Homework\_2 - Computer Network

张家成 2017013564

**Q1：The sender side of rdt3.0 simply ignores(that is, takes no action on) all received packets that are either in error or have the wrong value in the acknum ﬁeld of an acknowledgment packet. Suppose that in such circumstances, rdt3.0 were simply to retransmit the current data packet. Would the protocol still work?( Hint: Consider what would happen if there were only bit errors; there are no packet losses but premature timeouts can occur. Consider how many times the nth packet is sent, in the limit as n approaches inﬁnity.)**

**A1：**协议仍然是有效的，只不过重传次数过多会导致网络拥塞。问题的关键在于防止提前超时的发生，所以发送方选择多次重传数据，直到提前超时不再发生。

**Q2：Consider the GBN protocol with a sender window size of 4 and a sequence number range of 1,024. Suppose that at time t, the next in-order packet that the receiver is expecting has a sequence number of k. Assume that the medium does not reorder messages. Answer the following questions: a. What are the possible sets of sequence numbers inside the sender's window at time t? Justify your answer. b. What are all possible values of the ACK field in all possible messages currently propagating back to the sender at time t? Justify your answer.**

**A2：**

a： 如果所有这些ACK已经被发送方接收，那么发送方的窗口应该是[k, k+N-1]。假设下一步发送方没有收到任何ack，发送方的窗口包含k-1，因此发送方窗口为[k-N, k-1]。根据以上分析，发送方的起始位置在[k-4, k]范围内。也就是

[k-4, …, k-1], [k-3, …, k], [k-2, …, k+1], [k-1, …, k+2], [k, …, k+3]

b：之前的k-5包肯定已经被ACK确认，且发送者不会发送K包，因此可能的ACK 值域应为[k-4,…, k-1]。

**Q3：Answer true or false to the following questions and briefly justify your answer:**

**a. With the SR protocol, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.**

**b. With GBN, it is possible for the sender to receive an ACK for a packet that falls outside of its current window.**

**c. The alternating-bit protocol is the same as the SR protocol with a sender and receiver window size of 1.**

**d. The alternating-bit protocol is the same as the GBN protocol with a sender and receiver window size of 1.**

**A3：**

**a:** 正确。假设在t0时刻，窗口大小为3，发送方发送了包1，2，3. 在t1时刻(t1 > t0)，接收方对包1,2,3进行了ACK确认。在t2时刻(t2 > t1)， 发送方等待超时，重发了包1，2，3. 在t3时刻接收方又再次接收到了包1，2，3.在t4时刻，发送方接收到了t1时刻接收方发送过来的ACK，并且把它的窗口更新到了4，5，6。t5 时刻发送方接收到了接收方t2时刻发送过来的ACK，这些ACK在窗口以外。

**b：**正确。和a中的相同，GBN和SR协议都存在超时。

**c：**正确。窗口大小为1时，SR, GBN, 和替换位协议在功能上都是相同的。窗口大小为1排除了无序包存在的可能性。累积的ACK只是普通的ACK，因为它只能引用窗口中的单个包。

**d：**正确。解释同c

**Q4：Consider transferring an enormous file of L bytes from Host A to Host B. Assume an MSS of 536 bytes.**

**a. What is the maximum value of L such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.**

**b. For the L you obtain in (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network, and data-link header are added to each segment before the resulting packet is sent out over a 155 Mbps link. Ignore flow control and congestion control so A can pump out the segments back to back and continuously.**

**A4：**

**a：**L的最大值为2**32**

**b：**数据段的个数为232/536=8012998.68，取整就是8012999段。每一段数据都要负载66个字节的包头，所以总的数据大小为232+8012999\*66 = 4823825230 B

网速是155Mbps， 因此总的时间为 4823825230\*8 / (155 \* 1024 \* 1024) = 237.43s

**Q5：Consider sending a large file from a host to another over a TCP connection that has no loss. a. Suppose TCP uses AIMD for its congestion control without slow start. Assuming cwnd increases by 1 MSS every time a batch of ACKs is received and assuming approximately constant round-trip times, how long does it take for cwnd increase from 6 MSS to 12 MSS (assuming no loss events)? b. What is the average throughout (in terms of MSS and RTT) for this connection up through time = 6 RTT?**

**A5：**

**a：6 RTTs**

**b：6个RTTs 实现的吞吐量是（6+7+8+9+10+11）/6 = 8.5 MSS/RTT**

**Q6：Suppose that the UDP receiver computes the Internet checksum for the received UDP segment and finds that it matches the value carried in the checksum field. Can the receiver be absolutely certain that no bit errors have occurred? Explain.**

**A6：**UDP（用户数据报协议）中的接收方通过计算Internet 校验和并与校验字段中的值进行比较来验证接收到的段。和的反码被认为是校验和。所以，当这个校验和被用来检测数据包中的错误时，错误就会隐藏起来。在这种情况下，如果添加了两个16位的单词，那么就有翻转0和1的空间。如果位被翻转，和将是相同的。错误不能被检测到。

**Q7：Host A and B are communicating over a TCP connection, and Host B has already received from A all bytes up through byte 126. Suppose Host A then sends two segments to Host B back-to-back. The first and second segments contain 80 and 40 bytes of data, respectively. In the first segment, the sequence number is 127, the source port number is 302, and the destination port number is 80. Host B sends an acknowledgment whenever it receives a segment from Host A.**

**a. In the second segment sent from Host A to B, what are the sequence number, source port number, and destination port number?**

**b. If the first segment arrives before the second segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number, the source port number, and the destination port number?**

**c. If the second segment arrives before the first segment, in the acknowledgment of the first arriving segment, what is the acknowledgment number?**

**d. Suppose the two segments sent by A arrive in order at B. The first acknowledgment is lost and the second acknowledgment arrives after the first timeout interval. Draw a timing diagram, showing these segments and all other segments and acknowledgments sent. (Assume there is no additional packet loss.) For each segment in your figure, provide the sequence number and the number of bytes of data; for each acknowledgment that you add, provide the acknowledgment number**

**A7:**

**a：在第二段中，序列号是207， 源端口号是302，目的端口号是80**

**b：ACK 号是207， 源端口号是80，目的端口号是207**

**c：acknowledgement number 为127，表示它仍然在等待127字节。**

**d：**

