Chapter 5 End-to-End Protocols

1. When TCP sends a <SYN, SequenceNum=x> or <FIN, SequenceNum=x>, the consequent ACK has Acknowledgment = x+1, that is, SYNs and FINs each take up one unit in sequence number space. Is this necessary? If so, give an example of an ambiguity that would arise if the corresponding Acknowledgment were x instead of x+1; if not, explain why.

   Incrementing the Ack number for a FIN is essential, so that the sender of the FIN can determine that the FIN was received and not just the preceding data. For a SYN, any ACK of subsequent data would increment the acknowledgment number, and any such ACK would implicitly acknowledge the SYN as well (data cannot be ACKed until the connection is established). Thus, the incrementing of the sequence number here is a matter of convention and consistency rather than design necessity.

2. Consider an ARQ protocol that uses only negative acknowledgments (NAKs), but no positive acknowledgments (ACKs). Describe what timeouts would need to be scheduled. Explain why an ACK-based protocol is usually preferred to a NAK=based protocol.

   Assume a NAK is sent only when an out-of-order packet arrives. The receiver must now maintain a RESEND NAK timer in case the NAK, or the packed it NAK'ed, is lost. Unfortunately, if the sender sends a packet and is then idle for a while, and this packet is lost, the receiver has no way of noticing the loss. Either the sender must maintain a timeout anyway, requiring ACKs, or else some zero-data filler packets must be sent during idle times. Both are burdensome.
   Finally, at the end of the transmission a strict NAK-only strategy would leave the sender unsure about whether any packets got through. A final out-of-order filler packet, however, might solve this.

3. Suppose you are designing a sliding window protocol for a 1Mbps point-to-point link to the stationary satellite evolving around Earth at 3x10^{4}km altitude. Assuming that each frame carries 1KB of data. What is the maximum SWS? What is the minimum number of bits you need for the sequence number in the following cases when the pipe reaches its maximum capacity? Assume the speed of light is 3x108 meters per second.
   (a) RWS=1
   (b) RWS=SWS

   From the RTT-Bandwidth product, we obtain that 25 frames can be in transit (SWS=25) when the pipe reaches its capacity. With 25 frames, for case (a), we need 26 sequence numbers. Therefore 5 bits are needed for the sequence numbers; for case (b), we need 50 sequence numbers, and therefore 6 bits are needed.

4. In TCP, will the loss of an ACK of a packet always result in a retransmission of that packet? Why?
   No. Remember that ACK in TCP is cumulative. So if the ACK for the next packet in sequence arrives before this packet's retransmission timer expires, this packet does not get retransmitted.

5. Imagine a receiver in a TCP session with ***fixed*** window size of 5000 bytes. Suppose the ***next expected*** byte sequence number is 13000. Assume that the segment size is 1000 bytes.

Segments may get ***lost*** in the network but are ***not reordered***, i.e., they do not reach the receiver out of order.
For each of the following numbers, specify whether it is possible for the receiver to receive a packet with that ***sequence number***? If so, will the receiver discard or buffer the segment? What will be the corresponding number in the acknowledgment it sends?

SEQ #	Possible?	Discarded?	ACK #
11000	Yes	Yes	13000
13000	Yes	No	14000
15000	Yes	No	13000
18000	No		

6. Imagine a sender in a TCP session with ***fixed*** window size of 5000 bytes. Suppose its current sliding window ***begins*** at byte sequence number 13000. As above, size of each segment is 1000 bytes and they may get ***lost*** but are ***not reordered***.
For each of the following numbers, specify whether it is possible for the sender to receive a segment with that ***acknowledgment number***? If so, will the sender slide its window to the right and what will be the new ***beginning*** sequence number of its sliding window.

ACK #	Possible?	Slide?	SEQ #
11000	No		
13000	Yes	No	
15000	Yes	Yes	15000
18000	Yes	Yes	18000

7. You are hired to design a reliable byte-stream protocol that uses a sliding window (like TCP). This protocol will run over a 1-Gbps network. The RTT of the network is 140ms, and the maximum segment lifetime is 60 seconds. How many bits would you include in the AdvertisedWindow and SequenceNum fields of the protocol header?

The advertised window should be large enough to keep the pipe full; delay (RTT) × bandwidth here is 140ms × 1Gbps = 10Mb = 17.5 MB of data. This requires 25 bits for the AdvertisedWindow field. The sequence number field must not wrap around in the maximum segment lifetime. In 60 seconds, 7.5GB can be transmitted. 33 bits allows a sequence space of 8.6GB, and so will not wrap in 60 seconds.

8. If host A receives two SYN packets from the same port from remote host B, the second may be either a retransmission of the original or else, if B has crashed and rebooted, an entirely new connection request. Describe the difference as seen by host A between these two cases.
If a SYN packet is simply a duplicate, its ISN value will be the same as the initial ISN. If the SYN is not a duplicate, and ISN values are clockgenerated, then the second SYN's ISN will be different.

9. Suppose in TCP's adaptive retransmission mechanism, that **EstimatedRTT** is 4.0 at some point and subsequent measured RTTs all are 1.0. How long dowa it take before the **TimeOut** value, as calculated by the Jacobson/Karels algorithm, falls below 4.0? Assume a plausible initial value of Deviation; how sensitive is your answer to this choice? Use $\delta = 1/8$.