初始化后，每个结点向它的两个邻居发送其距离向量。图 4-30 中用从表的第一列到表的第二列的箭头说明了这一情况。例如，结点 $x$ 向两个结点 $y$ 和 $z$ 发送了它的距离向量 $\boldsymbol{D}_x = [0, 2, 7]$。在接收到该更新后，每个结点重新计算它自己的距离向量。例如，结点 $x$ 计算

$$D_x(x) = 0$$

$$D_x(y) = \min\{c(x,y) + D_y(y), c(x,z) + D_z(y)\} = \min\{2+0, 7+1\} = 2$$

$$D_x(z) = \min\{c(x,y) + D_y(z), c(x,z) + D_z(z)\} = \min\{2+1, 7+0\} = 3$$

第二列因此为每个结点显示了结点的新距离向量连同刚从它的邻居接收到的距离向量。注意到，例如结点 $x$ 到结点 $z$ 的最低费用估计 $D_x(z)$ 已经从 7 变成了 3。还应注意到，对于结点 $x$，结点 $y$ 在该 DV 算法的第 14 行中取得了最小值；因此在该算法的这个阶段，我们在结点 $x$ 得到了 $v^*(y) = y$ 和 $v^*(z) = y$。

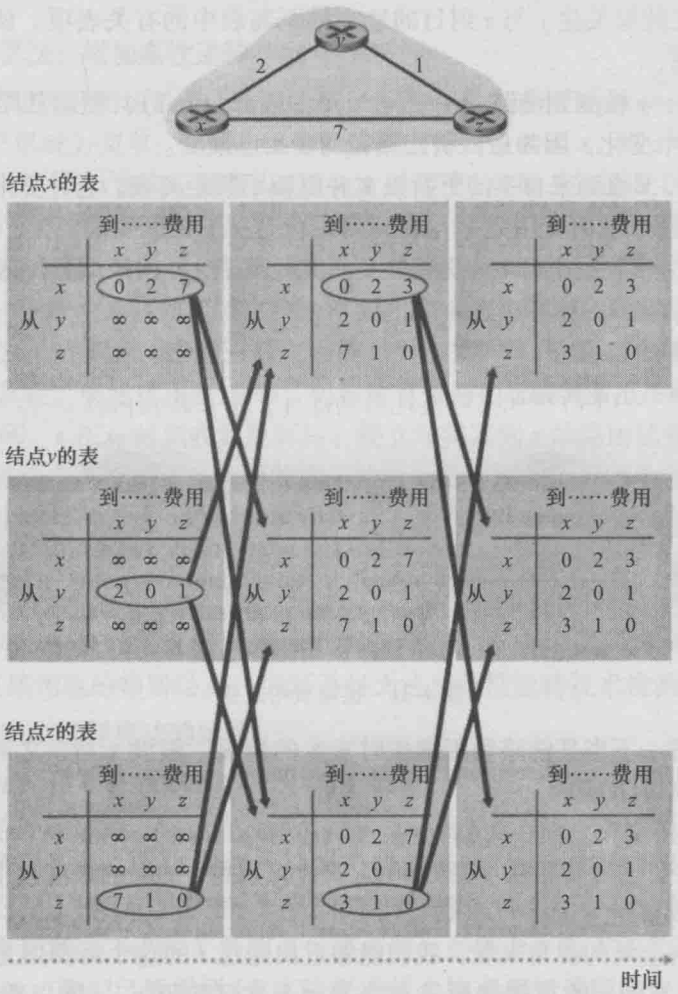


图 4-30 距离向量（DV）算法

在结点重新计算它们的距离向量之后，它们再次向其邻居发送它们的更新距离向量（如果它们已经改变的话）。图 4-30 中由从表第二列到表第三列的箭头说明了这一情况。注意到仅有结点 $x$ 和结点 $z$ 发送了更新：结点 $y$ 的距离向量没有发生变化，因此结点 $y$ 没有发送更新。在接收到这些更新后，这些结点则重新计算它们的距离向量并更新它们的路