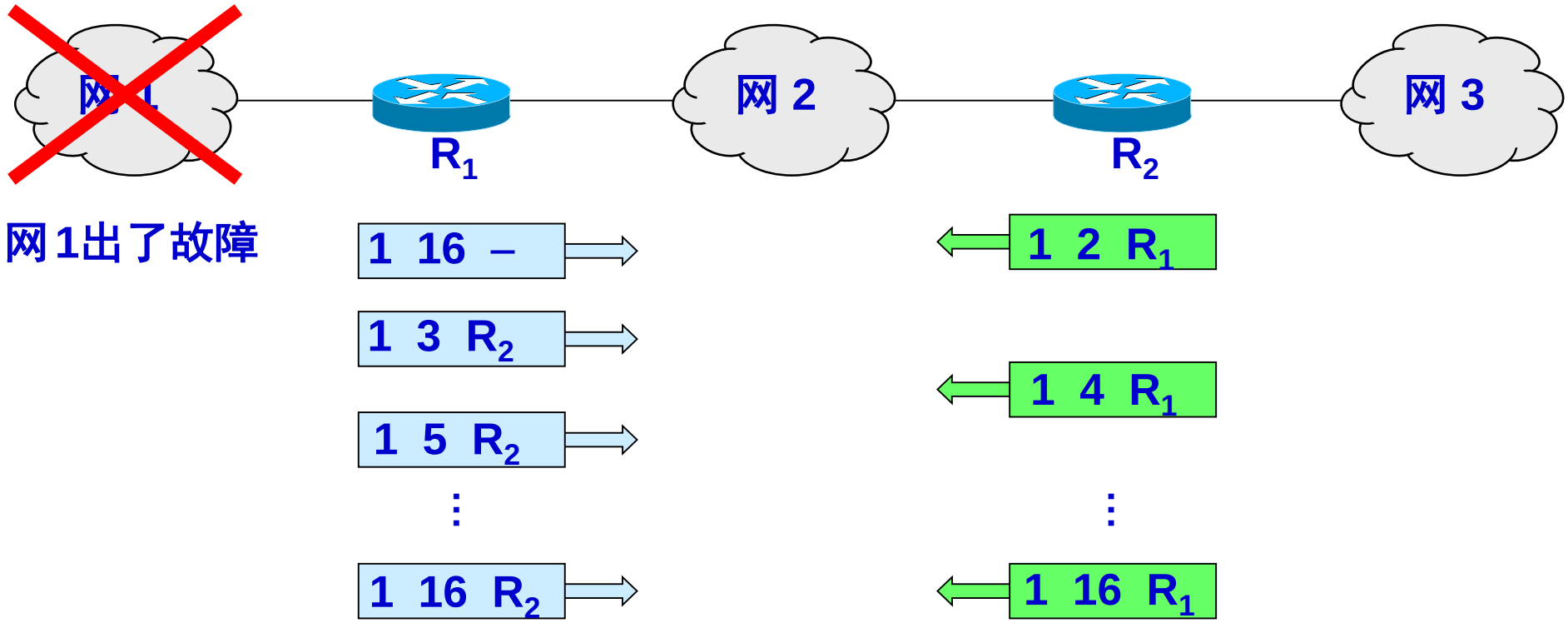
R_1 收到 R_2 的更新报文后，误认为可经过 R_2 到达网 1，于是更新自己的路由表，说：“我到网 1 的距离是 3，下一跳经过 R_2”。然后将此更新信息发送给 R_2。

正常情况



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”。

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。

4.5.3 内部网关协议 OSPF

1. 基本特点

- OSPF(Open Shortest Path First)(开放最短路径优先)协议，是为克服RIP的缺点在1989年开发出来的。
- “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。
- 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”。
- 最主要的特征：采用分布式的链路状态协议 (link state protocol)，而不是RIP那样的距离向量协议。

三个要点

(1) 和哪些路由器交换信息？

- 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) 交换什么信息？

- 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**。
  - “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)：费用、距离、时延、带宽等。

(3) 在什么时候交换信息？

- 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)

- 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）。
- 每个路由器都知道全网有多少路由器，哪些路由器是相连的，代价是多少等。
- 每个路由器根据数据状态数据库，构造出自己的路由表。
- 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。

2.链路状态协议

- **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。

链路状态路由算法

- 1. 每个路由器发现它的邻居节点，发送问候 (Hello) 分组，了解邻居节点的网络地址。
- 2. 设置它到每个邻居的成本度量（Metric）。
- 3. 构造数据库描述 (Database Description) 分组，向邻站给出自己的链路状态数据库所有项目的摘要信息。
- 4. 如果DD分组中的摘要自己都有，则邻站不做处理；如果没有或者是更新的，则发送链路状态请求 (Link State Request) 分组请求新信息。
- 5. 收到邻站的LSR分组后，发送链路状态更新 (Link State Update) 分组进行更新。
- 6. 更新完毕后，邻站返回一个链路状态确认 (Link State Acknowledgment)分组进行确认。

只要一个路由器的链路状态发生变化：

- 5. 泛洪发送链路状态更新 (Link State Update) 分组进行更新。
- 6. 更新完毕后，邻站返回一个链路状态确认 (Link State Acknowledgment)分组进行确认。

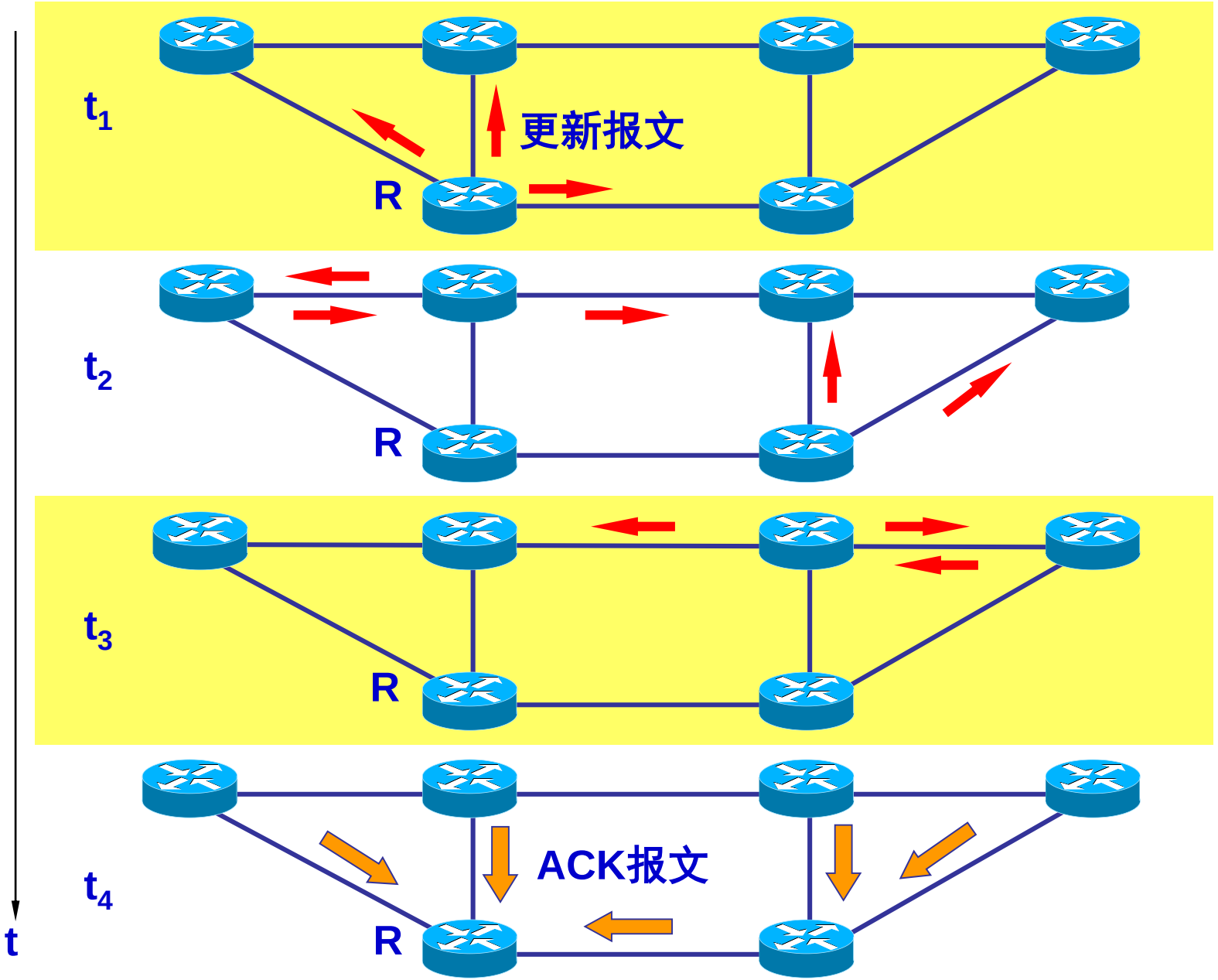
OSPF 的基本操作


确定可达性
问候
问候
达到数据库的同步
数据库描述
数据库描述
数据库描述
数据库描述
新情况下的同步
链路状态请求
链路状态更新
链路状态确认

OSPF 的五种分组类型

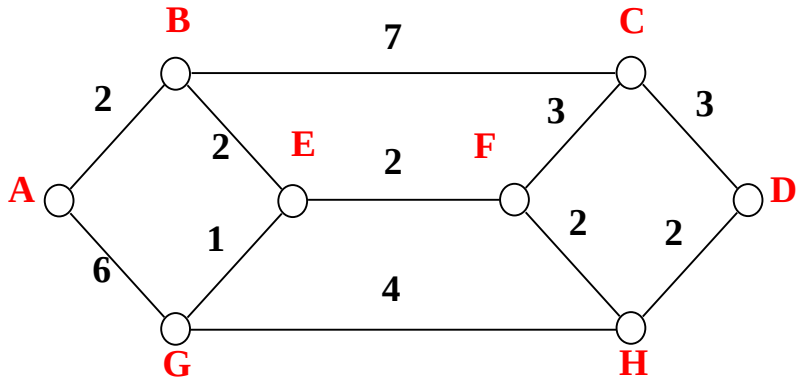
- 类型1，问候 (Hello) 分组，用来发现和维持邻站的可达性。
- 类型2，数据库描述 (Database Description) 分组，向邻站给出自己的链路状态数据库所有项目的摘要信息。
- 类型3，链路状态请求 (Link State Request) 分组，向对方请求发送某些链路状态项目的详细信息。
- 类型4，链路状态更新 (Link State Update) 分组，用洪泛法对全网更新链路状态。OSPF协议最核心的部分
- 类型5，链路状态确认 (Link State Acknowledgment)分组，对链路更新分组的确认。

OSPF 使用可靠的洪泛法发送更新分组



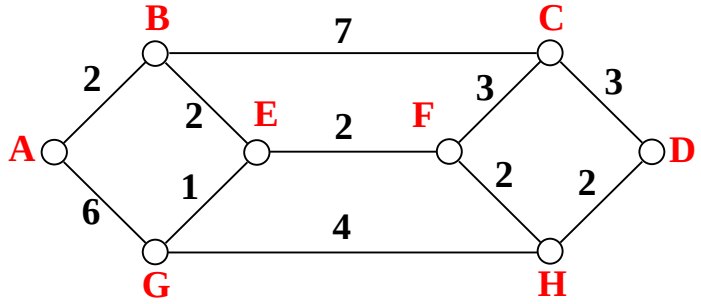
最短路径问题

- 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。
- 所谓路由最短可以是指：
  - 物理距离近
  - 分组传输延迟时间最小
  - 通信费用最低



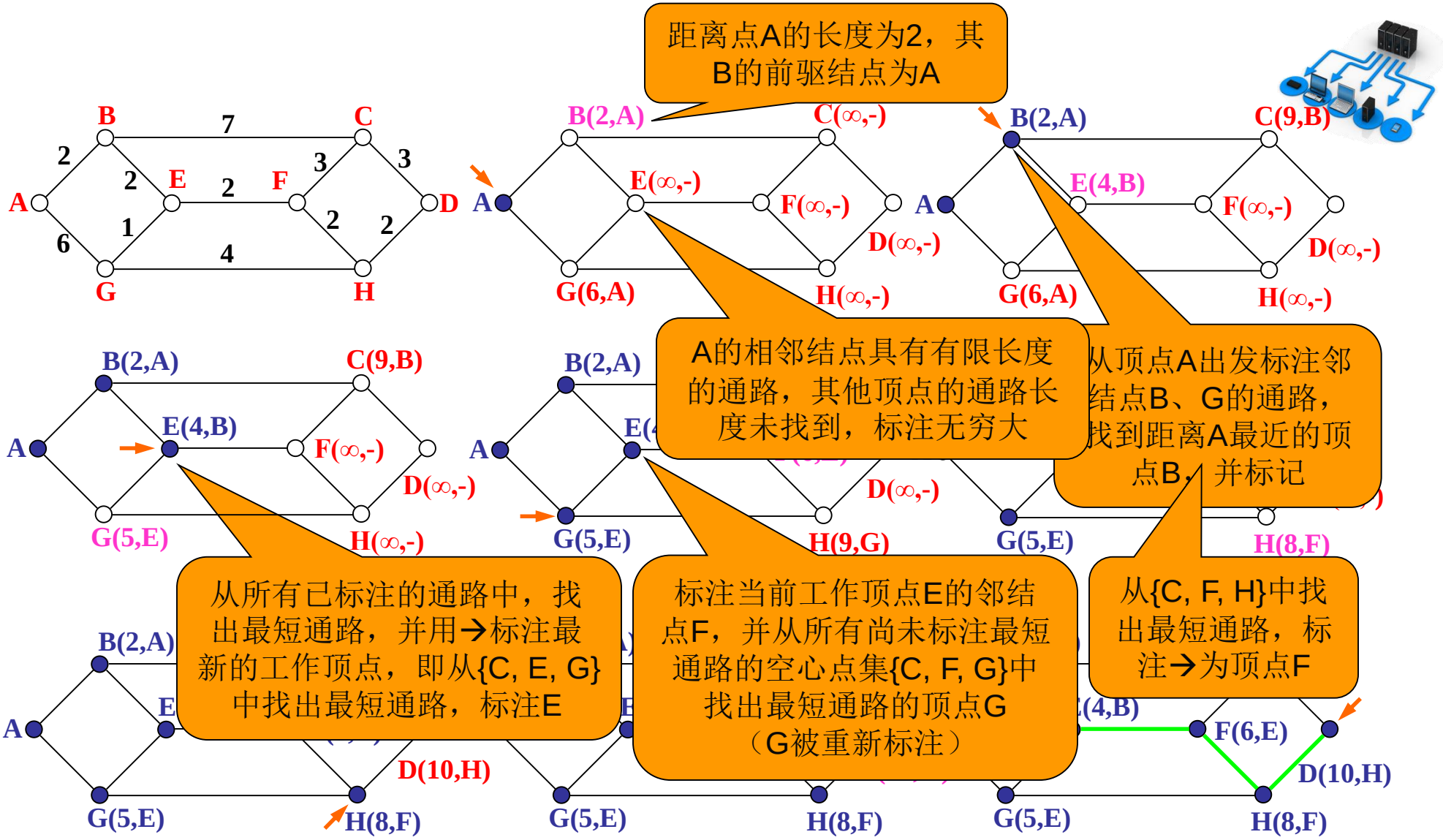
迪杰斯特拉(Dijkstra)算法

- **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

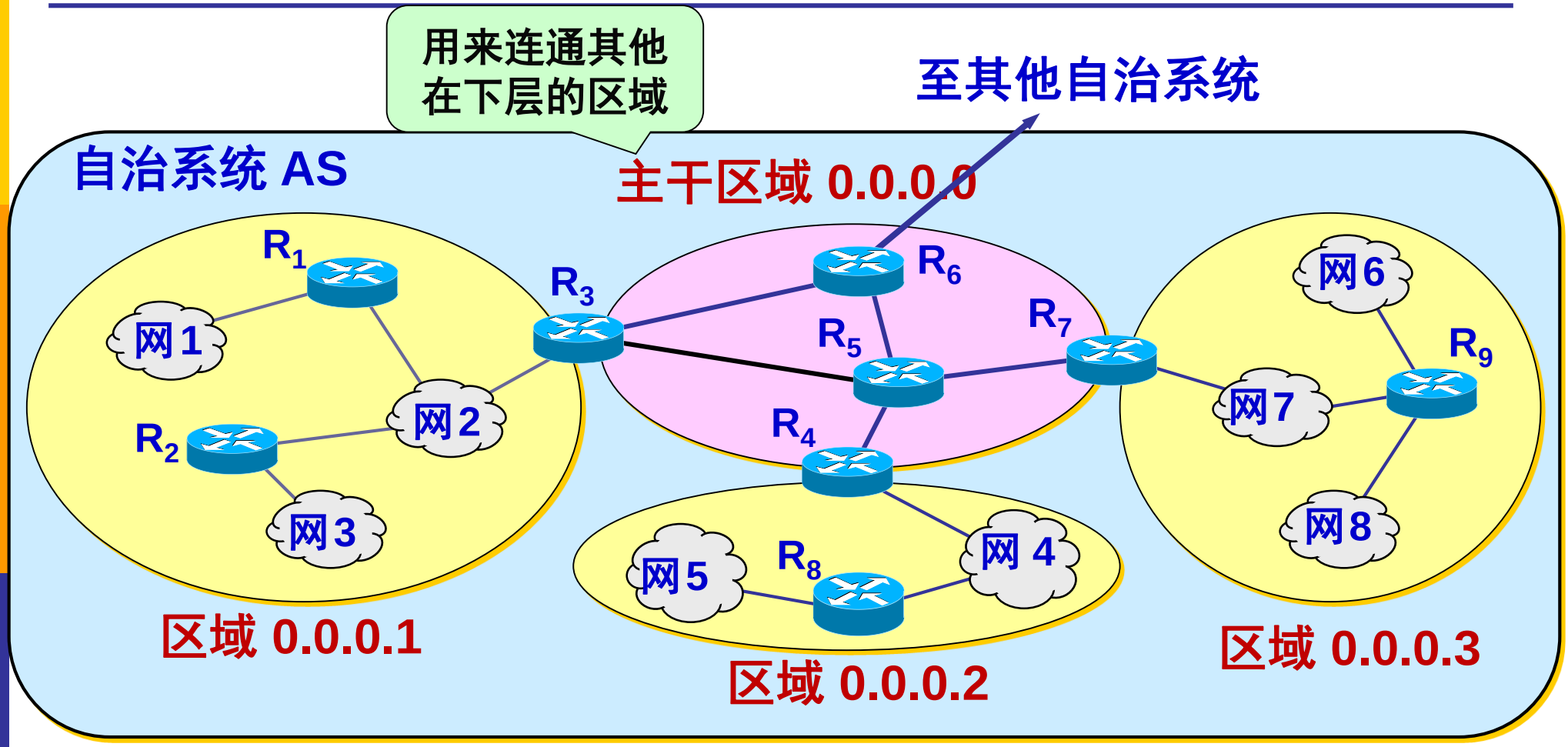
Dijkstra算法的C语言实现

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

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

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

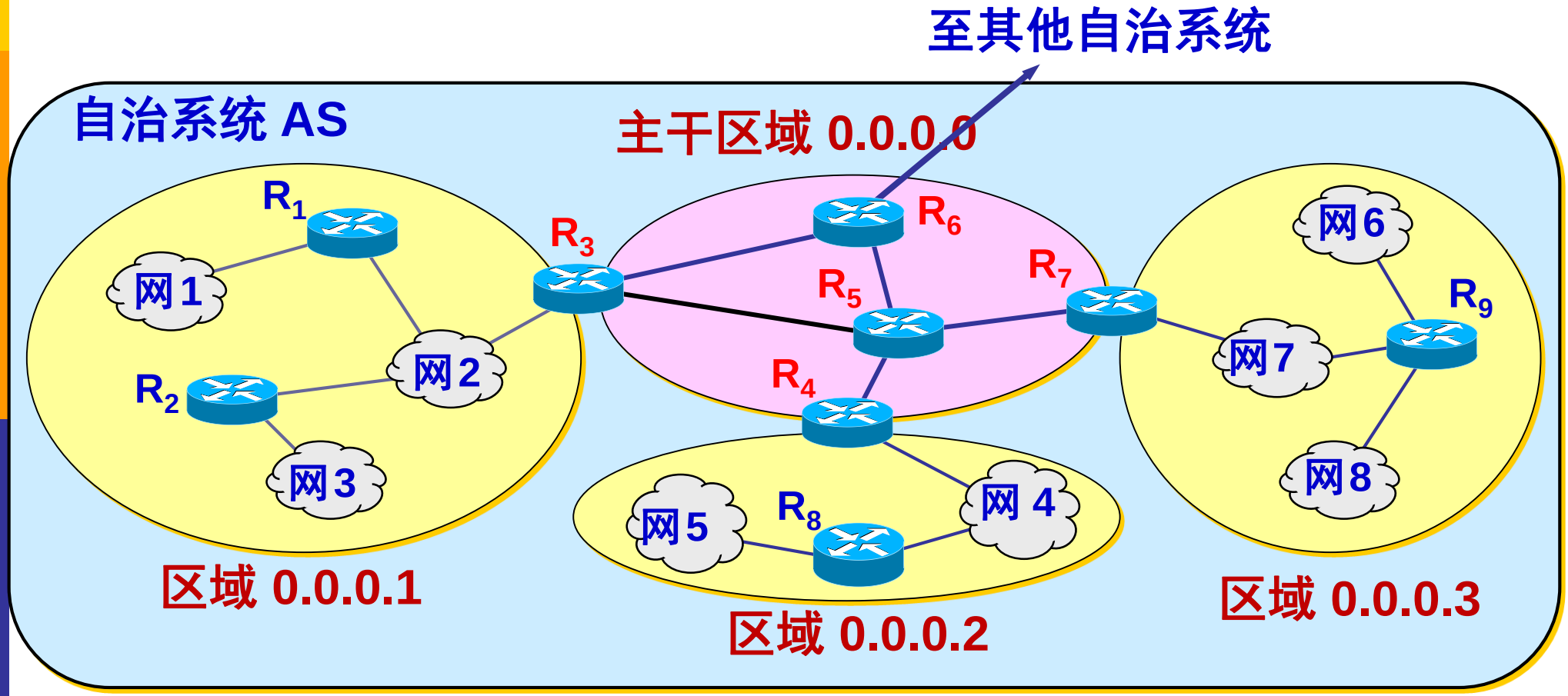
OSPF 的区域 (area)

- 为了使 OSPF 能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。
- 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

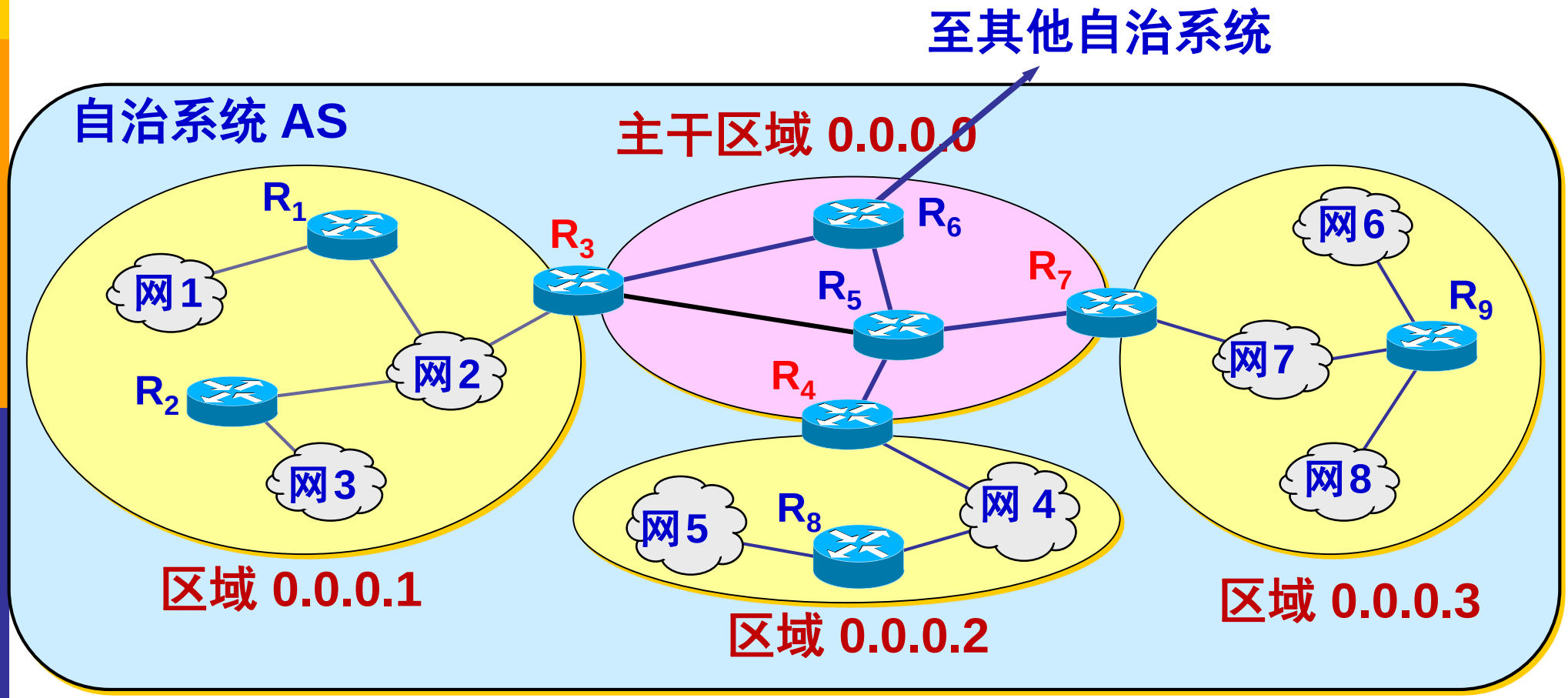
OSPF 划分为两种不同的区域



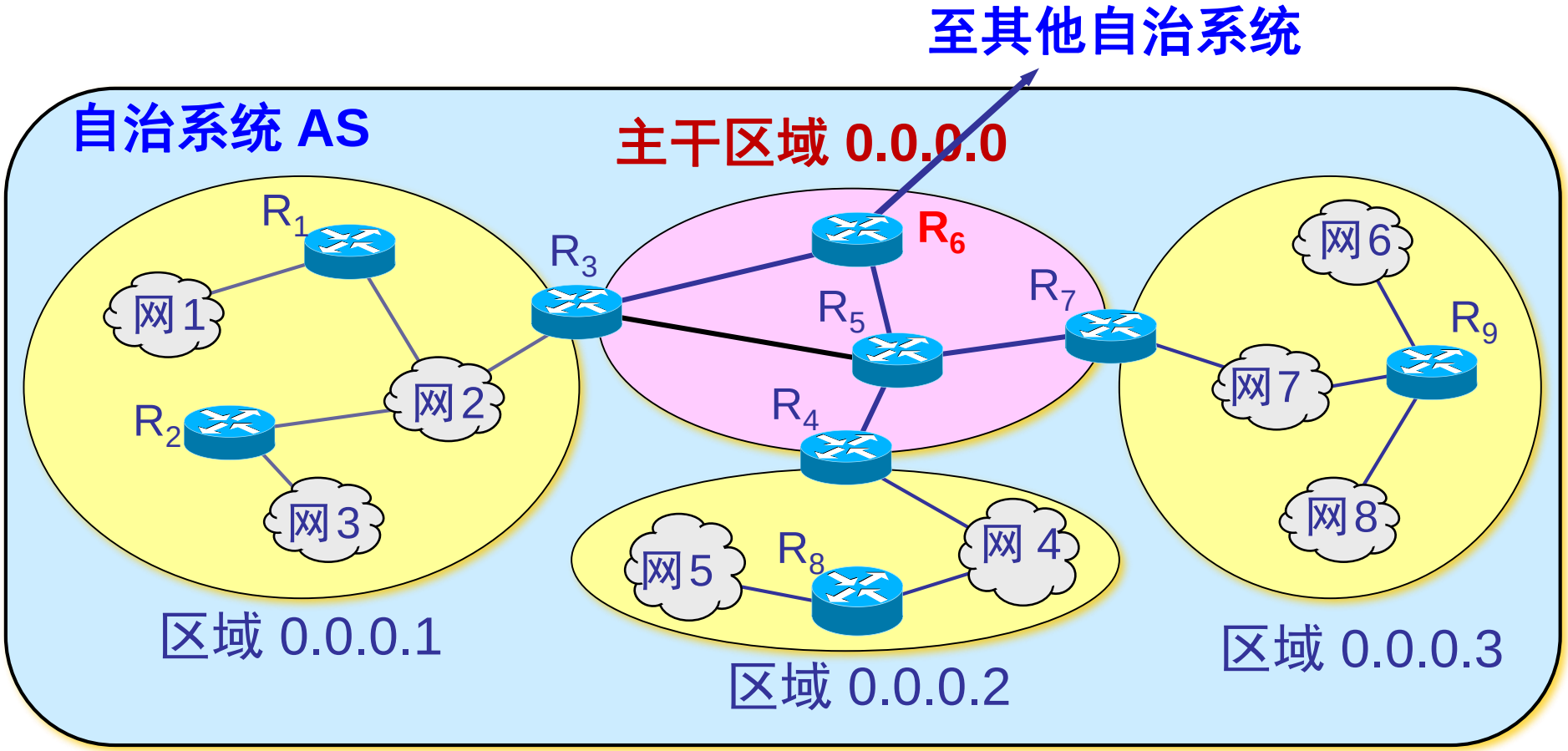
主干路由器
(backbone router)



区域边界路由器
(area border router)



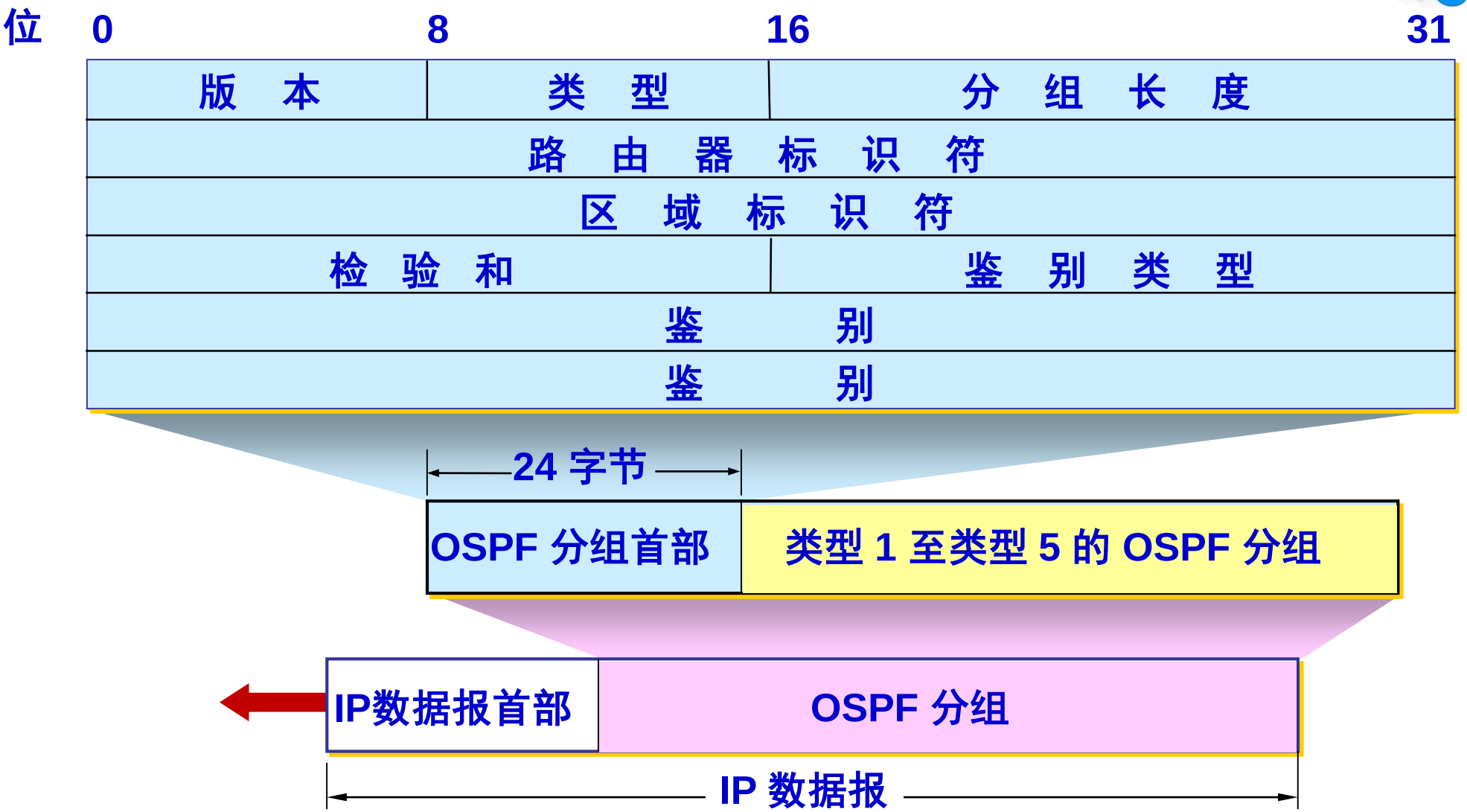
自治系统边界路由器


至其他自治系统
自治系统 AS
主干区域 0.0.0.0
R1
R2
R3
R4
R5
R6
R7
R8
R9
网1
网2
网3
网4
网5
网6
网7
网8
区域 0.0.0.1
区域 0.0.0.2
区域 0.0.0.3

3.OSPF 协议的报文格式

- 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。
  - 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。
- OSPF 分组使用24字节的固定长度首部

OSPF 分组



OSPF 分组用 IP 数据报传送

OSPF首部字段的意义

- 版本：当前版本号是2，
- 类型：OSPF 分组有五种类型，
- 分组长度：以字节为单位，
- 路由器标识符：标志发送该分组的路由器的接口的IP地址
- 区域标识符：分组属于区域的标识符，
- 校验和：用来检测分组中的差错，
- 鉴别类型：有0（不用）和1（口令）两种
- 鉴别：鉴别类型为1时填入8个字符的口令

OSPF 的特点

- 允许管理员给每条路由指派不同的代价(1至65535)
- 负载均衡 (load balancing)：到同一个网络有多条相同代价的路径，可以将通信量分配给这几条路径。
- 鉴别功能保证了仅在可信赖的路由器之间交换信息。
- 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，序号越大状态就越新。解决重复分组问题。

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，因而与整个互联网的规模并无直接关系。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。

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

4.5.4 外部网关协议 BGP

- BGP (Border Gateway Protocol)(边界网关协议) 是不同自治系统的路由器之间交换路由信息的协议，目前使用最多的版本是BGP-4
- 为什么不能使用内部网关协议？
  - 互联网的规模太大，使得自治系统之间路由选择非常困难。
    - 主干网路由器的项目表已超过了5万个网络前缀。对于自治系统之间的路由选择，要寻找最佳路由是很不现实的。
    - 当一条路径通过几个不同 AS 时，要想对这样的路径计算出有意义的代价是不太可能的。
    - 比较合理的做法是在 AS 之间交换“可达性”信息。例：“到达目的网络 N 可经过自治系统 AS_X”。
  - 自治系统之间的路由选择必须考虑有关策略。
    - 包括政治、安全或经济方面的考虑。

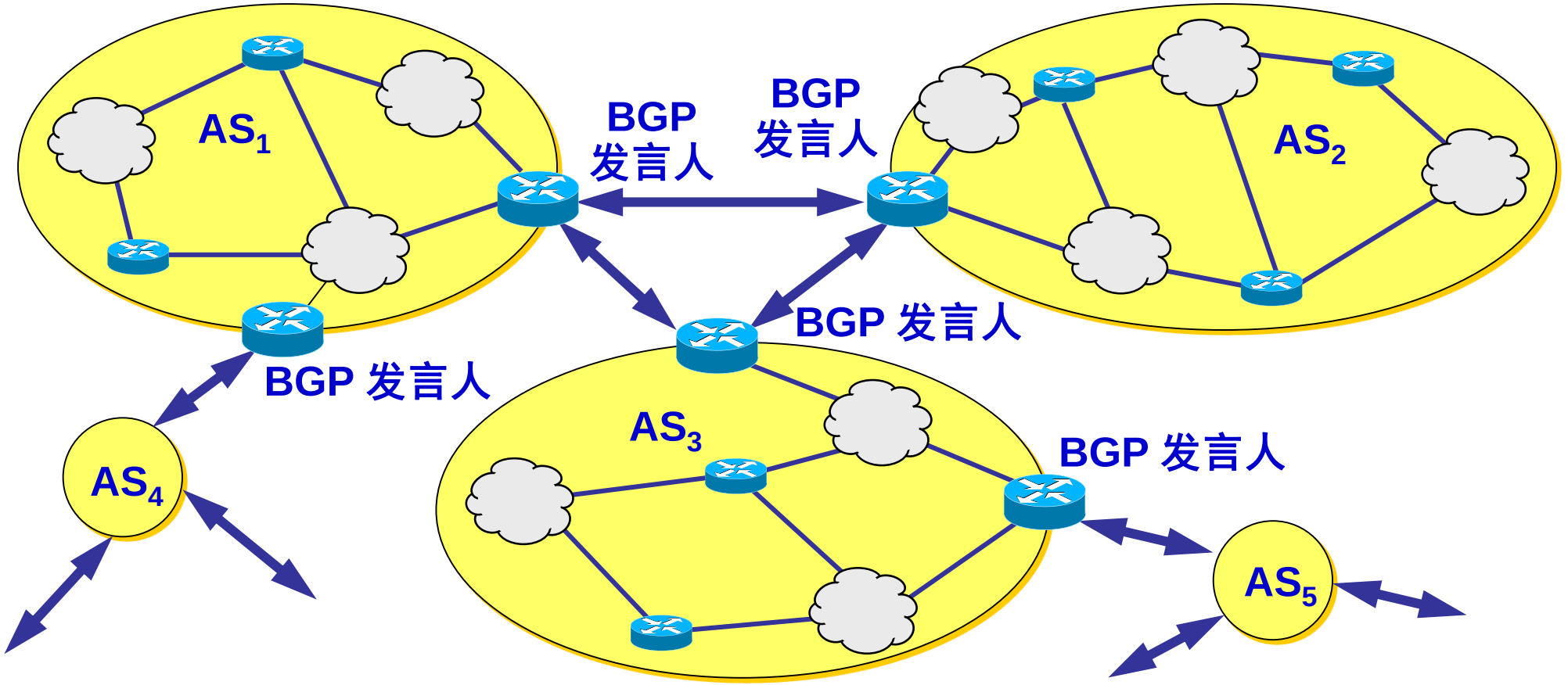
BGP 协议

- 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。
- BGP 采用了路径向量路由选择协议。

BGP 协议

- (1) 和谁交换信息？
  - 与其他AS的邻站BGP发言人交换信息。
    - 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” (BGP speaker) 。
    - 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。
- (2) 交换什么信息？
- (3)在什么时候交换信息？

BGP 发言人和自治系统 AS 的关系



BGP 协议

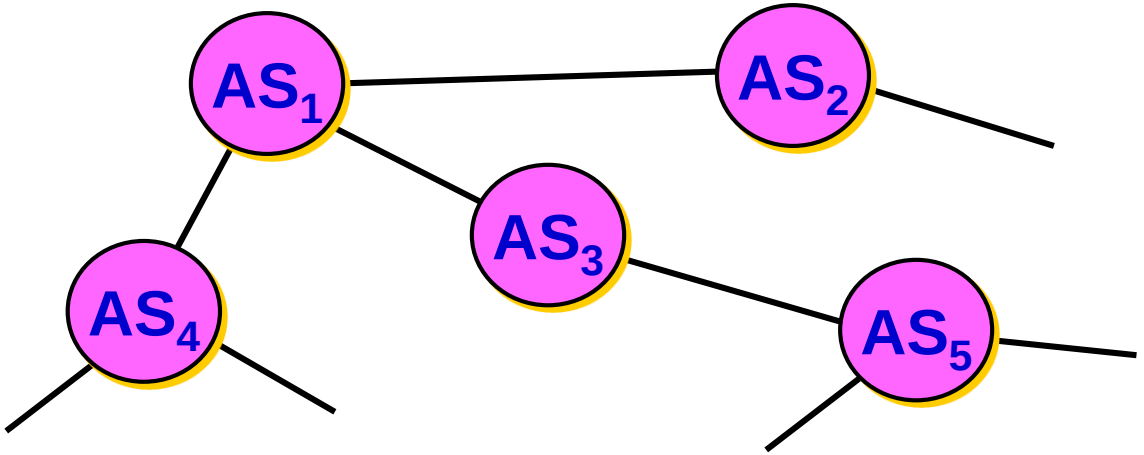
- (1) 和谁交换信息？
  - 与其他AS的邻站BGP发言人交换信息。
- (2) 交换什么信息？
  - 交换的信息是网络可达性的信息，即到达某个网络所要经过的一系列AS。
- (3)在什么时候交换信息？
  - 发生变化时更新有变化的部分。

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(端口号为179)，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，如增加新路由，撤销过时路由，报告差错等。
- 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(neighbor)或对等站(peer) 。
- BGP 发言人除了运行BGP协议外，还必须运行该自治系统所使用的内部网关协议。

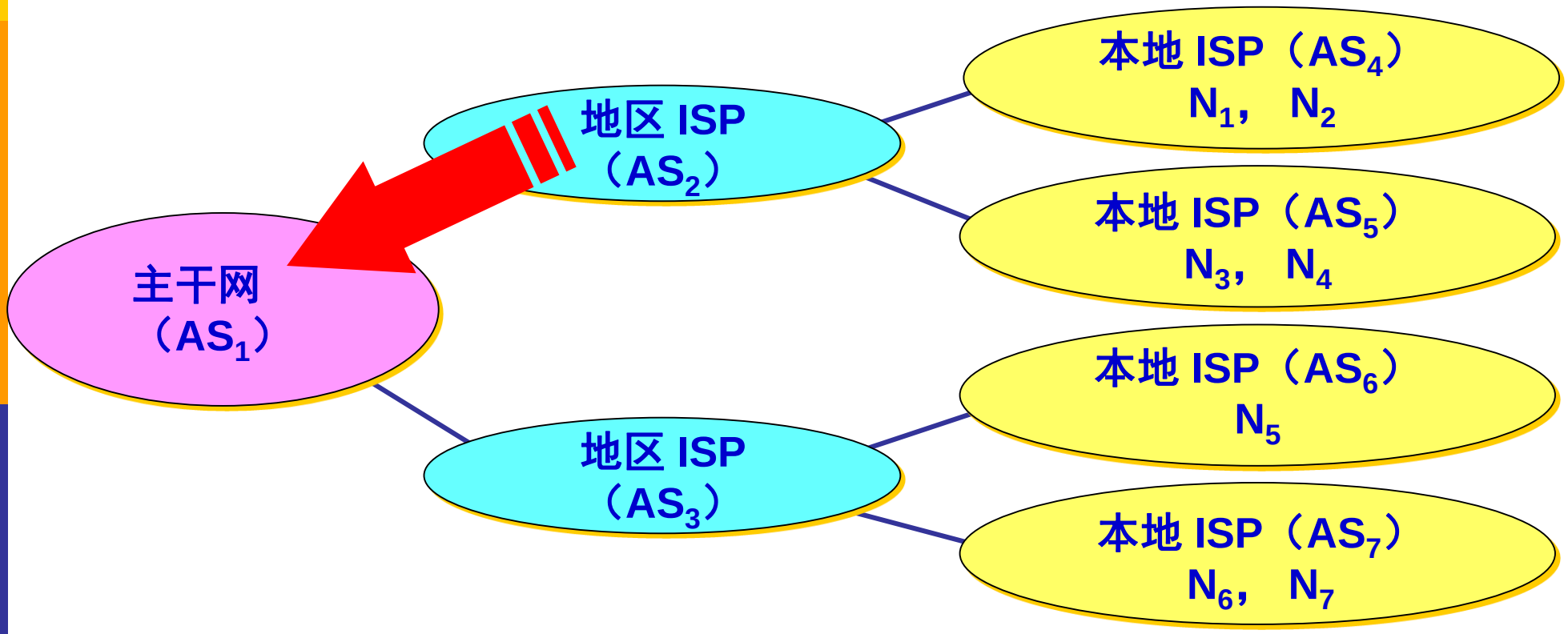
AS 的连通图举例

- BGP 所交换的网络可达性的信息就是要到达某个网络（用网络前缀表示）所要经过的一系列自治系统。
- 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。
- 例：AS_1的一个BGP发言人构造的自治系统连通图，是树形结构，不存在回路



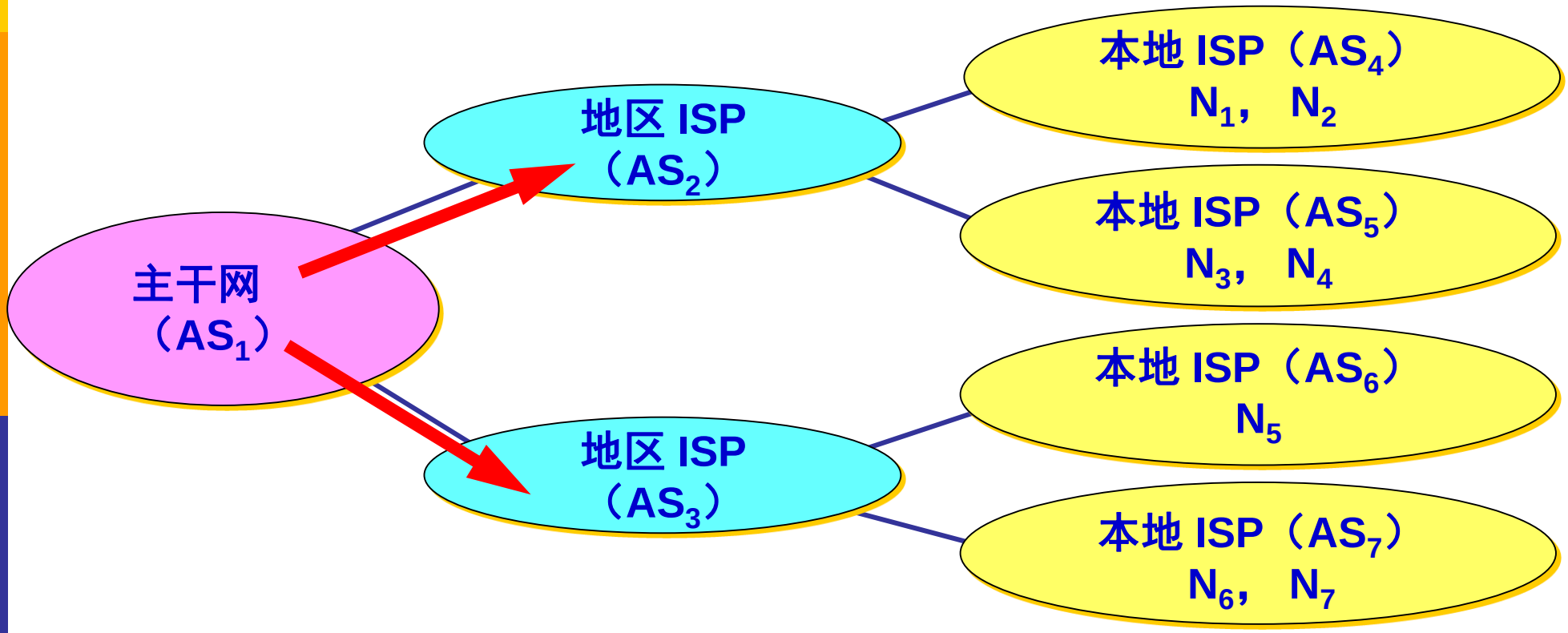
BGP 发言人交换路径向量

自治系统 AS_2 的 BGP 发言人通知主干网 AS_1 的 BGP 发言人：
“要到达网络 N_1、N_2、N_3 和 N_4 可经过 AS_2。”

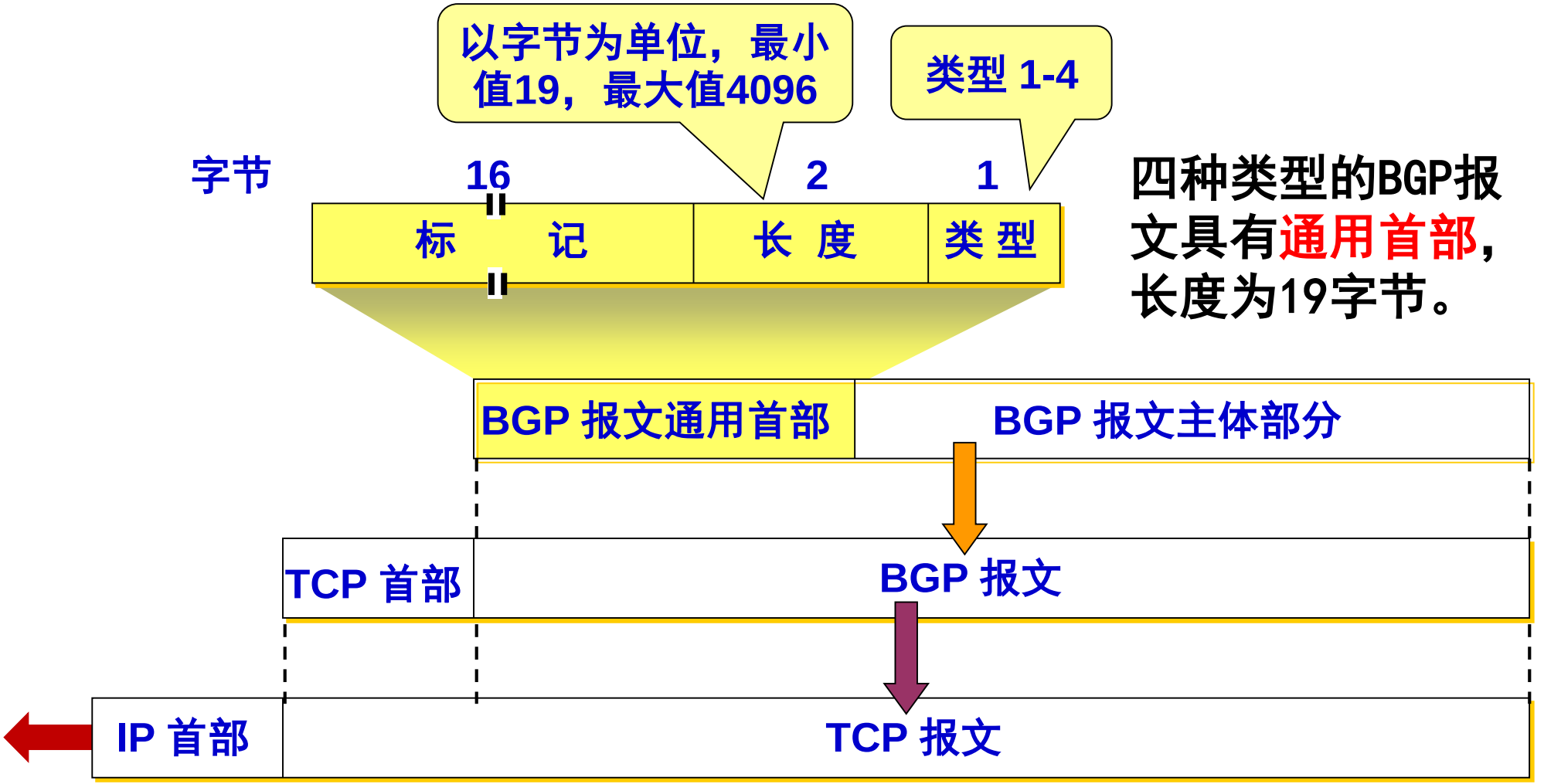


BGP 发言人交换路径向量

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）。”



BGP 的报文格式



四种类型的BGP报文具有通用首部，长度为19字节。

BGP 协议的特点

- BGP 协议交换路由信息的结点数量级是自治系统个数的量级，这要比这些自治系统中的网络数少很多。
- 每一个自治系统中 BGP 发言人（或边界路由器）的数目是很少的。这样就使得自治系统之间的路由选择不致过分复杂。
- 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，以及到达该目的网络所要经过的各个自治系统序列。
- 由于使用了路径向量的信息，就可以避免兜圈子的路由。
- 在 BGP 刚运行时，BGP 的邻站是交换整个 BGP 路由表。但以后只需要在发生变化时更新有变化的部分。这样做对节省网络带宽和减少路由器的处理开销都有好处。

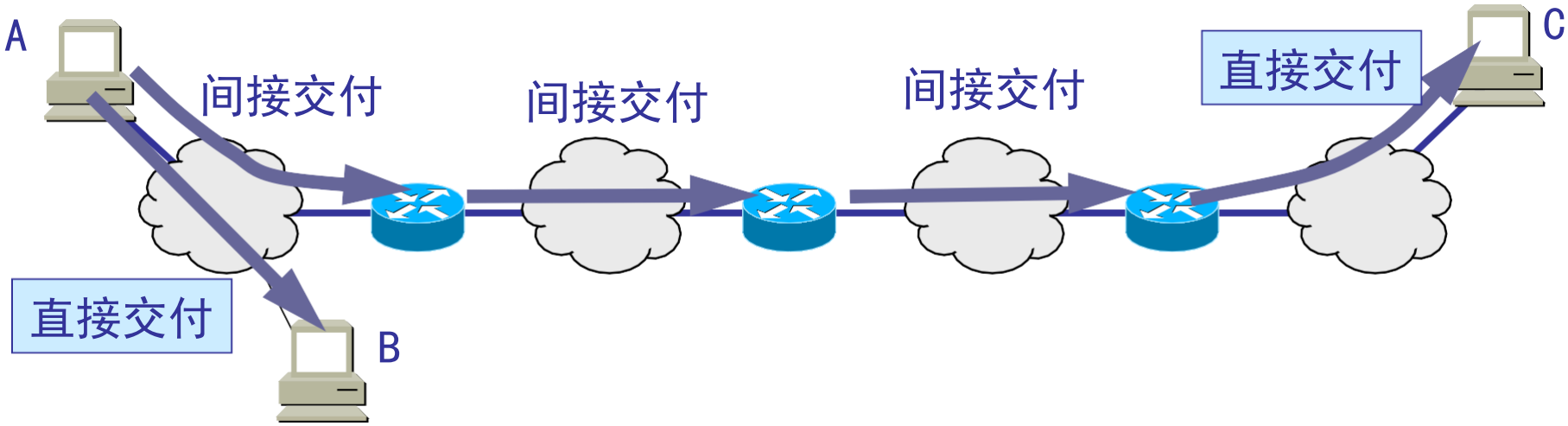
BGP-4 共使用四种报文

- (1) 打开 (OPEN) 报文，用来与相邻的另一个BGP发言人建立关系，使通信初始化。
- (2) 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。是BGP协议的核心内容。
- (3) 保活 (KEEPALIVE) 报文，用来周期性地证实邻站的连通性。
- (4) 通知 (NOTIFICATION) 报文，用来发送检测到的差错。

路由器在网际互连中作用

- 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。

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

路由器的构成

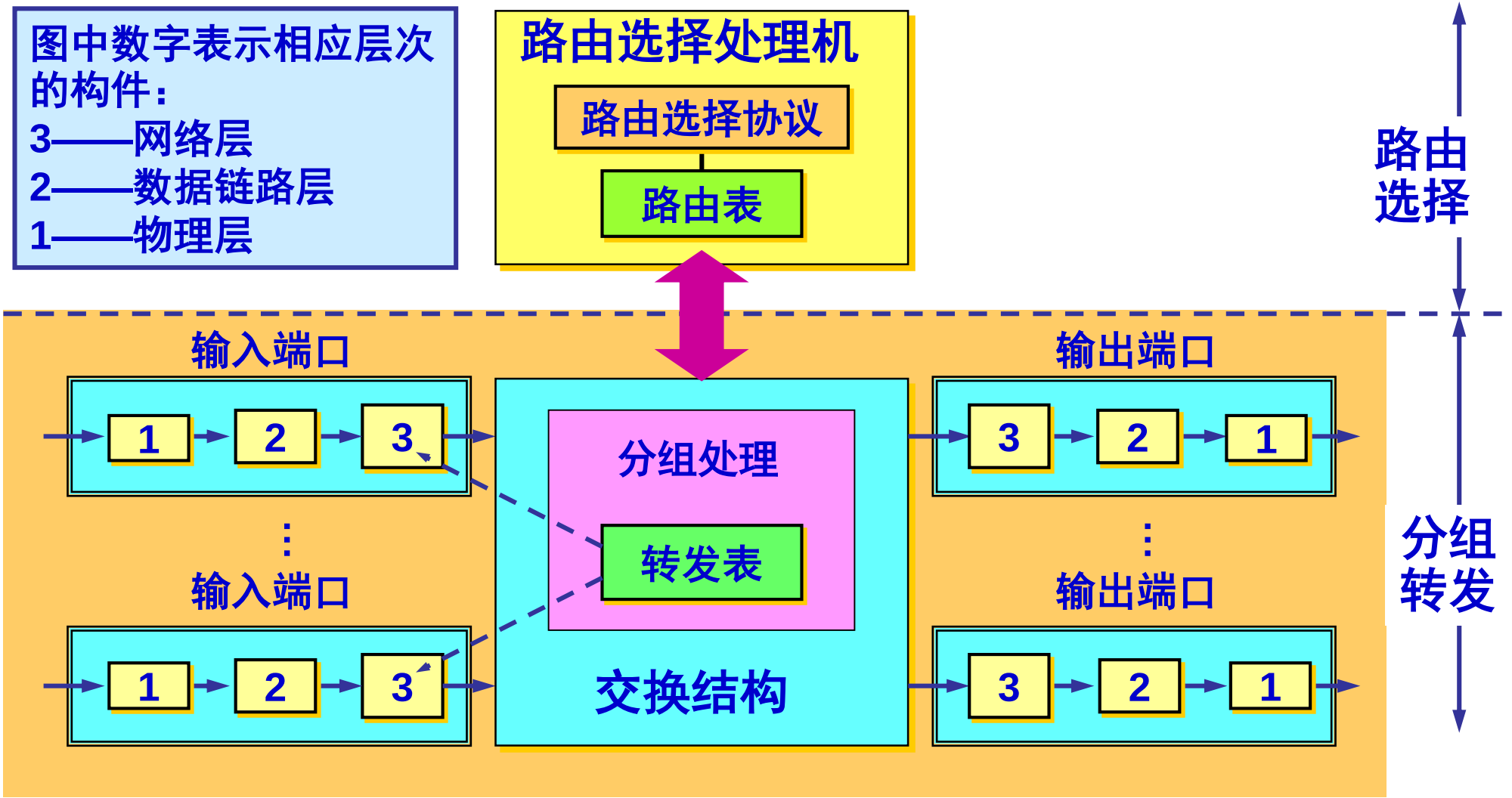
- 路由器是一种典型的网络层设备。
- 路由器是互联网中的关键设备。
- 路由器的主要作用是：
  - 连通不同的网络。
  - 选择信息传送的线路。选择通畅快捷的近路，能大大提高通信速度，减轻网络系统通信负荷，节约网络系统资源，提高网络系统畅通率，从而让网络系统发挥出更大的效益来。

1. 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。
- 路由器的转发分组正是网络层的主要工作。

典型的路由器的结构

图中数字表示相应层次的构件：
3——网络层
2——数据链路层
1——物理层



典型的路由器的结构

- 整个的路由器结构可划分为两大部分：
  - 路由选择部分
  - 分组转发部分
- 路由选择部分
  - 也叫做控制部分，其核心构件是路由选择处理机。
  - 路由选择处理机的任务是根据所选定的路由选择协议构造出路由表，同时经常或定期地和相邻路由器交换路由信息而不断地更新和维护路由表。

典型的路由器的结构

- **分组转发部分**由三部分组成：
  - **交换结构** (switching fabric)：又称为交换组织，其作用是根据**转发表** (forwarding table) 对分组进行处理。
  - **一组输入端口**
  - **一组输出端口**

（请注意：这里的端口就是硬件接口）

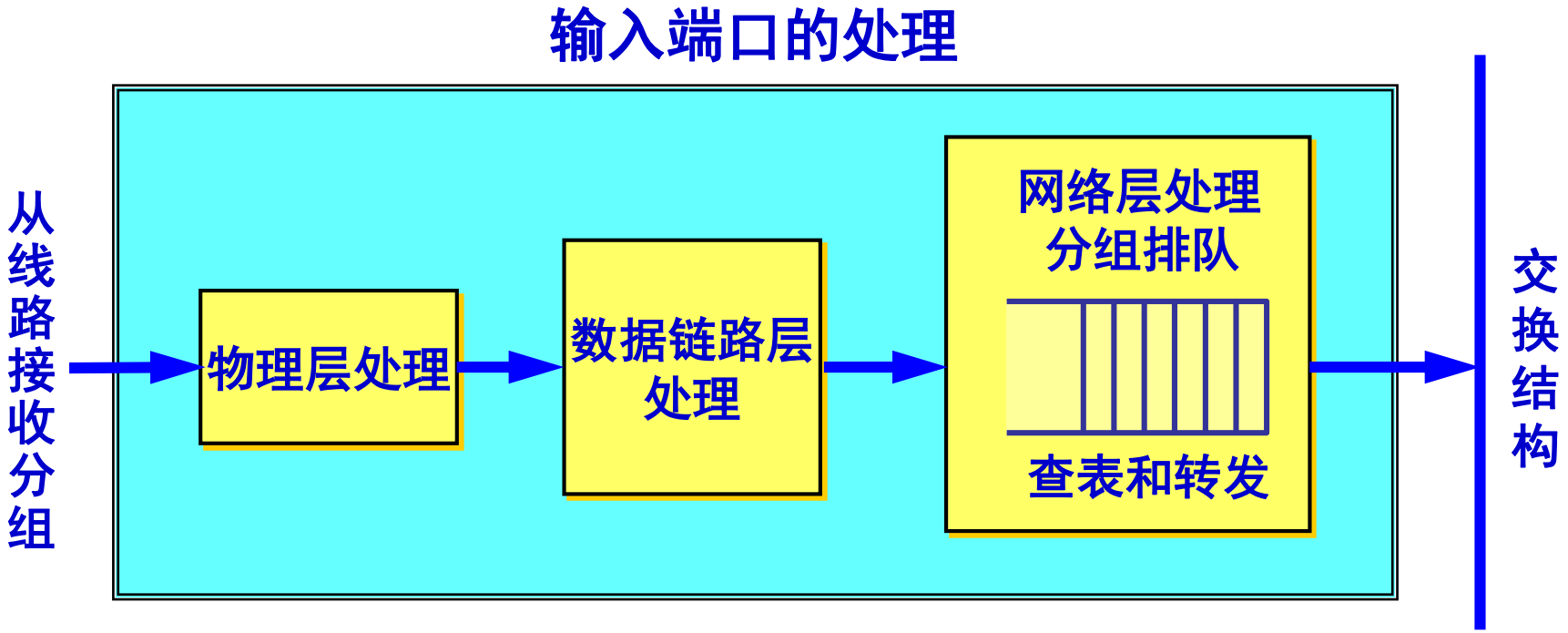
“转发”和“路由选择”的区别

- “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。

输入端口对线路上收到的分组的处理

- 路由器的输入端口里面装有物理层、数据链路层和网络层的处理模块。
- 数据链路层剥去帧首部和尾部后，将分组送到网络层的队列中排队等待处理。这会产生一定的时延。
- 输入端口中的查找和转发功能在路由器的交换功能中是最重要的。

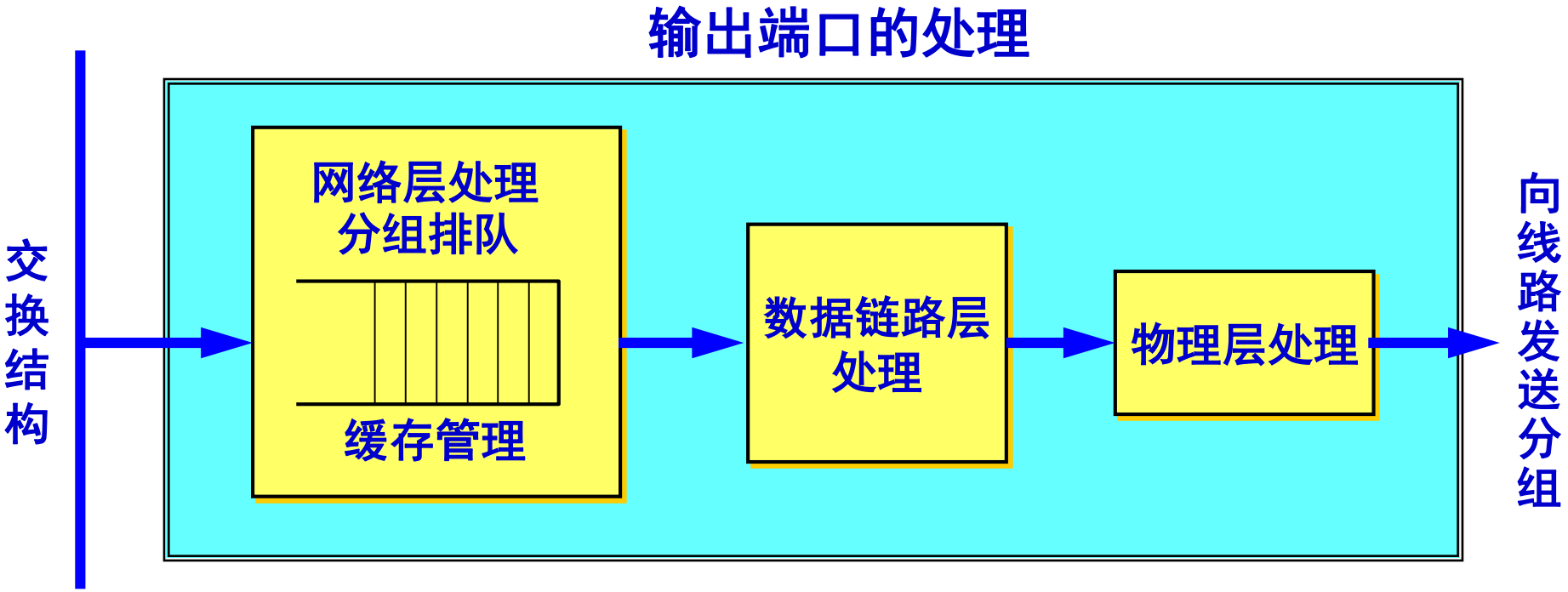
输入端口对线路上收到的分组的处理



输出端口将交换结构传送来的分组发送到线路

- 输出端口里面装有物理层、数据链路层和网络层的处理 模块。
- 输出端口从交换结构接收分组，然后把它们发送到路由器外面的线路上。
- 在网络层的处理模块中设有一个缓冲区（队列）。当交换结构传送过来的分组的速率超过输出链路的发送速率 时，来不及发送的分组就必须暂时存放在这个队列中。
- 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。

输出端口将交换结构传送来的分组发送到线路



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。

2. 交换结构

- 交换结构是路由器的关键构件。
- 正是这个交换结构把分组从一个输入端口转移到某个合适的输出端口。
- 实现交换有多种方法。常用交换方法有三种：
  - 通过存储器
  - 通过总线
  - 通过纵横交换结构

2. 交换结构

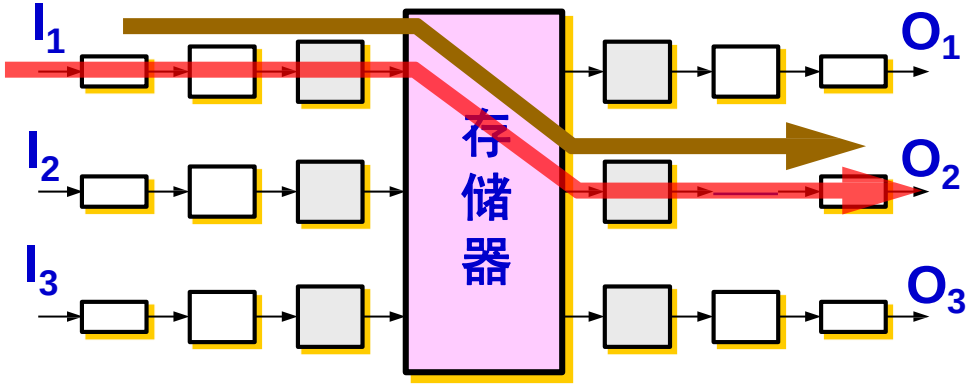
- **通过存储器**
  - 最早使用的路由器就是利用普通的计算机，用计算机的CPU作为路由器的路由选择处理机，路由器的输入和输出端口和操作系统中的I/O设备一样。
  - 当路由器的某个输入端口收到一个分组时，就用中 断方式通知路由选择处理机。然后分组就从输入端 口复制到存储器中。
  - 路由器处理机从分组首部提取目的地址，查找路由 表，再将分组复制到合适的输出端口的缓存中。
  - 若存储器的带宽（读或写）为每秒 *M* 个分组，那么 路由器的交换速率（即分组从输入端口传送到输出 端口的速率）一定小于 *M/2*。

2. 交换结构

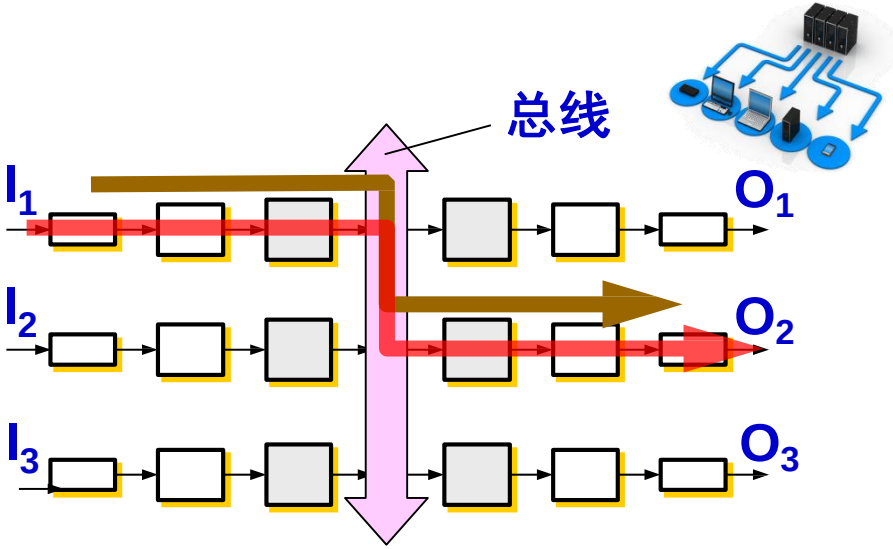
- **通过总线**
  - 数据报从输入端口通过**共享的总线**直接传送到合适 的输出端口，而**不需要路由选择处理机的干预**。
  - 因为每一个要转发的分组都要通过这一条总线，因 此路由器的转发带宽就受总线速率的限制。
  - 现代的技术已经可以将总线的带宽提高到每秒吉比 特的速率，因此许多的路由器产品都采用这种通过 总线的交换方式。

2. 交换结构

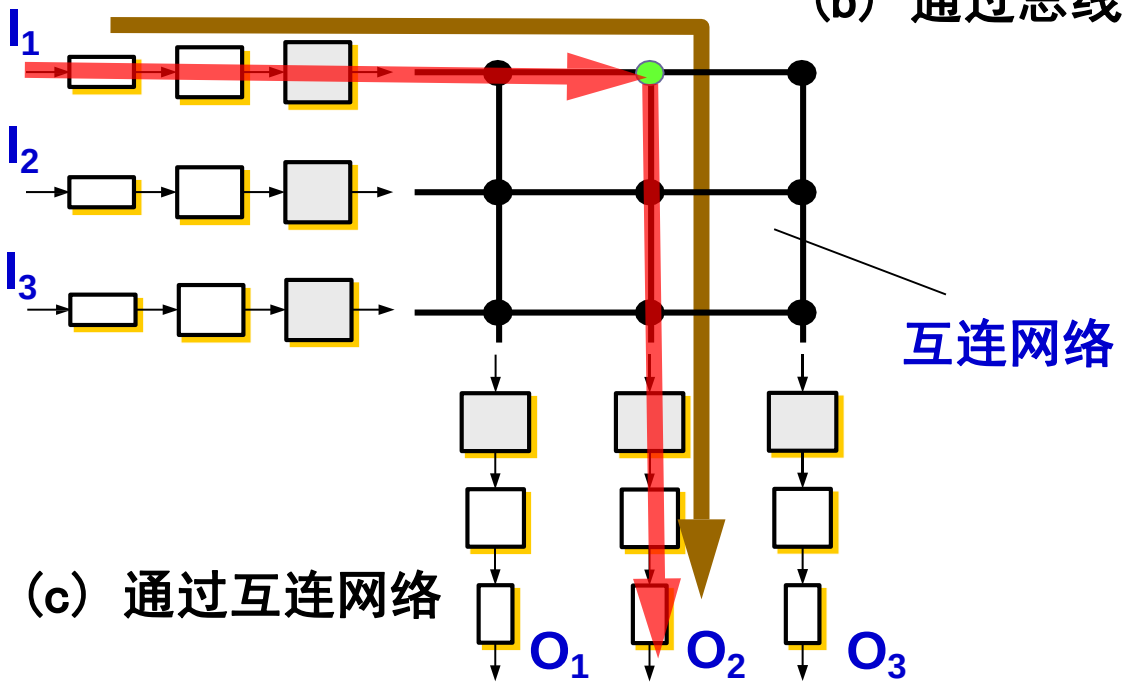
- **通过纵横交换结构 (crossbar switch fabric)**
  - 这种交换结构常称为互连网络 (interconnection network)
  - 它有 *2N* 条总线，可以使 *N* 个输入端口和 *N* 个输出端口相连接。
  - 当输入端口收到一个分组时，就将它发送到与该输入端口相连的水平总线上。
  - 若通向所要转发的输出端口的垂直总线是空闲的，则在这个结点将垂直总线与水平总线接通，然后将该分组转发到这个输出端口。
  - 但若该垂直总线已被占用（有另一个分组正在转发到同一个输出端口），则后到达的分组就被阻塞，必须在输入端口排队。



（a）通过存储器



（b）通过总线



（c）通过互连网络

三种常用的交换方法

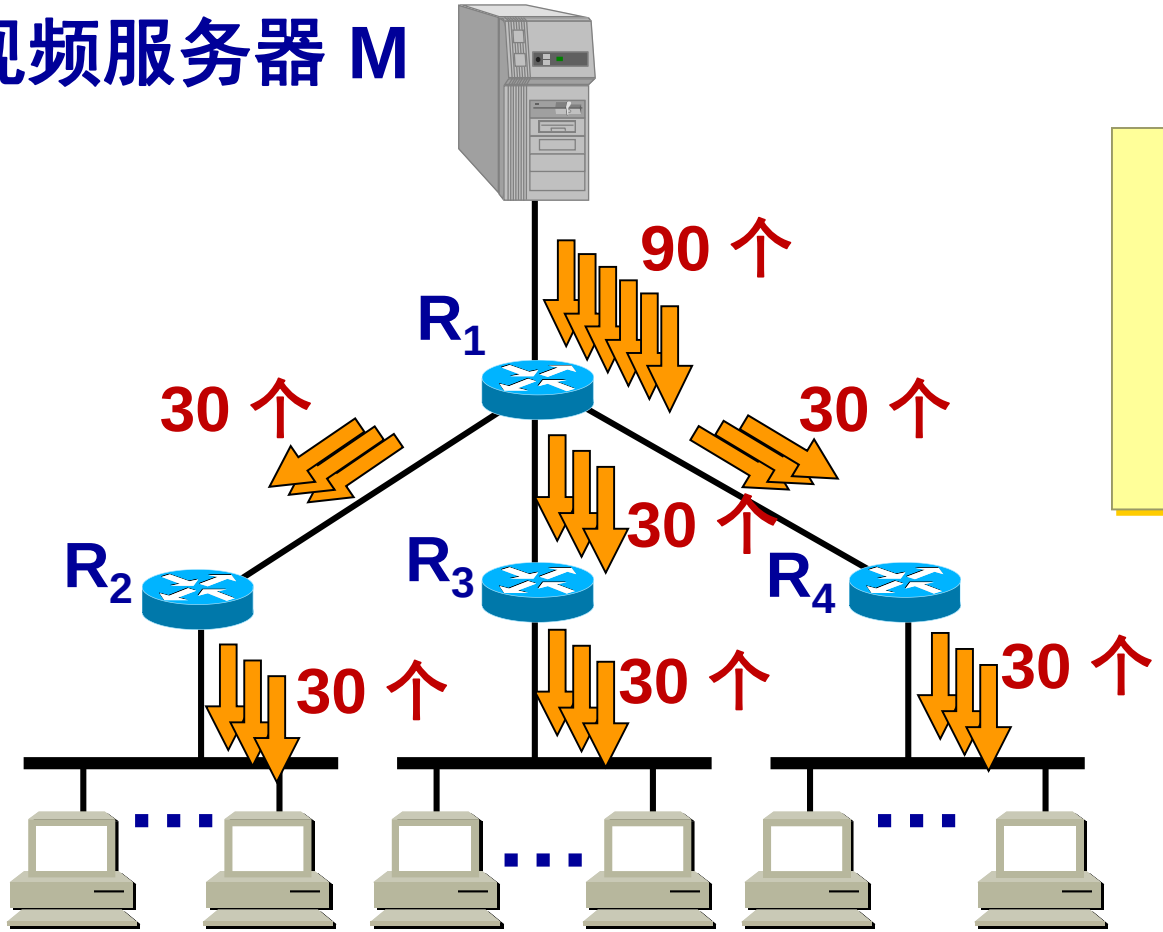
4.7 IP 多播

- 4.7.1 IP 多播的基本概念
- 4.7.2 在局域网上进行硬件多播
- 4.7.3 网际组管理协议 IGMP 和多播路由选择协议

4.7.1 IP 多播的基本概念

- 多播 (multicast，以前曾译为组播) 已成为互联网的一个热门课题。
- 目的：更好第支持一对多通信。
- 一对多通信：一个源点发送到许多个终点。
  - 例如，实时信息的交付（如新闻、股市行情等），软件更新，交互式会议及其他多媒体通信。

多播可大大节约网络资源

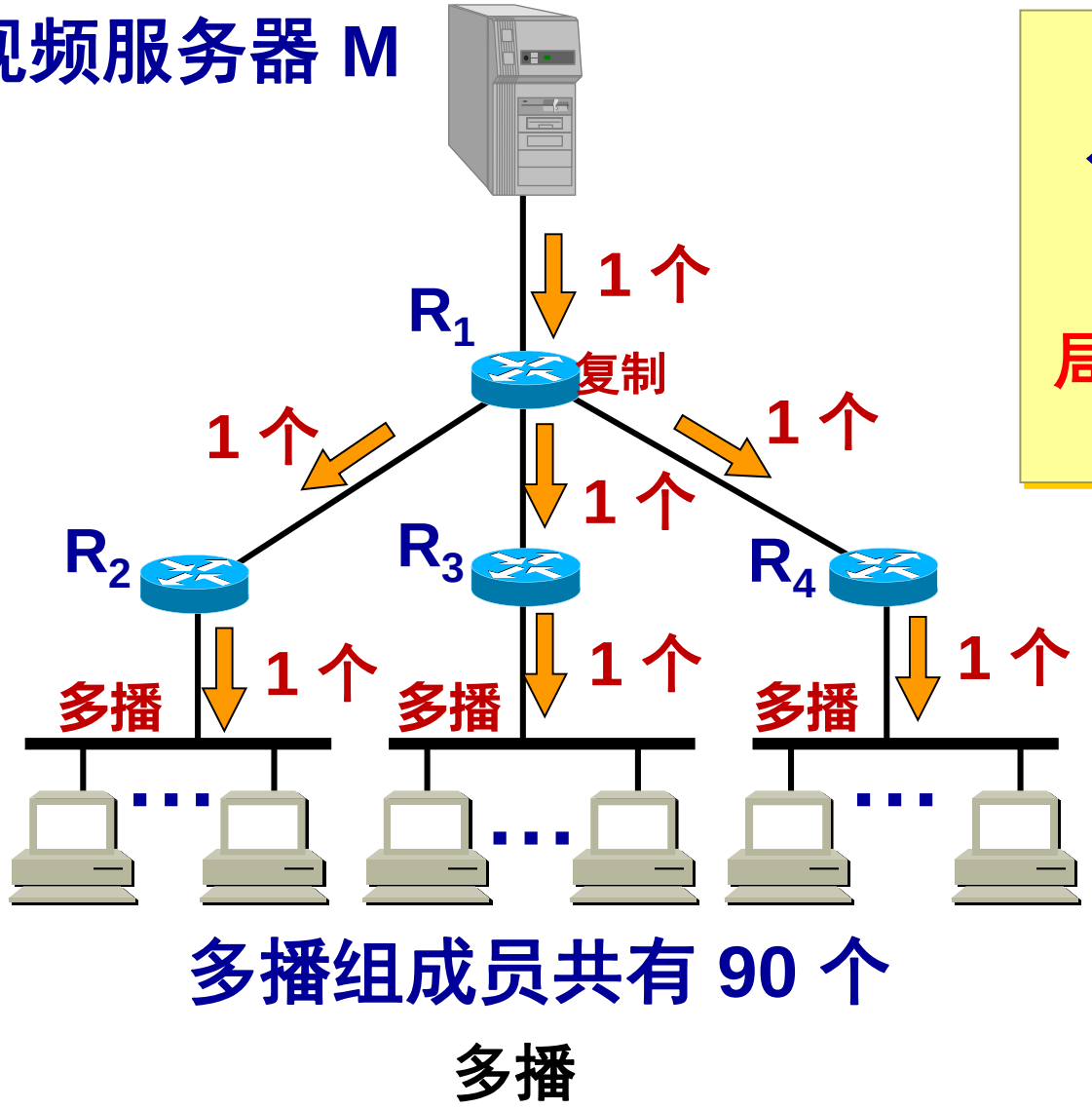


采用单播方式，
向 90 台主机传送
同样的视频节目
需要发送 90 个单播

共有 90 个主机接收视频节目

单播

多播可大大节约网络资源



采用多播方式，
只需发送一次到多播组。
路由器复制分组。

局域网具有硬件多播功能，
不需要复制分组。

当多播组的主机数很大时（如成千上万个），采用多播方式就可明显地减轻网络中各种资源的消耗。

IP 多播

- 互联网范围的多播要靠路由器来实现，这些路由器必须增加一些能够识别多播数据报的软件。
- 能够运行多播协议的路由器称为多播路由器(multicast router)。当然它也可以转发普通的单播IP数据报。
- 在互联网上进行多播就叫做 IP 多播。
- IP 多播 分为两种
  - 一种是只在本局域网上进行硬件多播
  - 另一种是在互联网的范围内进行多播

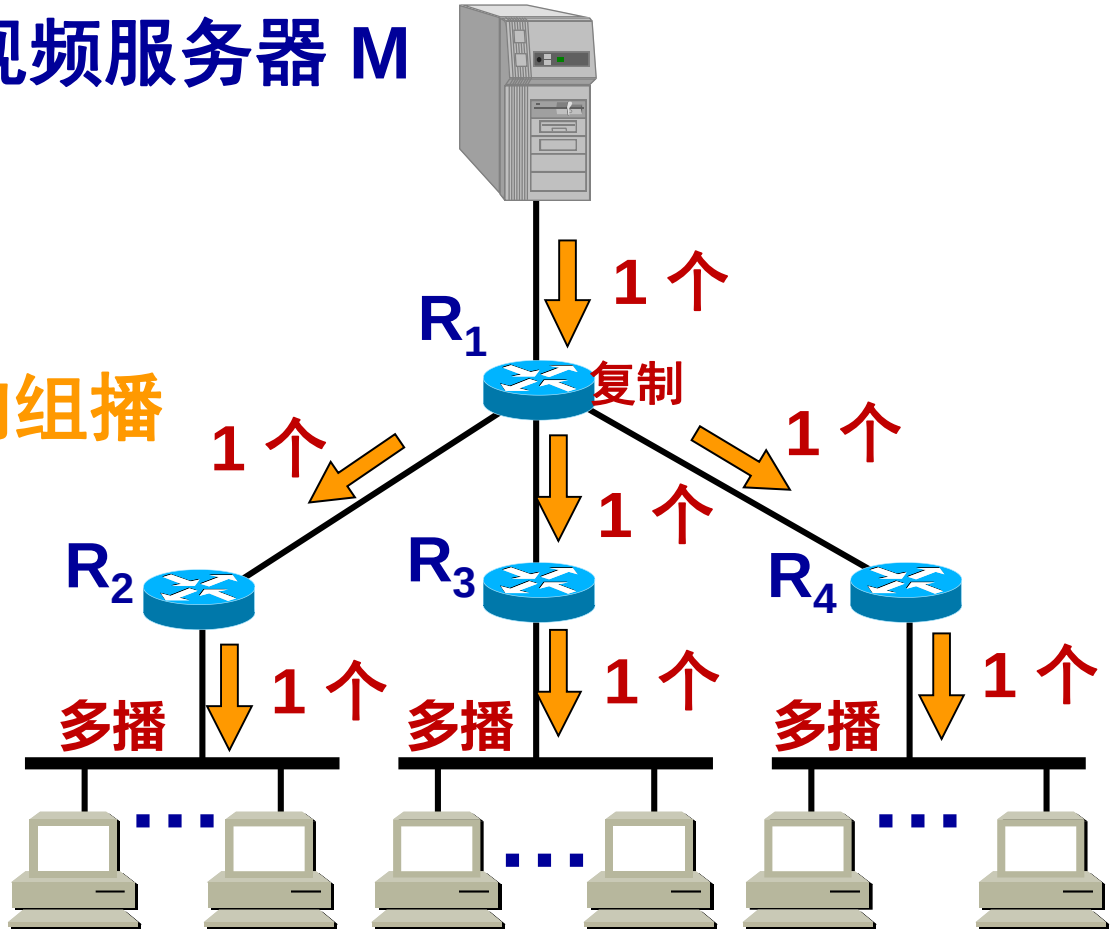
多播 IP 地址

- IP 多播所传送的分组需要使用多播 IP 地址。
- 在多播数据报的目的地址写入多播组的标识符。
- 多播组的标识符就是 IP 地址中的 D 类地址（多播地址）。
- D类 IP 地址的前四位是1110，因此范围是 224.0.0.0 到 239.255.255.255。
- 每一个 D 类地址标志一个多播组。
- 多播地址只能用于目的地址，不能用于源地址。

多播数据报

- 多播数据报和一般的 IP 数据报的区别就是它使用 D 类 IP 地址作为目的地址，并且首部中的协议字段值是 2，表明使用网际组管理协议 IGMP。
- 多播数据报也是“尽最大努力交付”，不保证一定能够交付多播组内的所有成员。
- 对多播数据报不产生 ICMP 差错报文。因此，若在 PING 命令后面键入多播地址，将永远不会收到响应。

IP 多播



多播组成员共有 90 个

多播

4.7.2 在局域网上进行硬件多播

- 互联网号码指派管理局 IANA 拥有的以太网地址块的高 24 位为 00-00-5E，因此 TCP/IP 协议使用的以太网地址块的范围是

  从 00-00-5E-00-00-00

  到 00-00-5E-FF-FF-FF

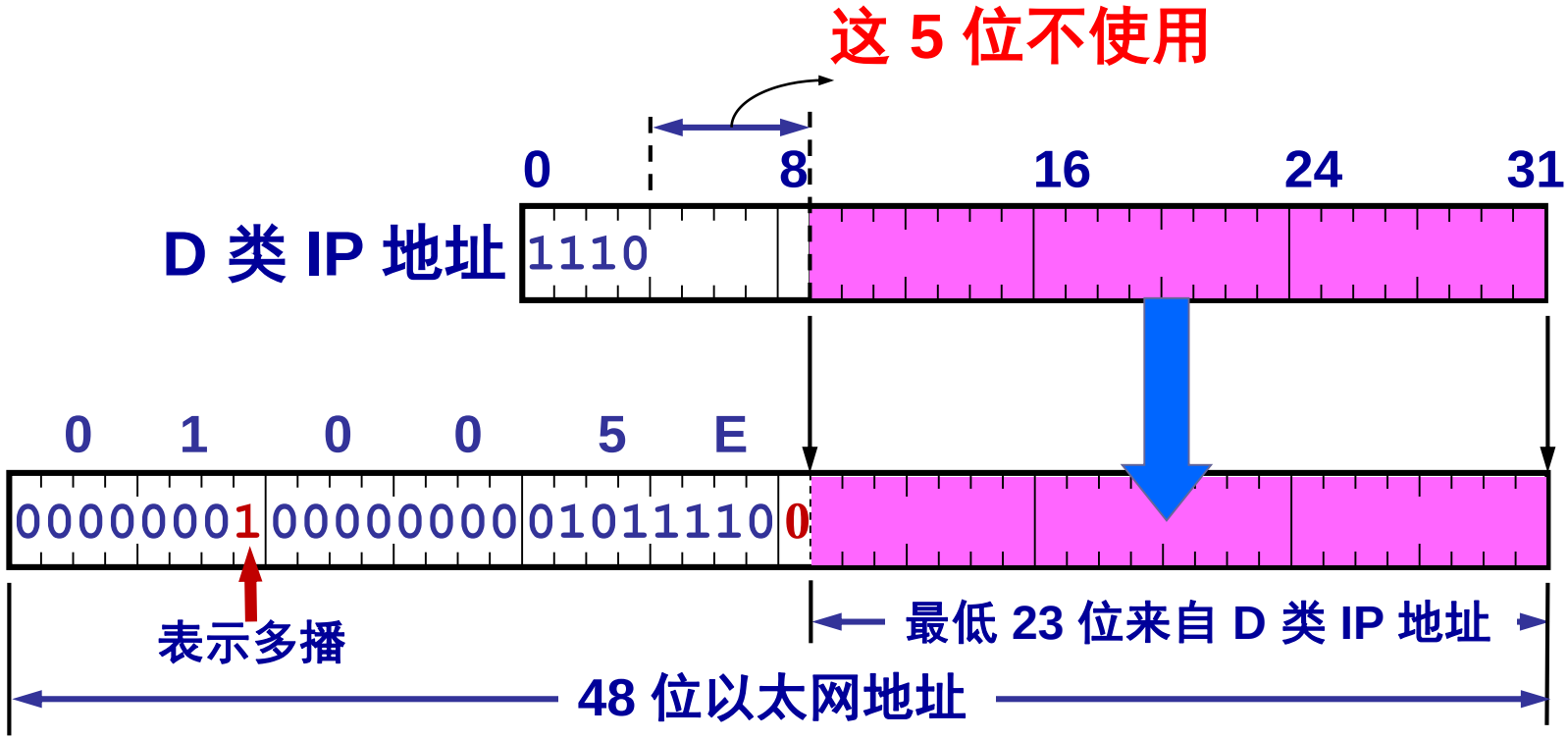
- 以太网硬件地址字段中的第1字节的最低位(I/G位)为1时为多播地址。
- IANA拥有的以太网多播地址的范围是

  从 01-00-5E-00-00-00

  到 01-00-5E-7F-FF-FF

- 只有23位可用作多播，和D类地址中的23位有一一对应的关系
- 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。

D 类 IP 地址与以太网多播地址的映射关系



由于多播IP地址与以太网硬件地址的映射关系不是唯一的，因此收到多播数据报的主机，还要在IP层利用软件进行过滤，把不是本主机要接收的数据报丢弃。

例如，IP多播地址224.128.64.32(E0-80-40-20)和224.0.64.32(E0-00-40-20) 转换成MAC地址都是 01-00-5E-00-40-20。

一个练习

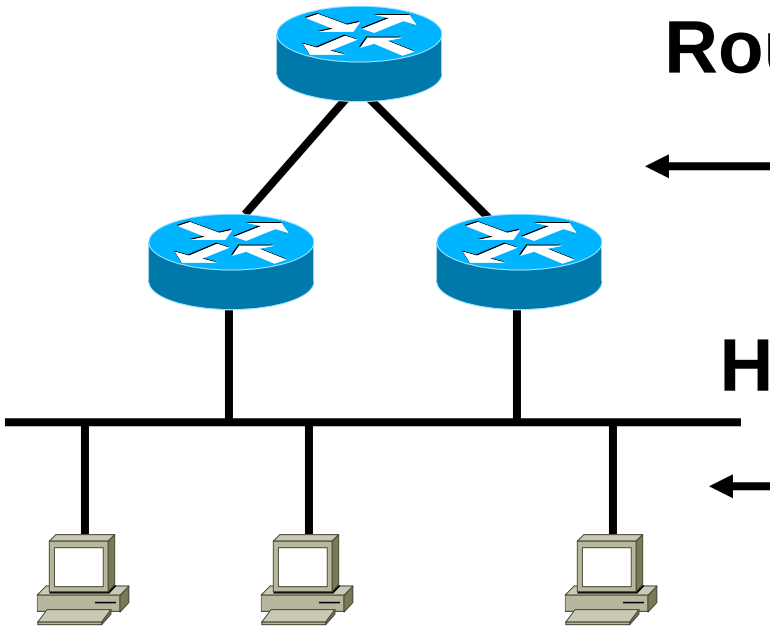
- IP多播地址236.130.97.33,其十六进制形式的以太网硬件多播地址是(_____)

4.7.3 网际组管理协议 IGMP 和多播路由选择协议

1. IP 多播需要两种协议

- 为了使路由器知道多播组成员的信息，需要利用**网际组管理协议 IGMP** (Internet Group Management Protocol)。
- 连接在局域网上的多播路由器还必须和互联网上的其他多播路由器协同工作，以便把多播数据报用最小代价传送给所有的组成员。这就需要使用**多播路由选择协议**。

多播相关协议



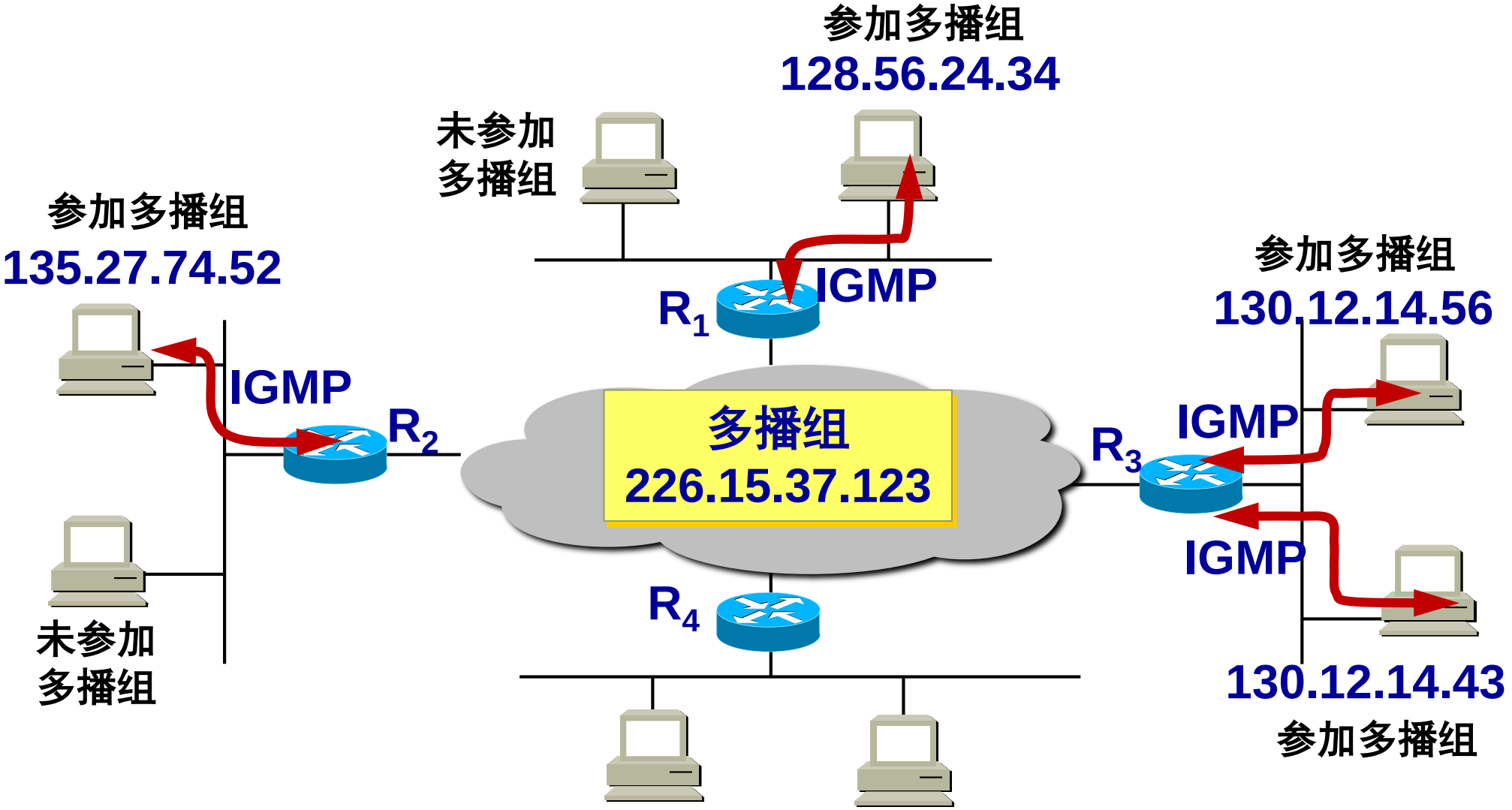
Router to Router

多播路由协议，如DVMRP，PIM
功能：生成多播路由表

Host to Router

多播管理协议，IGMP
功能：组成员的加入和退出

IGMP 使多播路由器知道多播组成员信息

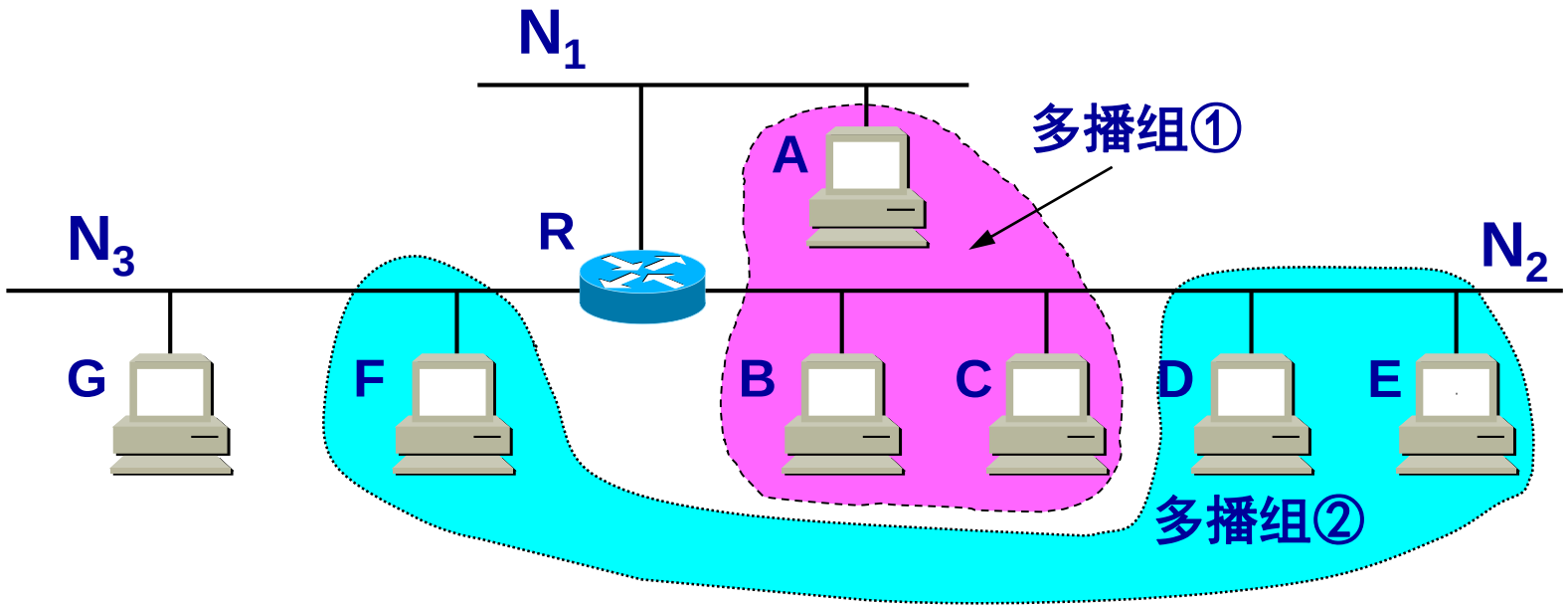


IGMP 的使用范围

- IGMP 并非在互联网范围内对所有多播组成员进行管理的协议。
- IGMP 不知道 IP 多播组包含的成员数，也不知道这些成员都分布在哪些网络上。
- IGMP 协议是让连接在本地局域网上的多播路由器知道本局域网上是否有主机（严格讲，是主机上的某个进程）参加或退出了某个多播组。

多播路由选择协议更为复杂

多播路由选择协议比单播路由选择协议复杂得多。



多播路由选择协议更为复杂

- 多播转发必须动态地适应多播组成员的变化（这时网络拓扑并未发生变化）。请注意，单播路由选择通常是在网络拓扑发生变化时才需要更新路由。
- 多播路由器在转发多播数据报时，不能仅仅根据多播数据报中的目的地址，而是还要考虑这个多播数据报从什么地方来和要到什么地方去。
- 多播数据报可以由没有加入多播组的主机发出，也可以通过没有组成员接入的网络。

2. 网际组管理协议 IGMP

- 1989 年公布的 RFC 1112（IGMPv1）早已成为了互联网的标准协议。
- 1997 年公布的 RFC 2236（IGMPv2，建议标准）对 IGMPv1 进行了更新。
- 2002 年 10 月公布了 RFC 3376（IGMPv3，建议标准），宣布 RFC 2236（IGMPv2）是陈旧的。

IGMP 是整个网际协议 IP 的一个组成部分

- 和 ICMP 相似，IGMP 使用 IP 数据报传递其报文（即 IGMP 报文加上 IP 首部构成 IP 数据报），但它也向 IP 提供服务。
- 因此，我们不把 IGMP 看成是一个单独的协议，而是属于整个网际协议 IP 的一个组成部分。

IGMP 工作可分为两个阶段

- **第一阶段：加入多播组。**
  - 当某个主机加入新的多播组时，该主机应向多播组的多播地址发送 IGMP 报文，声明自己要成为该组的成员。
  - 本地的多播路由器收到 IGMP 报文后，还要利用多播路由选择协议将组成员关系转发给互联网上的其他多播路由器。

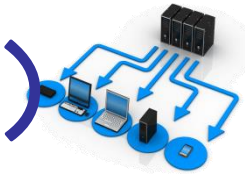
IGMP 可分为两个阶段

- **第二阶段：探询组成员变化情况。**
  - 因为组成员关系是**动态**的，因此**本地多播路由器**要**周期性地探询本地局域网上的主机，**以便知道这些主机是否还继续是组的成员。
  - **只要对某个组有一个主机响应，那么多播路由器就认为这个组是活跃的。**
  - 但一个组在经过几次的探询后仍然没有一个主机响应，则不再将该组的成员关系转发给其他的多播路由器。

IGMP 采用的一些具体措施

- 在主机和多播路由器之间的所有通信都是使用 IP 多播。
- 多播路由器在探询组成员关系时，只需要对所有的组发送一个请求信息的询问报文，而不需要对每一个组发送一个询问报文。默认的询问速率是每 125 秒发送一次。
- 当同一个网络上连接有几个多播路由器时，它们能够迅速和有效地选择其中的一个来探询主机的成员关系。

IGMP 采用的一些具体措施（续）



- 在 IGMP 的询问报文中有一个数值 *N*，它指明一个最长响应时间（默认值为 10 秒）。当收到询问时，主机在 0 到 *N* 之间随机选择发送响应所需经过的时延。对应于最小时延的响应最先发送。
- 同一个组内的每一个主机都要监听响应，只要有本组的其他主机先发送了响应，自己就可以不再发送响应了。

3. 多播路由选择

- 一个多播组中的成员是动态变化的，随时会有主机加入或离开这个多播组。
- 多播路由选择实际上就是要找出**以源主机为根结点**的**多播转发树。**
- 在多播转发树上的路由器不会收到重复的多播数据报。
- 对不同的多播组对应于不同的多播转发树。
- 同一个多播组，对不同的源点也会有不同的多播转发树。

3. 多播路由选择

- 多播路由选择协议在转发多播数据报时使用三种方法：
  - (1) 洪泛与剪除
  - (2) 隧道技术 (tunneling)
  - (3) 基于核心的发现技术

(1) 洪泛与剪除

- 这种方法适合于较小的多播组，而所有的组成员接入的局域网也是相邻接的。
- 一开始，路由器转发多播数据报使用洪泛的方法（这就是广播）。
- 为了避免兜圈子，采用了叫做**反向路径广播 RPB** (Reverse Path Broadcasting) 的策略。

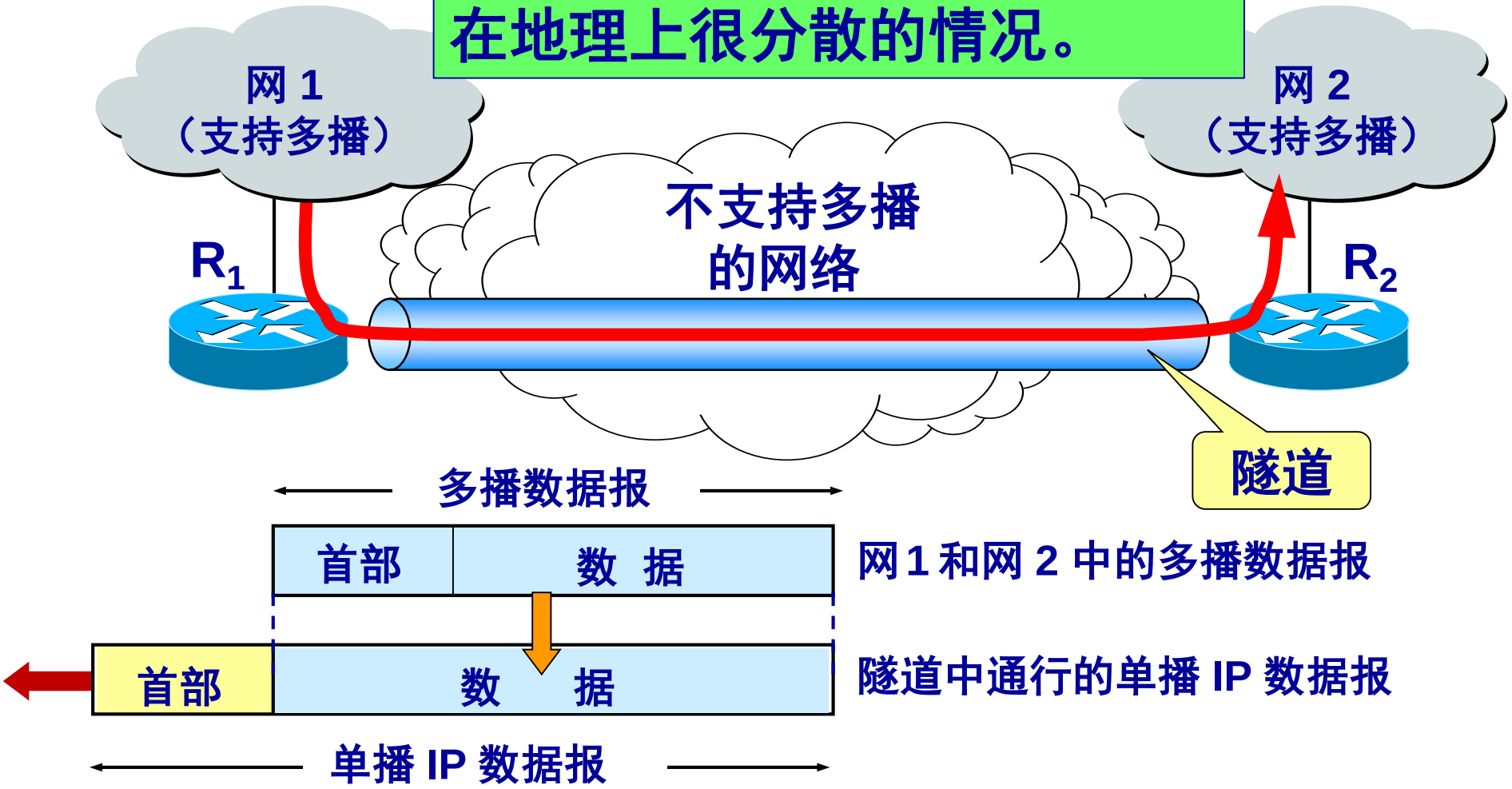
RPB 的要点

- 路由器收到多播数据报时，先检查它是否是从源点经最短路径传送来的。
- 若是，就向所有其他方向转发刚才收到的多播数据报（但进入的方向除外），否则就丢弃而不转发。
- 最后就得出了用来转发多播数据报的多播转发树，以后就按这个多播转发树转发多播数据报。避免了多播数据报的兜圈子，同时每一个路由器也不会接收重复的多播数据报。
- 如果在多播转发树上的某个路由器发现它的下游树枝（即叶节点方向）已没有该多播组的成员，就应把它和下游的树枝一起剪除。
- 当某个树枝有新增加的组成员时，再接入到多播转发树上。

反向路径广播 RPB 和剪除

(2) 隧道技术 (tunneling)

隧道技术适用于多播组的位置在地理上很分散的情况。



隧道技术在多播中的应用

(3) 基于核心的发现技术

- 这种方法对于多播组的大小在较大范围内变化时都适合。
- 这种方法是对每一个多播组 G 指定一个核心 (core) 路由器，给出它的 IP 单播地址。
- 核心路由器按照前面讲过的方法创建出对应于多播组 G 的转发树。

几种多播路由选择协议

- 目前还没有整个互联网使用的多播路由选择协议。

- 距离向量多播路由选择协议 DVMRP (Distance Vector Multicast Routing Protocol)
- 基于核心的转发树 CBT (Core Based Tree)
- 开放最短通路优先的多播扩展 MOSPF (Multicast Extensions to OSPF)
- 协议无关多播-稀疏方式 PIM-SM (Protocol Independent Multicast-Sparse Mode)
- 协议无关多播-密集方式 PIM-DM (Protocol Independent Multicast-Dense Mode)

总结

- 网络层提供的服务
- 网际协议IP
  - 分类的IP地址
  - 划分子网和构成超网
  - IP数据报的格式
  - 路由器转发分组的流程
- 路由选择协议
  - 内部网关协议 RIP、OSPF
  - 外部网关协议 BGP
- 地址解析协议ARP、网际控制报文协议ICMP、网际组管理协议IGMP