距离矢量路由算法

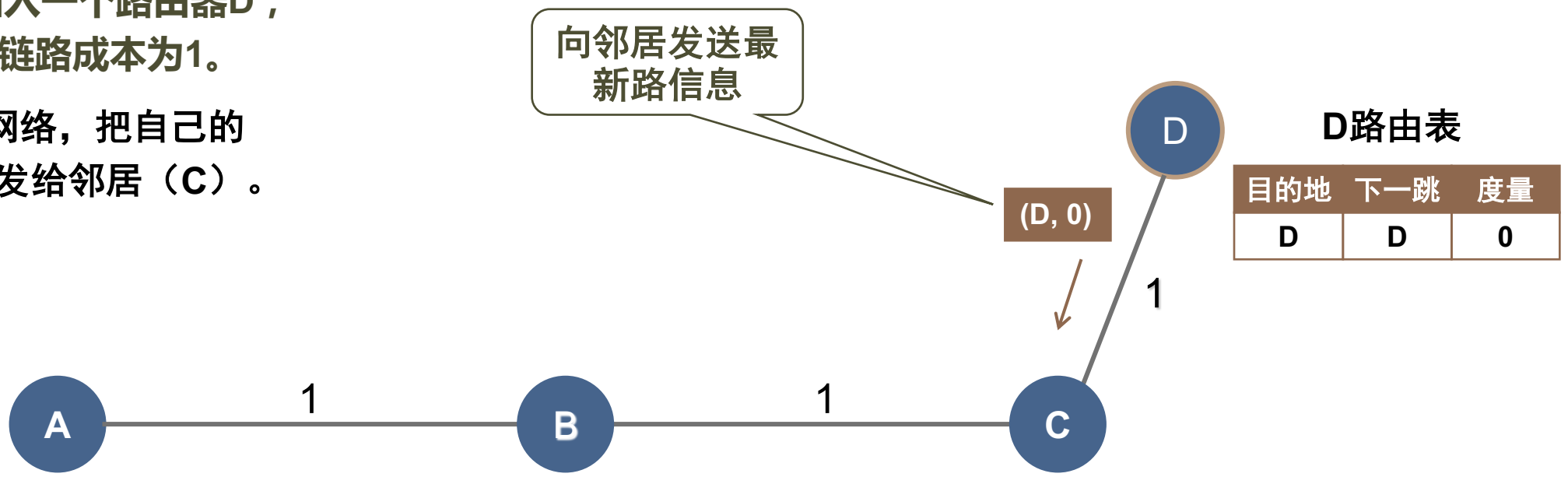
VS.

算法特点

距离矢量算法优点

- t3：网络新加入一个路由器D，与C是邻居，链路成本为1。

① D刚加入网络，把自己的路由信息发给邻居（C）。



D路由表

目的地	下一跳	度量
D	D	0

A路由表

目的地	下一跳	度量
A	A	0
B	B	1
C	B	2

B路由表

目的地	下一跳	度量
A	A	1
B	B	0
C	C	1

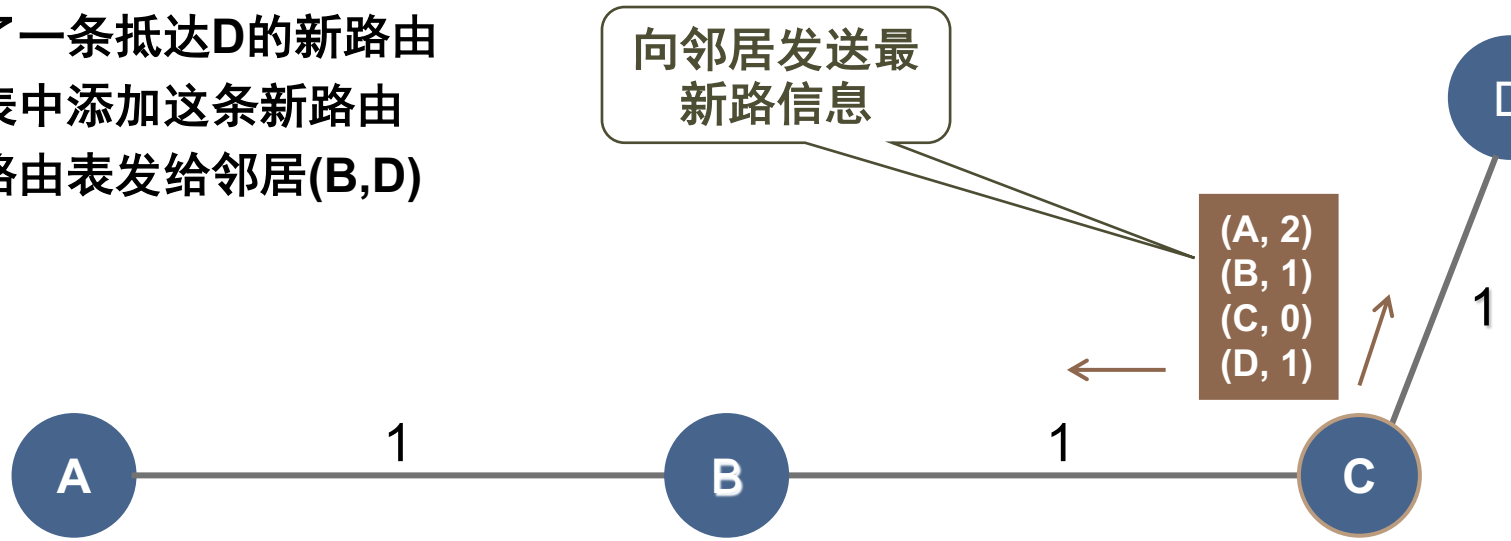
C路由表

目的地	下一跳	度量
A	B	2
B	B	1
C	C	0

路由发生变化的节点

距离矢量算法优点

- t4：C收到D的路由信息
  - ① C发现多了一条抵达D的新路由
  - ② C在路由表中添加这条新路由
  - ③ C把新的路由表发给邻居(B,D)



D路由表

目的地	下一跳	度量
D	D	0

A路由表

目的地	下一跳	度量
A	A	0
B	B	1
C	B	2

B路由表

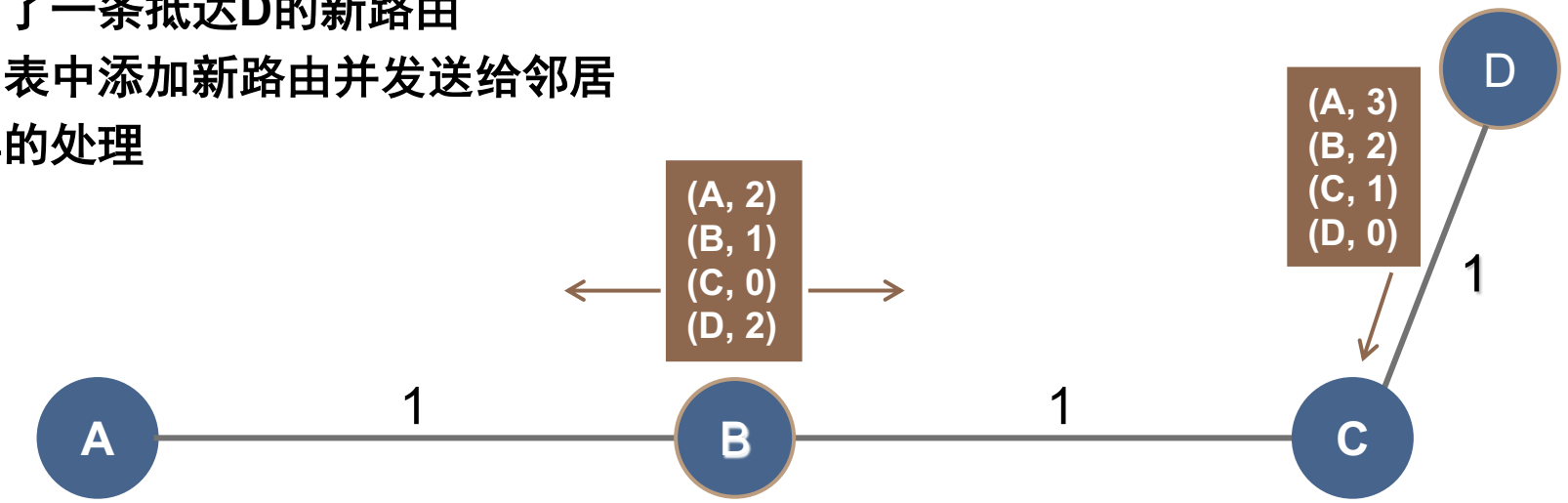
目的地	下一跳	度量
A	A	1
B	B	0
C	C	1

C路由表

目的地	下一跳	度量
A	B	2
B	B	1
C	C	0
D	D	1

距离矢量算法优点

- t5：B和D收到C的路由信息后
  - ① B发现多了一条抵达D的新路由
  - ② B在路由表中添加新路由并发送给邻居
  - ③ D做同样的处理



D路由表

目的地	下一跳	度量
A	C	3
B	C	2
C	C	1
D	D	0

A路由表

目的地	下一跳	度量
A	A	0
B	B	1
C	B	2

B路由表

目的地	下一跳	度量
A	A	1
B	B	0
C	C	1
D	C	2

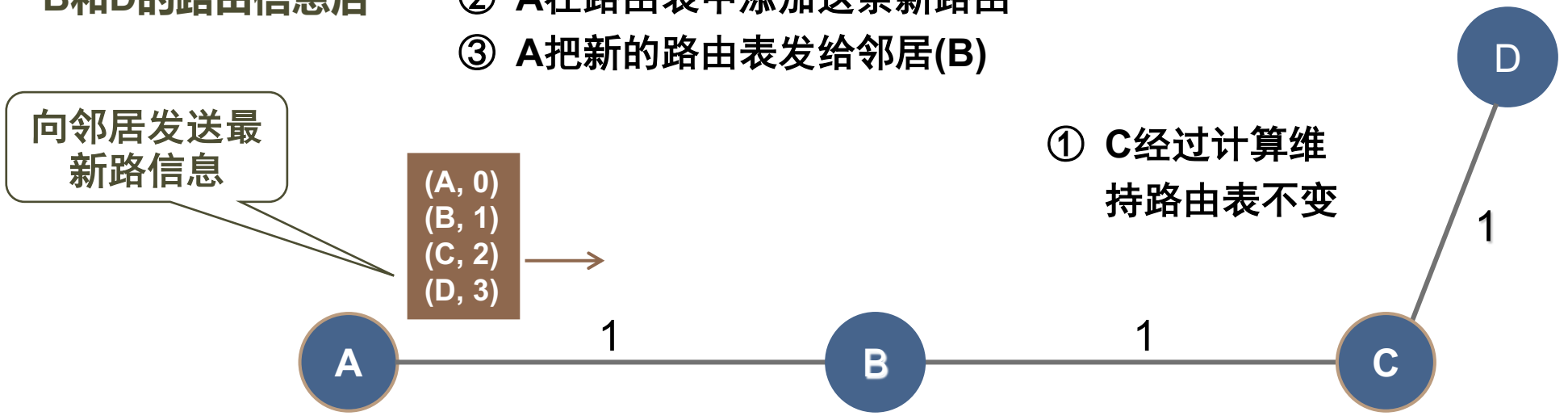
C路由表

目的地	下一跳	度量
A	B	2
B	B	1
C	C	0
D	D	1

距离矢量算法优点

- t6：A和C分别收到B和D的路由信息后

① A发现多了一条抵达D的新路由
② A在路由表中添加这条新路由
③ A把新的路由表发给邻居(B)



D路由表

目的地.	下一跳	度量
A	C	3
B	C	2
C	C	1
D	D	0

A路由表

目的地.	下一跳	度量
A	A	0
B	B	1
C	B	2
D	B	3

B路由表

目的地.	下一跳	度量
A	A	1
B	B	0
C	C	1
D	C	2

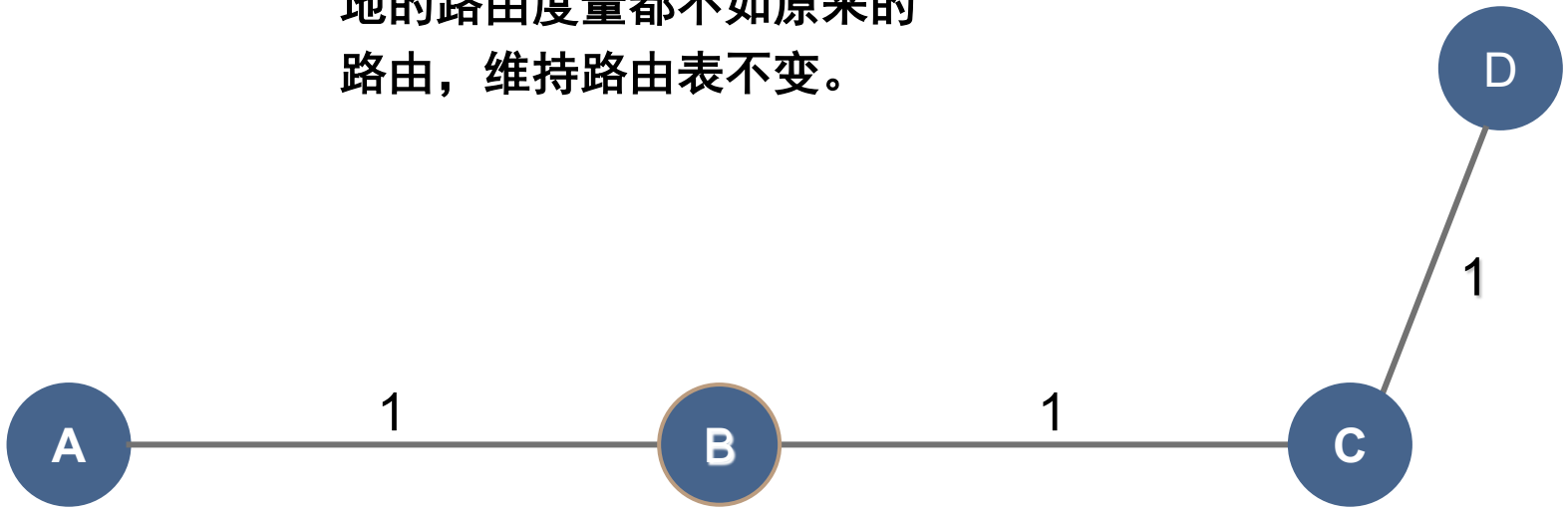
C路由表

目的地.	下一跳	度量
A	B	2
B	B	1
C	C	0
D	D	1

距离矢量算法优点

- t7：B收到来自A的路由信息后

① B发现经过A到达所有目的地的路由度量都不如原来的路由，维持路由表不变。



D路由表

目的地.	下一跳	度量
A	C	3
B	C	2
C	C	1
D	D	0

A路由表

目的地.	下一跳	度量
A	A	0
B	B	1
C	B	2
D	B	3

B路由表

目的地.	下一跳	度量
A	A	1
B	B	0
C	C	1
D	C	2

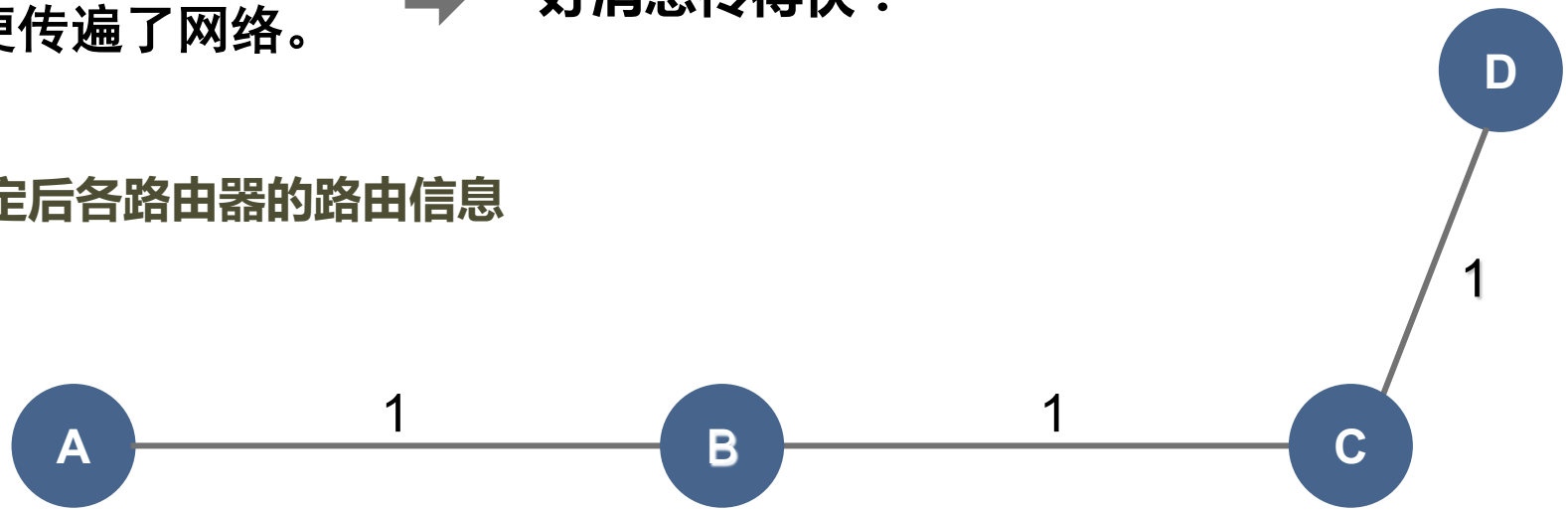
C路由表

目的地.	下一跳	度量
A	B	2
B	B	1
C	C	0
D	D	1

距离矢量算法优点

经过有限的几次传播，新节点和新路由便传遍了网络。 ➡ 好消息传得快！

- t8：路由稳定后各路由器的路由信息



D路由表

目的地	下一跳	度量
A	C	3
B	C	2
C	C	1
D	D	0

A路由表

目的地	下一跳	度量
A	A	0
B	B	1
C	B	2
D	D	3

B路由表

目的地	下一跳	度量
A	A	1
B	B	0
C	C	1
D	D	2

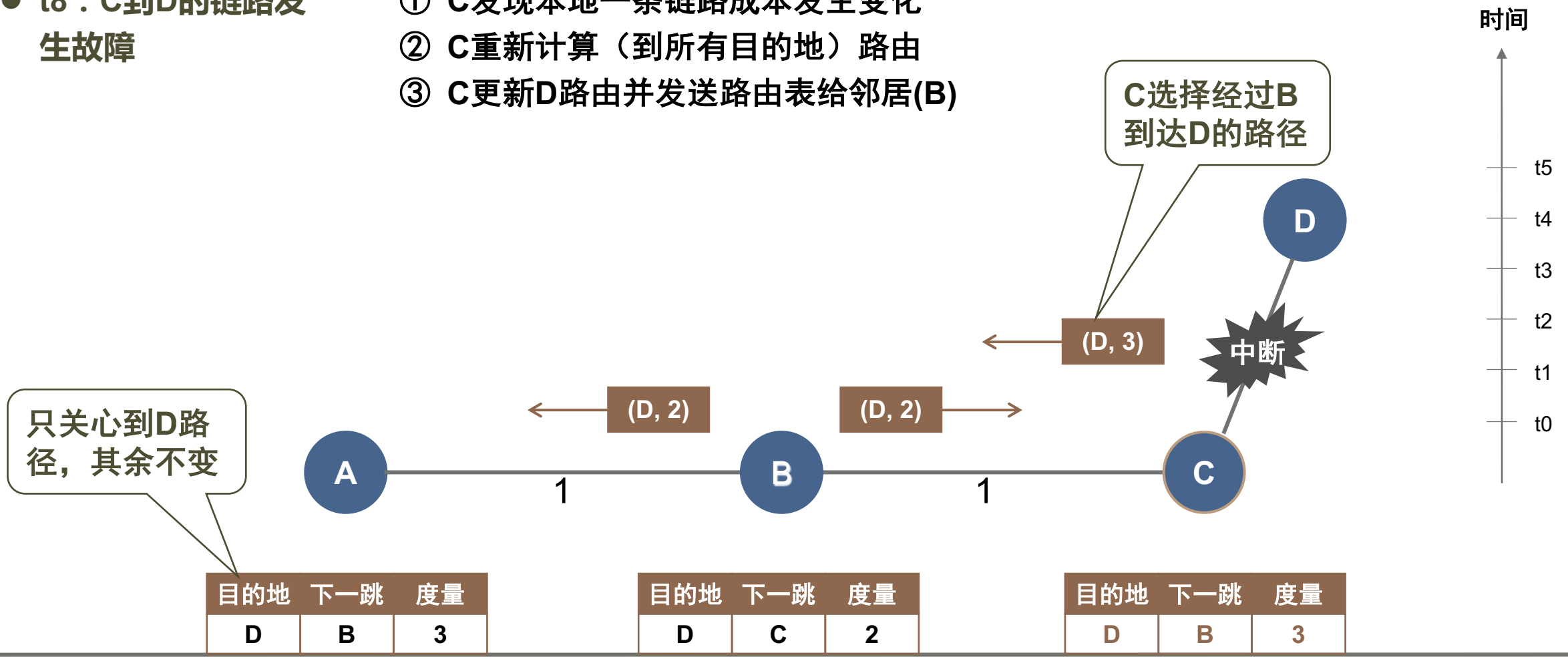
C路由表

目的地	下一跳	度量
A	B	2
B	B	1
C	C	0
D	D	1

距离矢量算法缺点

- t8：C到D的链路发生故障

① C发现本地一条链路成本发生变化
② C重新计算（到所有目的地）路由
③ C更新D路由并发送路由表给邻居(B)



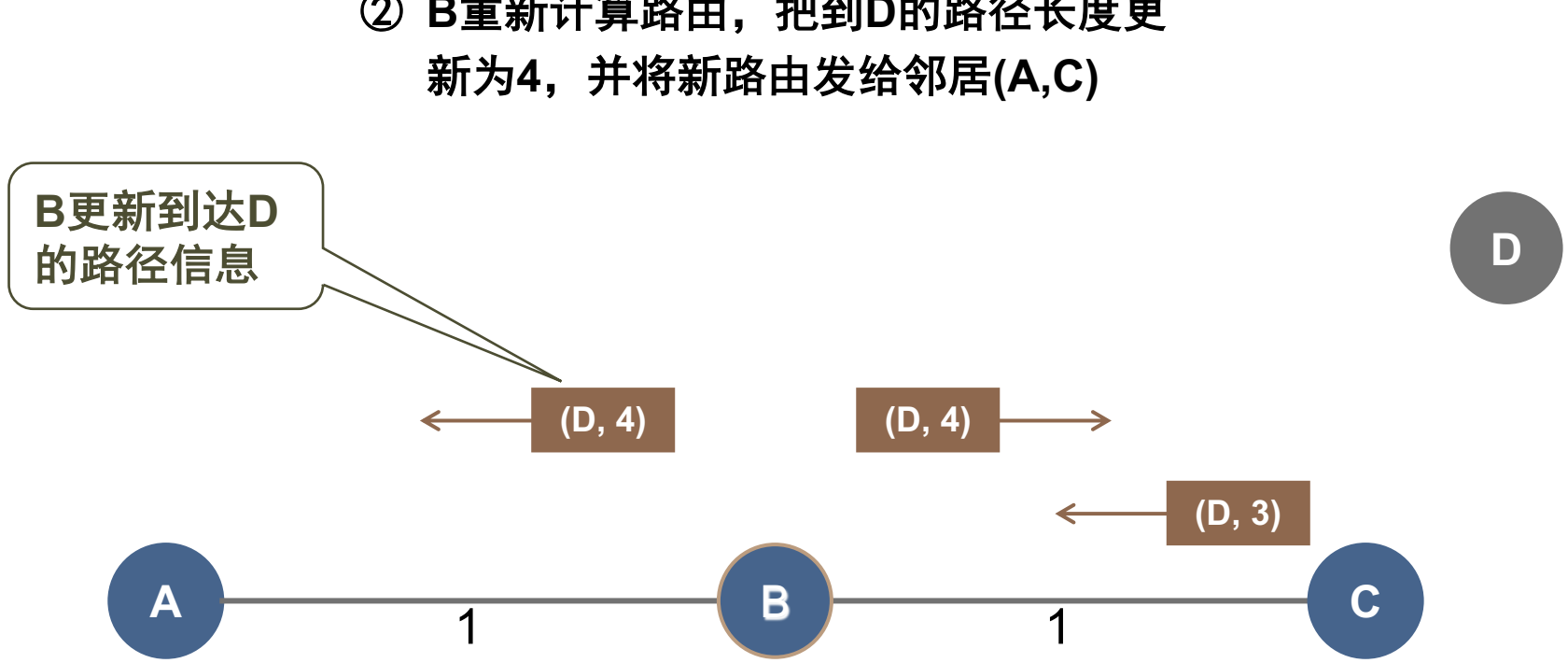
注：此后仅考虑路由表中抵达D的路由

距离矢量算法缺点

- t9：B收到C的路由包

① B发现C到D的路径长度由1变成了3

② B重新计算路由，把到D的路径长度更新为4，并将新路由发给邻居(A,C)



目的地	下一跳	度量
D	B	3

目的地	下一跳	度量
D	C	4

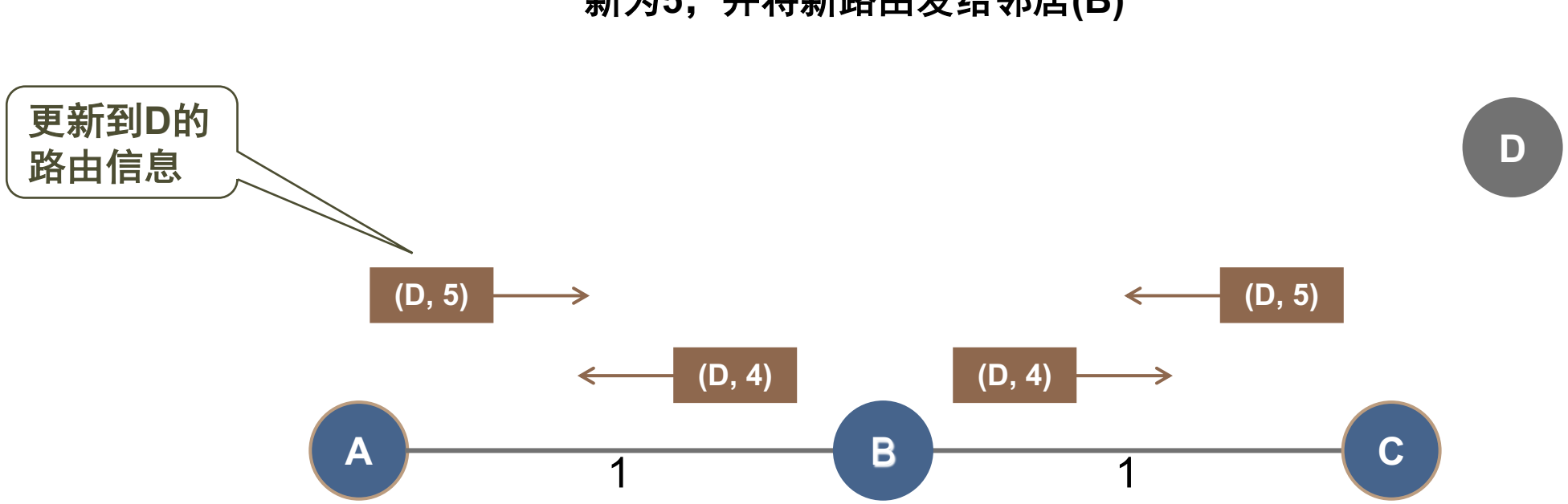
目的地	下一跳	度量
D	B	3

注：仅考虑路由表中抵达D的路由

距离矢量算法缺点

- t10：A和C收到B的路由包

① A和C均发现B到D的路径长度由2变成了4
② A和C重新计算路由，把到D的路径长度更新为5，并将新路由发给邻居(B)



目的地	下一跳	度量
D	B	5

目的地	下一跳	度量
D	C	4

目的地	下一跳	度量
D	B	5

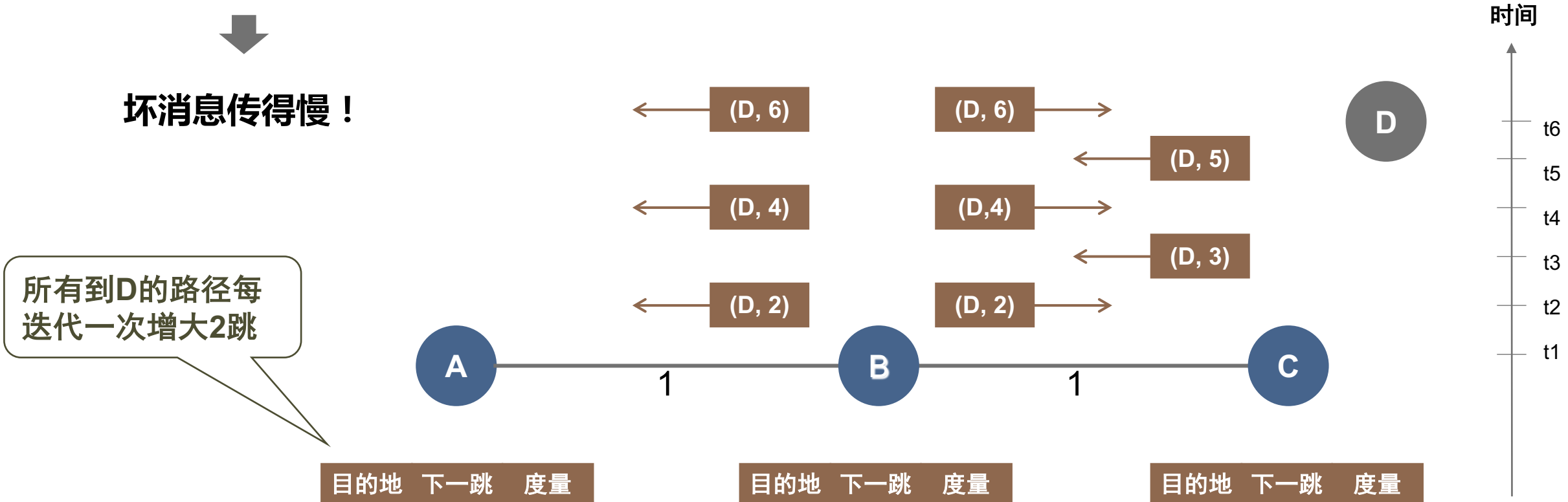
注：仅考虑路由表中抵达D的路由

距离矢量算法缺点

无穷计算问题：算法陷入无限迭代（慢收敛）

坏消息传得慢！

算法每次迭代都修改到D的路由度量，导致邻居跟着迭代更新，如此循环。。。



目的地	下一跳	度量
D	B	3,5,7..

目的地	下一跳	度量
D	C	2,4,6..

目的地	下一跳	度量
D	D	∞,3,5..

注：仅考虑路由表中抵达D的路由

无穷计算问题的本质

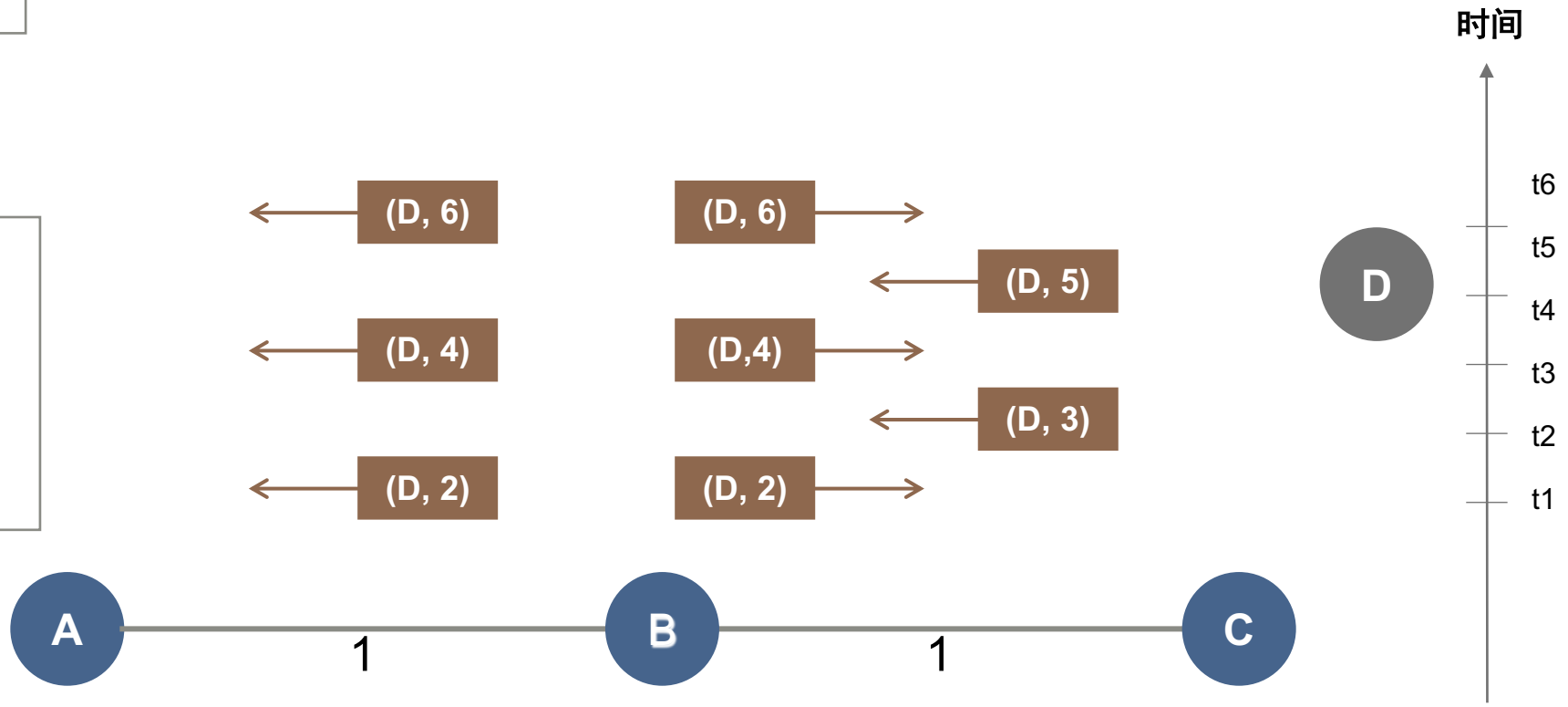
为什么会出现这种无穷迭代情况？

交换路由信息 → ① 可达目的地 ② 相应路径长度

解决方案

① 路由更新包包括完整的路径信息

② 不发布下一跳是对方的路由信息



注：仅考虑路由表中抵达D的路由