

路由算法概述



旅行者 vs. 旅行路线

?

北京的“小明”要去南京看望“小芳”怎么走

链路状态
路由算法

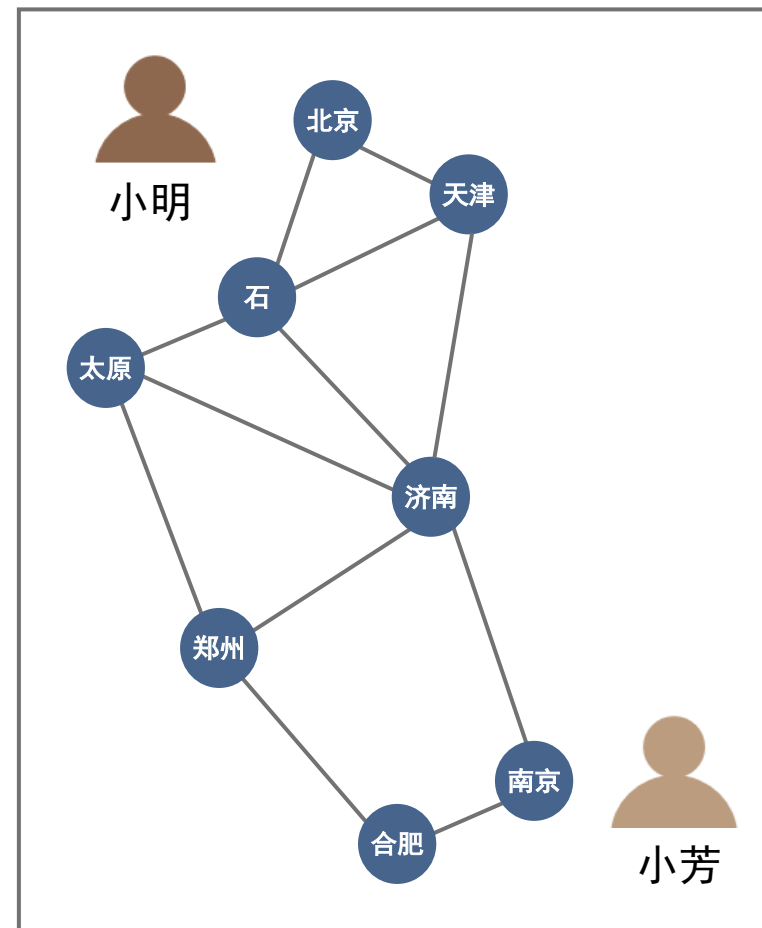
有地图

自己独立计算从北京出发到目的地的旅行路线，并且知道该条线路的长短。

没地图

看自己有几个邻居，期望通过和邻居交流获得抵达目的地的所有线路，并从中选择一条最好的。

距离矢量
路由算法



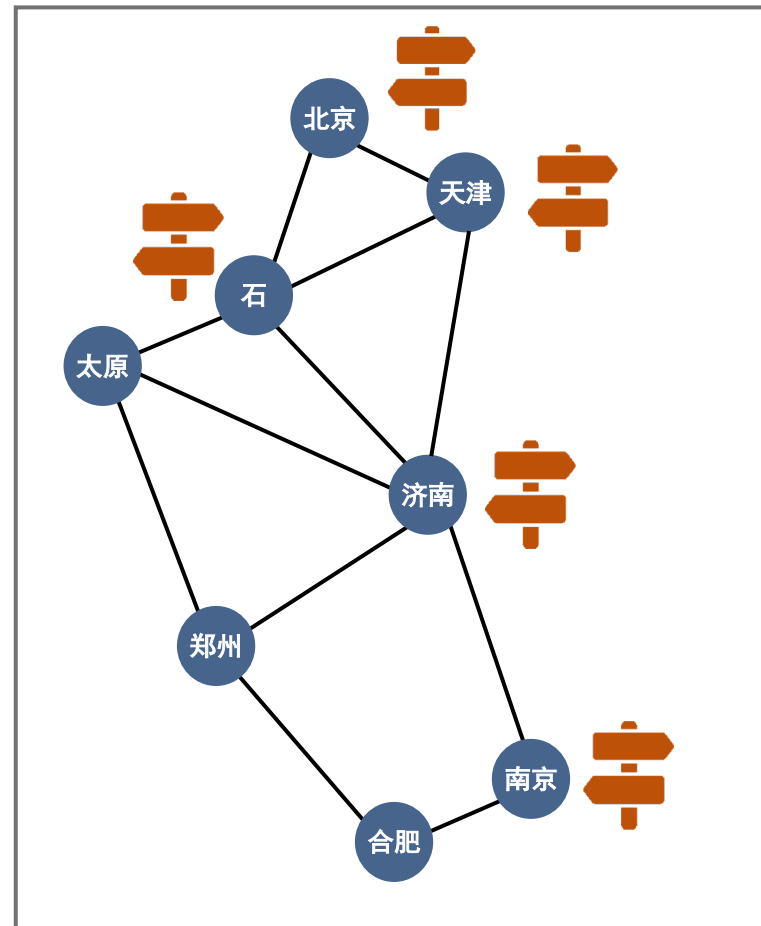
走一步看一步的旅行

假设

- 出发地北京，目的地南京
- 没有同行者商量
- 每一站都有旅行信息咨询机构（相当于在交叉路口有方向指示牌）

小明可能的旅游线路：

- 沿着唯一的路出发，在每个交叉路口寻找方向指示牌
- 第一个路标：“到西安太原向右，到天津向左，其余向左”，选择了省缺路线向左；
- 到天津后发现第二个路标：“到石家庄向右，到北京向后，其余向前”，选择向前；
- 如此这般，在每个路口根据路标指示的方向前进
- 最终看到了通向“南京大学”的路标。



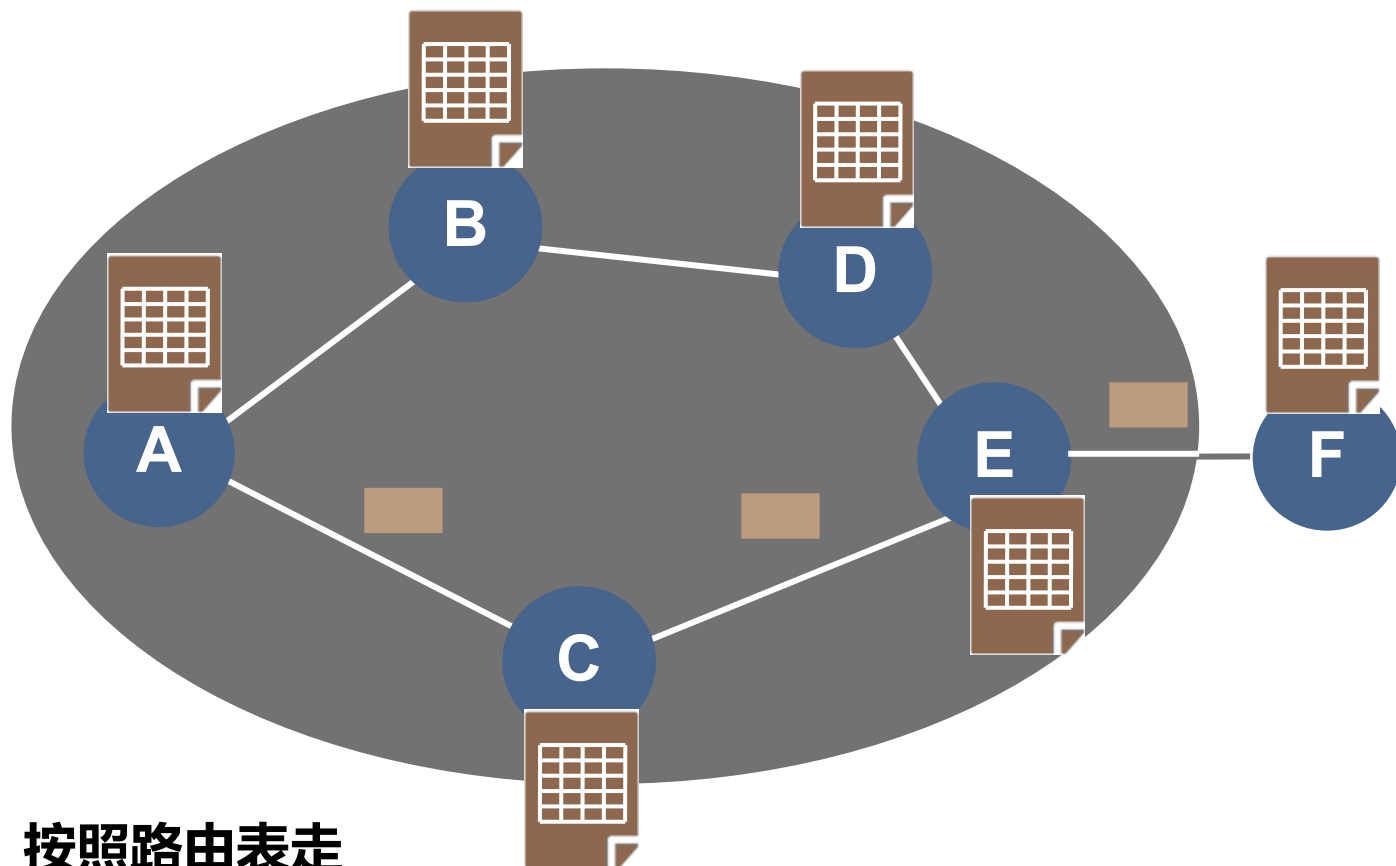
包在互联网中传递

假设

- 每个包有完整的目标地址
- 每个包独立选择下一跳
- 下一跳的选择依赖路由表

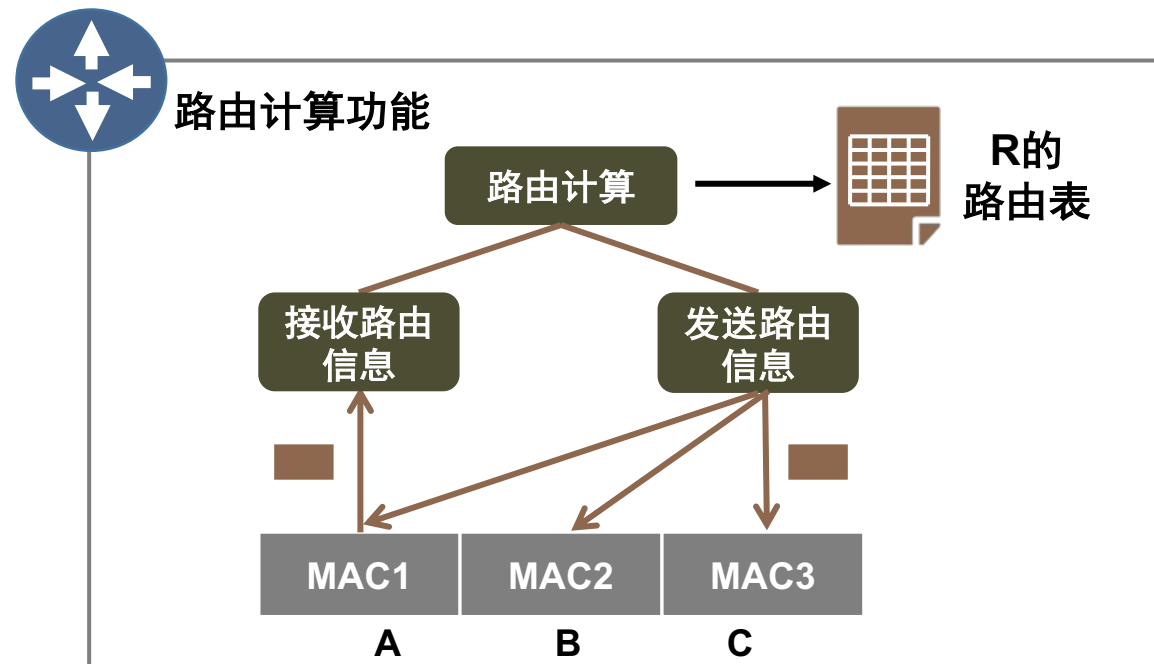
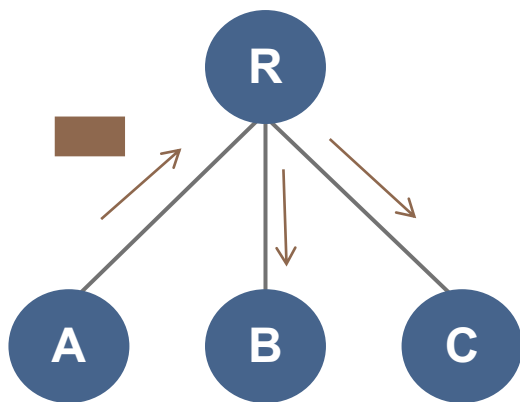
旅行者小明 = 网络层的包
旅行路线 = 包的路径
旅游咨询台 = 沿途路由器
交叉路口路标 = 路由器路由表

只要路由表上的信息是一致的和完备的，按照路由表走就一定能抵达目的地，而且这是一条最“好”的路径。



路由器的内部结构

假设：某个互联网中某个路由器R和邻居的连接关系如下：

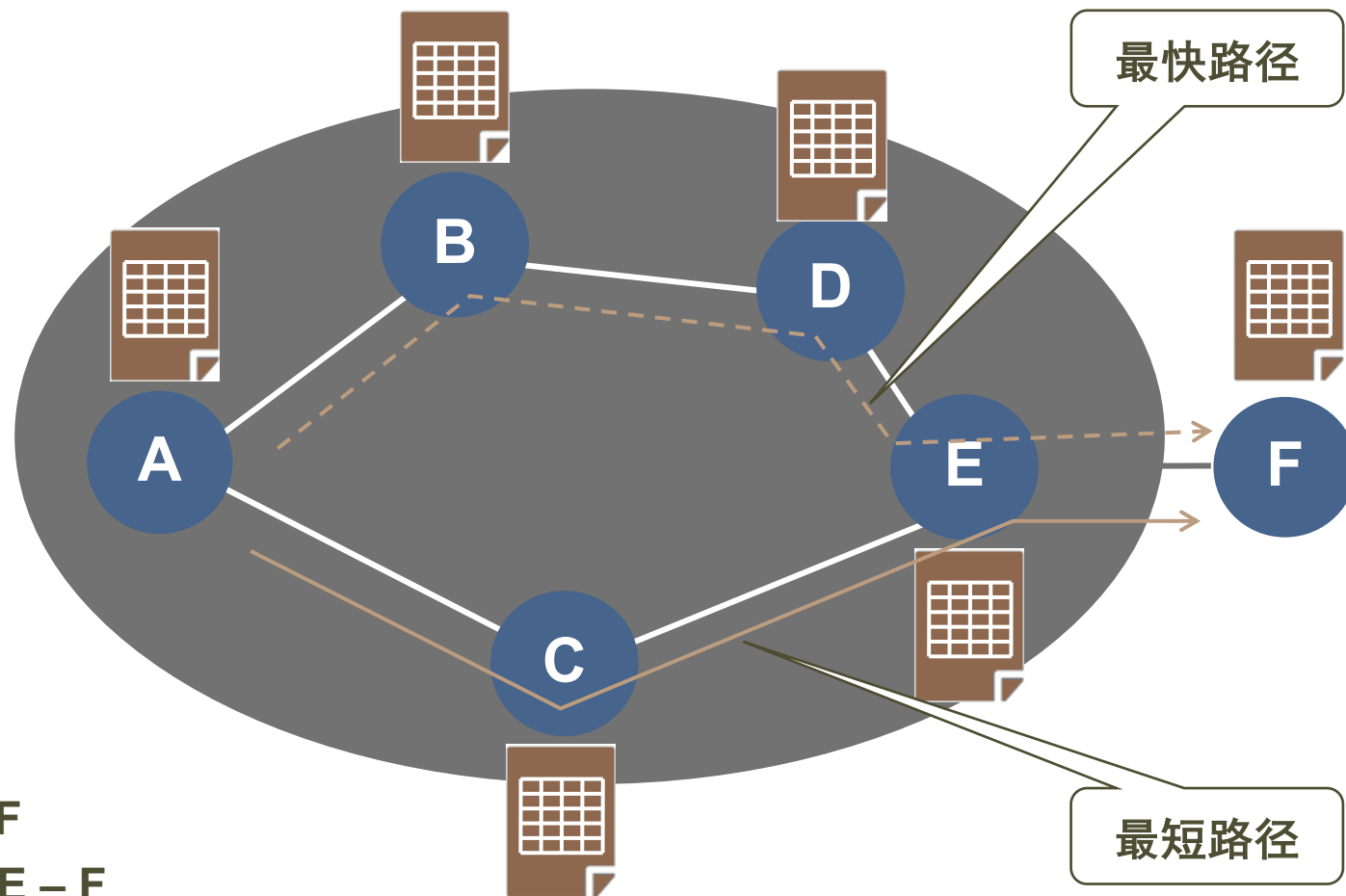


- 路由器之间只能通过消息交换获得更多的拓扑和路由信息
- 网络所采用的路由算法决定了各路由器之间交换怎样的路由信息

路由计算给出的最“好”路径

“好”的标准

- 最短路径
- 最快路径
- 最宽路径
- 最可靠路径
- 最低成本路径
-



例如：考虑从A到F的路由

- 最短路径（跳数最少）A – C – E – F
- 最快路径（延迟最短）A – B – D – E – F



路由算法的分类

非自适应算法——静态路由

- 不是根据实际测量或估计的网络当前流量和拓扑结构作路由决策

自适应算法——动态路由

- 根据网络拓扑结构、流量的变化来改变其路由选择



自适应路由的前提：节点间交换网络状态信息

- 信息越多，做出的路由决策越好
- 信息越多，网络负担越大性能下降越快

优点

- 可提高网络性能
- 有助于拥塞控制

缺点

- 路由决策复杂
- 依赖于状态信息
- 不能太快和太慢

