

链路状态路由算法

概述

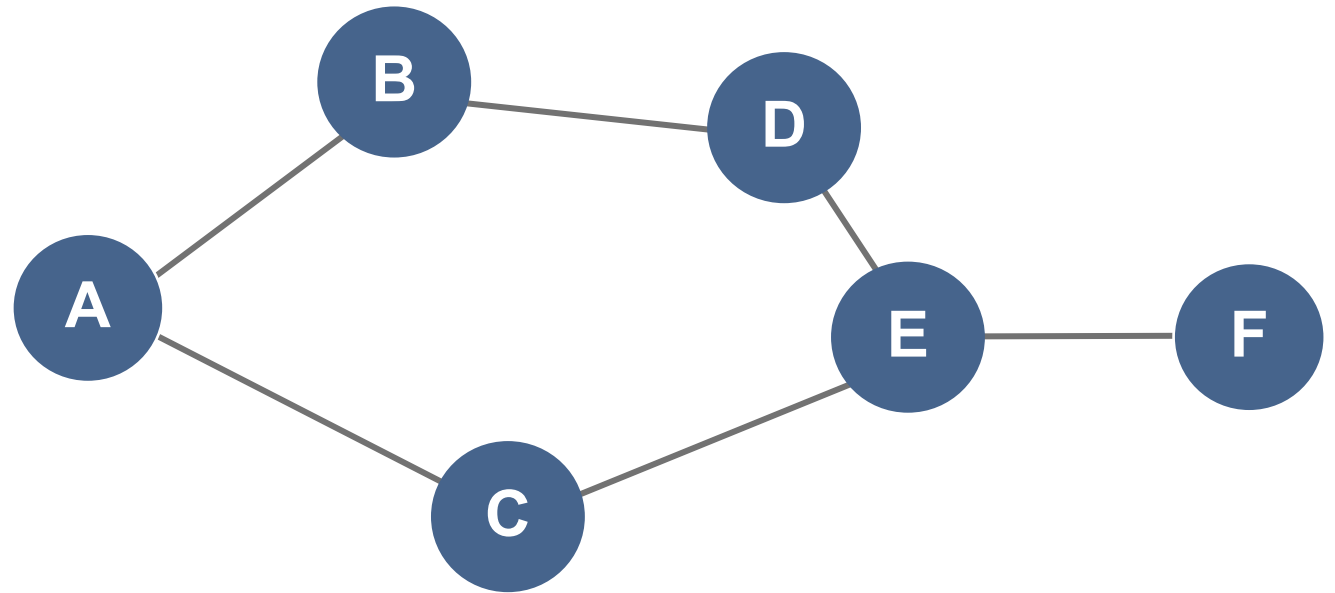


链路状态路由算法

链路状态路由算法：每个节点都有网络完整拓扑图，每个节点维护到邻居的连通性与链路成本。节点向网络中所有其他节点广播自己和邻居的连接信息，每当接收到来自其他节点信息时用Dijkstra算法重新计算路由表。

Dijkstra 算法

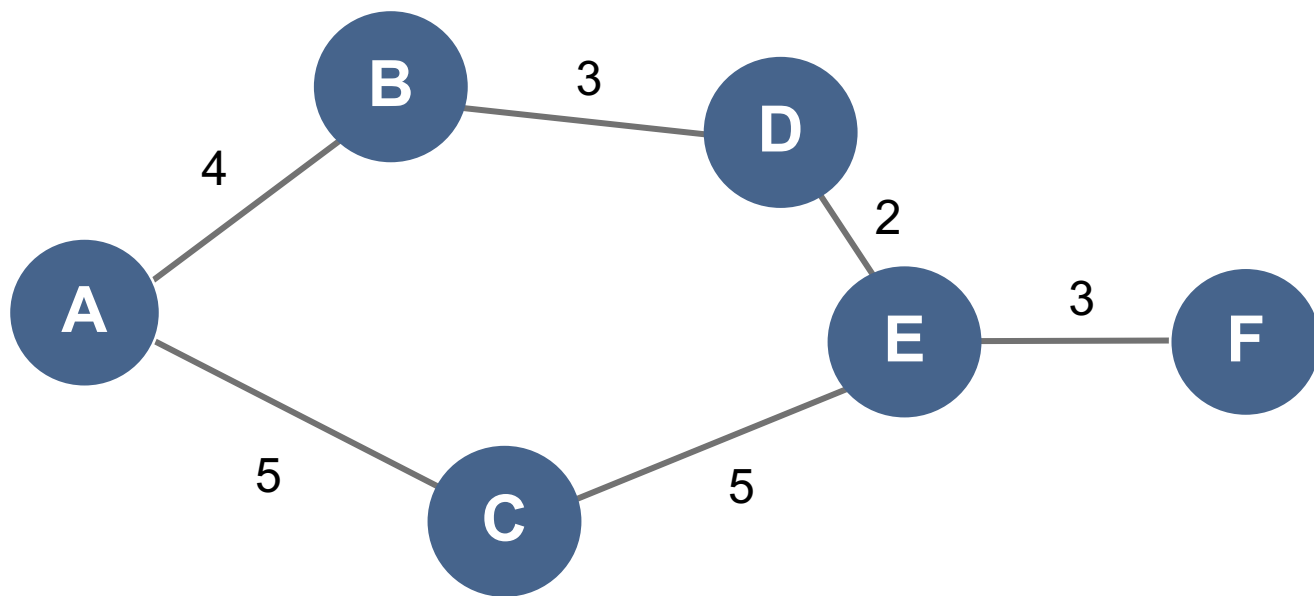
- 计算从一个节点(源, 假设为A)到网络中所有其他节点的最小成本路径。
- 算法迭代进行
- 经第k次迭代后, 就能找出到k个目标节点的最小成本路径。



链路状态与链路成本

如果链路成本与链路带宽成反比。

带宽越高成本越低 → 路由选择高带宽的路径



例如：路径A-B-D-E-F比路径A-C-E-F的成本低（意味着端到端带宽更高）。

链路成本：反应路由度量的函数

$$E = f(\text{距离/时延/成本})$$

链路时延测量

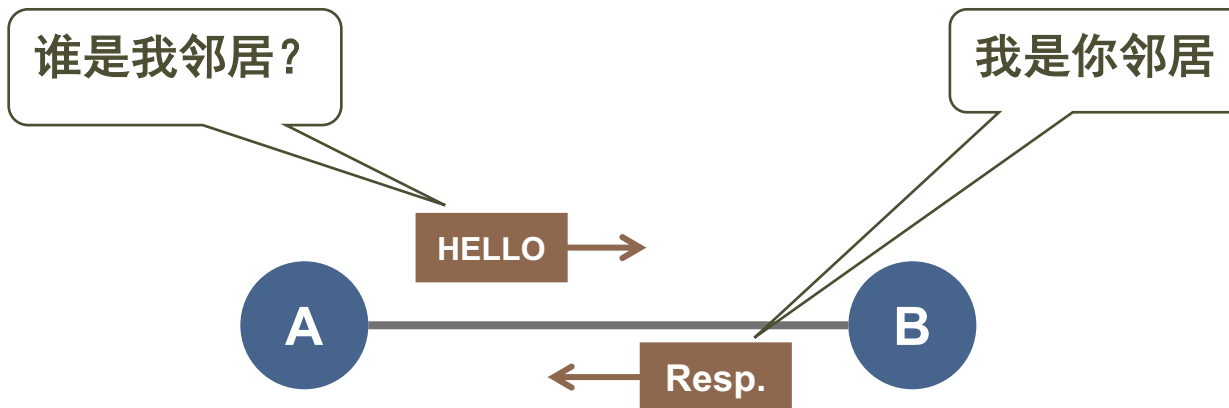
- 给邻居节点发送一个盖上时间戳的包，记录其离开时间
- 收到邻居返回的确认时，计算该包的来回时延



链路状态信息的获取

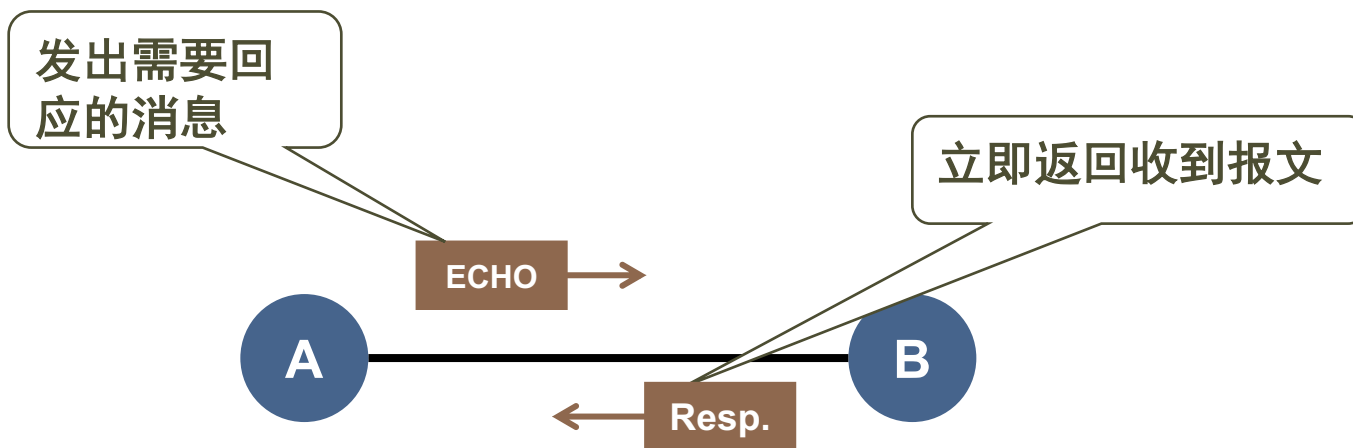
① 发现邻接节点

在一跳范围内广播一个HELLO消息，邻居的Response消息把自己的地址带了过来。



② 测量链路成本

收到ECHO消息后立即返回，用来测试链路的往返时间（时延）。



链路状态信息包的发送

③ 封装链路状态包

把本地链路（与邻居相连）成本以及邻居ID封装在一个报文中。

本机ID
序号
生存期
邻居1 成本
邻居2 成本
.....

④ 广播链路状态信息

发送链路状态包时机

- 定期发送
- 出现重大事件时

- 本机ID：指出链路状态包的发送方
- 序号：标识了源节点发出的链路状态包次序
- 生存期：标识了本链路状态包信息的有效期
- 邻居/成本：表示本机与邻居的链路成本

防止广播通信中的重复包：接收节点可根据(源节点ID，序号)来判定此次入境包是否含有最新的链路状态信息。



链路状态路由的计算

A发送的本地链路状态包

A
100
300s
B 4
C 5

A收到的来自其他路由器的链路状态包

B
100
300s
A 4
D 3

C
100
300s
A 5
E 5

D
100
300s
B 3
E 2

E
100
300s
C 5
D 2
F 3

F
100
300s
E 3

⑤ 计算新路由

- 根据接收到的所有节点链路状态包构建表示网络拓扑的子网
- 采用Dijkstra算法计算从本地到所有其他节点的最短路径

