报，该数据报应首先被转发到相邻路由器 A；该表还指出沿着最短路径到目的子网 w 为两跳距离。类似地，该表指出了子网 z 经由路由器 B 为 7 跳距离。虽然 RIP 版本 2 允许使用类似于我们在 4.4 节中学习的路由聚合技术来聚合子网表项，但原则上 AS 内的每个子网应在转发表中占一行。在图 4-36 中的表以及后续的表，都只是部分完成了。

目的子网	下一台路由器	到目的地的跳数
w	A	2
y	B	2
z	B	7
x	—	1
…	…	…

图 4-36 收到来自路由器 A 的通告之前路由器 D 中的转发表

现在假定 30 秒以后，路由器 D 收到来自路由器 A 的如图 4-37 所示的通告。注意到该通告正是来自路由器 A 的路由选择表信息！该信息特别指明了子网 z 离路由器 A 仅有 4 跳距离。一旦收到该通告，路由器 D 将该通告（图 4-37）与旧路由选择表（图 4-36）合并。特别是路由器 D 知道了通过路由器 A 到子网 z 比通过路由器 B 到达路径更短。因此，路由器 D 更新其转发表以记下该更短的最短路径，如图 4-38 中所示。你也许会问，到子网 z 的最短路径怎么会变得更短呢？可能是分布式的距离向量算法还处在收敛过程中（参见 4.5.2 节），或者也许是新的链路和/或路由器加入了该 AS，因此改变了在 AS 中的最短路径。

目的子网	下一台路由器	到目的地的跳数
z	C	4
w	—	1
x	—	1
…	…	…

图 4-37 来自路由器 A 的通告

目的子网	下一台路由器	到目的地的跳数
w	A	2
y	B	2
z	A	5
…	…	…

图 4-38 收到路由器 A 的通告后路由器 D 中的转发表

我们下面考虑 RIP 实现方面的几个问题。前面讲过 RIP 路由器大约每 30 秒相互交互通告。如果一台路由器一旦超过 180 秒没有从邻居听到报文，则该邻居不再被认为是可达的；即要么其邻居死机了，要么连接的链路中断了。当这种情况发生时，RIP 修改本地路由选择表，然后通过向相邻路由器（那些仍然可达的路由器）发送通告来传播该信息。路由器也可通过使用 RIP 请求报文，请求其邻居到指定目的地的费用。路由器在 UDP 上使用端口 520 相互发送 RIP 请求与响应报文。封装在标准 IP 数据报中的 UDP 报文段在路由器之间传输。RIP 使用一个位于网络层协议（IP）之上的运输层协议（UDP）来实现网络层功能（一种路由选择算法），这个事实看起来似乎相当令人费解（事实如此！）。若再深入一些观察 RIP 的实现原理将能更明白这一点。

图 4-39 概略地说明了 RIP 在一个 UNIX 系统中通常是如何实现的，例如一台用作路由器的 UNIX 工作站。一个称为 routed 的进程执行 RIP，即维护路由选择信息并与相邻路由器中的 routed 进程交换报文。因为 RIP 是被当作一个应用层进程来实现的（虽然它是一个能操作 UNIX 内核中的转发表的特殊进程），它能在一个标准套接字上发送和接收报文，并且使用一个标准的运输层协议。如显示的那样，RIP 是一个运行在 UDP 上的应用层协议（参见第 2 章）。如果读者有意关注 RIP（或我们将很快学习的 OSPF 和 BGP 协议）的实现，请参阅［Quagga 2012］。