

# 第五章 网络层（2）

---

袁华: [hyuan@scut.edu.cn](mailto:hyuan@scut.edu.cn)

华南理工大学计算机科学与工程学院

广东省计算机网络重点实验室（CCNL）

# 本节的主要内容 (5.2.4节, P285)

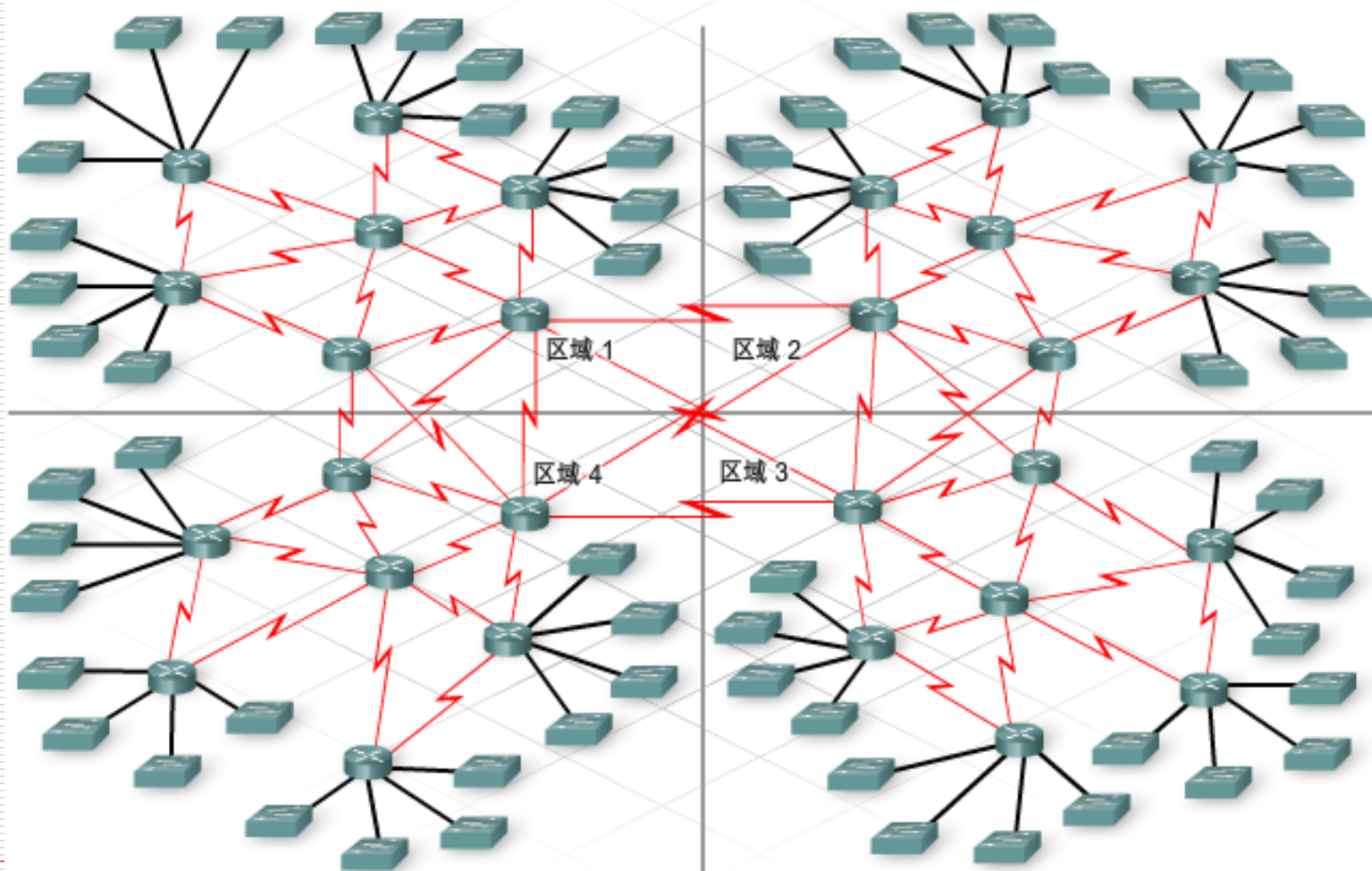
---

- 动态路由选择算法
- 距离矢量路由
  - DV实例: Rip
- DV问题及解决办法



# 为什么需要路由选择协议？

试想一下为这样一个网络维护静态路由配置！



# 动态路由算法

---

## □ 距离矢量路由 (DV)

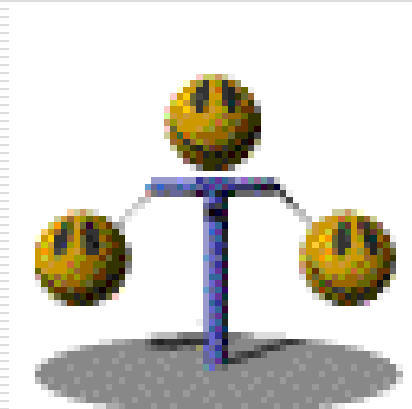
■ 实例: RIP

## □ 链路状态路由 (LS)

■ 实例: OSPF

## □ 混合路由 (Hybrid)

■ 实例: EIGRP



# 动态性如何实现？

---

## □ 天王盖地虎——宝塔镇河妖

- 路由器之间要通信，他们必须说着相同的语言，  
即运行同一种路由选择协议
- 

## □ 初次见面，请多关照

- 一台新的路由器必须主动介绍它自己 (say hello)
- 

## □ 好久不见，近况如何？

- 周期性地联系，以了解其它路由器的健康状况  
(keep alive)

# 距离矢量路由选择 P285

- 距离矢量路由选择：每个路由器维护一张表，表中列出了当前已知的到每个目标的最佳距离，以及为了到达那个目标，应该从哪个目标转发的线路（端口）
- D-V 算法是动态的和分布式的，它常被用于小型网络，RIP是一个典型的 DV
  - RIP: Routing information protocol, 路由选择信息协议，1988，RFC1058

# DV的工作原理 P286

- 每个路由器（节点）**维护**两个向量， $D_i$ 和 $S_i$ ，分别表示从该路由器到所有其它路由器的距离及相应的下一跳（next hop）
- 在邻居路由器之间**交换**路由信息（矢量）
- 每个路由器（节点）根据收到的矢量信息，**更新**自己的路由表

# DV工作原理（续）

- $d_{i1}$ : 从节点i到节点1的度量（代价）
- $s_{i1}$ : 沿着从节点i到节点1的最优路径上的下一跳
- $n$ : 网络中的节点数

$$D_i = \begin{bmatrix} d_{i1} \\ d_{i2} \\ d_{i3} \\ \dots \\ d_{in} \end{bmatrix}$$

$$S_i = \begin{bmatrix} s_{i1} \\ s_{i2} \\ s_{i3} \\ \dots \\ s_{in} \end{bmatrix}$$



# 更新路由表

□ 当邻居间交换了矢量信息之后:

■ 更新距离:  $d_{ij} = \text{Min}[d_{ix} + d_{xj}] \quad (x \in A)$

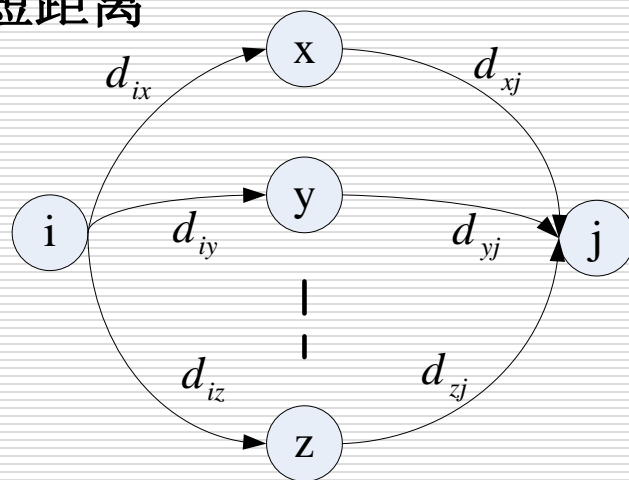
□ A—节点i的邻居集合

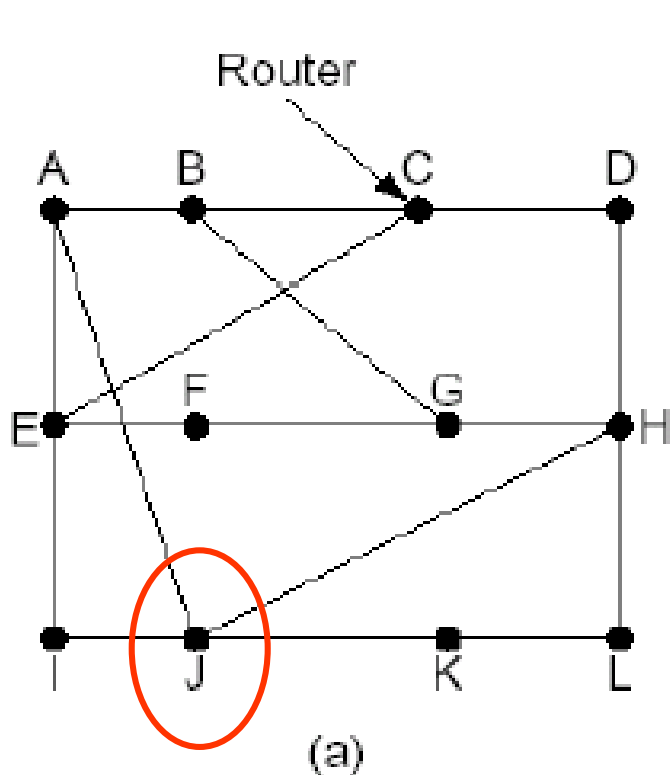
□  $d_{ij}$ —从节点i到节点j的最短距离

□  $d_{ix}$ —从节点i到节点x的最短距离

□  $d_{xj}$ —从节点x到节点j的最短距离

■ 更新下一跳:  $S_{ij} = x$





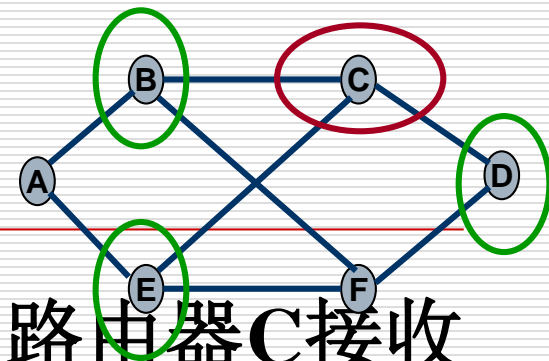
P286, 图5-9

						New estimated delay from J	
To	A	I	H	K			Line
A	0	24	20	21		8	A
B	12	36	31	28		20	A
C	25	18	19	36		28	I
D	40	27	8	24		20	H
E	14	7	30	22		17	I
F	23	20	19	40		30	I
G	10	31	6	31		18	H
H	17	20	0	19		12	H
I	21	0	14	22		10	I
J	0	11	7	10		0	-
K	24	22	22	0		6	K
L	29	33	9	9		15	K
						New routing table for J	

JA delay is 8      JI delay is 10      JH delay is 12      JK delay is 6

Vectors received from J's four neighbors

# 课堂练习



- 一个网络拓扑如下，某个时刻，路由器C接收到三个邻居发过来的矢量如下：

A    B    C    D    E    F

From router B: ( 5,   0, 8, 12, 6,   2)

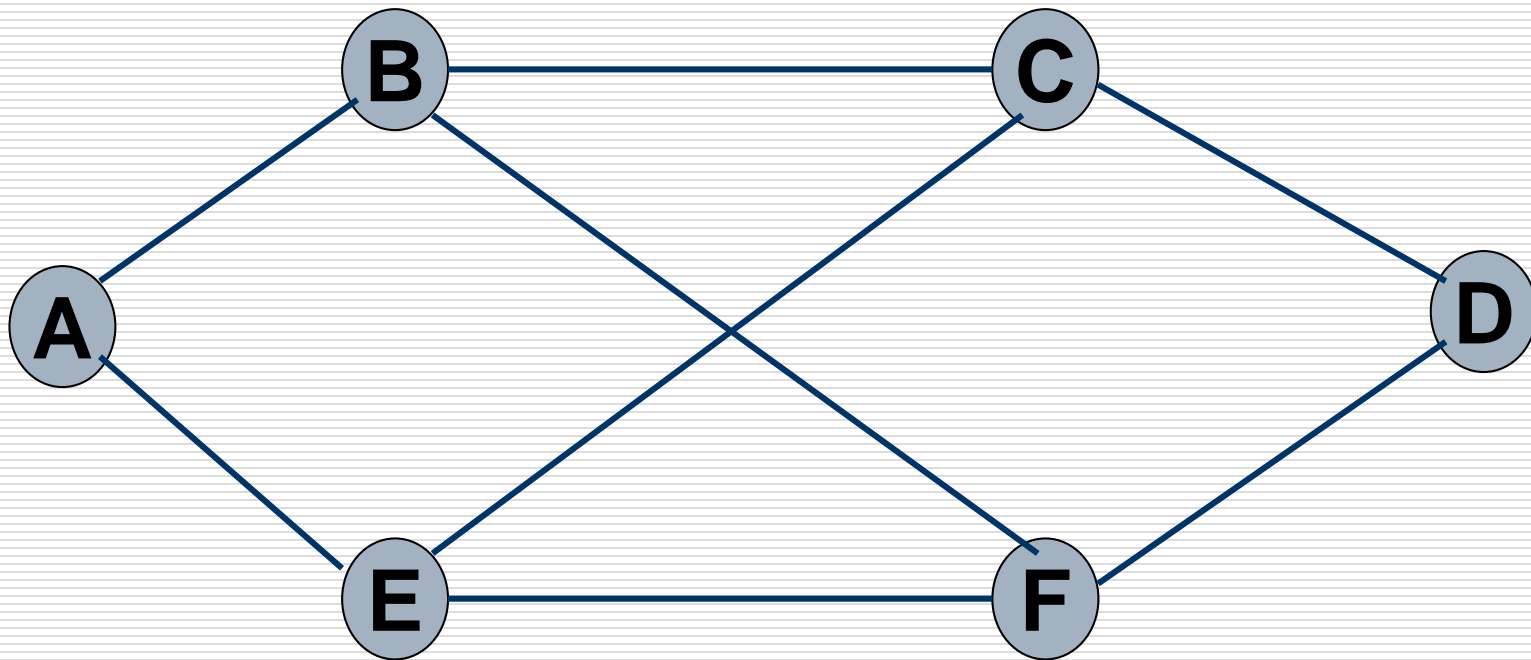
From router D: (16, 12, 6,   0, 9, 10)

From router E: ( 7,   6, 3,   9, 0,   4)

- 现在，路由器C到 B、D 和 E 的代价分别是 6、3 和 5，试回答路由器C更新后的路由表

# 网络拓扑

---



# 参考答案

- 分析：通过路由器 B、D 和 E 进行转发的话,路由器C的矢量分别是：

Via router B: (11, 6, 14, 18, 12, 8)

Via router D: (19, 15, 9, 3, 12, 13)

Via router E: (12, 11, 8, 14, 5, 9)

所以，路由器C更新后的路由表如下：

(11, 6, 0, 3, 5, 8)

( B, B, -, D, E, B)

# D-V算法的特点

## □ 优点

- 简单



## □ 缺点

- 交换的信息太大了
- 路由信息传播慢，可能导致路径信息不一致
- 收敛慢，度量计数到无穷
- 不适合大型的网络

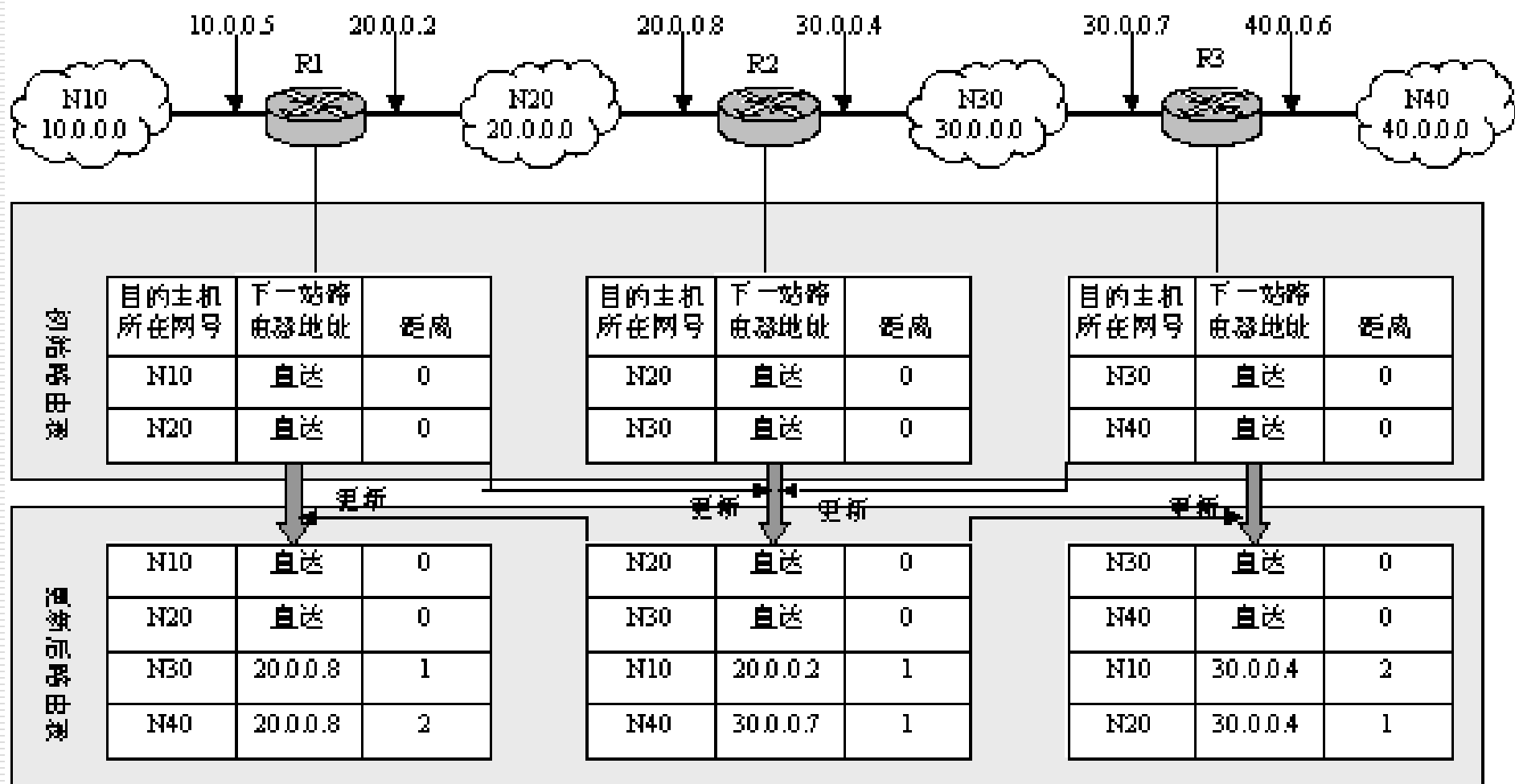


# RIP的主要特点

---

- RIP 是一种典型的 D-V 路由选择协议
- RIP 采用了跳数（hop）作为量度（metric）
- 当量度超过 15 跳，目的被认为不可达
- 默认地，每30秒钟交换一次矢量/向量信息（全部路由表）

# RIP的工作原理

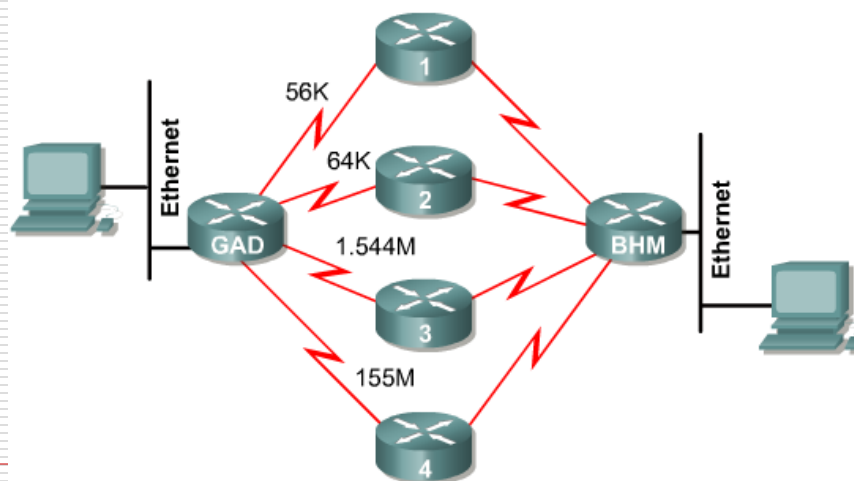




# RIP的主要缺陷 P286



- ❑ 不能到达量度超过15跳的目标网络
- ❑ RIP的度量（代价）是跳数，即沿途经过的路由器的个数，有时候，并不合理，不能真正反映网络的状况
- ❑ 实际运行中，会遇到度量计数到无穷、收敛慢等问题



# DV路由可能遇到的问题

---

## □ 问题表现

- 路由环路（**routing loop**）
- 计数到无穷问题（**Count to infinite**）
- 收敛慢的问题（**slow Convergence**）

## □ 原因

- 相信错误的路由信息导致

# DV路由的主要问题 P286

□ 好消息跑得快，坏消息传得慢(It reacts rapidly to good news, but leisurely to bad news) (好事传千里，坏事不出门)

A	B	C	D	E	
•	•	•	•	•	Initially
	1	•	•	•	After 1 exchange
	1	2	•	•	After 2 exchanges
	1	2	3	•	After 3 exchanges
	1	2	3	4	After 4 exchanges

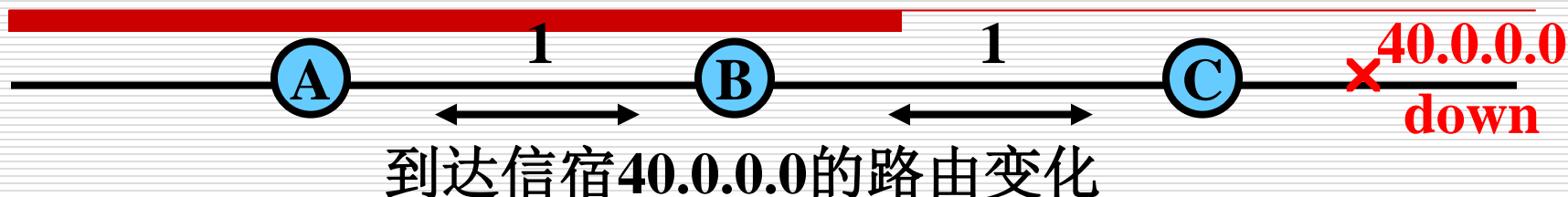
(a)

A	B	C	D	E	
•	1	2	3	4	Initially
	3	2	3	4	After 1 exchange
	3	4	3	4	After 2 exchanges
	5	4	5	4	After 3 exchanges
	5	6	5	6	After 4 exchanges
	7	6	7	6	After 5 exchanges
	7	8	7	8	After 6 exchanges
	•	•	•	•	⋮

(b)

计数到无穷

# 错误路由消息的传播



时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	2	B→C, 1+1=2

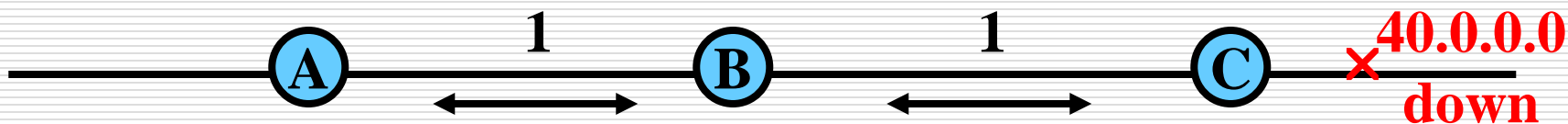
C与B之间的对话:

C 我得不到信宿40.0.0.0的任何路由信息, 你能告诉我如何到达信宿吗?

B 我可以到达信宿, 距离为1。 (传播了一条过时的错误信息)

C 既然如此, 我选择经过你到达信宿的路径, 距离为2。

# 路由环

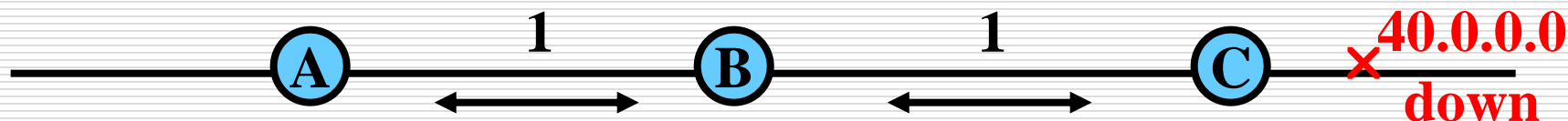


到达信宿40.0.0.0的路由变化

时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	2	B→C, 1+1=2
第 1 步	2	3	2	C→B, 2+1=3
第 2 步	4	3	4	B→C, 3+1=4 B→A, 3+1=4
...				

这条错误的路由信息在C与B之间不断复制和修改，并在网络中传播（殃及A），形成路径传播的环路。

# 计数到无穷



到达信宿40.0.0.0的路由变化

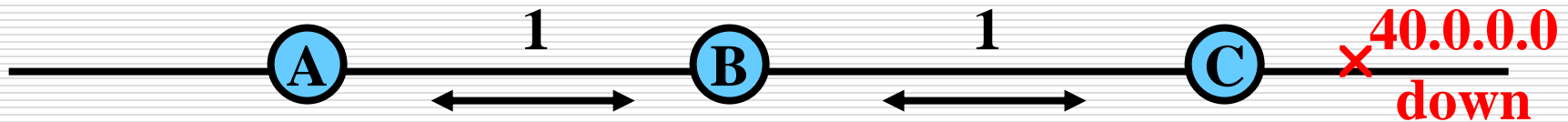
时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	2	B→C, 1+1=2
第 1 步	2	3	2	C→B, 2+1=3
第 2 步	4	3	4	B→A, B→C, 3+1=4
第 3 步	4	5	4	C→B, 4+1=5
...				
第 13 步	14	15	14	C→B, 14+1=15
第 14 步	16	15	16	B→A, B→C, 15+1=16
...				

# 解决办法

---

- 定义路径度量（代价）的最大值
- 提高收敛速度
  - 水平分割（**Split Horizon**）
  - 毒性逆转（**Poison Reverse**）
  - 抑制定时器（**Hold-Down Timers**）
  - 触发更新（**Triggered Updates**）

# 定义路径代价的最大值



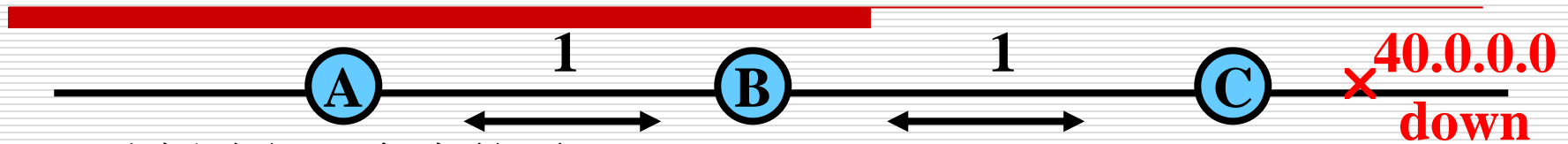
到达信宿40.0.0.0的路由变化（定义Hop最大值为16）

时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	2	B→C, 1+1=2
第 1 步	2	3	2	C→B, 2+1=3
第 2 步	4	3	4	B→A, B→C, 3+1=4
第 3 步	4	5	4	C→B, 4+1=5
...				
第 13 步	14	15	14	C→B, 14+1=15
第 14 步	16	15	16	B→A, B→C, 15+1=16
第 15 步	不可达	16	不可达	C→B, 15+1=16
第 16 步		不可达		抛弃

收敛!



# 水平分割



## □ 分析路径环产生的原因

- B向C提供了一条过时的、错误的路由信息。

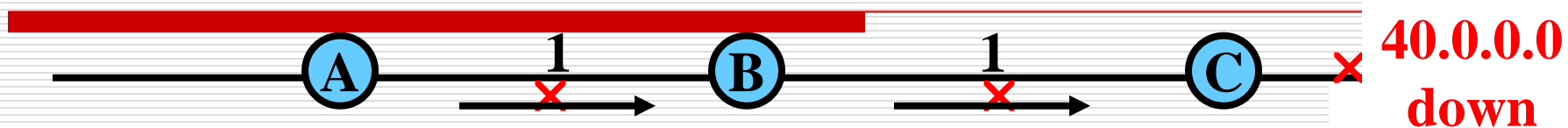
## □ 能否避免事件发生？

- B必须经由C方可到达网络40.0.0.0，B不可能向C提供任何有价值的路由信息。
- 修改B对C提供的路由，禁止B向C提供关于此信宿的路由信息。

## □ 解决办法

- B告诉C一条在正常情况下不真实的消息：网络40.0.0.0不可达（距离为 $\infty$ ）。

# 水平分割如何加快收敛？



链路断开时C与B之间的对话：

C 我得不到信宿40.0.0.0的任何路由信息，你能告诉我如何到达信宿吗？

B 我不能到达信宿，距离为 $\infty$ 。

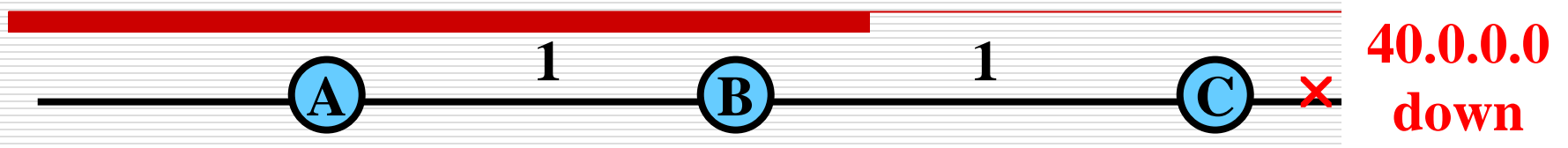
C 既然如此，我认为信宿不可达。

到达信宿40.0.0.0的路由变化

收敛！

时间	A	B	C	刷新
初始	2	1/ $\infty$	0	信宿可达
40.0.0.0 断开	2	1/ $\infty$	$\infty$	B→C, $\infty$ (虚假)
第 1 步	2	$\infty$	$\infty$	C→B, $\infty$
第 2 步	$\infty$	$\infty$	$\infty$	B→A, $\infty$

# 毒性逆转



## □ 方法

- 当C发现网络40.0.0.0发生故障时，主动将到达信宿的距离改为 $\infty$ 。

## □ 结果

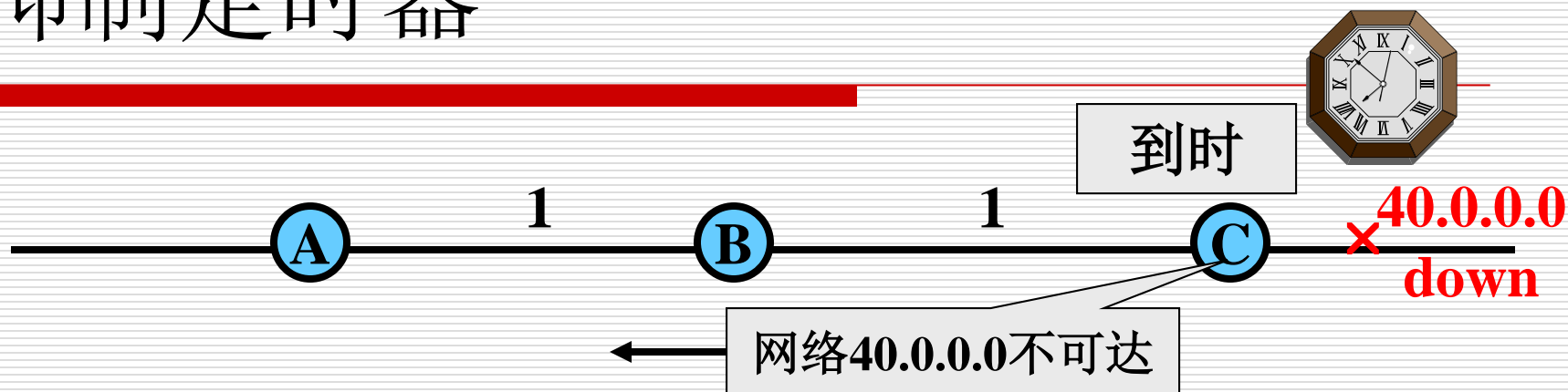
- 如果无其他到达信宿的路径，算法迅速收敛为信宿不可达。
- 如果存在其他到达信宿的路径，C根据传播过来的信息再做修改。

到达信宿40.0.0.0的路由变化

收敛!

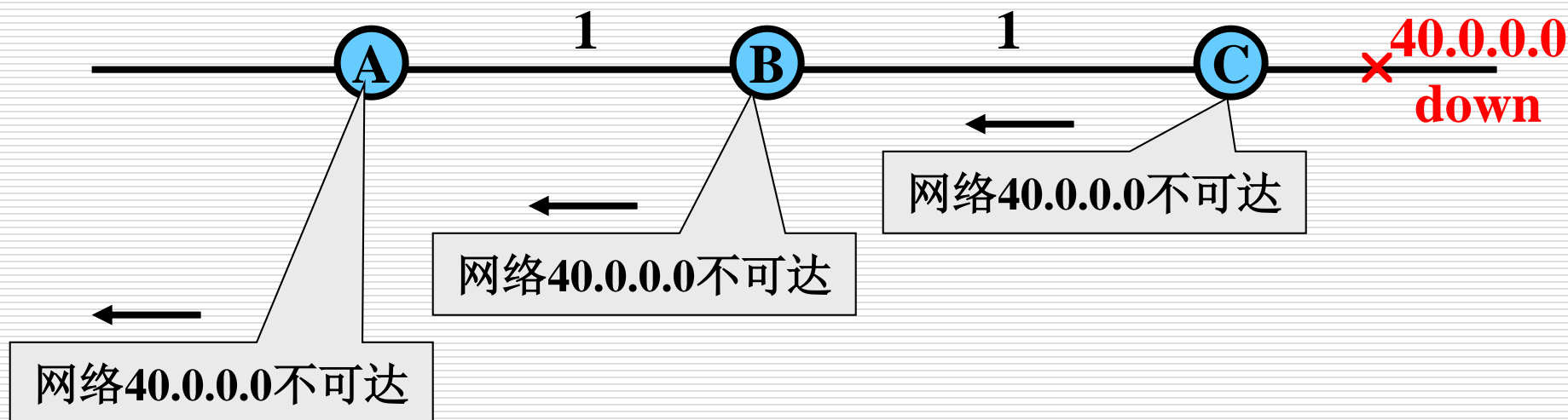
时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	$\infty$	C 主动改距离为 $\infty$
第 1 步	2	$\infty$	$\infty$	C $\rightarrow$ B, $\infty$
第 2 步	$\infty$	$\infty$	$\infty$	B $\rightarrow$ A, $\infty$

# 抑制定时器



- 当C发现网络40.0.0.0发生故障时，启动抑制计时器
- 在抑制计时期间内，C的策略
  - 如果网络状态转变，down  $\Rightarrow$  up，关闭计时器，保留原有路由信息；
  - 如果收到来自B的关于信宿的路由信息，且路径比原有路径短，则关闭计时器，更新路由信息；
  - 如果无上述两种情况发生，计时器到时，更新路由为信宿不可达。

# 触发更新



- 当C发现网络40.0.0.0发生故障时，不等下一刷新周期到来，立刻更改路由为“信宿不可达”
- 引起全网的连锁反映，迅速刷新

# 路由面临的复杂情况

- ❑ **站得高才能看得远**，确定全局最佳路径，但是站得高需要付出代价。
- ❑ 途经线路、站点以及目的网络都是动态变化的，最佳路径也要跟随发生变化，需要及时获取状态变化信息。
- ❑ 在站得不够高、跟得不够紧的情况下，只能直接获取近邻信息，远处信息通过逐站信息传播而间接获取，有可能传播、学习到错误的、过时的信息。
- ❑ 最坏情况，全网传播和学习过时的信息，永远无法达到稳定状态：算法不收敛。

# 本节小结

---

## □ DV-路由算法的工作原理

### ■ 典型实例：RIP

## □ DV运行中的问题

### ■ 解决的办法

- 水平分割
- 毒性逆转
- 抑制定时器
- 触发更新



---

# Thank you all!

