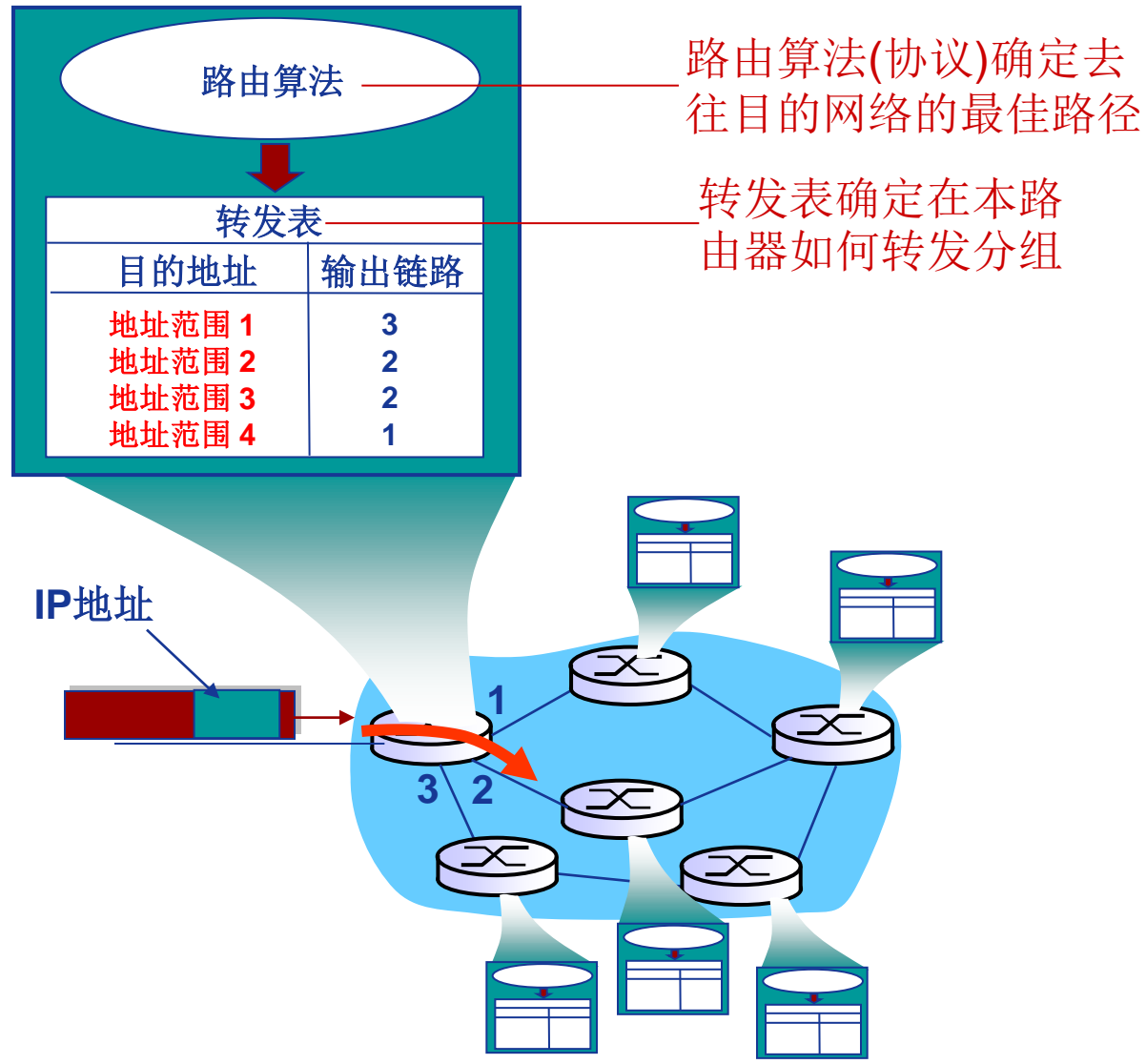


本讲主题

路由算法

路由与转发



网络抽象：图

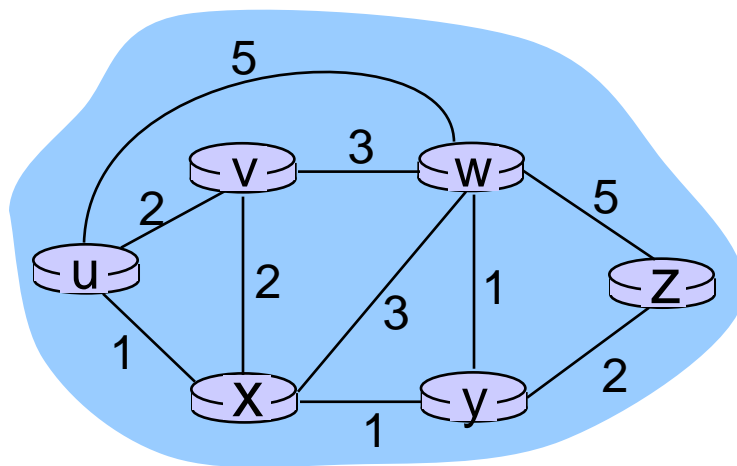


图: $G = (N, E)$

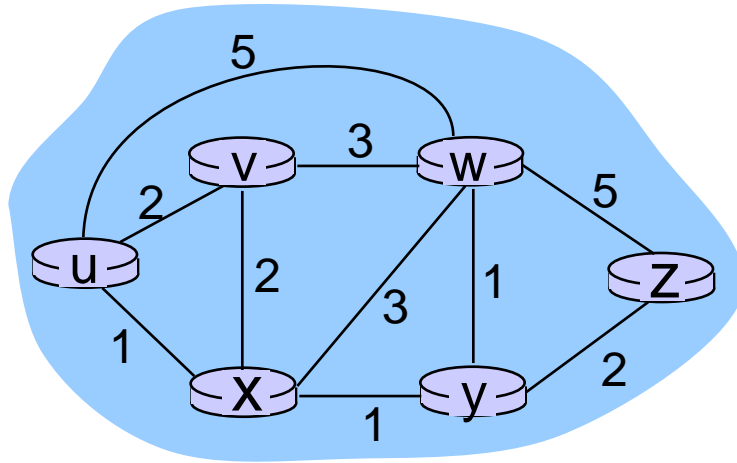
N = 路由器集合 = $\{ u, v, w, x, y, z \}$

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

图抽象：费用(Costs)



$c(x, x') =$ 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: $(x_1, x_2, x_3, \dots, x_p) = c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$

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

路由算法分类

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

本讲主题

链路状态路由算法

网络抽象：图

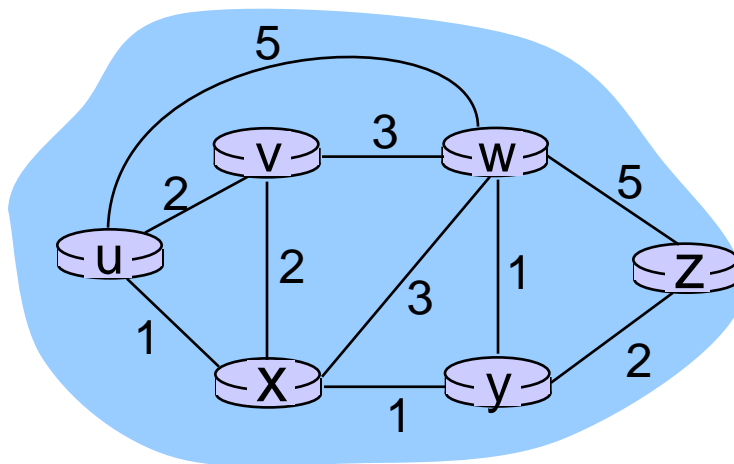


图: $G = (N, E)$

N = 路由器集合 = $\{ u, v, w, x, y, z \}$

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

链路状态路由算法

Dijkstra 算法

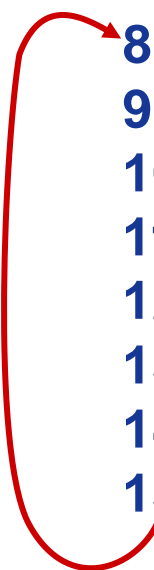
- ❖ 所有结点(路由器)掌握网络拓扑和链路费用
 - 通过“链路状态广播”
 - 所有结点拥有相同信息
- ❖ 计算从一个结点(“源”)到达所有其他结点的最短路径
 - 获得该结点的转发表
- ❖ 迭代: k 次迭代后, 得到到达 k 个目的结点的最短路径

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 $=\infty$
- ❖ $D(v)$: 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

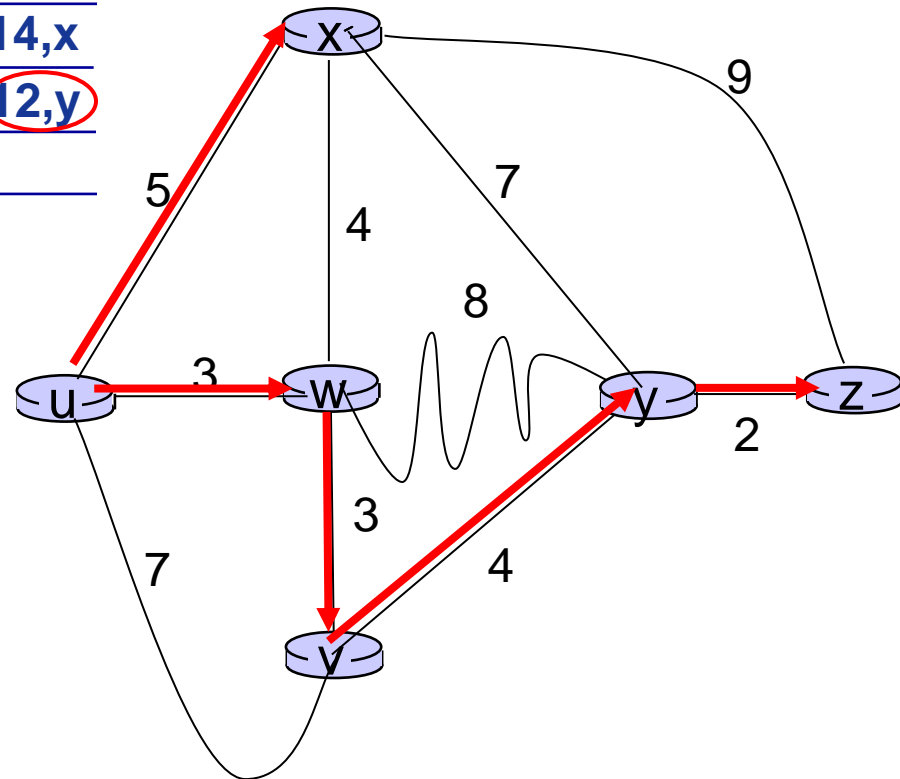
Dijkstra 算法

```
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  $D(v) = \infty$ 
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
```



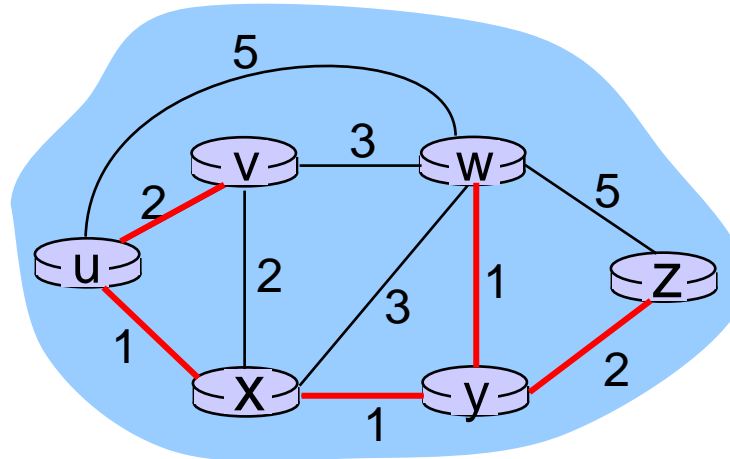
Dijkstra 算法:例1

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



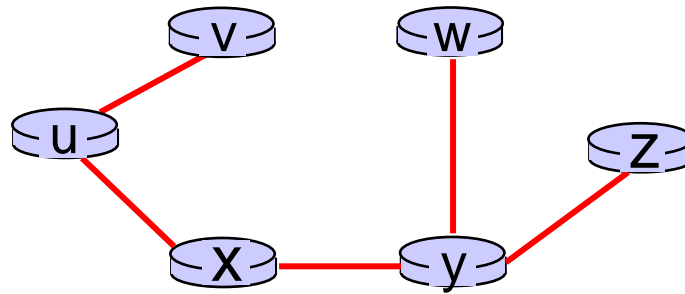
Dijkstra 算法:例2

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					



Dijkstra 算法:例2

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)

本讲主题

距离向量路由算法(1)

距离向量(Distance Vector)路由算法

Bellman-Ford方程(动态规划)

令：

$d_x(y)$:= 从x到y最短路径的费用（距离）

则：

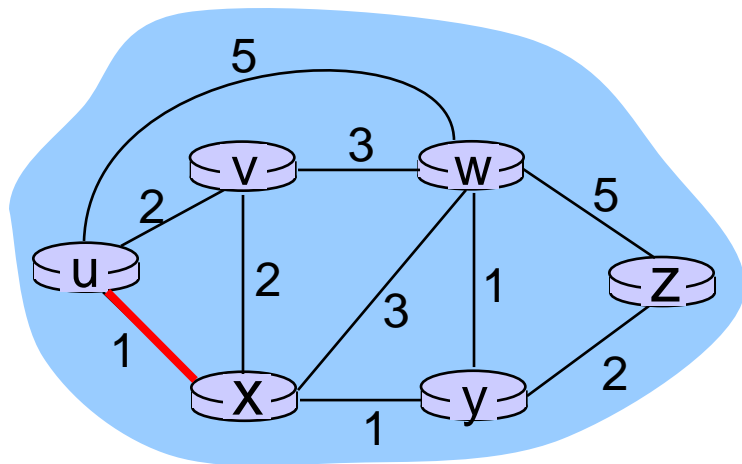
$$d_x(y) = \min_v \{ c(x,v) + d_v(y) \}$$

从邻居v到达目的y的费用（距离）

x到邻居v的费用

在x的所有邻居v中取最小值

Bellman-Ford 举例



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

根据B-F方程:

$$\begin{aligned} d_u(z) &= \min \{ c(u,v) + d_v(z), \\ &\quad c(u,x) + d_x(z), \\ &\quad c(u,w) + d_w(z) \} \\ &= \min \{ 2 + 5, \\ &\quad \mathbf{1 + 3}, \\ &\quad 5 + 3 \} = \mathbf{4} \end{aligned}$$

重点: 结点获得最短路径的下一跳, 该信息用于转发表中!

距离向量路由算法

❖ $D_x(y)$ = 从结点x到结点y的最小费用估计

- x维护距离向量(DV): $D_x = [D_x(y): y \in N]$

❖ 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: $D_v = [D_v(y): y \in N]$

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:

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

❖ $D_x(y)$ 将最终收敛于实际的最小费用 $d_x(y)$

距离向量路由算法

异步迭代:

❖ 引发每次局部迭代的因素

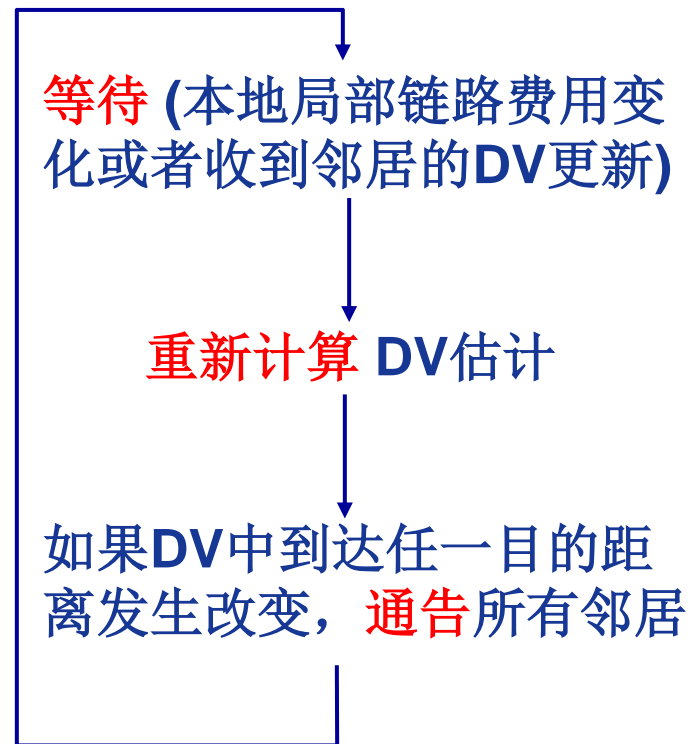
- 局部链路费用改变
- 来自邻居的DV更新

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:



本讲主题

距离向量路由算法（2）

距离向量路由算法：举例

**node x
table**

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

**node y
table**

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

**node z
table**

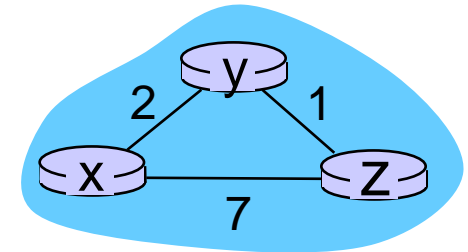
		cost to		
		x	y	z
from	x	∞	∞	∞
	y	∞	∞	∞
	z	7	1	0

cost to

		x	y	z
from	x	0	2	3
	y	2	0	1
	z	7	1	0

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} \\ = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} \\ = \min\{2+1, 7+0\} = 3$$



time

距离向量路由算法：举例

**node x
table**

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

**node y
table**

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

**node z
table**

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

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

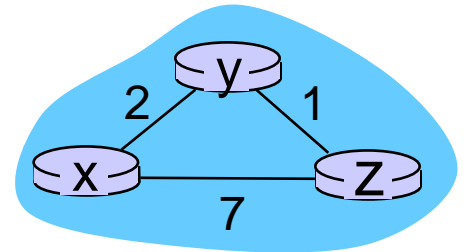
from	cost to		
	x	y	z
x	0	2	7
y	2	0	1
z	7	1	0

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

from	cost to		
	x	y	z
x	0	2	3
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



time

本讲主题

层次路由

层次路由

将任意规模网络抽象为一个图计算路由-过于理想化

- ❖ 标识所有路由器

- ❖ “扁平”网络

——在实际网络（尤其是大规模网络）中，**不可行！**

网络规模：考虑6亿目的结点的网络

- ❖ 路由表几乎无法存储！

- ❖ 路由计算过程的信息（e.g. 链路状态分组、DV）交换量巨大，会淹没链路！

管理自治：

- ❖ 每个网络的管理可能都期望自主控制其网内的路由

- ❖ 互联网(internet) = 网络之网络(network of networks)

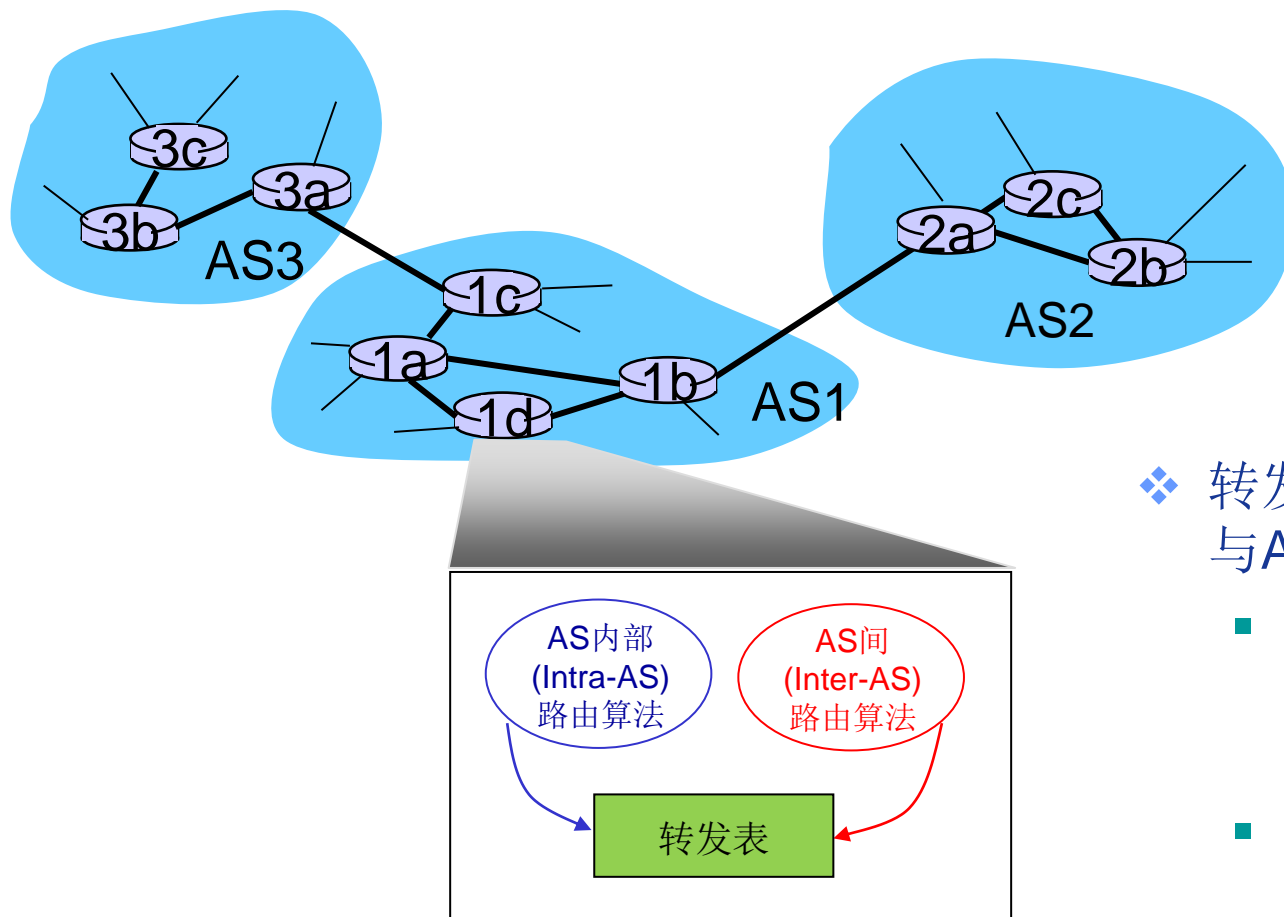
层次路由

- ❖ 聚合路由器为一个区域：
自治系统**AS**
(autonomous systems)
- ❖ 同一**AS**内的路由器运行相同的路由协议(算法)
 - 自治系统内部路由协议
(“intra-AS” routing protocol)
 - 不同自治系统内的路由器可以运行不同的**AS**内部路由协议

网关路由器(**gateway router**):

- ❖ 位于**AS**“边缘”
- ❖ 通过链路连接其他**AS**的网关路由器

互连的AS



- ❖ 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口

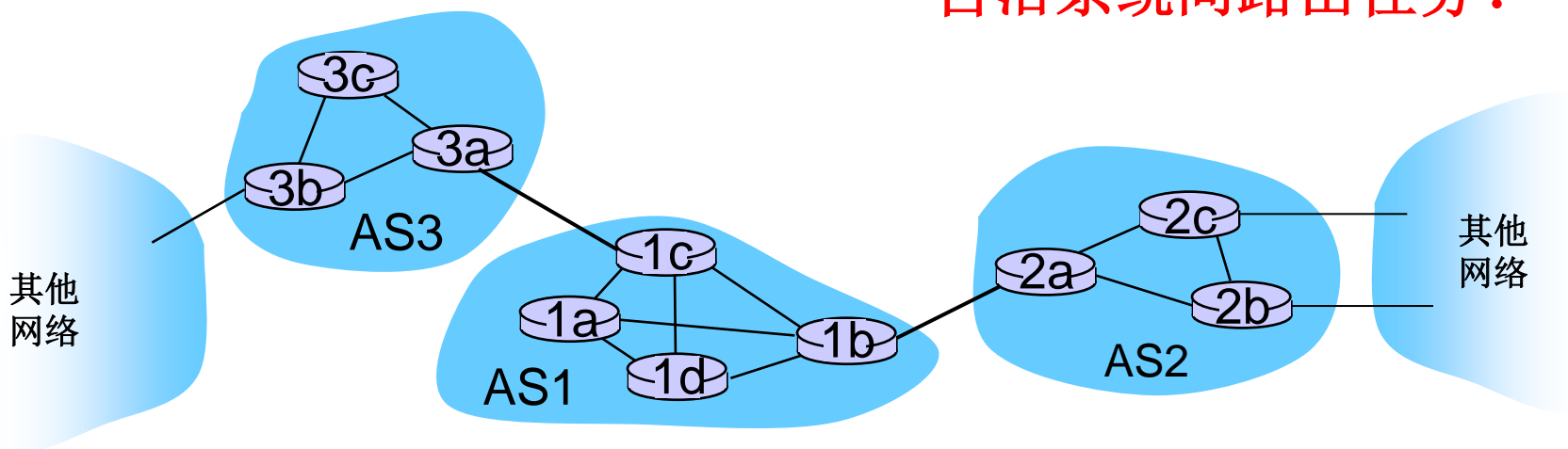
自治系统间(Internet-AS)路由任务

- ❖ 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:

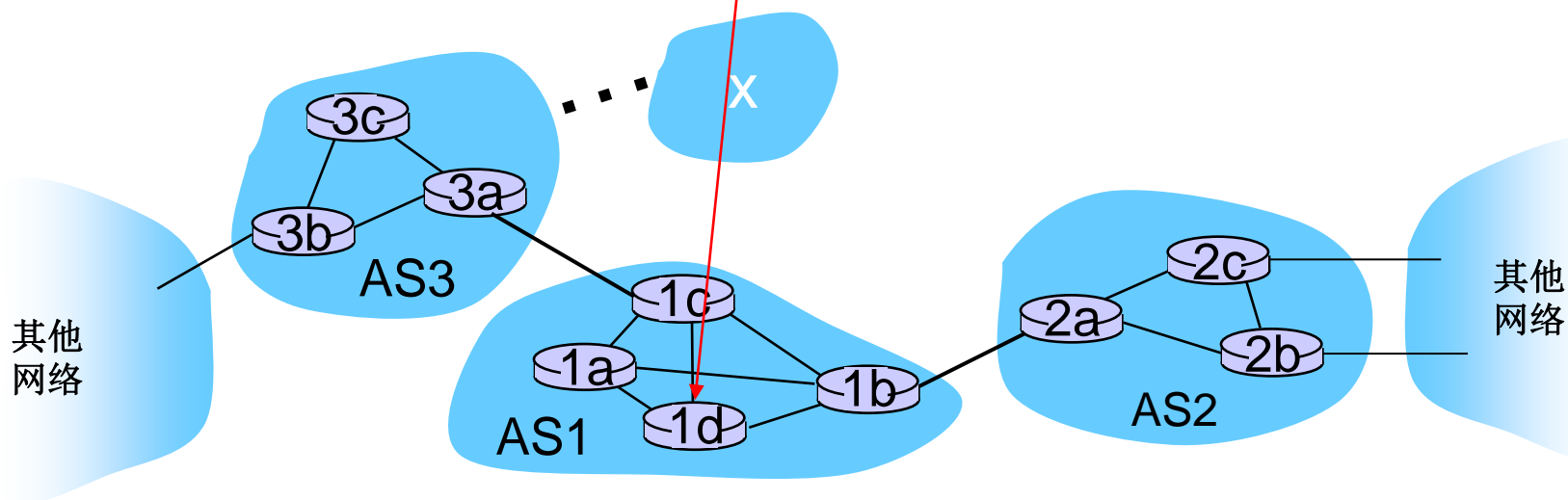
1. 学习到哪些目的网络可以通过AS2到达，哪些可以通过AS3到达
2. 将这些网络可达性信息传播给AS1内部路由器

自治系统间路由任务！



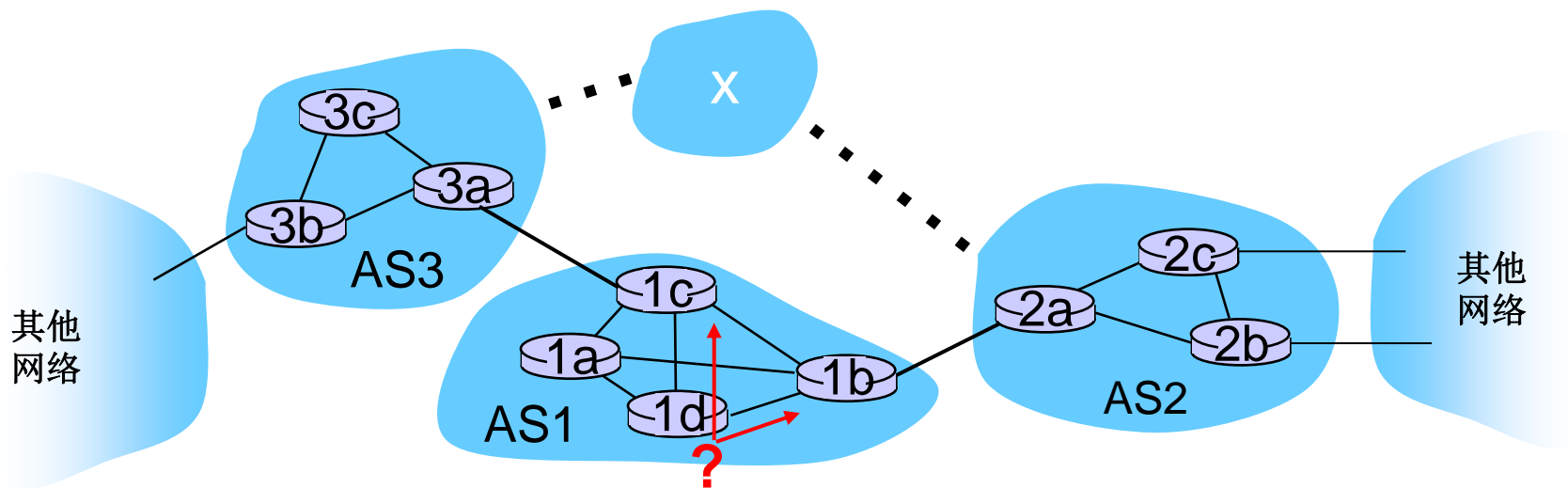
例：路由器1d的转发表设置

- ❖ 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传播该可达性信息
- ❖ 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！
 - 在转发表中增加入口：(x, 1)



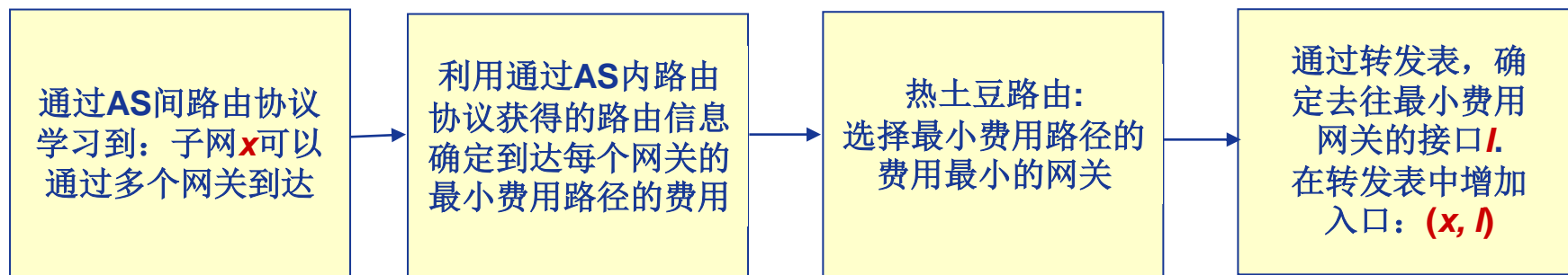
例：在多AS间选择

- ❖ 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！



例：在多AS间选择

- ❖ 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。



本讲主题

RIP协议简介

AS内部路由

- ❖ Internet采用层次路由
- ❖ AS内部路由协议也称为内部网络协议**IGP** (interior gateway protocols)
- ❖ 最常见的AS内部路由协议:
 - 路由信息协议: RIP(Routing Information Protocol)
 - 开放最短路径优先: OSPF(Open Shortest Path First)
 - 内部网关路由协议: IGRP(Interior Gateway Routing Protocol)
 - Cisco私有协议

本讲主题

OSPF协议简介

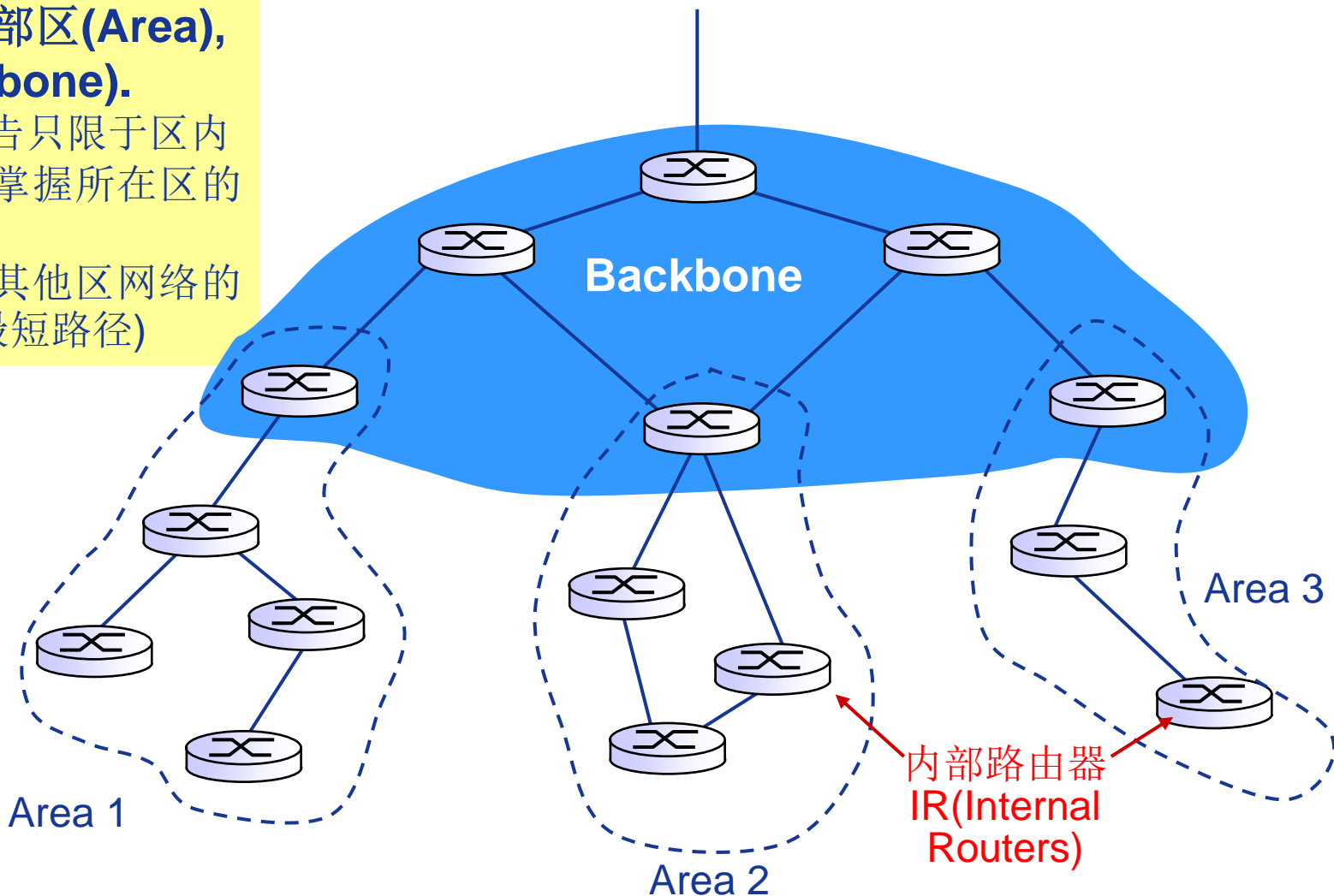
OSPF (Open Shortest Path First)

- ❖ “开放”：公众可用
- ❖ 采用链路状态路由算法
 - LS分组扩散（通告）
 - 每个路由器构造完整的网络(AS)拓扑图
 - 利用Dijkstra算法计算路由
- ❖ OSPF通告中每个入口对应一个邻居
- ❖ OSPF通告在**整个AS**范围泛洪
 - OSPF报文直接封装到**IP**数据报中
- ❖ 与OSPF极其相似的一个路由协议：**IS-IS路由协议**

分层的OSPF

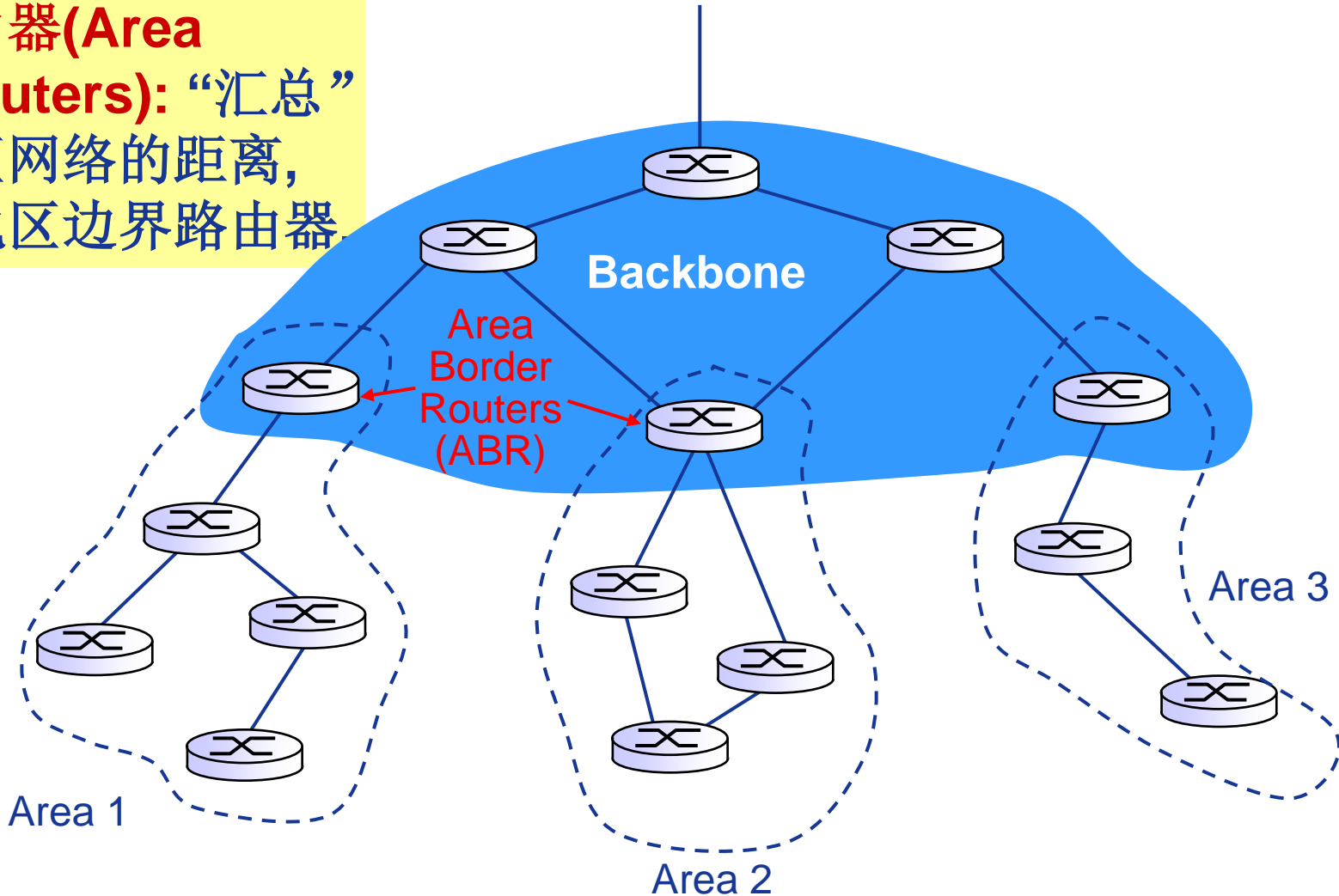
两级分层: 局部区(Area), 主干区(Backbone).

- 链路状态通告只限于区内
- 每个路由器掌握所在区的详细拓扑
- 只知道去往其他区网络的“方向” (最短路径)



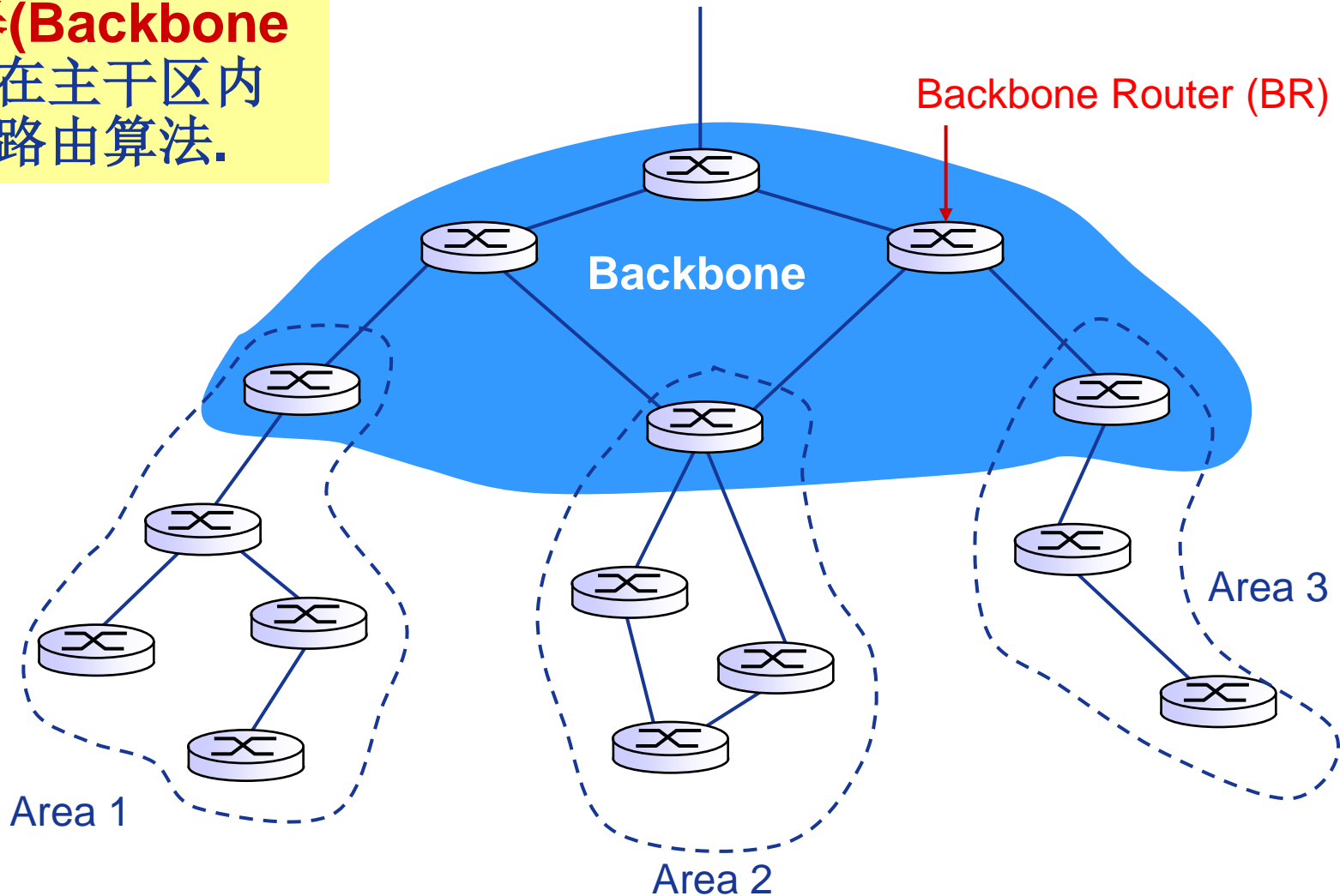
分层的OSPF

区边界路由器(**Area Border Routers**): “汇总”到达所在区网络的距离, 通告给其他区边界路由器.



分层的OSPF

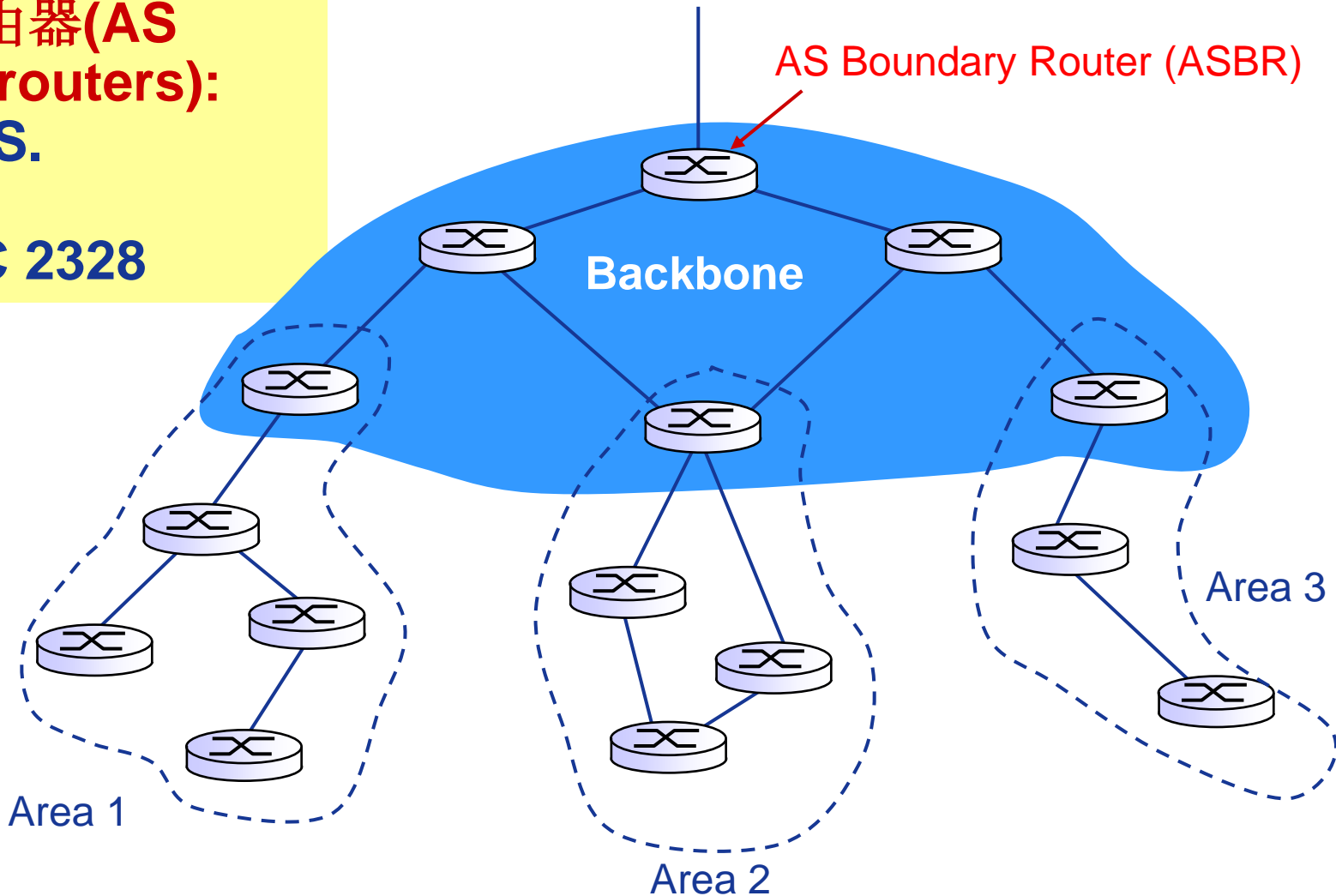
主干路由器(**Backbone Routers**): 在主干区内运行**OSPF**路由算法.



分层的OSPF

AS边界路由器(AS boundary routers):
连接其他AS.

参考: RFC 2328



本讲主题

BGP协议简介（1）

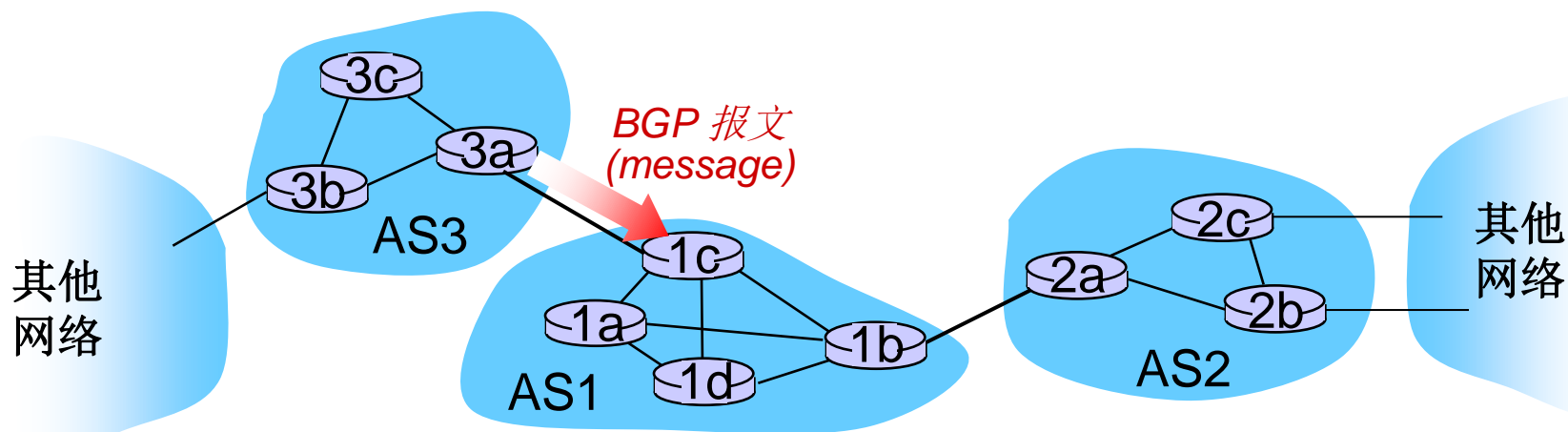
Internet AS间路由协议: BGP

- ❖ 边界网关协议**BGP (Border Gateway Protocol)**: 事实上的标准域间路由协议
 - 将Internet “粘合” 为一个整体的关键
- ❖ **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.
 - 基于可达性信息与策略, 确定到达其他网络的 “好” 路径.
- ❖ 容许子网向Internet其余部分通告它的存在:
“我在这儿!”

BGP基础

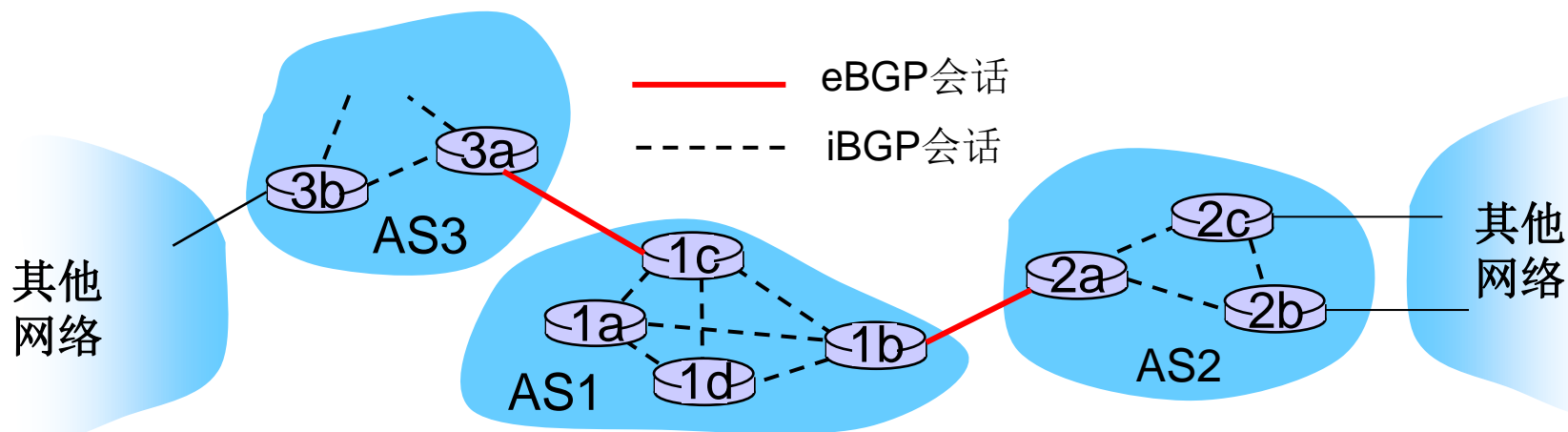
❖ 当AS3通告一个前缀给AS1时:

- AS3**承诺**可以将数据报转发给该子网
- AS3在通告中会**聚合**网络前缀



BGP基础: 分发路径信息

- ❖ 在3a与1c之间, AS3利用eBGP会话向AS1发送前缀可达性信息.
 - 1c则可以利用iBGP向AS1内的所有路由器分发新的前缀可达性信息
 - 1b可以（也可能不）进一步通过1b-到-2a的eBGP会话，向AS2通告新的可达性信息
- ❖ 当路由器获得新的前缀可达性时，即在其转发表中增加关于该前缀的入口（路由项）。



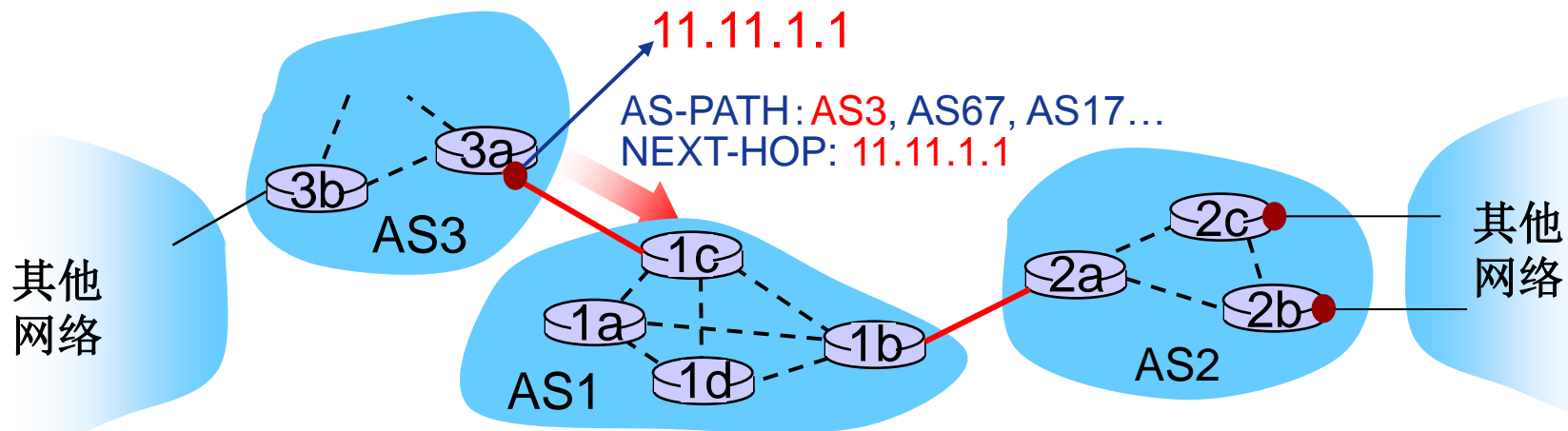
路径属性与BGP路由（route）

❖ 通告的前缀信息包括BGP属性

- 前缀+属性= “路由”

❖ 两个重要属性:

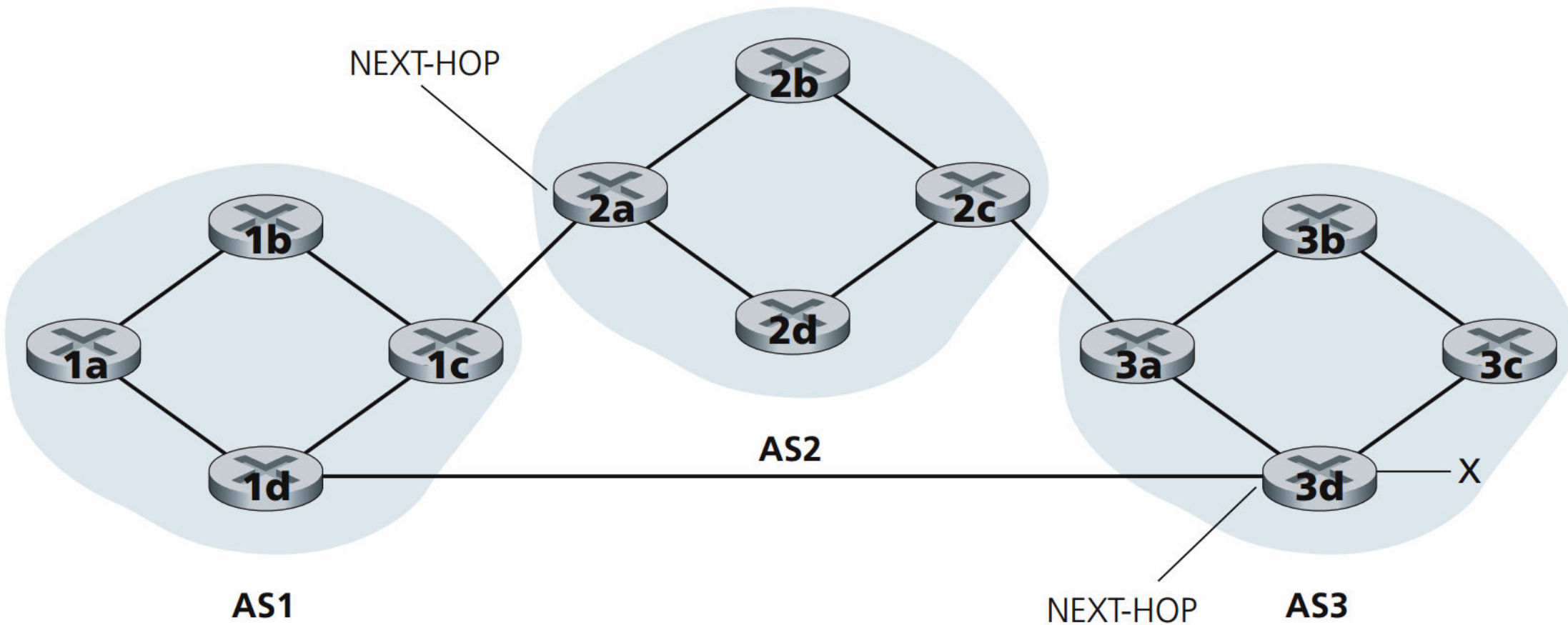
- **AS-PATH(AS路径):** 包含前缀通告所经过的AS序列: e.g., AS 67, AS 17
- **NEXT-HOP(下一跳):** 开始一个AS-PATH的路由器接口, 指向下一跳AS.
 - 可能从当前AS到下一跳AS存在多条链路



hot potato routing:
least cost to NEXT-HOP

5.4

ROUTING AMONG THE ISPS: BGP
1b → x } 2a 2 ✓
3d: 3



BGP routes {
NEXT-HOP
 IP address of leftmost interface for router 2a; AS2 AS3; x
 IP address of leftmost interface of router 3d; AS3; x
AS-PATH destination prefix

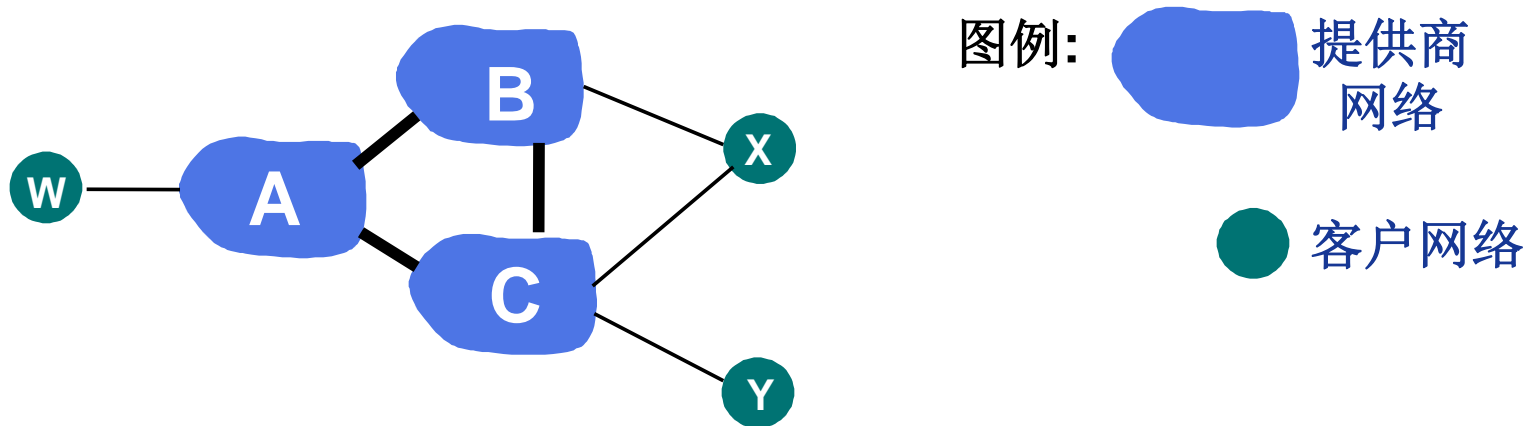
本讲主题

BGP协议简介（2）

BGP路由选择

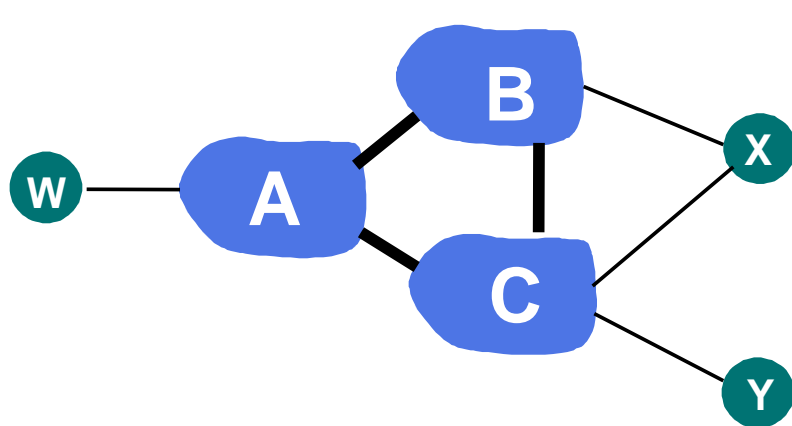
- ❖ 网关路由器收到路由通告后，利用其输入策略(import policy)决策接受/拒绝该路由
 - e.g., 从不将流量路由到AS x
 - 基于策略(policy-based) 路由
- ❖ 路由器可能获知到达某目的AS的多条路由，基于以下准则选择：
 1. 本地偏好(preference)值属性: 策略决策(policy decision)
 2. 最短AS-PATH
 3. 最近NEXT-HOP路由器: 热土豆路由(hot potato routing)
 4. 附加准则

BGP路由选择策略



- ❖ 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是双宿网络(dual-homed network/AS): 连接两个其他AS
 - X不期望经过他路由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

BGP路由选择策略



图例：



提供商
网络



客户网络

- ❖ 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！

为什么采用不同的AS内与AS间路由协议？

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