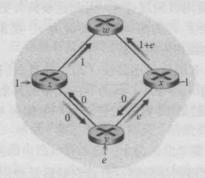
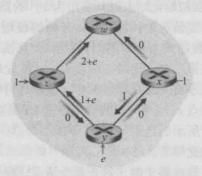
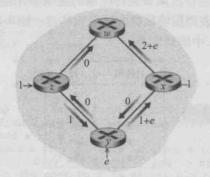
路径现在是顺时针的。类似地,x 确定其到w 的新的最低费用路径也是顺时针的,产生如图 4-29b 中所示的费用。当 LS 算法下次运行时,结点x、y 和z 都检测到一条至w 的逆时针方向零费用路径,它们都将其流量引导到逆时针方向的路由上。下次 LS 算法运行时,x、y 和z 都将其流量引导到顺时针方向的路由上。



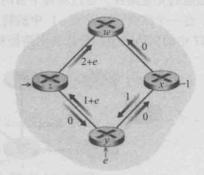
a) 初始路由选择



b) x, y检测到w的更好路径, 顺时针



c) x, y, z检测到w的更好路径, 逆时针



d) x, y, z检测到w的更好路径, 顺时针

图 4-29 拥塞敏感的路由选择的振荡

如何才能防止这样的振荡(它不只是出现在链路状态算法中,而且也可能出现在任何使用拥塞或基于时延的链路测度的算法中)。一种解决方案可能强制链路费用不依赖于所承载的流量,但那是一种不可接受的解决方案,因为路由选择的目标之一就是要避开高度拥塞(如高时延)的链路。另一种解决方案就是确保并非所有的路由器都同时运行 LS 算法。这似乎是一个更合理的方案,因为我们希望即使路由器以相同周期运行 LS 算法,在每台路由器上算法执行的时机也将是不同的。有趣的是,研究人员近来已注意到了因特网上的路由器能在它们之间进行自同步 [Floyd Synchronization 1994]。这就是说,即使它们初始时以同一周期但在不同时刻执行算法,算法执行时机最终会在路由器上变为同步并保持之。避免这种自同步的一种方法是,让每台路由器发送链路通告的时间随机化。

学习过 LS 算法之后,我们接下来考虑目前在实践中使用的其他重要的路由选择算法,即距离向量路由选择算法。

## 4.5.2 距离向量路由选择算法

距离向量(Distance-Vector, DV)算法是一种迭代的、异步的和分布式的算法,而LS