

送给它的默认路由器。我们将源主机的默认路由器称作源路由器 (source router)，把目的主机的默认路由器称作目的路由器 (destination router)。一个分组从源主机到目的主机的路由选择问题显然可归结为从源路由器到目的路由器的路由选择问题。这是本节的重点。

因此，路由选择算法的目的是简单的：给定一组路由器以及连接路由器的链路，路由选择算法要找到一条从源路由器到目的路由器的“好”路径。通常，一条好路径指具有最低费用的路径。然而我们将看到，实践中现实世界还关心诸如策略之类的问题（例如，诸如“属于组织 Y 的路由器 X 不应转发任何来源于组织 Z 网络的分组”之类的规则），这也使得概念简单、性能优秀的算法变得复杂。然而这些概念简单、性能优秀的算法的理论奠定了当今网络路由选择实践的基础。

可以用图来形式化描述路由选择问题。我们知道图 (graph) $G = (N, E)$ 是一个 N 个结点和 E 条边的集合，其中每条边是取自 N 的一对结点。在网络层路由选择的环境中，图中的结点表示路由器，这是做出分组转发决定的点；连接这些结点的边表示这些路由器之间的物理链路。这样一个计算机网络图的抽象显示在图 4-27 中。若要查看某些表示实际网络图的图，参见 [Dodge 2012; Cheswick 2000]；对于基于不同的图模型建模因特网的好坏的讨论，参见 [Zegura 1997; Faloutsos 1999; Li 2004]。

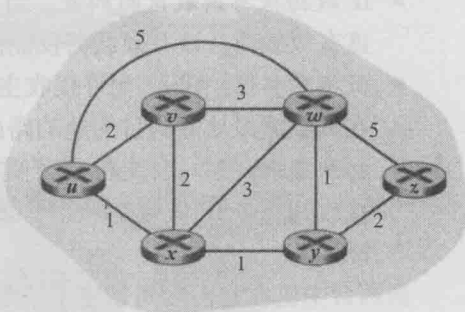


图 4-27 一个计算机网络的抽象图模型

如图 4-27 所示，一条边还有一个值表示它的费用。通常，一条边的费用可反映出对应链路的物理长度（例如一条越洋链路费用可能比一条短途陆地链路费用高）、链路速度，或与该链路相关的金融上的费用。为了我们的目的，我们只将这些链路费用看成是给定的，而不必操心这些值是如何确定的。对于 E 中的任一条边 (x, y) ，我们用 $c(x, y)$ 表示结点 x 和 y 间边的费用。如果不属于 E 的结点对 (x, y) ，置 $c(x, y) = \infty$ 。此外，我们考虑都是无向图（即图的边没有方向），因此边 (x, y) 与边 (y, x) 是相同的并且 $c(x, y) = c(y, x)$ 。如果 (x, y) 属于 E ，结点 y 也被称为结点 x 的邻居 (neighbor)。

在图抽象中为各条边指派了费用后，路由选择算法的自然目标是找出从源到目的地的最低费用路径。为了使问题更为精确，回想在图 $G = (N, E)$ 中的一条路径 (path) 是一个结点的序列 (x_1, x_2, \dots, x_p) ，这样每一个对 $(x_1, x_2), (x_2, x_3), \dots, (x_{p-1}, x_p)$ 是 E 中的边。路径 (x_1, x_2, \dots, x_p) 的费用只是沿着路径所有边的费用的总和，即 $c(x_1, x_2) + c(x_2, x_3) + \dots + c(x_{p-1}, x_p)$ 。给定任何两个结点 x 和 y ，通常在这两个结点之间有许多条路径，每条路径都有一个费用。这些路径中的一条或多条是最低费用路径 (least-cost path)。因此最低费用路径问题是清楚的：找出源和目的地之间的具有最低费用的一条路。例如，在图 4-27 中，源结点 u 和目的结点 w 之间的最低费用路径是 (u, x, y, w) ，具有的路径费用是 3。注意到若在图中的所有边具有相同的费用，则最低费用路径也就是最短路径 (shortest path)，即在源和目的地之间的具有最少链路数量的路径。

作为一个简单练习，试找出图 4-27 中从结点 u 到结点 z 的最低费用路径，并要反映出你是如何算出该路径的。如果你像大多数人一样，通过考察图 4-27，跟踪几条从 u 到 z 的路由，你就能找出从 u 到 z 的路径，然后以某种方式来确信你所选择的路径就是所有可能