

The University of Queensland
School of Information Technology and Electrical Engineering
Semester Two, 2009

COMS3200/7201 – Tutorial 3 - Solutions

Question 1

Why does UDP exist? Would it not have been enough to just let user processes send raw IP packets?

The main components of the UDP header are the source and destination port numbers which allow packets to be associated with particular processes. Use of raw IP packets wouldn't allow the receiver to determine the process to which the packet was directed.

Question 2

(PD 5.5) When closing a TCP connection, why is the two-segment-lifetime timeout not necessary on the transition from LAST_ACK to CLOSED?

The two-segment-lifetime timeout results from the uncertainty of the sender of the last ACK as to whether it was received by the other side. That other side will be waiting in the LAST_ACK state - if it does receive the ACK, it knows there will be no further communication - it can move immediately from LAST_ACK to CLOSED.

Question 3

(variant of PD5.9) You are hired to design a reliable byte-stream protocol that uses a sliding window (like TCP). This protocol will run over a 100Mbps network. The round-trip-time (RTT) of the network is 100ms, and the maximum segment lifetime (MSL) is 60 seconds.

(a) How many bits would you include in the AdvertisedWindow and SequenceNumber fields of your protocol header?

(b) How would you determine the MSL and RTT given above and how certain could you be of those values?

(a) The advertised window should be large enough to keep the pipe full; delay (RTT) x bandwidth here is 100ms x 100Mbps = 10Mb = 1.25MB of data. This requires 21 bits ($2^{21} = 2,097,152$) for the AdvertisedWindow field. The sequence number field must not wrap around in the maximum segment lifetime. In 60 seconds, 750MB can be transmitted. 30 bits allows a sequence space of 1024MB (1 gigabyte), and so will not wrap in 60 seconds. (If the maximum segment lifetime were not an issue, the sequence number field would still need to be large enough to support twice the maximum window size (see pages 108-109 in text).)

(b) The RTT can be determined based on the specifications of the links and intermediate nodes on the network. This is reasonably certain, but will change on future changes in the size of the network. The MSL can be determined similarly, but is less certain as it depends on the size and complexity of the network, and on how long it takes routing loops to be resolved.

Question 4

(variant of PD5.12) If host A receives two SYN packets from the same port from remote host B, how do you determine whether the second is a retransmission or an entirely new connection request?

If a SYN packet is a duplicate its sequence number will be the same as the initial sequence number. If the SYN is not a duplicate, its sequence number should be different. (This will definitely be the case if initial sequence numbers are clock-generated.)

Question 5

Initial TCP sequence numbers specified during connection are often generated according to a 32 bit clock incremented every 4 microseconds. How often do these sequence numbers wrap around?

The clock has $2^{32} = 4294967296$ possible values. If the clock is incremented every 4 microsec-

onds, the sequence numbers will wrap around every

$$4.29 \times 10^9 \times 4 \times 10^{-6} = 17.2 \times 10^3 \text{ seconds} \approx 4 \frac{3}{4} \text{ hours}$$

Remember, this is the initial sequence number. Future sequence numbers for the connection are determined by counting bytes.

Question 6

TCP Maximum segment lifetime (MSL) – the maximum time a segment exists out on the network is taken to be 120 seconds. When starting or recovering from a crash, TCP entities must wait 120 seconds before sending any segments. Why? How might this restriction be overcome?

There must not be any possibility of a segment being transmitted that is confused with a segment from a previous connection. If the system has no record of what the last sequence number used was, it must wait the full 120 second before sending any segments. This could be overcome by having the system keep a non-volatile record (i.e. one that can survive a crash) of the last (or a recent) sequence number. Upon recovery, the system would be able to generate a new sequence number that it knows won't match the segments generated in any previous connection.