Solutions to Homework Questions on Transport Layer Part 1

Q1. Which protocol – Go-Back-N or Selective-Repeat - makes more efficient use of network bandwidth? Why?

A1. Selective repeat makes more efficient use of network bandwidth since it only retransmits those messages lost at the receiver (or prematurely timed out). In Go-Back-N, the sender retransmits the first lost (or prematurely timed out) message as well as all following messages (without regard to whether or not they have been received).

Q2. Consider a reliable data transfer protocol that uses only negative acknowledgements. Suppose the sender sends data only infrequently. Would a NAK-only protocol be preferable to a protocol that uses ACKs? Why? Now suppose the sender has a lot of data to send and the end-to-end connection experiences few losses. In this second case, would a NAK-only protocol be preferable to a protocol that uses ACKs? Why?

A2. In a NAK only protocol, the loss of packet x is only detected by the receiver when packet x+1 is received. That is, the receiver receives x-1 and then x+1, only when x+1 is received does the receiver realizes that x was missed. If there is a long delay between the transmission of x and the transmission of x+1, then it will be a long time until x can be recovered, under a NAK only protocol.

On the other hand, if data is being sent often, then recovery under a NAK-only scheme could happen quickly. Moreover, if errors are infrequent, then NAKs are only occasionally sent (when needed), and ACK are never sent – a significant reduction in feedback in the NAK-only case over the ACK-only case.

Q3. If the RTT from London to Cape Sydney is 120ms and all links in the network have a 155 Mbits/second data-rate, how much data can fit in the "pipe"? Express your answer in bytes.

A3. 120 ms x 155 Mbits/sec = 18.6 x 10_6 bits = 2,325,000 bytes will fit in the pipe.

Q4. A reliable transport protocol is using Selective Repeat with 8-bit sequence numbers. What is the largest allowable sender window that will prevent the risk of accepting duplicate data as new in the receiver?

A4. Sender window must be less than or equal to half of the sequence space, which is 2^8 = 256. Therefore, the largest allowable window is 128.

Q5. Two 16-bit words 1011 0101 1010 1000 and 0101 1001 0000 0101 are received, along with another 16-bit word, 1101 0001 0101 0001, which is the UDP checksum of the first two words. Will the receiver detect an error?

A5. The sum of the first two words is 0000 1110 1010 1110 and the UDP checksum should be 1111 0001 0101 0001 (1's complement of the sum). This is not the same as the received UDP checksum. The receiver will detect an error.

Q6. Is it possible for an application to enjoy reliable data transfer even when the application runs over UDP? If so, how?

A6. Only if the application itself implements reliability tools, such as ACK, retransmission, timer, etc.

Q7. 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 sure that no bit errors have occurred? Explain. Would things be different with TCP?

A7. No, the receiver cannot be absolutely certain that no bit errors have occurred. This is because of the manner in which the checksum for the packet is calculated. If the corresponding bits (that would be added together) of two 16-bit words in the packet were 0 and 1 then even if these get flipped to 1 and 0 respectively, the sum still remains the same. Hence, the 1s complement the receiver calculates will also be the same. This means the checksum will verify even if there was transmission error. Since TCP uses the same checksum mechanism, the above would hold true with TCP as well.

End of homework