

的路径中具有最低费用的路径。(你考察过 u 到 z 之间的所有 17 条可能的路径吗? 很可能没有!) 这种计算就是一种集中式路由选择算法的例子, 即路由选择算法在一个位置 (即你的大脑中) 运行, 具有该网络的完整信息。对路由选择算法的一种广义分类方式是根据该算法是全局式的还是分散式的来加以区分。

- **全局式路由选择算法** (global routing algorithm) 用完整的、全局性的网络知识计算出从源到目的地之间的最低费用路径。也就是说, 该算法以所有结点之间的连通性及所有链路的费用为输入。这就要求该算法在真正开始计算以前, 要以某种方式获得这些信息。计算本身可在某个场点 (集中式全局路由选择算法) 进行, 或可在多个场点重复进行。然而这里的主要区别在于, 全局式算法具有关于连通性和链路费用方面的完整信息。实践中, 具有全局状态信息的算法常被称作**链路状态** (Link State, LS) **算法**, 因为该算法必须知道网络中每条链路的费用。我们将在 4.5.1 节中学习 LS 算法。
- **分散式路由选择算法** (decentralized routing algorithm) 以迭代、分布式的方式计算出最低费用路径。没有结点拥有关于所有网络链路费用的完整信息, 而每个结点仅有与其直接相连链路的费用知识即可开始工作。然后, 通过迭代计算过程并与相邻结点 (即与该结点相连链路的另一端的结点) 交换信息, 一个结点逐渐计算出到达某目的结点或一组目的结点的最低费用路径。我们将在后面的 4.5.2 节学习一个称为**距离向量** (Distance-Vector, DV) **算法**的分散式路由选择算法。之所以叫做 DV 算法, 是因为每个结点维护到网络中所有其他结点的费用 (距离) 估计的向量。

路由选择算法的第二种广义分类方式是根据算法是静态的还是动态的进行分类。在**静态路由选择算法** (static routing algorithm) 中, 随着时间的流逝, 路由的变化是非常缓慢的, 通常是人工干预进行调整 (如人为手工编辑一台路由器的转发表)。**动态路由选择算法** (dynamic routing algorithm) 能够当网络流量负载或拓扑发生变化时改变路由选择路径。一个动态算法可周期性地运行或直接响应拓扑或链路费用的变化而运行。虽然动态算法易于对网络的变化做出反应, 但也更容易受诸如路由选择循环、路由振荡之类问题的影响。

路由选择算法的第三种分类方式是根据它是负载敏感的还是负载迟钝的进行划分。在**负载敏感算法** (load-sensitive algorithm) 中, 链路费用会动态地变化以反映出底层链路的当前拥塞水平。如果当前拥塞的一条链路和高费用相联系, 则路由选择算法趋向于绕开该拥塞链路来选择路由。而早期的 ARPAnet 路由选择算法就是负载敏感的 [McQuillan 1980], 所以遇到了许多难题 [Huitema 1998]。当今的因特网路由选择算法 (如 RIP、OSPF 和 BGP) 都是**负载迟钝的** (load-insensitive), 因为某条链路的费用不明显地反映其当前 (或最近) 的拥塞水平。

4.5.1 链路状态路由选择算法

前面讲过, 在链路状态算法中, 网络拓扑和所有的链路费用都是已知的, 也就是说可用作 LS 算法的输入。实践中这是通过让每个结点向网络中所有其他结点广播链路状态分组来完成的, 其中每个链路状态分组包含它所连接的链路的特征和费用。在实践中 (例如使用因特网的 OSPF 路由选择协议, 讨论见 4.6.1 节), 这经常由**链路状态广播** (link state broadcast) 算法 [Perlman 1999] 来完成。我们将在 4.7 节中讨论广播算法。结点广播的