计 算 机 网 络

西北工业大学 软件学院

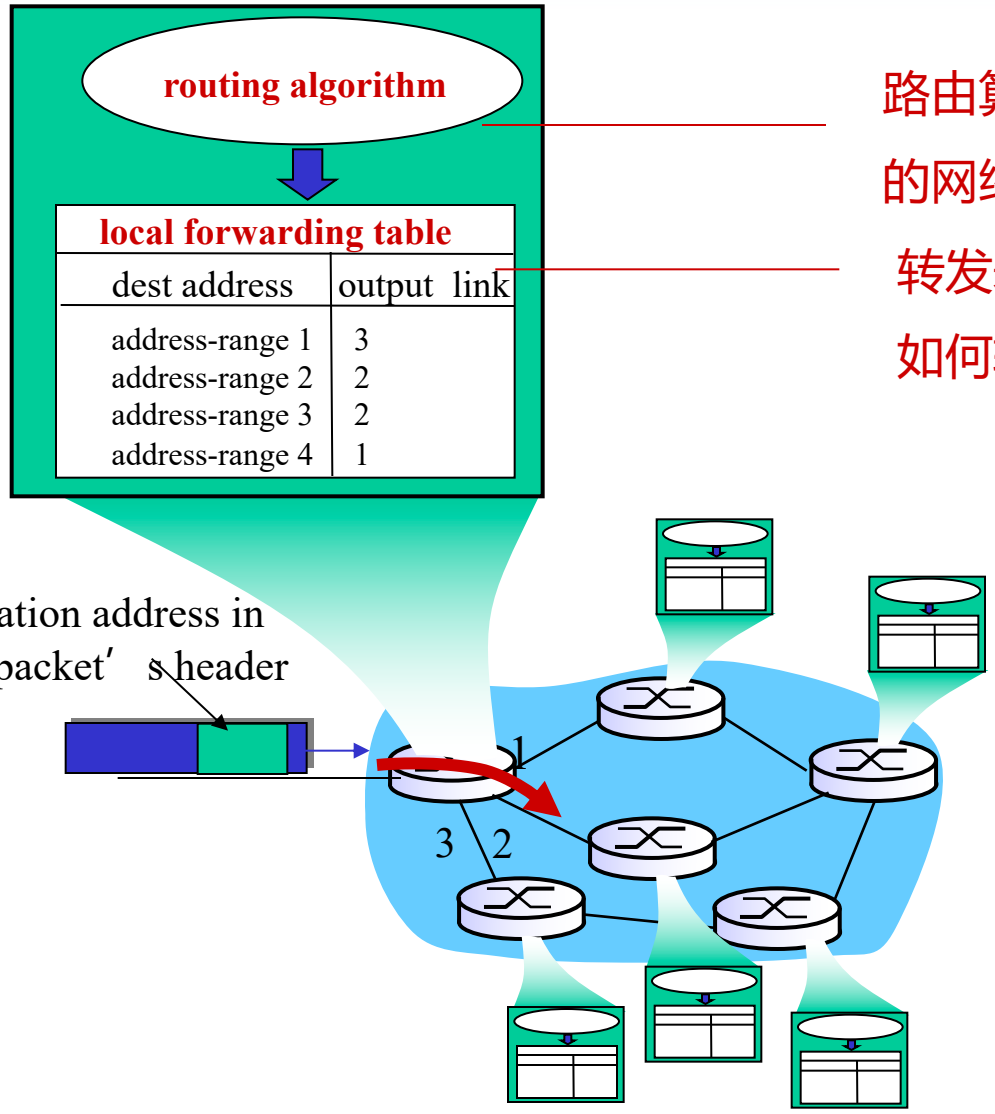
计算机网络

第5章 网络层：控制平面

路由算法

Connecting the dots---

5 概述



路由算法(协议)确定去往目的网络的最佳路径

转发表确定在本路由器如何转发分组

5 概述

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。

5 概述

- 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。

5 概述

- 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。

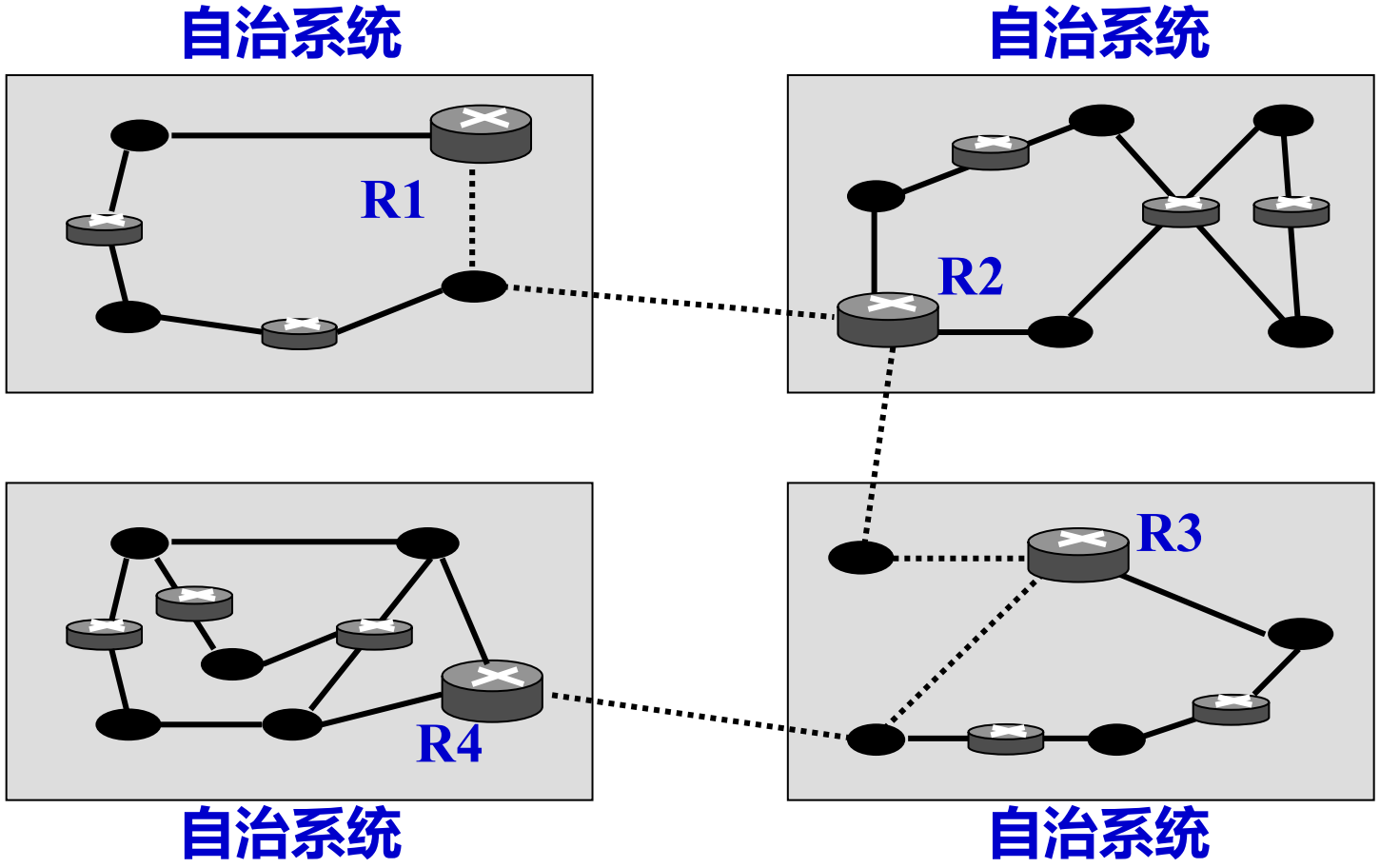
5 概述

- 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。

5 概述

- 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。
- 现在对自治系统 AS 的定义是强调下面的事实：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。

5 概述



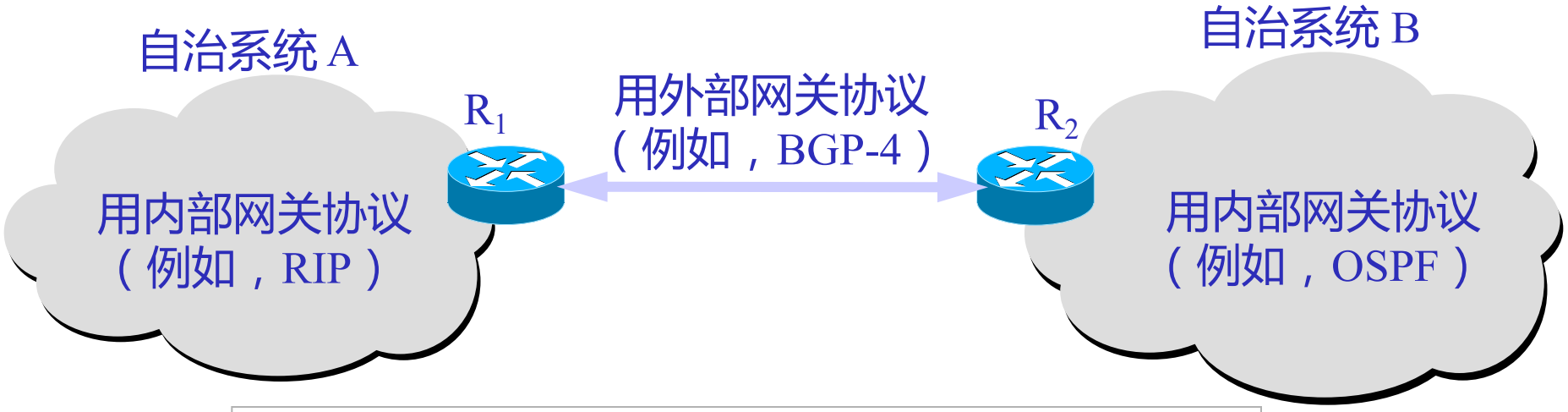
5 概述

- 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。

5 概述

- 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。

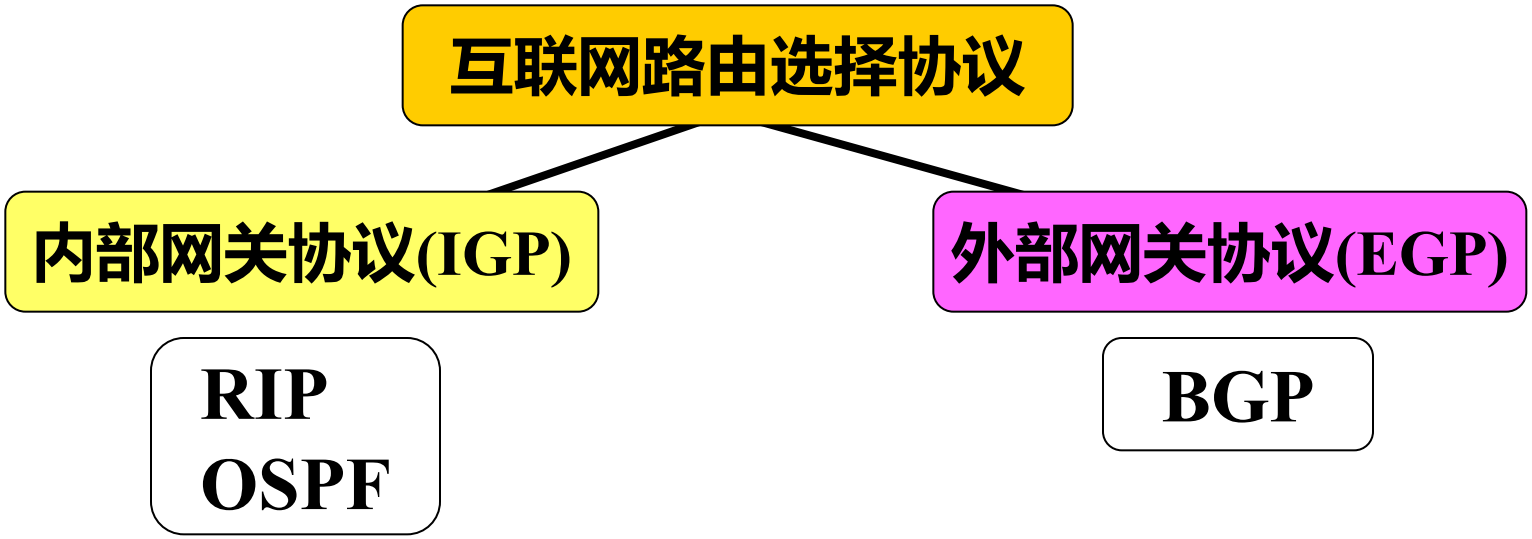
5 概述



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)

5 概述

- 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。



5.1 路由选择协议

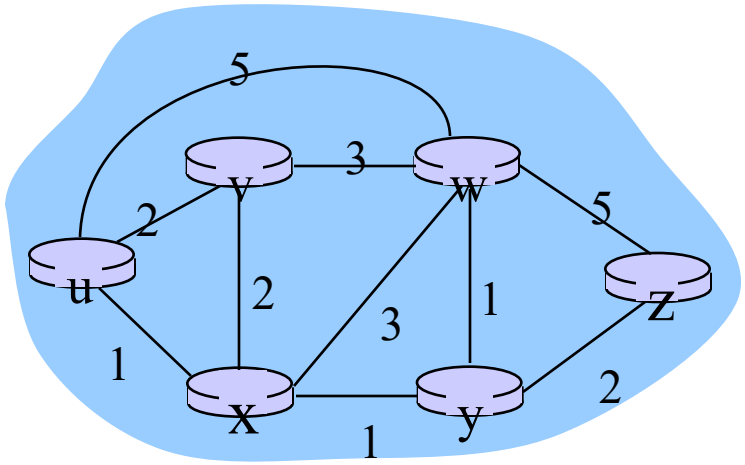


图: G = (N, E)

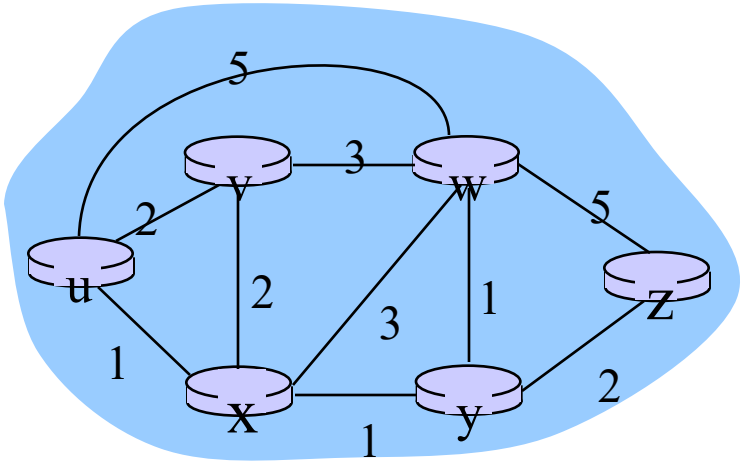
N = 路由器集合= { u, v, w, x, y, z }

E = 链路集合 ={ (u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z) }

附注: 图的抽象在网络领域应用很广泛

E.g.:P2P，其中，N是 peers集合，而E是TCP连接集合

5.1 路由选择协议



c(x, x′) = 链路(x, x')的费用

e.g., c(w, z) = 5

每段链路的费用可以总是1， 或者是带宽的倒数、拥塞程度等

路径费用：**(x1, x2, x3,…, xp) = c(x1,x2) + c(x2,x3) + … + c(xp-1,xp)**

关键问题: 源到目的（如u到z）的最小费用路径是什么？

路由算法: 寻找最小费用路径的算法

5.1 路由选择协议

静态路由 vs 动态路由？

静态路由：

- 手工配置
- 路由更新慢
- 优先级高

动态路由：

- 路由更新快
- 定期更新
- 及时响应链路费用或网络拓扑变化

全局信息 vs 分散信息？

全局信息：

- 所有路由器掌握完整的网络拓扑和链路费用信息
- **E.g.** 链路状态(LS)路由算法

分散(decentralized)信息：

- 路由器只掌握物理相连的邻居以及链路费用
- 邻居间信息交换、运算的迭代过程
- **E.g.** 距离向量(DV)路由算法

5.1.1链路状态路由算法

链路状态路由算法

- 路由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。
- 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)。
- 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。

5.1.1链路状态路由算法

- 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）。

5.1.1链路状态路由算法

Dijkstra 算法

□所有结点(路由器)掌握网络拓扑和链路费用

•通过“链路状态广播”

•所有结点拥有相同信息

□计算从一个结点(“源”)到达所有其他结点的最短路径

•获得该结点的转发表

□迭代: k次迭代后，得到到达k个目的结点的最短路径

符号:

□c(x,y): 结点x到结点y链路费用；如果x和y不直接相连，则=∞

□D(v): 从源到目的v的当前路径费用值

□p(v): 沿从源到v的当前路径，v的前序结点

□N’: 已经找到最小费用路径的结点集合

5.1.1链路状态路由算法

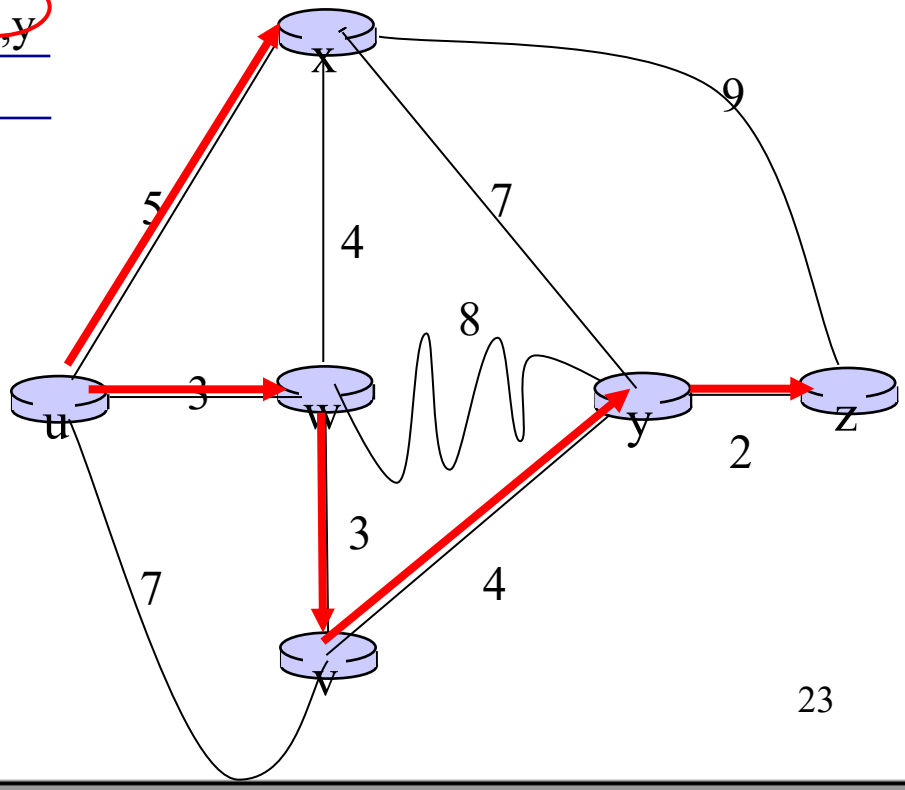
```
1  初始化:
2   N' = {u}
3   for 所有结点v
4     if v毗邻u
5        then D(v) = c(u,v)
6     else D(v) = ∞
7
8  Loop
9    找出不在 N'中的w ， 满足D(w)最小
10   将w加入N'
11   更新w的所有不在N'中的邻居v的D(v) :
12     D(v) = min( D(v), D(w) + c(w,v) )
13   /*到达v的新费用或者是原先到达v的费用，或者是
14     已知的到达w的最短路径费用加上w到v的费用 */
15 until 所有结点在N'中
```

5.1.1链路状态路由算法

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					

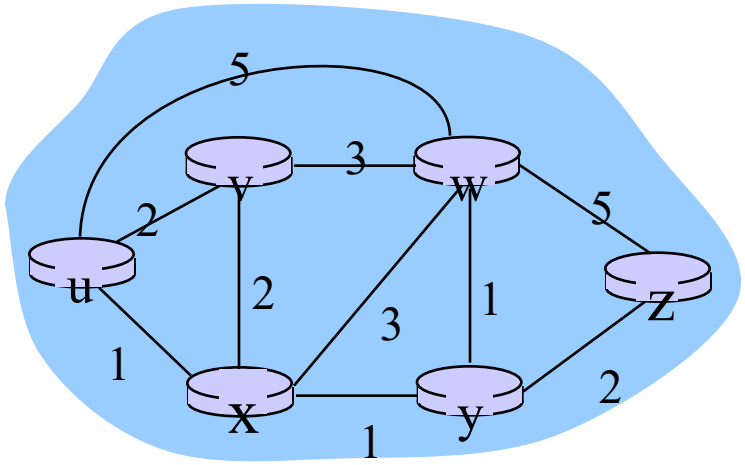
注意:

新的最短路径一定是从某条已知的最短路径衍生出来



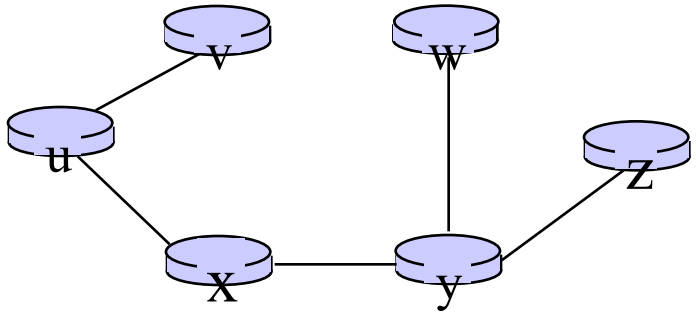
5.1.1链路状态路由算法

Step	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					



5.1.1链路状态路由算法

u的最终最短路径树:



u的最终转发表

目的	链路
v	(u,v)
x	(u,x)
y	(u,x)
w	(u,x)
z	(u,x)

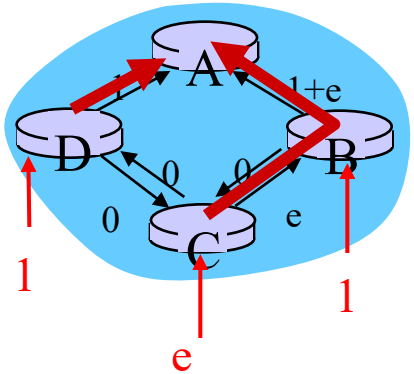
5.1.1链路状态路由算法

算法复杂性: n个结点

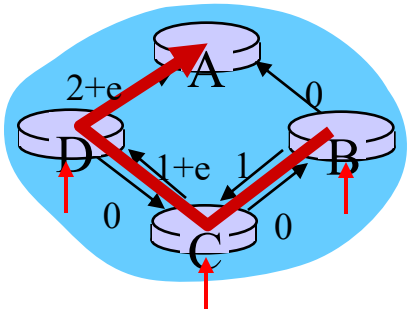
- 每次迭代: 需要检测所有不在集合N'中的结点w
- n(n+1)/2次比较: O(n2)
- 更高效的实现: O(nlogn)

存在震荡(oscillations)可能:

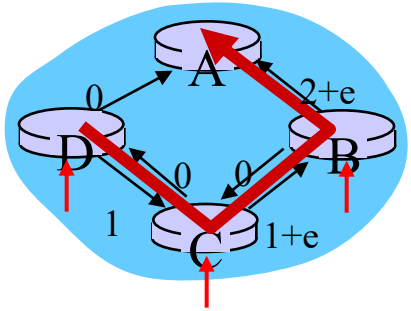
- e.g., 假设链路费用是该链路承载的通信量:



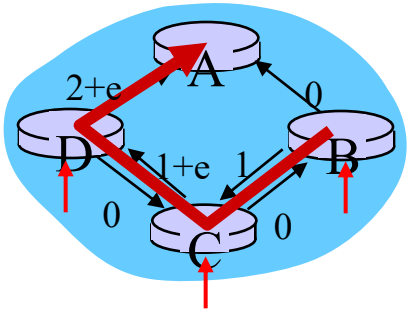
初始



基于这些费用
计算新的路由



基于这些费用
计算新的路由



基于这些费用
计算新的路由

5.1.2距离向量路由算法

Bellman-Ford方程(动态规划)

令：

dx(y):=从x到y最短路径的费用（距离）

则：

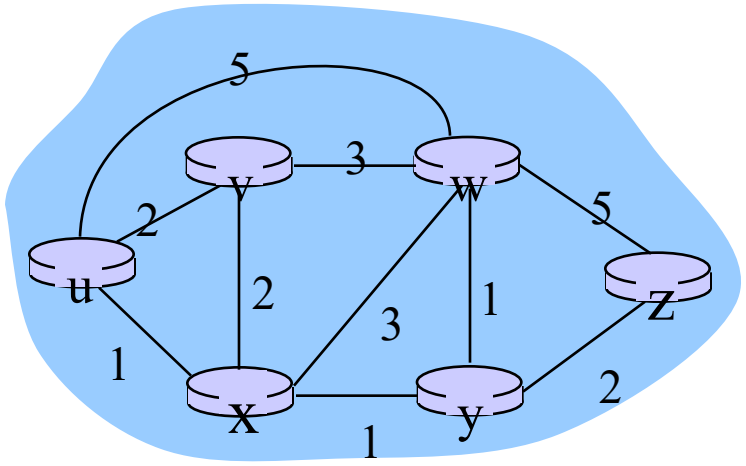
$$d_x(y) = \min_v \{c(x,v) + d_v(y)\}$$

从邻居v到达目的y的费用

x到邻居v的费用

遍历所有的邻居v

5.1.2距离向量路由算法



显然: dv(z) = 5, dx(z) = 3, dw(z) = 3

du(z) = min { c(u,v) + dv(z),

c(u,x) + dx(z),

c(u,w) + dw(z) }

= min {2 + 5,

1 + 3,

5 + 3} = 4

重点：结点获得最短路径的下一跳, 该信息用于转发表中！

5.1.2距离向量路由算法

Dx(y) = 从结点x到结点y的最小费用估计

- x维护距离向量(DV): **D**x = [Dx(y): y ϵ N]
- 结点x:
- 已知到达每个邻居的费用: c(x,v)
- 维护其所有邻居的距离向量: **D**v = [Dv(y): y ϵ N]

核心思想:

- 每个结点不定时地将其自身的DV估计发送给其邻居
- 当x接收到邻居的新的DV估计时，即依据B-F更新其自身的距离向量估计：

$$D_x(y) \leftarrow \min_v\{c(x,v) + D_v(y)\} \text{ for each node } y \in N$$

- Dx(y)将最终收敛于实际的最小费用 dx(y)

5.1.2距离向量路由算法

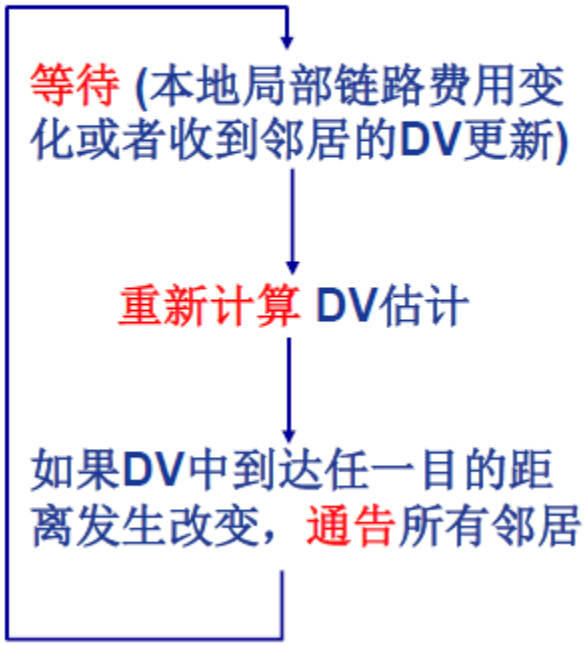
异步迭代:

- 引发每次局部迭代的因素
- 局部链路费用改变
- 来自邻居的DV更新

分布式:

- 每个结点只当DV变化时才通告给邻居
- 邻居在必要时（其DV更新后发生改变）再通告它们的邻居

每个结点：



5.1.2距离向量路由算法

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\}$

$= \min\{2+0 , 7+1\} = 2$

node y table

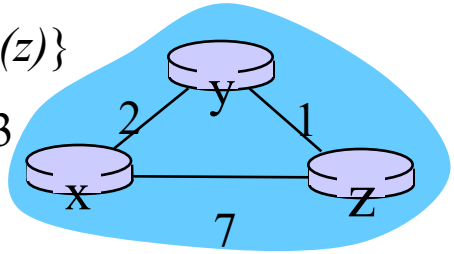
from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\}$

$= \min\{2+1 , 7+0\} = 3$

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0



5.1.2距离向量路由算法

node x table

from \ cost to	x	y	z
x	0	2	7
y	∞	∞	∞
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	7	1	0

from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node y table

from \ cost to	x	y	z
x	∞	∞	∞
y	2	0	1
z	∞	∞	∞

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

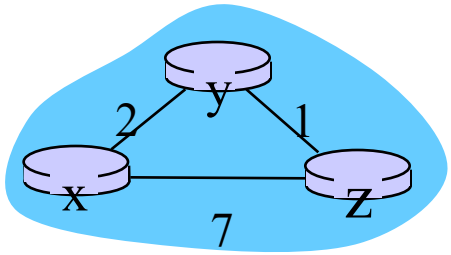
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0

node z table

from \ cost to	x	y	z
x	∞	∞	∞
y	∞	∞	∞
z	7	1	0

from \ cost to	x	y	z
x	0	2	7
y	2	0	1
z	3	1	0

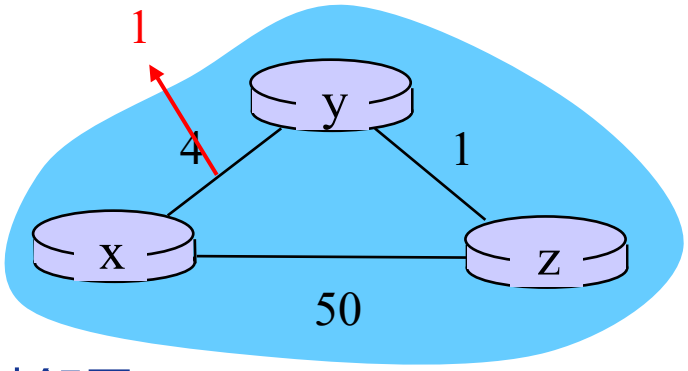
from \ cost to	x	y	z
x	0	2	3
y	2	0	1
z	3	1	0



5.1.2距离向量路由算法

链路费用变化:

- 结点检测本地链路费用变化
- 更新路由信息，重新计算距离向量
- 如果DV改变，通告所有邻居



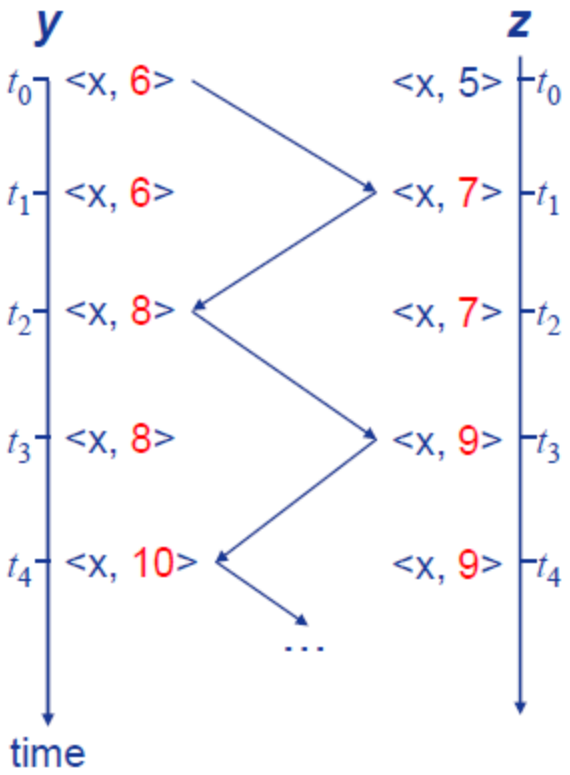
t0：*y*检测到链路费用改变 ，更新DV，通告其邻居.

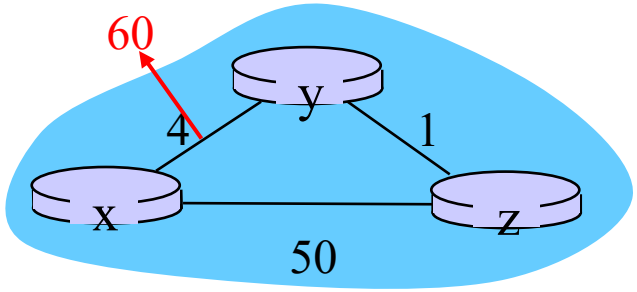
t1：*z*收到*y*的DV更新，更新其距离向量表，计算到达*x*的最新最小费用，更新其DV，并发送给其所有邻居.

t2：*y*收到*z*的DV更新，更新其距离向量表，重新计算*y*的DV，未发生改变，不再向*z*发送*DV*.

"☺好消息传播快！"　　"坏消息会怎么样呢？"

5.1.2距离向量路由算法





坏消息传播慢！

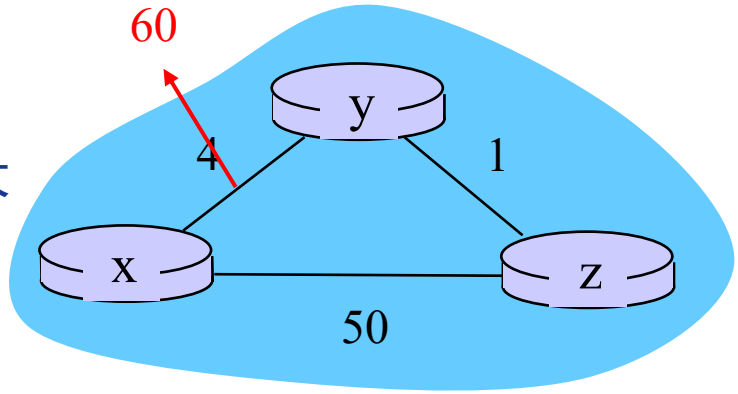
— “无穷计数 (count to infinity)” 问题！

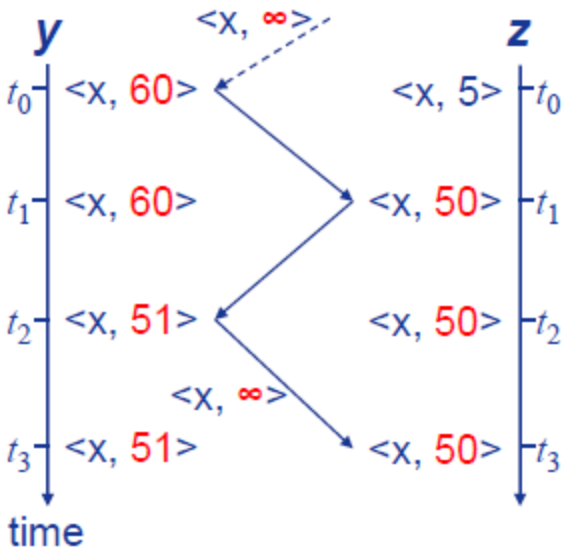
5.1.2距离向量路由算法

毒性逆转(poisoned reverse):

□如果一个结点(e.g. Z)到达某目的(e.g.X)的最小费用路径是通过某个邻居(e.g.Y)，则：

□通告给该邻居结点到达该目的的距离为无穷大



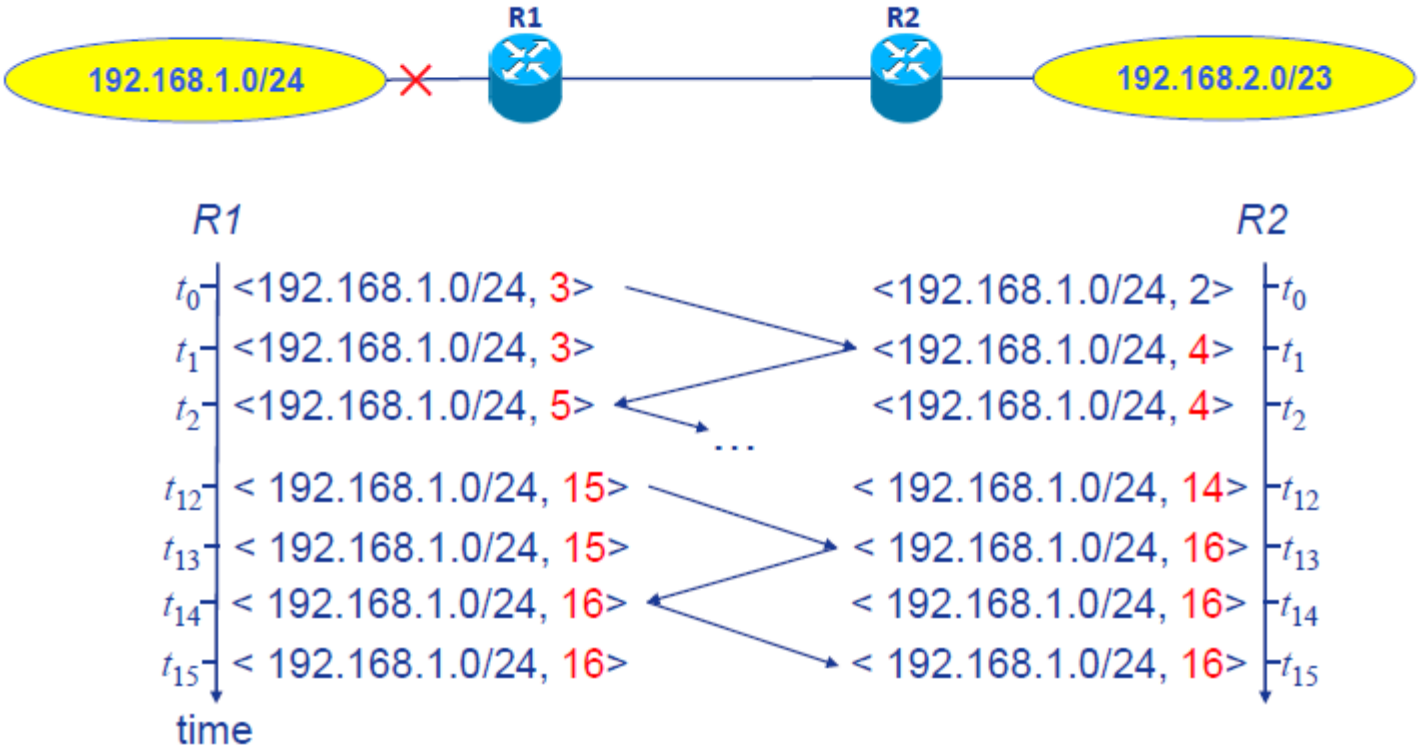


毒性逆转能否彻底解决无穷计数问题？

5.1.2距离向量路由算法

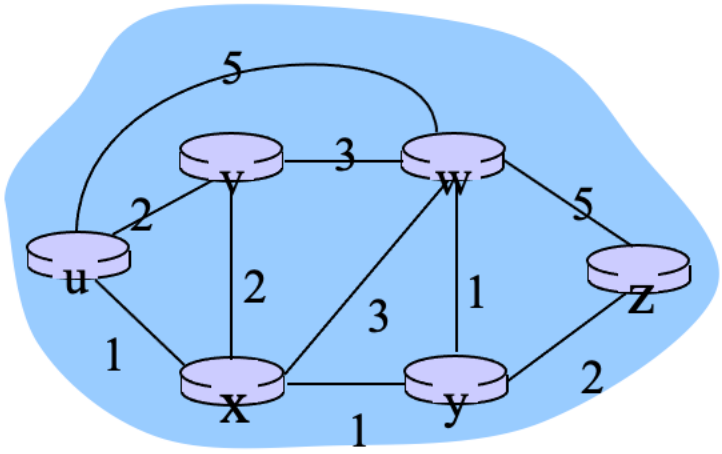
定义最大度量(maximum metric):

- 定义一个最大的有效费用值，如15跳步，16跳步表示∞

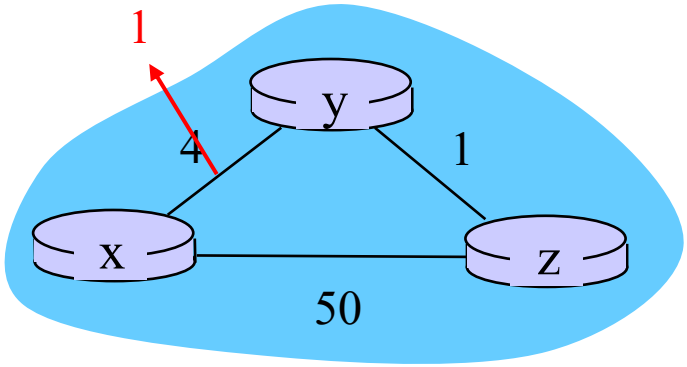


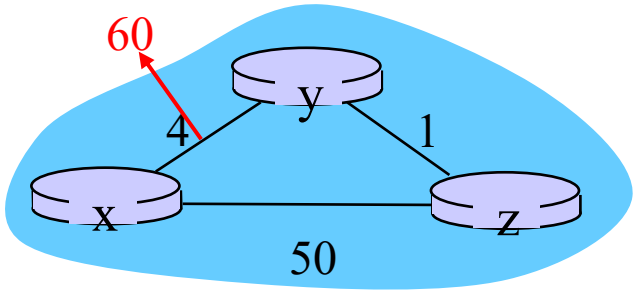
回 顾

Step	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	uxv	2,u	3,y			4,y
3	uxvv		3,y			4,y
4	uxvvw					4,y
5	uxvvwz					



回 顾





5.2 内部路由协议 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。

5.2 内部路由协议 OSPF

- “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”。

5.2 内部路由协议 OSPF

- 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。先向相邻的发送，相邻的又向其相邻的发送。。。。。
- 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)。
- 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。

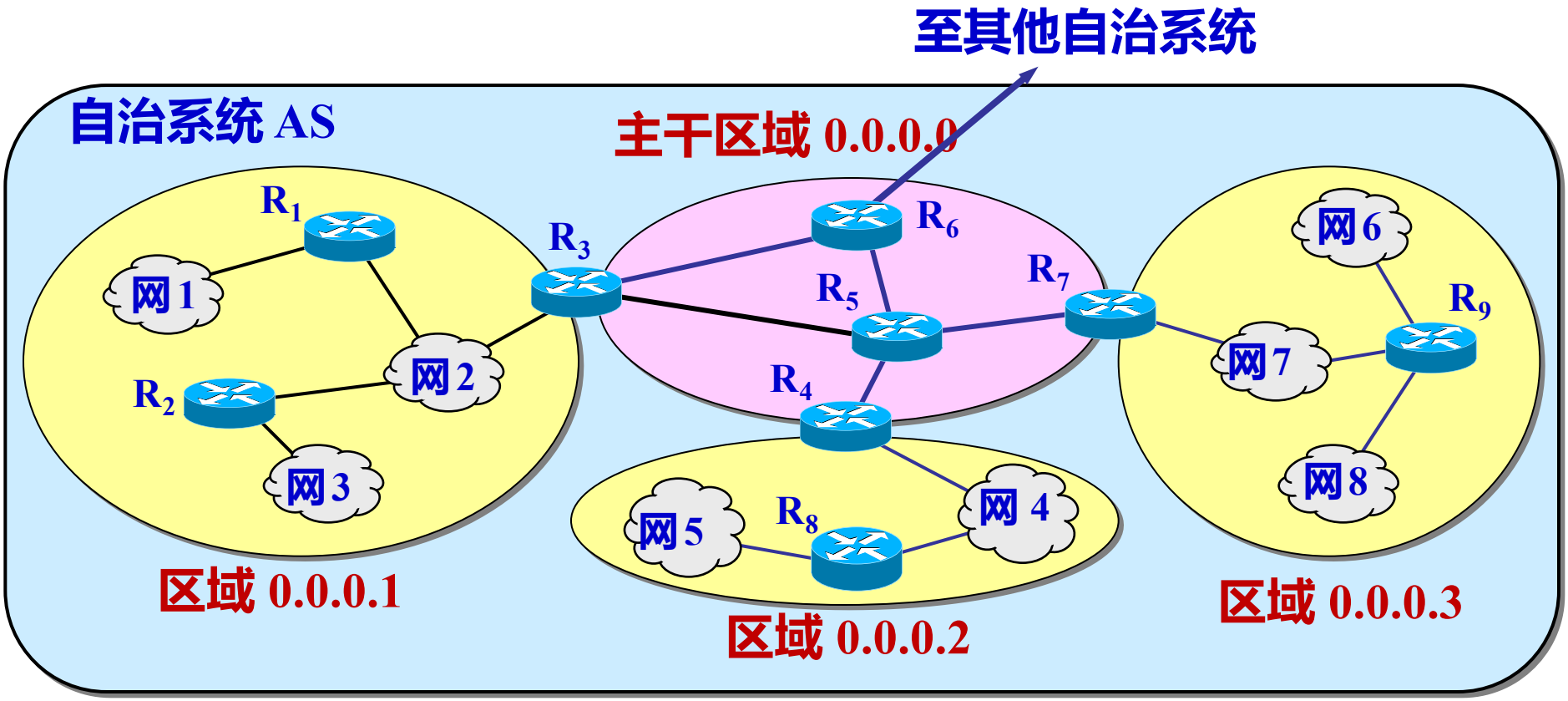
5.2 内部路由协议 OSPF

- 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。

5.2 内部路由协议 OSPF

- 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。
- 见下图例子：

5.2 内部路由协议 OSPF

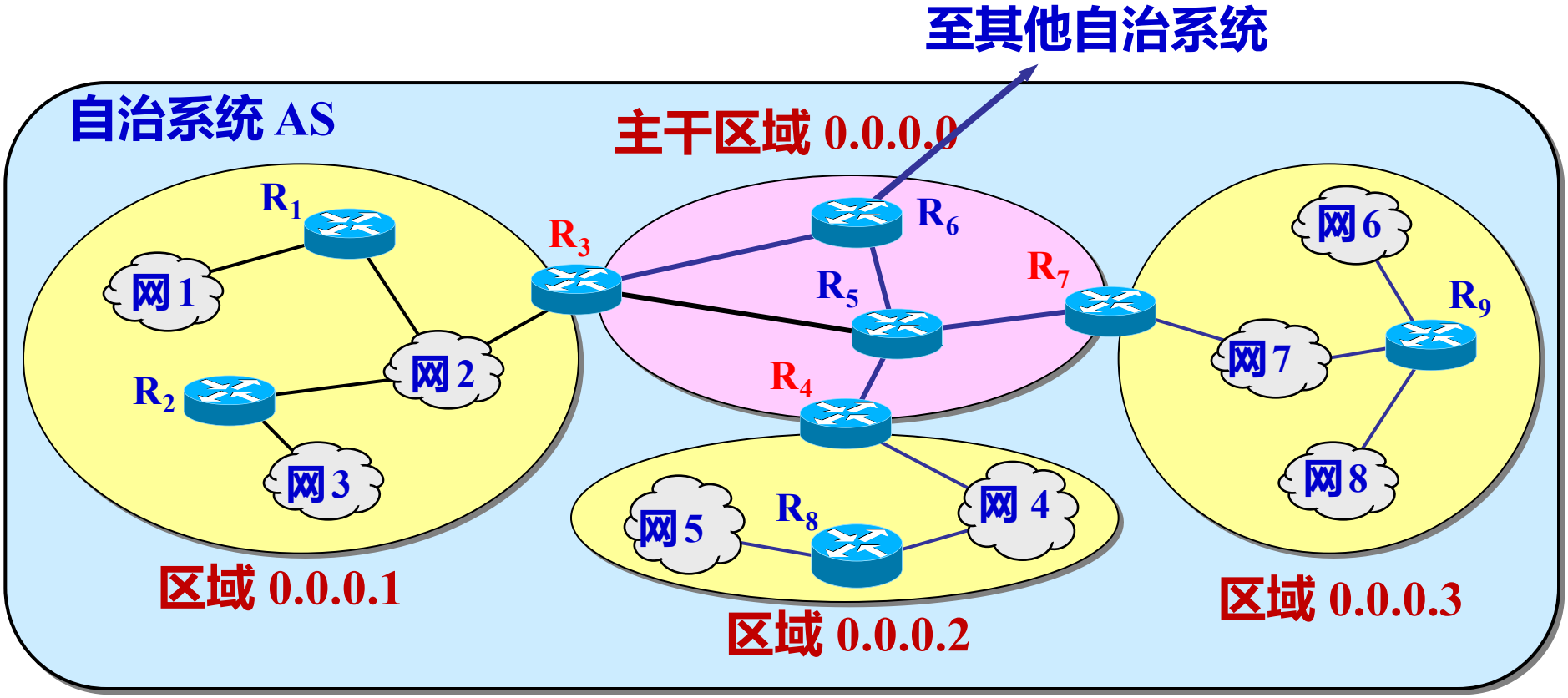


OSPF 划分为两种不同的区域

5.2 内部路由协议 OSPF

- 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。在上层的区域叫作主干区域 (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。

5.2 内部路由协议 OSPF



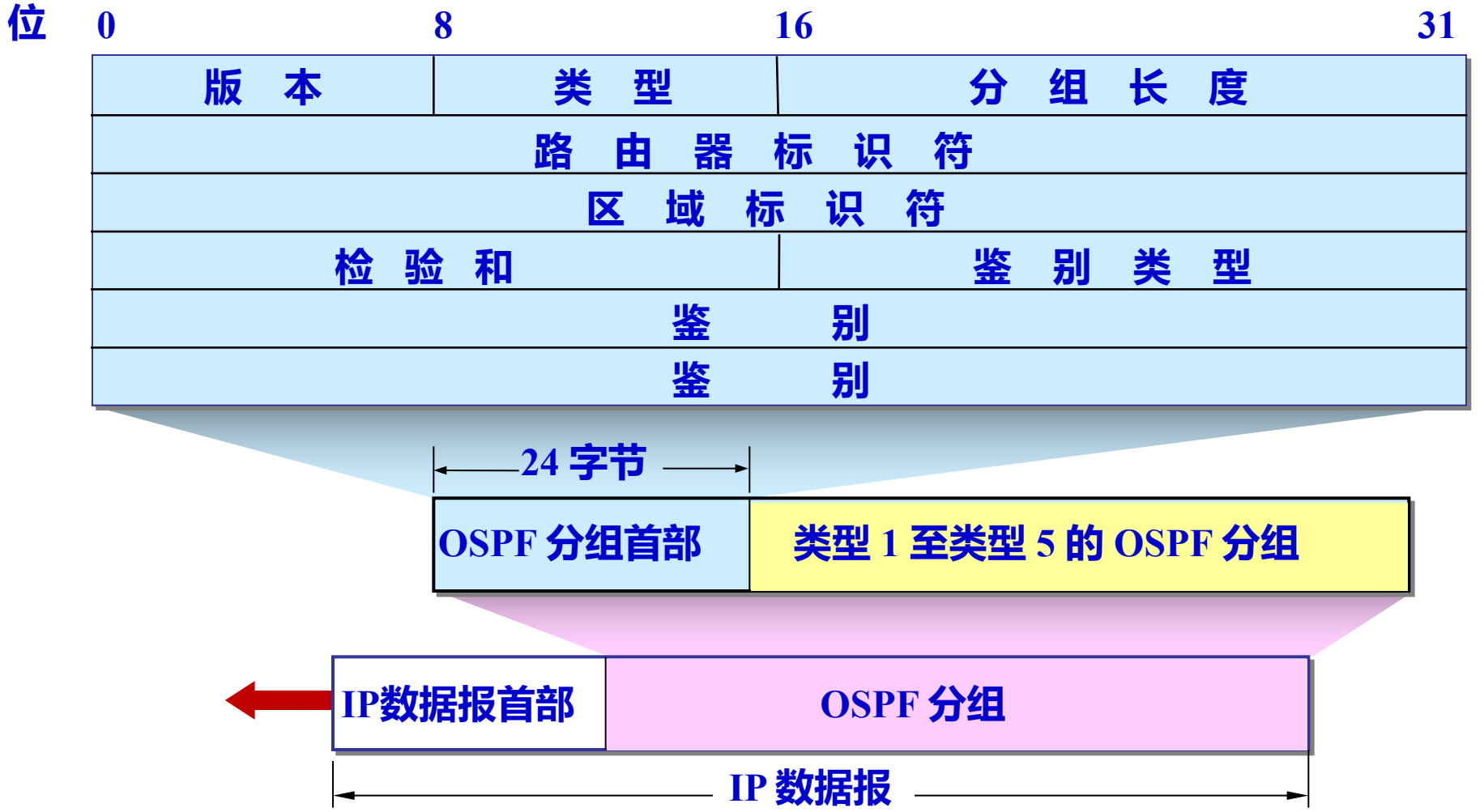
5.2 内部路由协议 OSPF

- 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。
- 数据报很短的另一好处是可以不必将长的数据报分片传送。
- 但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。

5.2 内部路由协议 OSPF

- OSPF 对不同的链路可根据 IP 分组的不同服务类型 TOS 而设置成不同的代价。因此，OSPF 对于不同类型的业务可计算出不同的路由。
- 如果到同一个目的网络有多条相同代价的路径，那么可以将通信量分配给这几条路径。这叫作多路径间的负载平衡。
- 所有在 OSPF 路由器之间交换的分组都具有鉴别的功能。
- 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。全部序号空间在600年不会产生重复号！

5.2 内部路由协议 OSPF



OSPF 分组用 IP 数据报传送

5.2 内部路由协议 OSPF

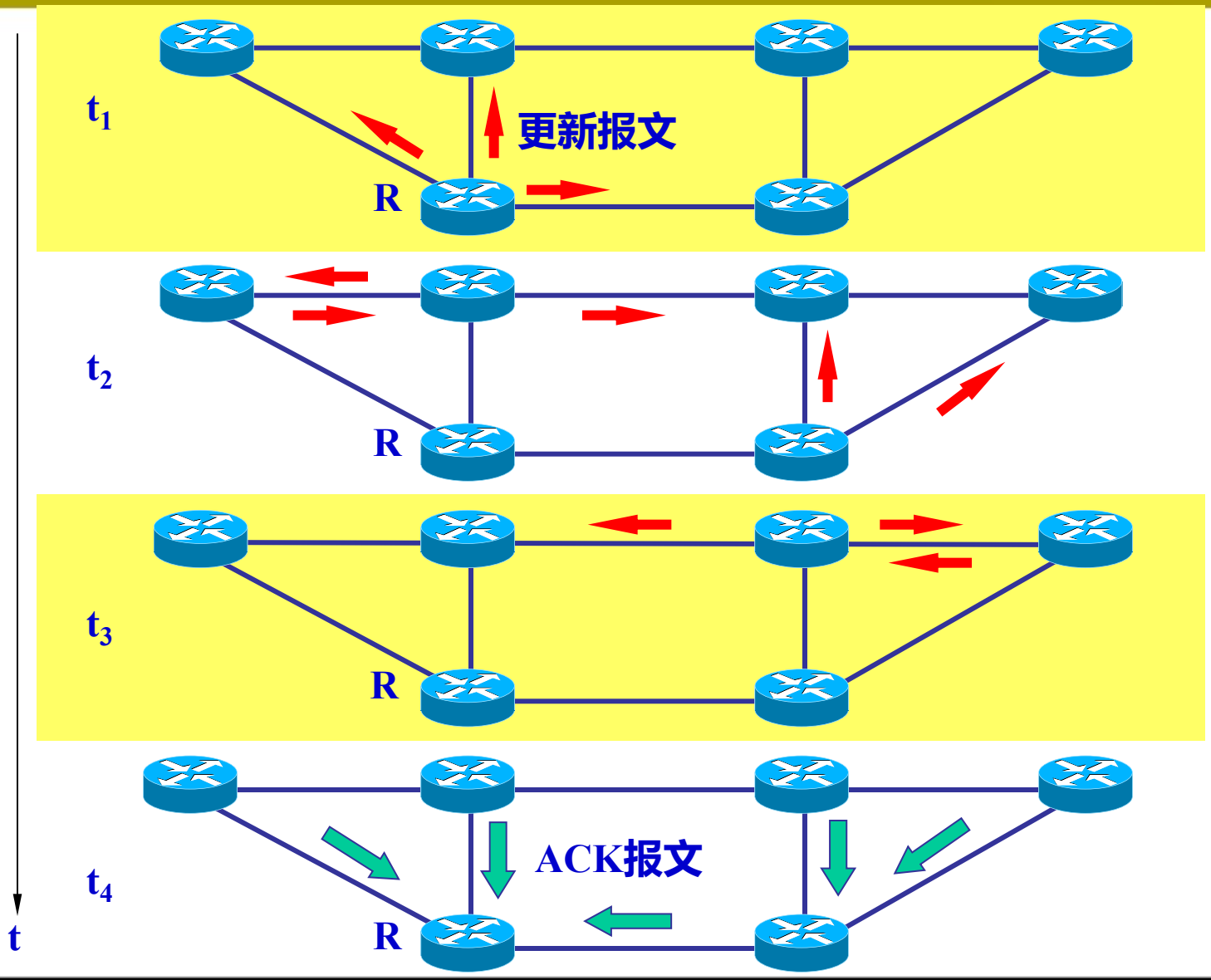
- 类型1，问候 (Hello) 分组。维护临站可达性。
- 类型2，数据库描述 (Database Description) 分组。
- 类型3，链路状态请求 (Link State Request) 分组。
- 类型4，链路状态更新 (Link State Update) 分组，用洪泛法对全网更新链路状态。
- 类型5，链路状态确认 (Link State Acknowledgment) 分组。

注：两个相邻路由器每隔10秒就发送一次问候信息，确知临站是可达的，若40秒没收到相邻路由器的问候，则可认为该路由器不可达，立即修改链路状态数据库，更改路由表。

5.2 内部路由协议 OSPF

OSPF 的基本操作

OSPF 使用可靠的洪泛法发送更新分组



5.2 内部路由协议 OSPF

- 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 的其他特点

5.2 内部路由协议 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。

5.2 内部路由协议 RIP

- 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

5.2 内部路由协议 RIP

- 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。

5.2 内部路由协议 RIP

和那些路由器交换信息？交换什么信息？何时交换？

- (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。

5.2 内部路由协议 RIP

- 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。

5.2 内部路由协议 RIP

- 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。

5.2 内部路由协议 RIP

- 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。

路由器之间交换信息与路由表更新

5.2 内部路由协议 RIP

【例】已知路由器 R_6 有表 (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 (b)所示。试更新路由器 R_6 的路由表

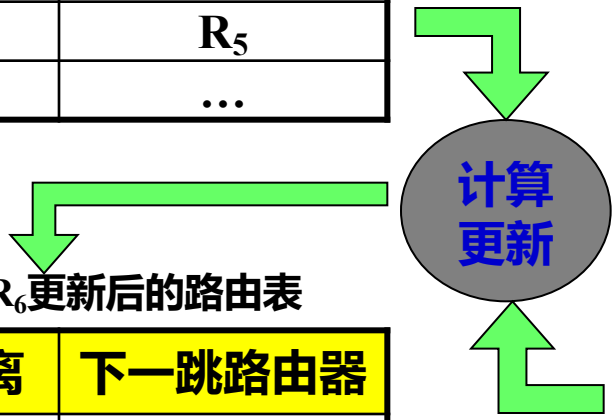
。

表 (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 (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

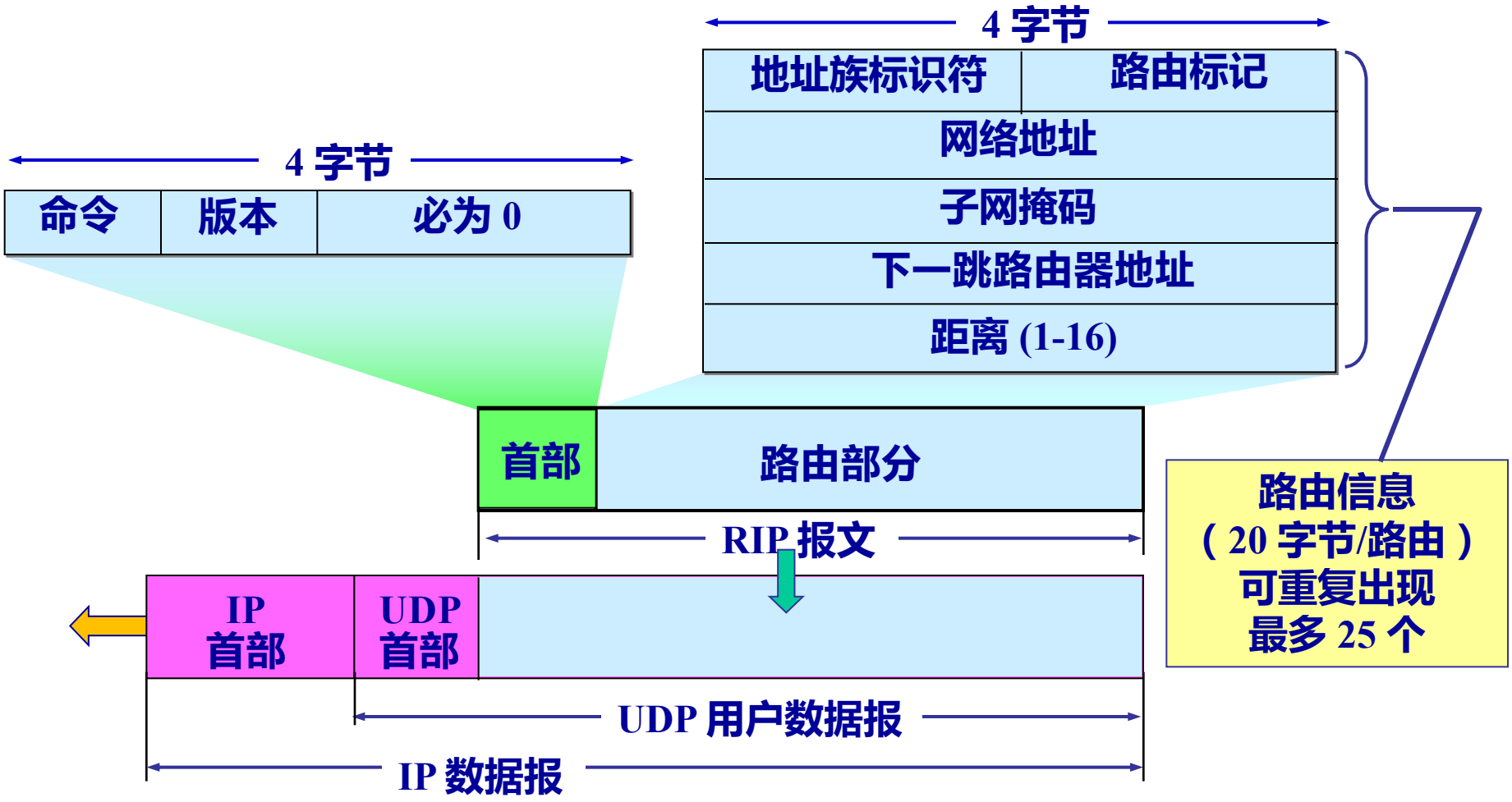
表 (c) 修改后的表(b)

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(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
…	…	…

5.2 内部路由协议 RIP



5.2 内部路由协议 RIP

- 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。

5.3 外部路由协议 BGP

- BGP 是不同自治系统的路由器之间交换路由信息的协议。
- BGP 较新版本是 2006 年 1 月发表的 BGP-4（BGP 第 4 个版本），即 RFC 4271 ~ 4278。
- 可以将 BGP-4 简写为 BGP。

5.3.1 外部路由协议 BGP

- 互联网的规模太大，使得自治系统之间路由选择非常困难。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。
- 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。

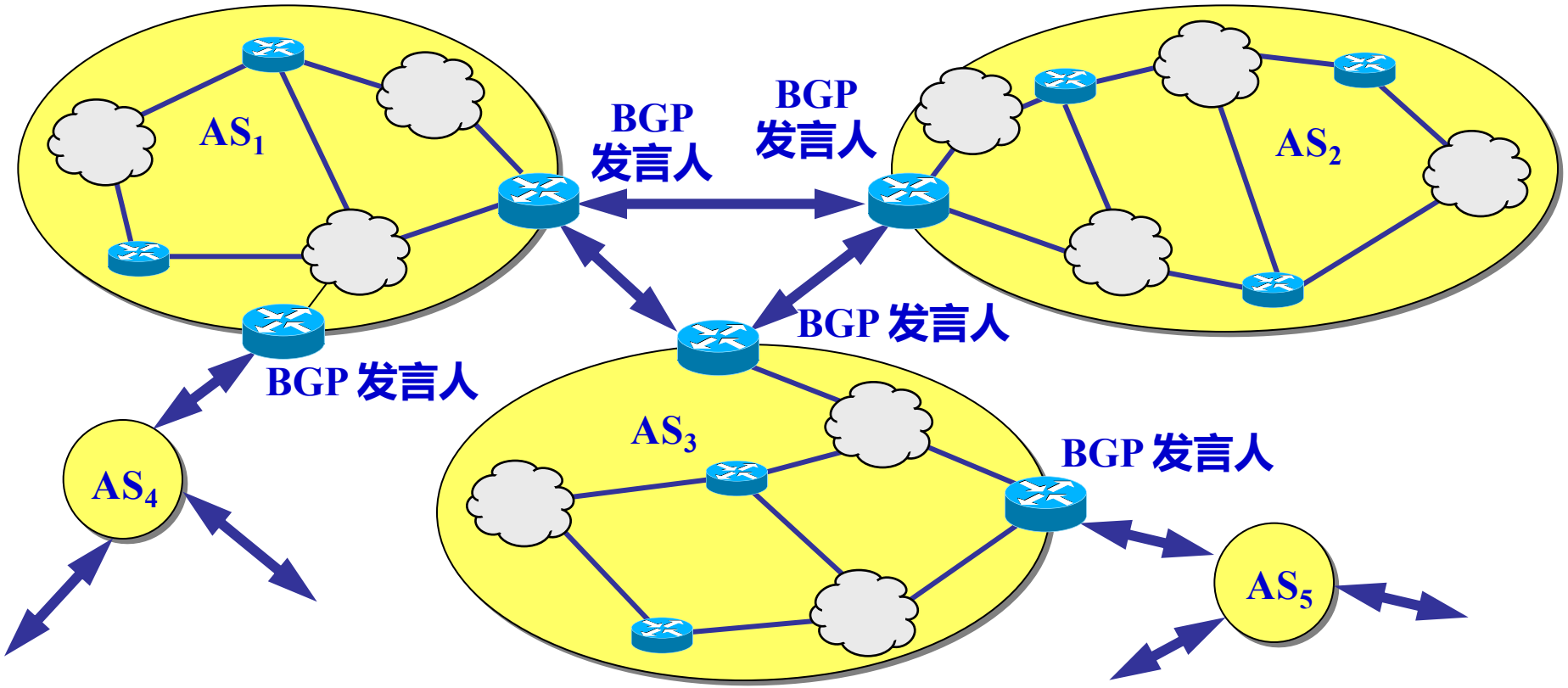
5.3.1 外部路由协议 BGP

- 在配置BGP时，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。

5.3.1 外部路由协议 BGP

- 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(neighbor)或对等站(peer) 。

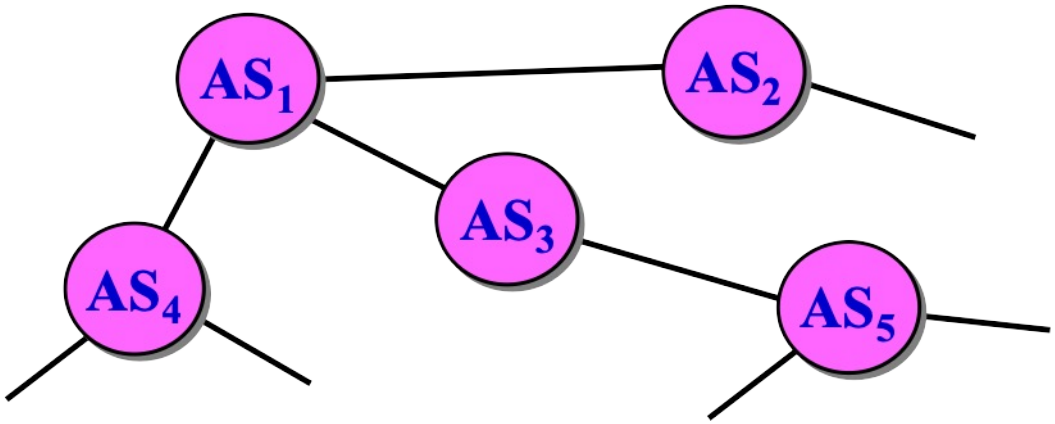
5.3.1 外部路由协议 BGP



BGP 发言人和自治系统 AS 的关系

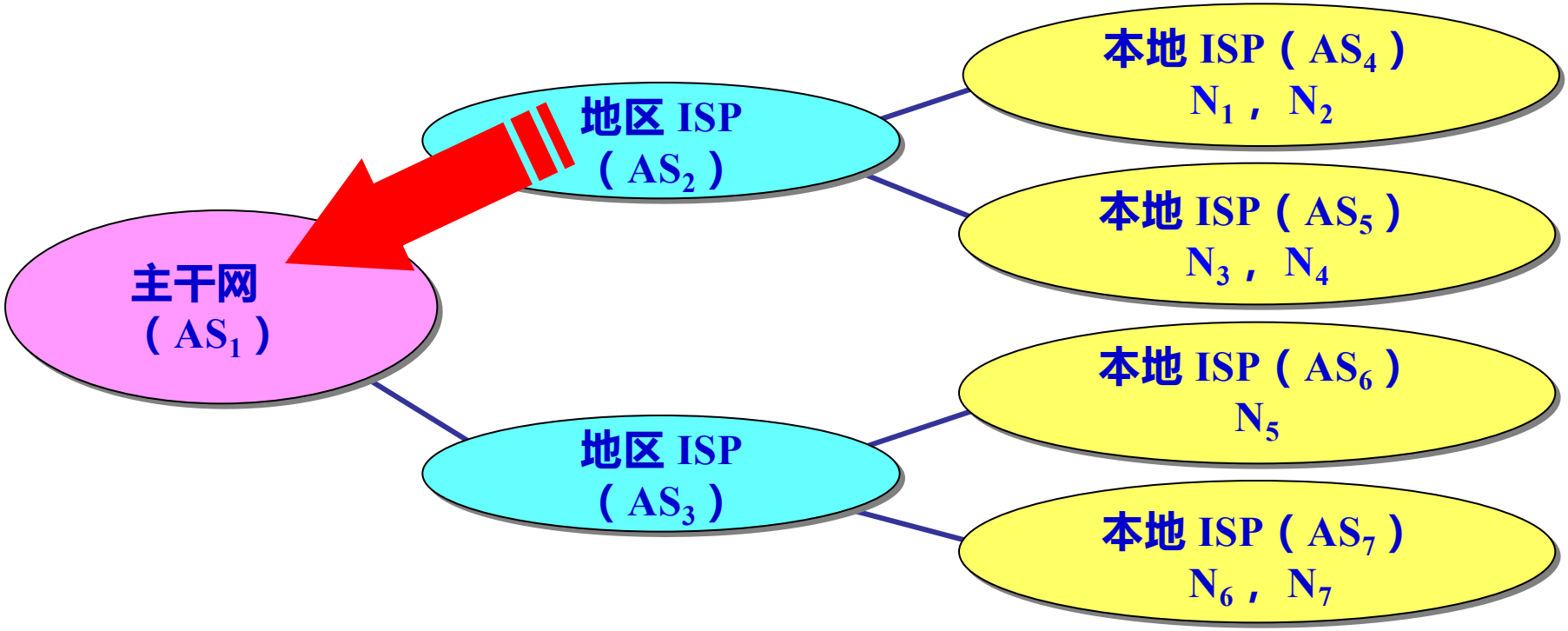
5.3.1 外部路由协议 BGP

- 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。



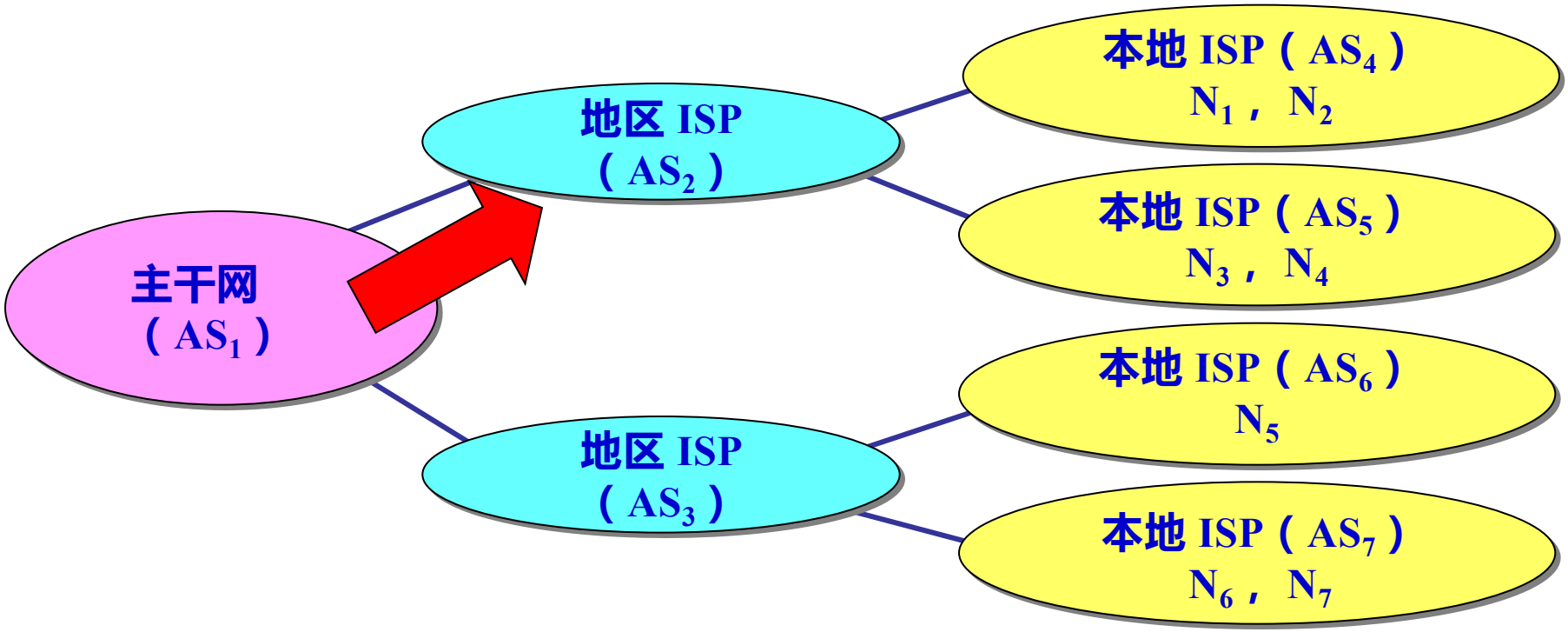
5.3.1 外部路由协议 BGP

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。”



5.3.1 外部路由协议 BGP

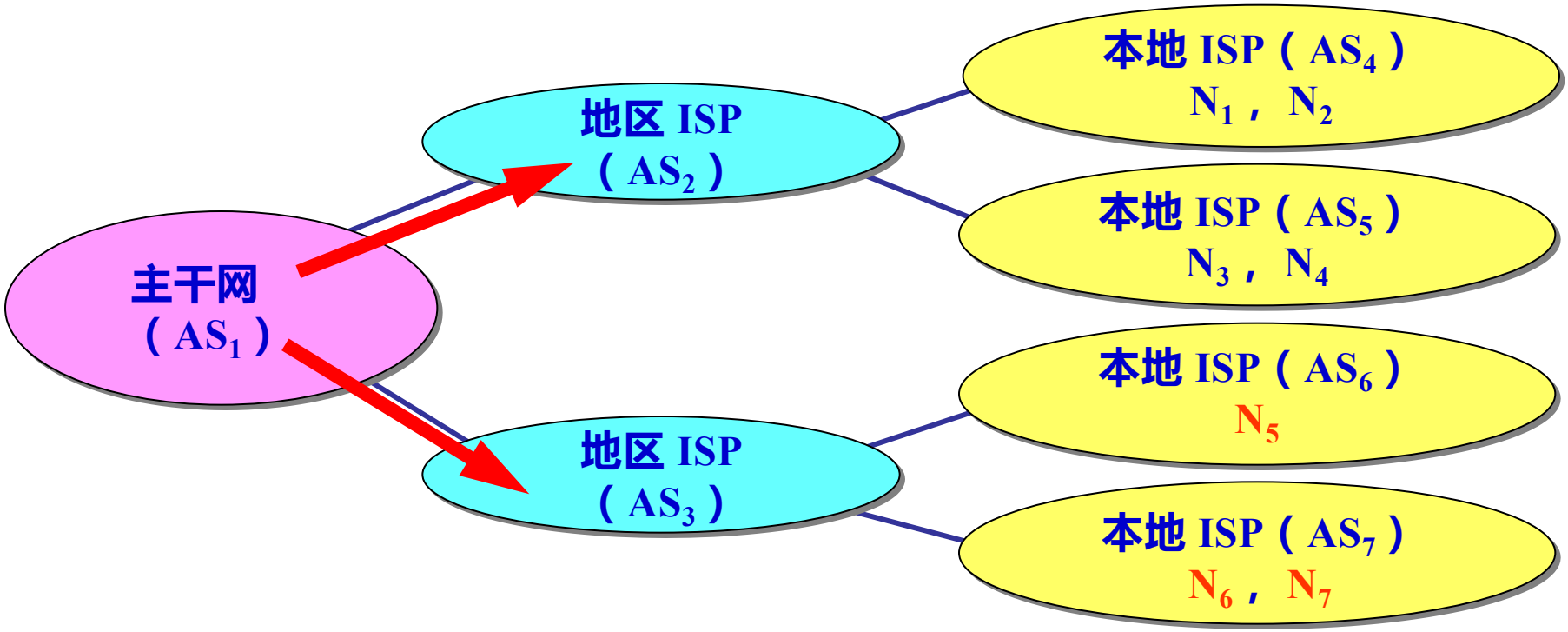
主干网收到这个通知后，就发出通知：“要到达网络 N1、 N2、N3 和 N4可沿路径（AS1, AS2）。”



BGP 发言人交换路径向量

5.3.1 外部路由协议 BGP

主干网还可发出通知：“要到达网络 N5、N6 和 N7 可沿路径（AS1, AS3）。”



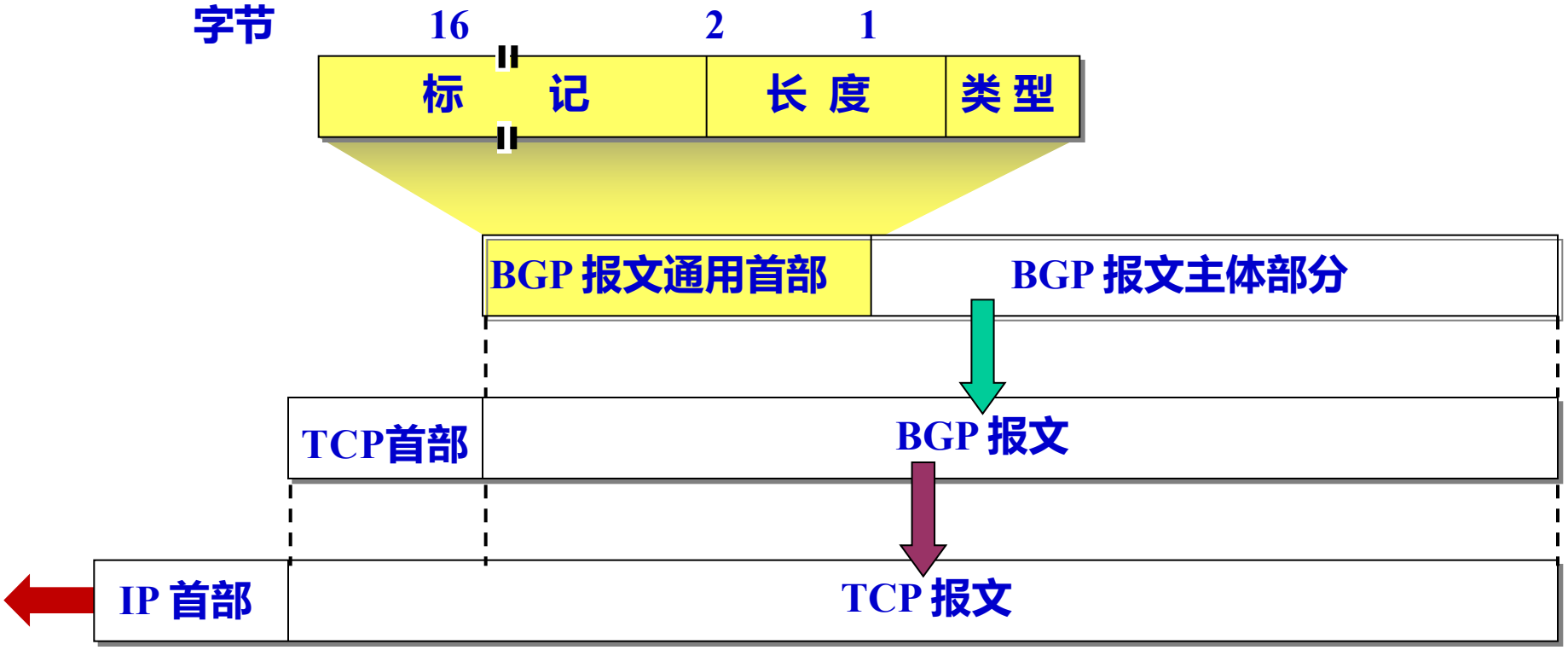
5.3.1 外部路由协议 BGP

- BGP 协议交换路由信息的结点数量级是自治系统数的量级，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。
- 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。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。

5.3.1 外部路由协议 BGP

- (1) 打开 (OPEN) 报文，用来与相邻的另一个BGP发言人建立关系。
- (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。
- (3) 保活 (KEEPALIVE) 报文，用来确认打开报文和周期性地证实邻站关系。
- (4) 通知 (NOTIFICATION) 报文，用来发送检测到的差错。

5.3.1 外部路由协议 BGP



BGP 报文结构

5.3.1 外部路由协议 BGP

策略(policy):

- inter-AS: 期望能够管理控制流量如何被路由，谁路由经过其网络等.
- intra-AS: 单一管理，无需策略决策

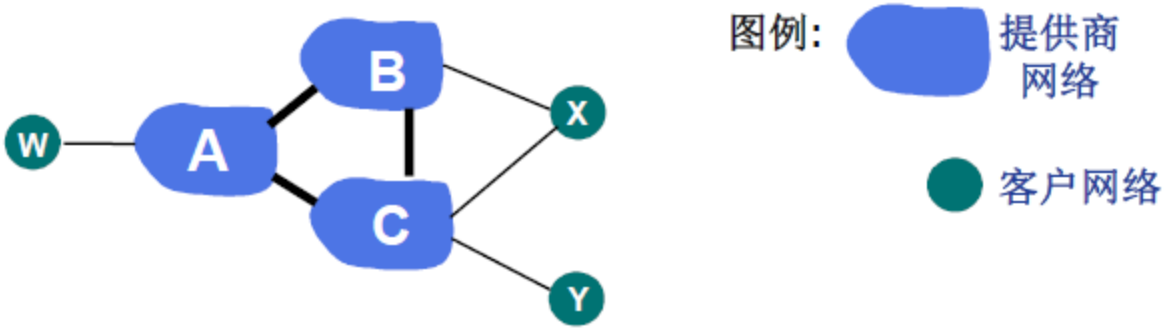
规模(scale):

- 层次路由节省路由表大小，减少路由更新流量
- 适应大规模互联网

性能(performance):

- intra-AS: 侧重性能
- inter-AS: 策略主导

5.3.1 外部路由协议 BGP

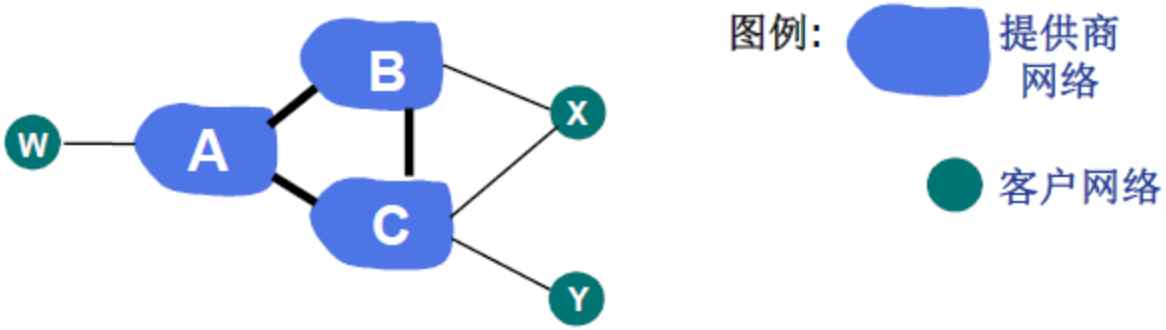


- A,B,C是提供商网络/AS(provider network/AS)
- X,W,Y是客户网络(customer network/AS)
- W,Y是桩网络(stub network/AS): 只与一个其他AS相连
- X是多宿网络(multi-homed network/AS): 连接两个其他AS

•X不期望经过他来路由B到C的流量

•… 因此，X不会向B通告任何一条到达C的路由

5.3.1 外部路由协议 BGP



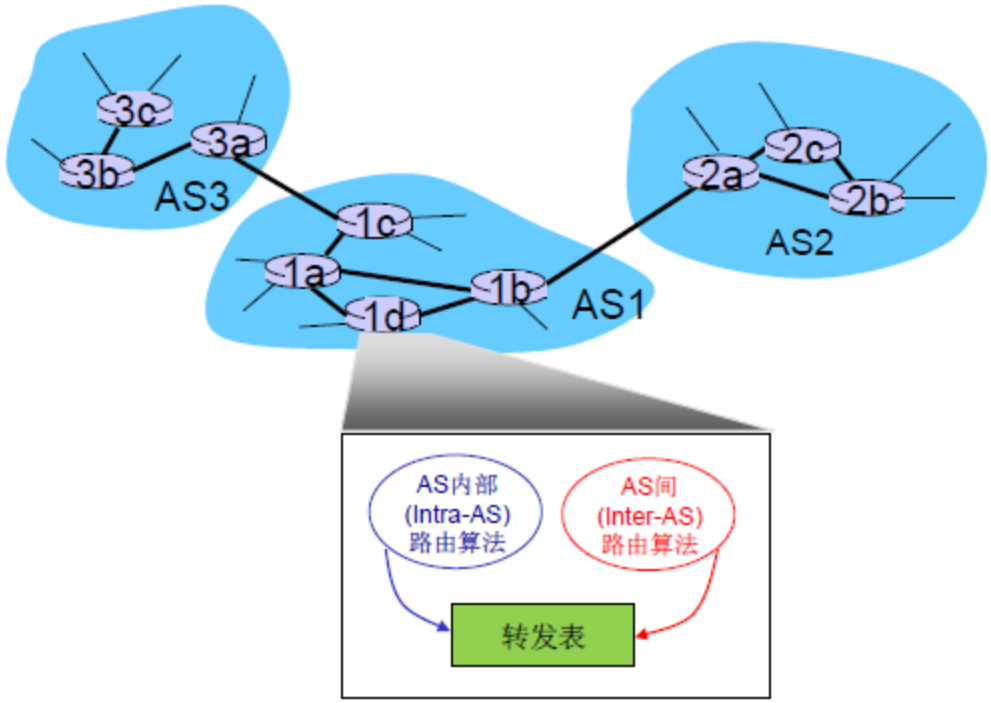
- A向B通告一条路径：AW
- B向X通告路径：BAW
- B是否应该向C通告路径BAW呢?

•绝不! B路由CBAW的流量没有任何“收益”，因为W和C均不是B的客户。

•B期望强制C通过A向W路由流量

•B期望只路由去往/来自其客户的流量 ！

5.3.2 层次路由



- 转发表由AS内部路由算法与AS间路由算法共同配置
  - AS内部路由算法设置AS内部目的网络路由入口(entries)
  - AS内部路由算法与AS间路由算法共同设置AS外部目的网络路由入

- BGP为每个AS提供了一种手段:
  - eBGP: 从邻居AS获取子网可达性信息.
  - iBGP: 向所有AS内部路由器传播子网可达性信息

.

5.3.2 层次路由

假设AS1内某路由器收到一个目的地址在AS1之外的数据报:

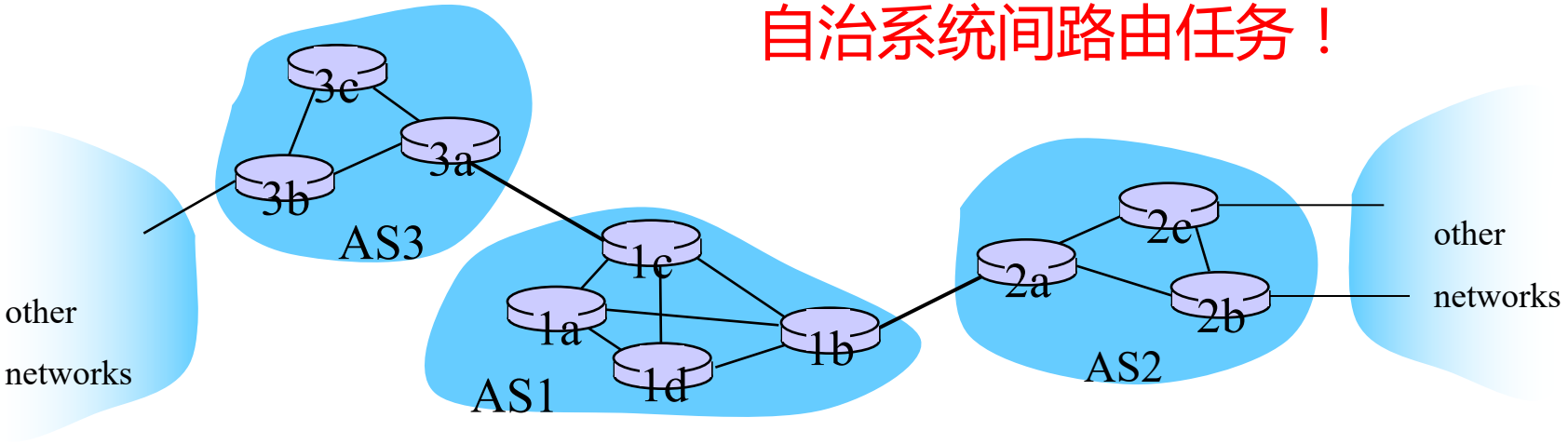
路由器应该将该数据报转发给哪个网关路由器呢？

AS1必须:

1.学习到哪些目的网络可以通过AS2到达，哪些可以通过AS3到达

2.将这些网络可达性信息传播给AS1内部路由器

自治系统间路由任务！



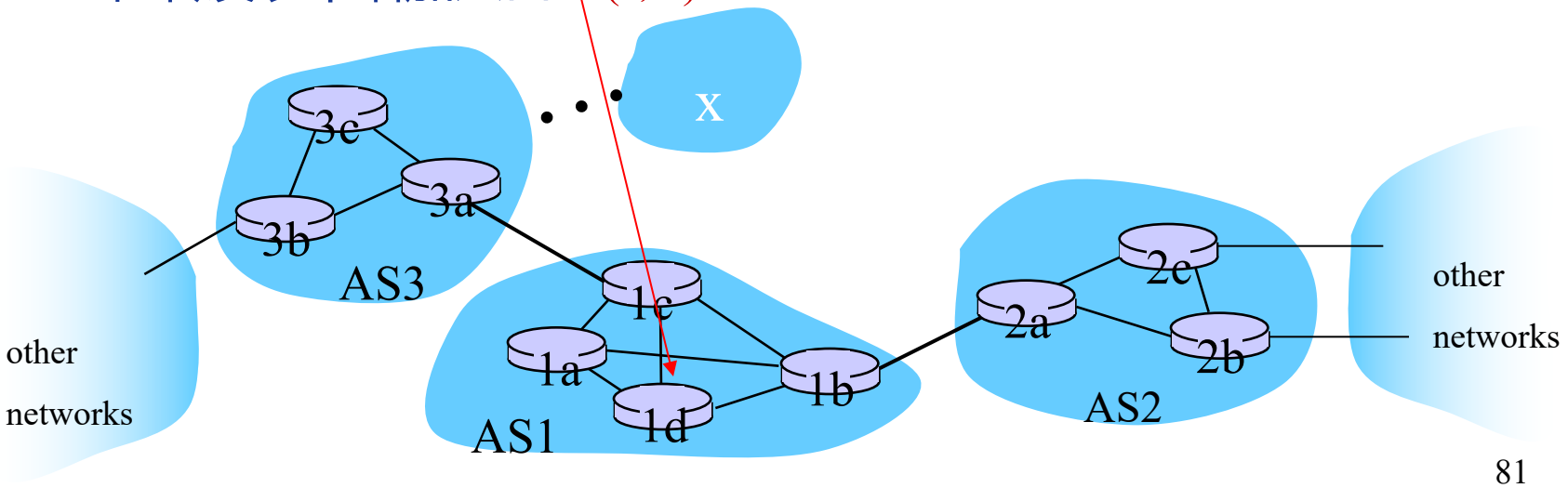
5.3.2 层次路由

假设AS1学习到(通过AS间路由协议)：子网*x*可以通过AS3 (网关 1c)到达，但不能通过AS2到达

- AS间路由协议向所有内部路由器通告该可达性信息

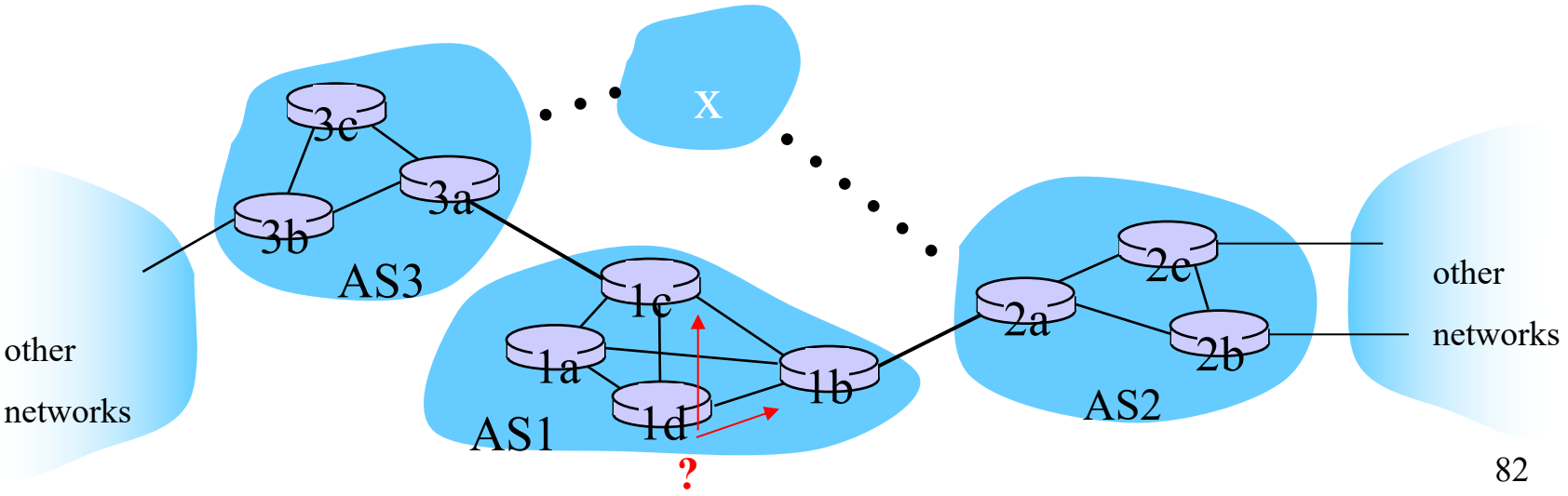
路由器1d：利用AS内部路由信息，确定其到达1c的最小费用路径接口*I*

- 在转发表中增加入口：(*x*, *I*)



5.3.2 层次路由

- 假设AS1通过AS间路由协议学习到：子网*x*通过AS3和AS2均可到达
- 为了配置转发表，路由器1d必须确定应该将去往子网*x*的数据报转发给哪个网关？
- 这个任务也是由AS间路由协议完成!

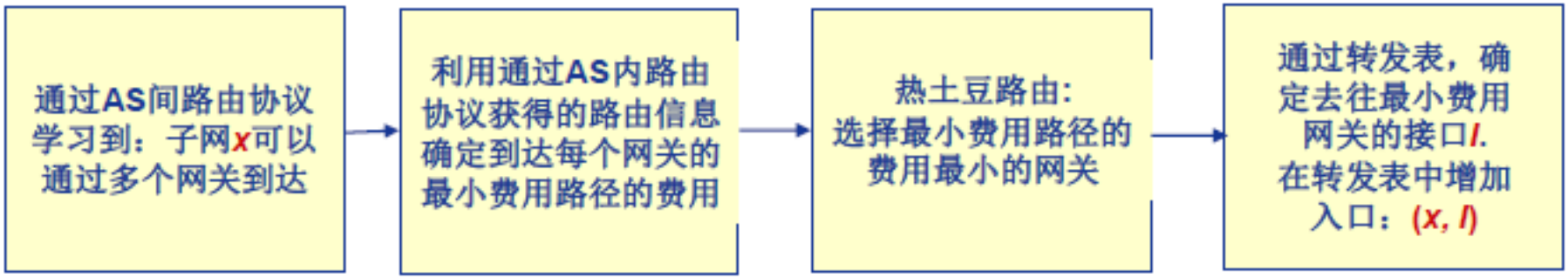


5.3.2 层次路由

- 假设AS1通过AS间路由协议学习到：子网*x*通过AS3和AS2均可到达
- 为了配置转发表，路由器1d必须确定应该将去往子网*x*的数据报转发给哪个网关？

•这个任务也是由AS间路由协议完成!

- 热土豆路由: 将分组发送给最近的网关路由器.



重要内容：

- 链路状态选路算法与距离向量选路算法；RIP，OSPF，BGP4的技术特点，应用场景。

- 作业
- P16、P17、P20