P53. 在3.7节对TCP未来的讨论中，我们注意到为了取得10Gbps的吞吐量，TCP仅能容忍2×10^{-10}的报文段丢失率（或等价为每5 000 000 000个报文段有一个丢包事件）。给出针对3.7节中给定的RTT和MSS值的对2×10^{-10}值的推导。如果TCP需要支持一条100Gbps的连接，所能容忍的丢包率是多少？

P54. 在3.7节中对TCP拥塞控制的讨论中，我们隐含地假定TCP发送方总是有数据要发送。现在考虑下列情况，某TCP发送方发送大量数据，然后在t_1时刻变得空闲（因为它没有更多的数据要发送）。TCP在相对长的时间内保持空闲，然后在t_2时刻要发送更多的数据。当TCP在t_2开始发送数据时，让它使用在t_1时刻的cwnd和ssthresh值，将有什么样的优点和缺点？你建议使用什么样的方法？为什么？

P55. 在这个习题中我们研究是否UDP或TCP提供了某种程度的端点鉴别。

a. 考虑一台服务器接收到在一个UDP分组中的请求并对该请求进行响应（例如，如由DNS服务器所做的那样）。如果一个具有IP地址X的客户用地址Y进行哄骗的话，服务器将向何处发送它的响应？

b. 假定一台服务器接收到具有IP源地址Y的一个SYN，在用SYNACK响应之后，接收一个具有IP源地址Y和正确确认号的ACK。假设该服务器选择了一个随机初始序号并且没有“中间人”，该服务器能够确定该客户的确位于Y吗？（并且不在某个其他哄骗为Y的地址X。）

P56. 在这个习题中，我们考虑由TCP慢启动阶段引入的时延。考虑一个客户和一个Web服务器直接连接到速率R的一条链路。假定该客户要取回一个对象，其长度正好等于$15S$，其中S是最大段长度（MSS）。客户和服务器之间的往返时间表示为RTT（假设为常数）。忽略协议首部，确定在下列情况下取回该对象的时间（包括TCP连接创建）：

a. $4S/R > S/R + \text{RTT} > 2S/R$

b. $S/R + \text{RTT} > 4S/R$

c. $S/R > \text{RTT}$

编程作业

实现一个可靠运输协议

在这个编程作业实验中，你将要编写发送和接收运输层的代码，以实现一个简单的可靠数据运输协议。这个实验有两个版本，即比特交替协议版本和GBN版本。这个实验应当是有趣的，因为你的实现将与实际情况下所要求的差异很小。

因为可能没有你能够修改其操作系统的独立机器，你的代码将不得不在模拟的硬件/软件环境中执行。然而，为你提供例程的编程接口（即从上层和下层调用你的实体的代码），非常类似于在实际UNIX环境中做那些事情的接口。（实际上，在本编程作业中描述的软件接口比起许多教科书中描述的无限循环的发送方和接收方要真实得多。）停止和启动定时器也是模拟的，定时器中断将激活你的定时器处理例程。

Wireshark实验：探究TCP

在这个实验中，你将使用Web浏览器访问来自某Web服务器的一个文件。如同在前面的Wireshark实验中一样，你将使用Wireshark来俘获到达你计算机的分组。与前面实验不同的是，你也能够从该Web服务器下载一个Wireshark可读的分组踪迹，记载你从服务器下载文件的过程。在这个服务器踪迹文件里，你将发现自己访问该Web服务器所产生的分组。你将分析客户端和服务器端踪迹文件，以探究TCP的方方面面。特别是你将评估在你的计算机与该Web服务器之间TCP连接的性能。你将跟踪TCP窗口行为、推断分组丢失、重传、流控和拥塞控制行为并估计往返时间。

与所有的Wireshark实验一样，该实验的全面描述能够在本书Web站点http://www.awl.com/kurose-ross上找到。