套接字编程作业

在第2章结尾给出了四个套接字编程作业。下面给出第5个应用ICMP的作业，ICMP的是本章讨论的协议。

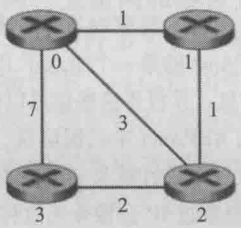
作业5：ICMP ping

ping是一种流行的网络应用程序，用于测试位于远程的某个特定的主机是否开机和可达。它也经常用于测量客户主机和目标主机之间的时延。它的工作过程是：向目标主机发送ICMP“回显请求”分组（即ping分组），并且侦听ICMP“回显响应”应答（即pong分组）。ping测量RRT、记录分组丢失和计算多个ping-pong交换（往返时间的最小、平均、最大和标准差）的统计汇总。

在本实验中，你将用Python语言编写自己的ping应用程序。你的应用程序将使用ICMP。但为了保持程序的简单，你将不完全遵循RFC 1739中的官方规范。注意到你将仅需要写该程序的客户程序，因为服务器侧所需的功能构建在几乎所有的操作系统中。你能够在Web站点http://www.awl.com/kurose-ross找到本作业的全面细节，以及该Python代码的重要片段。

编程作业

在本编程作业中，你要写一组的“分布式”过程，以为下图所示的网络实现一个分布式异步距离向量路由选择算法。



你要写出下列例程，这些例程将在为该作业提供的模拟环境中异步“执行”。对于结点0，你将写出这样的例程：

- rtinit0()。在模拟开始将调用一次该例程。rtinit0()无参数。它应当初始化结点0中的距离表，以反映出到达结点1、2和3的直接费用分别为1、3和7。在上图中，所有链路都是双向的，两个方向的费用皆相同。在初始化距离表和结点0的例程所需的其他数据结构后，它应向其直接连接的邻居（在本情况中为结点1、2和3）发送它到所有其他网络结点的最低费用路径的费用信息。通过调用例程tolayer2()，这种最低费用信息在一个路由选择更新分组中被发送给相邻结点，就像在完整编程作业中描述的那样。路由选择更新分组的格式也在完整编程作业中进行描述。
- rtupdate0(struct rtpkt *rcvdpkt)。当结点0收到一个由其直接相连邻居之一发给它的路由选择分组时，调用该例程。参数*rcvdpkt是一个指向接收分组的指针。rtupdate0()是距离向量算法的“核心”。它从其他结点i接收的路由选择更新分组中包含结点i到所有其他网络结点的当前最短路费用值。rtupdate0()使用这些收到的值来更新其自身的距离表（这是由距离向量算法所规定的）。如果它自己到另外结点的最低费用因此更新而发生改变的话，则结点0通过发送一个路由选择分组来通知其直接相连邻居这种最低费用的变化。我们在距离向量算法中讲过，仅有直接相连的结点才交换路由选择分组。因此，结点1和结点2将相互通信，但结点1和结点3将不相互通信。

为结点1、2、3定义类似的例程。因此你总共将写出8个过程：rtinit0()、rtinit1()、rtinit2()、rtinit3()、rtupdate0()、rtupdate1()、rtupdate2()和rtupdate3()。这些例程将共同实现一个分布式的、与图中所示拓扑和费用相关的距离表的异步计算。

你可在网址http://www.awl.com/kurose-ross处找到该编程作业的全部详细资料，以及你创建模拟硬件/软件环境所需的C程序代码。一个Java版的编程作业也可供使用。