

第五章 网络层

RIP为什么会衰败

DV路由可能遇到的问题

路由环路（routing loop）

计数到无穷问题（Count to infinite）

收敛慢的问题（slow Convergence）



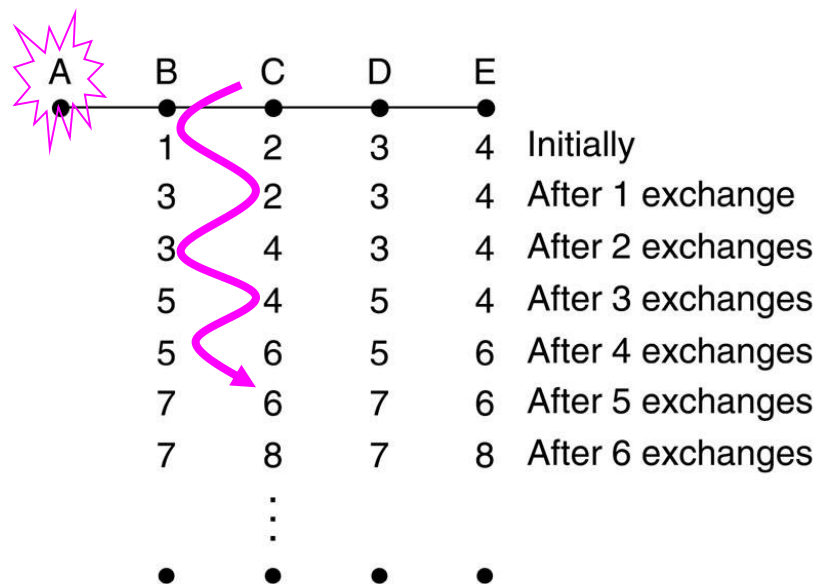
相信错误的路由
信息导致

好消息跑得快，坏消息传得慢

(It reacts rapidly to good news, but leisurely to bad news)



(a)

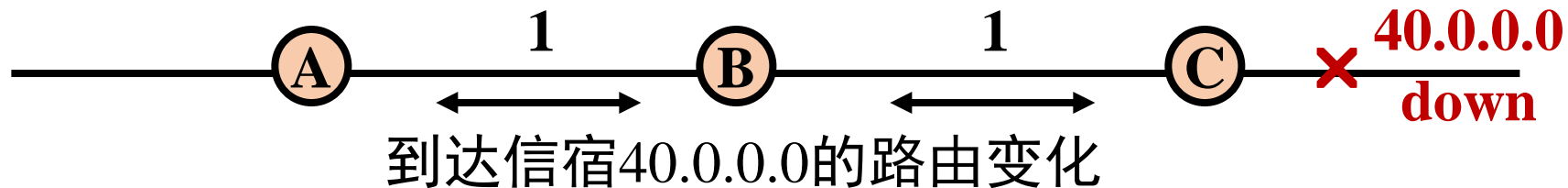


(b)

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

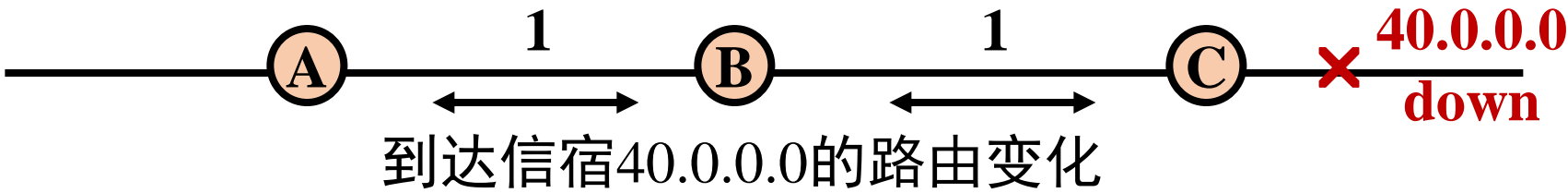
C与B之间的对话：

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。

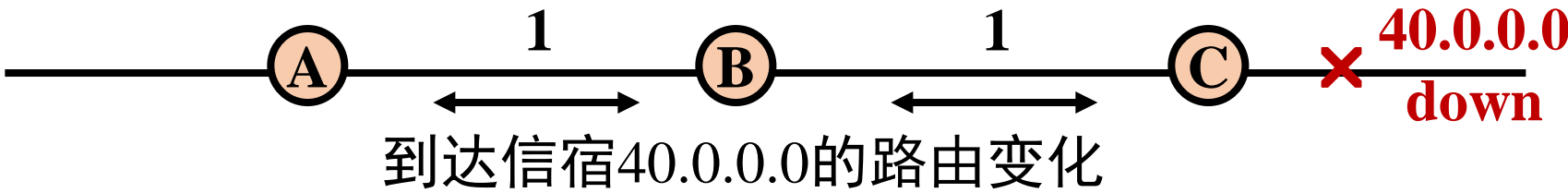
路由环



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



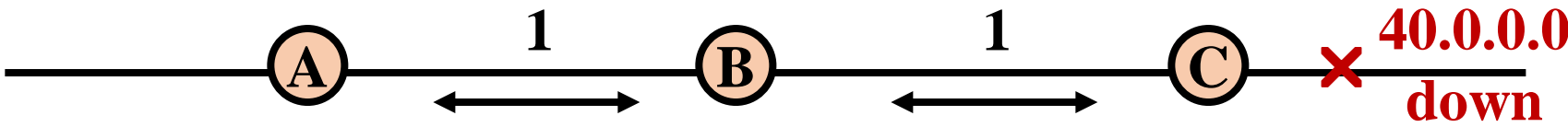
时间	A	B	C	刷新
初始	2	1	0	信宿可达
40.0.0.0 断开	2	1	2	B→C, 1+1=2
第 1 步	2	3	2	C→B, 2+1=3
第 2 步	4	3	4	B→A, B→C, 3+1=4
第 3 步	4	5	4	C→B, 4+1=5
...				
第 13 步	14	15	14	C→B, 14+1=15
第 14 步	16	15	16	B→A, B→C, 15+1=16
...				



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）

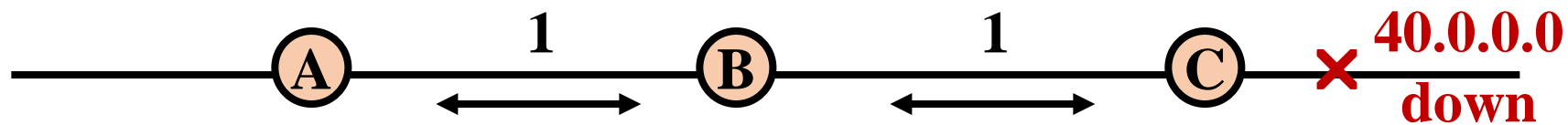
定义路径代价的最大值



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		扔弃

收敛!

水平分割



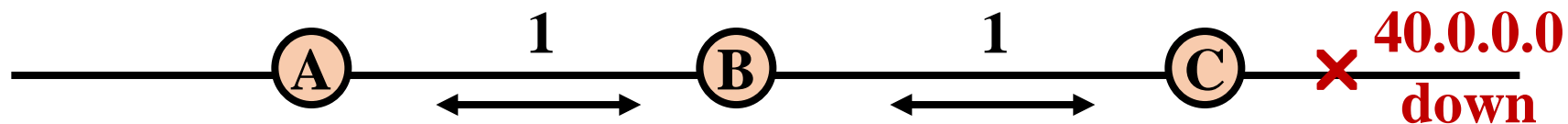
分析路径环产生的原因

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。

水平分割

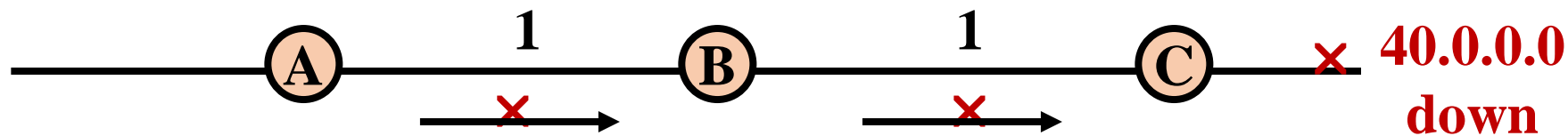


解决办法

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 ∞ ）。



水平分割如何加快收敛？



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/∞	∞	B→C, ∞ (虚假)
第 1 步	2	∞	∞	C→B, ∞
第 2 步	∞	∞	∞	B→A, ∞

链路断开时C与B之间的对话：

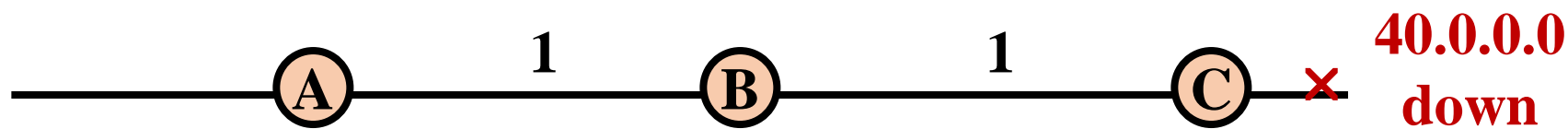
C 我得不到信宿40.0.0.0的任何路由信息，你能告诉我如何到达信宿吗？

B 我不能到达信宿，距离为 ∞ 。

C 既然如此，我认为信宿不可达。



毒性逆转



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	∞	C 主动改距离为 ∞
第 1 步	2	∞	∞	C \rightarrow B, ∞
第 2 步	∞	∞	∞	B \rightarrow A, ∞

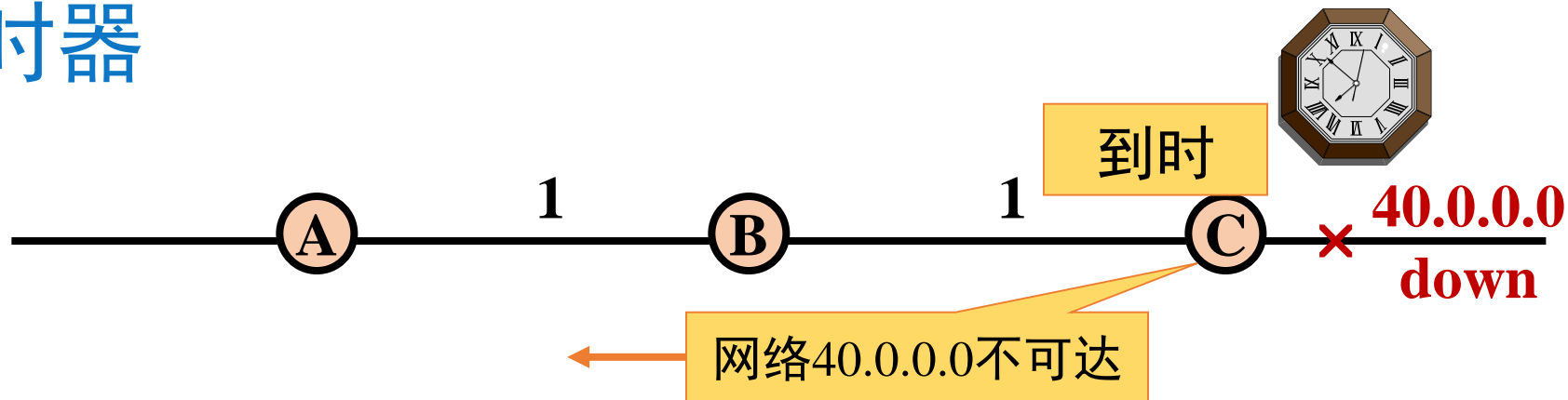
□ 方法

- 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 ∞ 。

□ 结果

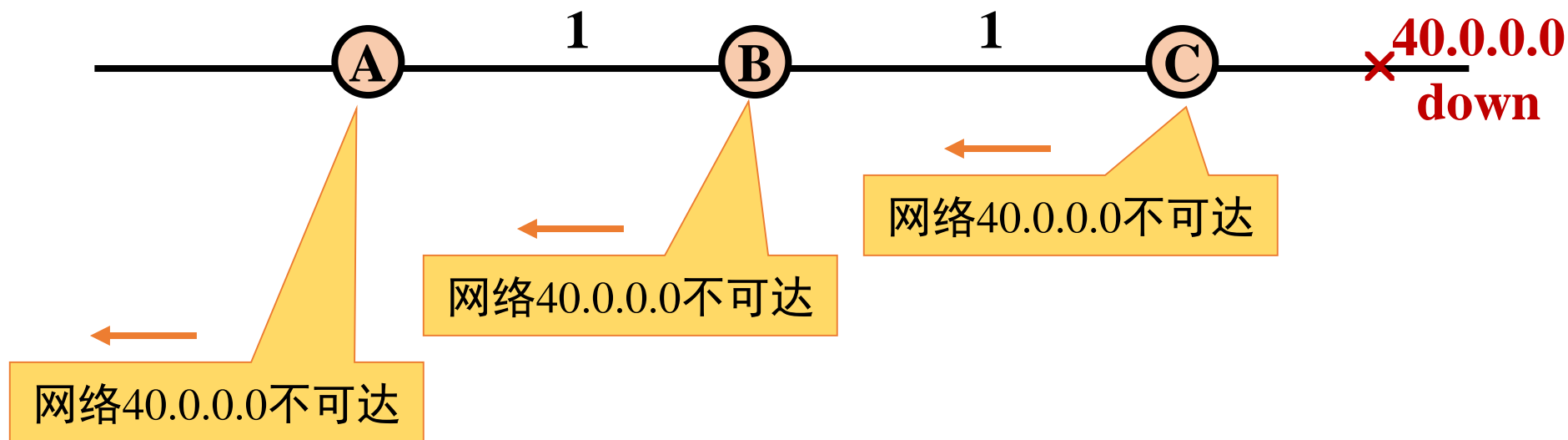
- 如果无其他到达信宿的路径，算法迅速收敛为信宿不可达。
- 如果存在其他到达信宿的路径，C根据传播过来的信息再做修改。

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。



小结

- DV、RIP的主要问题是计算到无穷，产生路由环
- 解决的方法：定义路径代价的最大数
 - 水平分割
 - 毒性逆转
 - 抑制定时器
 - 触发更新

思考题

- DV的主要问题是什么？
- 定义一个路径代价的最大数产生了什么后果？
- 可以采用哪些方法来加快缓解路由环等问题？
- RIP会彻底消失吗？

谢谢观看

致谢

本课程课件中的部分素材来自于：（1）清华大学出版社出版的翻译教材《计算机网络》（原著作者：Andrew S. Tanenbaum, David J. Wetherall）；（2）思科网络技术学院教程；（3）网络上搜到的其他资料。在此，对清华大学出版社、思科网络技术学院、人民邮电出版社、以及其它提供本课程引用资料的个人表示衷心的感谢！

对于本课程引用的素材，仅用于课程学习，如有任何问题，请与我们联系！