**第四次作业**

**P32.**

答：

1. 当检测到数据包丢失时，阈值被设置为拥塞窗口值的一半。当在第16轮传输中检测到丢失时，拥塞窗口的大小为42。因此，在第18轮传输中，**阈值为21**。
2. 当检测到数据包丢失时，阈值被设置为拥塞窗口值的一半。当在第22轮传输中检测到丢失时，拥塞窗口的大小为26。因此，在第24轮传输过程中，**阈值为13**。
3. TCP 的 “慢开始” 在 **[1, 6] 和 [23, 26]** 这两个区间内运行。
4. 当丢包发生时，阈值将被设置为拥塞窗口当前值（8）的一半，拥塞窗口将被设置为新的阈值 +3 MSS。因此，**阈值和窗口的新值将分别为：4和7。**
5. 在第16轮传输之后，**数据包的丢失被一个三倍的重复ACK所识别**。因为：如果有超时，那么拥塞窗口大小将下降到1。
6. 在第22轮传输后，**由于超时而检测到段损失**，因此拥塞窗口大小被设置为1。
7. TCP的 congestion avoidance（拥塞避免）是在区间 **[6, 16] 和 [17, 22]** 运行的。
8. **阈值最初为32**，因为正是在这个窗口大小时，“慢开始”停止，“拥塞避免”开始。
9. 段传输的过程如下：

* 在第1个传输轮中，发送数据包1（1个包）；
* 在第2个传输轮中发送数据包2 – 3（2个包）；
* 在第3个传输轮中发送数据包4 – 7（4个包）；
* 在第4个传输轮中发送数据包8 – 15（8个包）；
* 在第5个传输轮中发送数据包16 – 31（16个包）；
* 第6个传输轮中发送数据包32 – 63（32个包）；
* 第7个传输轮中发送数据包64 – 96（32个包）。

因此，数据包70是在第7轮传输中发送的。

**P33.**

答：

本题答案如图1所示。在图1(a)中，连接1和连接2之间的损失线性下降的比率是相同的，在这种情况下，吞吐量从未脱离AB线段，未能达到 equal share。在图1(b)中，连接1和连接2之间的损失的线性下降的比率是2:1。也就是说，每当有损失的时候，连接1的窗口减少的量是连接2的两倍。在这种情况下，最终，经历了足够多的损失和随后的增加后，连接1的吞吐量将变为0，而全部的链接带宽将被分配给连接2。

 

1. **损失下降比例相同 （b）损失下降比例2：1**

**图1.**

**P34.**

答：

假设数据包n、n+1和n+2被发送，数据包n被收到并发送ACK。如果数据包n+1和n+2沿端到端路径重新排序（即以n+2，n+1的顺序接收），那么收到数据包n+2将产生n的重复ACK，在只等待第二次重复ACK的策略下，将触发重传。通过等待三重重复ACK：必须是在数据包n之后的两个数据包被正确收到，而n+1没有被收到。三重重复ACK方案的设计者可能认为，等待两个数据包（而不是1个）是在需要时触发快速重传，但在面对数据包重新排序时不会过早重传的正确权衡。

**P35.**

答：

如果TCP是一个停止-等待（Stop-Wait）协议，那么超时间隔的翻倍就足以作为一个用拥塞控制机制。然而，TCP使用流水线（不是一个停止-等待协议），它允许发送方有多个未确认的分段。超时间隔的加倍这种做法，即使在端到端路径高度拥塞的情况下，也不能阻止TCP发送者向网络发送大量的首次发送的数据包。因此，需要一个拥塞控制机制，以便在出现网络拥堵的迹象时阻止 "从上面的应用中收到的数据 "的流动。

**P36.**

答：

1. 需要1个RTT将CongWin增加到2MSS；2个RTT增加到3MSS；3个RTT增加到4MSS；4个RTT增加到5MSS；5个RTT增加到6MSS。
2. 在第一个RTT中，发送了1个MSS；在第二个RTT中，发送了2个MSS；在第三个RTT中，发送了3个MSS；在第四个RTT中，发送了4个MSS；在第五个RTT中，发送了5个MSS。

因此，直到第5个RTT时间，1+2+3+4+5=15个MSS被发送（和确认）。因此，我们可以说，到第5个RTT时间的平均吞吐量是（15MSS）/（5RTT）=3 MSS/RTT。

**P37.**

答：

由于接收方的接收缓冲区可以容纳整个文件，所以不存在溢出的危险。另外，由于没有丢失，并且在计时器到期前返回确认，TCP拥塞控制不会对发送方进行限制。然而，主机A的进程不会连续向套接字传递数据，因为发送缓冲区会很快填满，一旦发送缓冲区被填满，该进程将以平均速率或R<<S的速率传递数据。