

距离矢量路由算法

概述



没有地图如何旅行

假设

- 出发地北京，目的地南京
- 没有同行者商量
- 每个交叉路口有方向指示牌

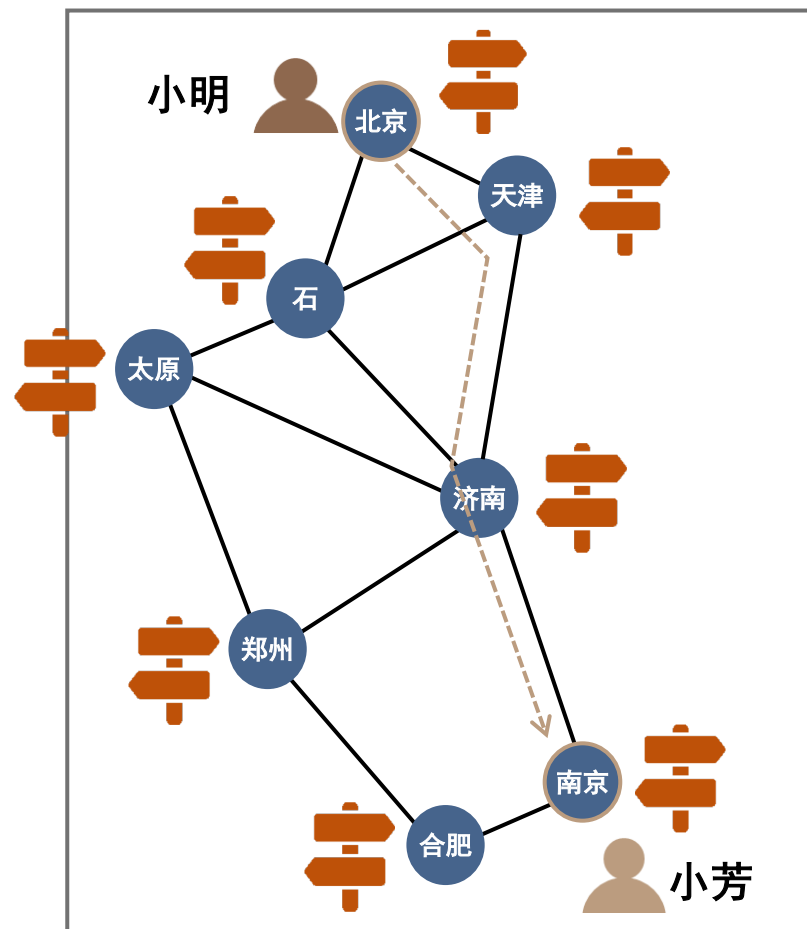
只要路标上的信息是一致的和完备的，小明一定能够从北京出发到达目的地南京，而且他走的是一条最好的路径。

旅行者小明 = 网络层的包

旅行路线 = 包的路径

旅游咨询台 = 沿途路由器

交叉路口路标 = 路由器路由表



没有网络完整拓扑信息时传递包

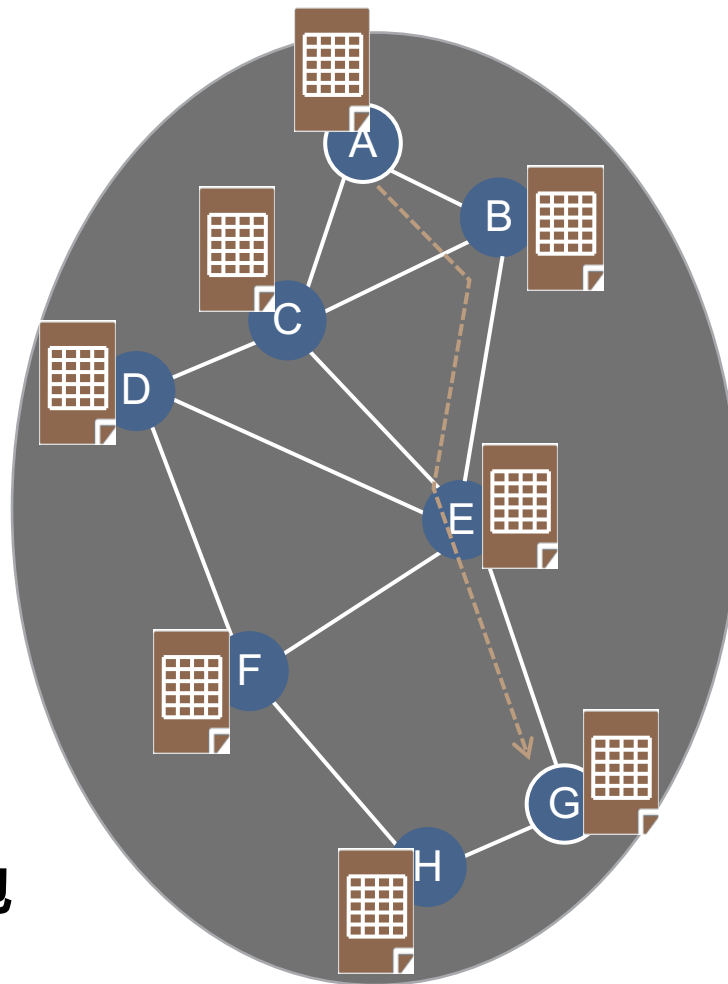
假设

- 每个包有完整的目标地址
- 每个包独立选择下一跳
- 只能依赖路由表

?

没有完整的网络拓扑信息，如何计算路由

只要路由表的信息是一致的和完备的，按照路由表转发包就一定能将包传递到目的地，而且这是一条最好的路径。



距离矢量路由算法 (DV)

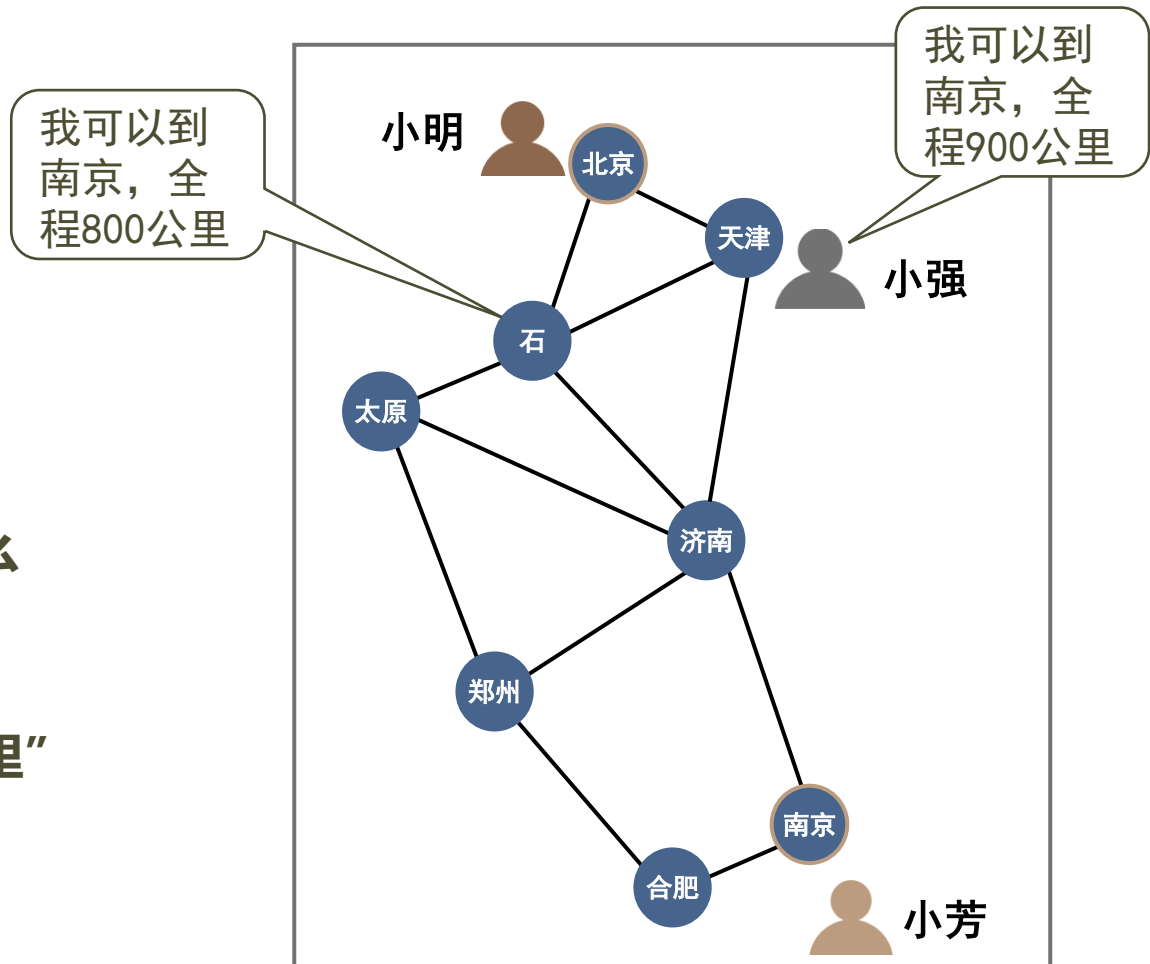
基本思想

只要一个节点不是孤立点，就可以通过和邻居交换路由信息来获得网络全部的路由信息。

假设：小明要去南京看小芳，没有地图，不知道怎么去。

- 小明有个同学小强在天津
- 小强说“从我这里出发可以到南京，900公里”
- 石家庄同学说“从我这里出发可以到南京，800公里”

试问：小明怎么走？



距离矢量算法特点

分布的

- 每个节点接收来自与其直接邻接节点的路由信息，并执行路由计算
- 将计算结果回传给直接邻接的节点

通过和邻居交换路由信息来计算全网路由

迭代的

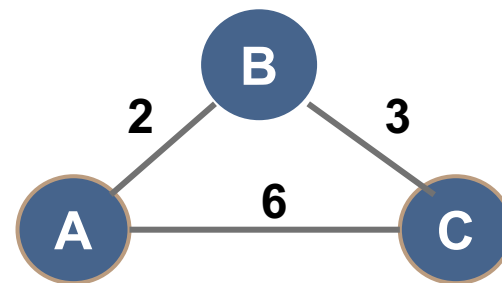
- 计算过程循环进行
- 直到相邻节点没有可交换的路由信息为止

当各节点路由稳定后，算法收敛。

异步的

- 并不要求所有节点相互锁步操作

A, B和C各自独立运行路由算法

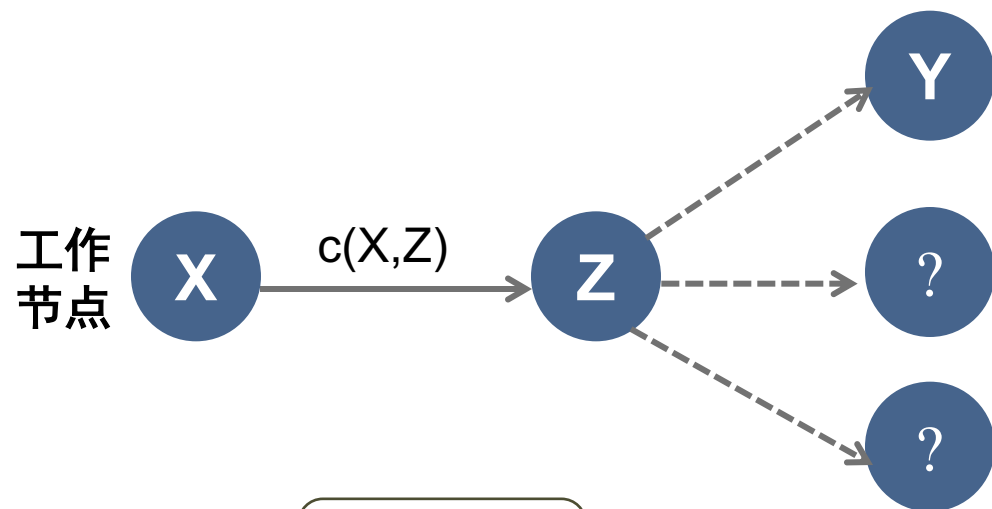


?

A有两个邻居，A到C应该走哪条路？



距离矢量算法——距离



假设：X和Z是邻居，连接两点的边成本为 $c(X, Z)$

试问：X经邻居Z到达Y的距离，表示为

$D^x(Y, Z)$

X本地测量获得

从Z广播的路由报文中获得

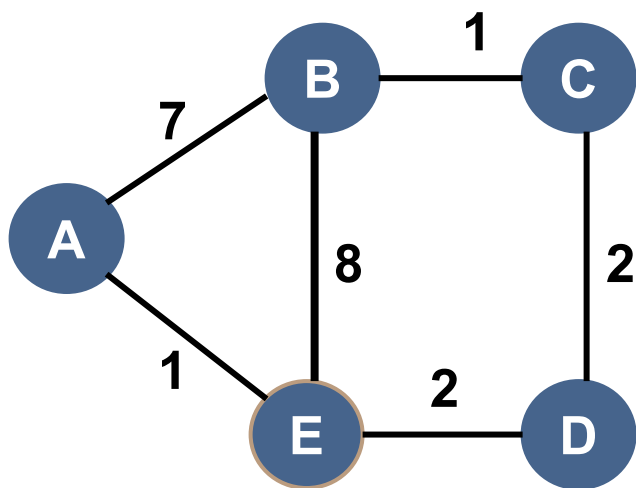
$$D^x(Y, Z) = c(X, Z) + \min_w \{D^z(Y, w)\}$$

节点X经过Z到达Y的距离 = X到邻居Z的距离 + Z到Y的最短距离

w为Z的所有直接邻居(包括X)



距离矢量算法——距离矩阵



E经过A到达D的距离:

$$\begin{aligned} D^E(A,D) &= c(E,D) + D^D(A, w) \\ &= 2+3 = 5 \end{aligned}$$

节点E的距离表 (从本地到达所有目的地距离)			
目标节点	经过邻居节点的成本		
	A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

所有抵达B
的路径

所有抵达D
的路径

所有抵达A
的路径

所有抵达C
的路径



距离矢量算法——生成路由表

节点E的距离表
(从本地到达所有目的地距离)

目标节点	经过邻居节点的成本 A	B	D
A	1	14	5
B	7	8	5
C	6	9	4
D	4	11	2

从距离矩阵中选择获得最小距离的那个邻居作为通向目的地的下一站，即路由表中最短路径的下一跳（next-hop）。

节点E的路由表

目标地址	出境线路	路径距离
A	A	1
B	D	5
C	D	4
D	D	2

