COMP90007 Internet Technologies
Week 8 Workshop

Semester 2, 2018

Suggested solutions

Question 1

Why does the maximum packet lifetime, *T*, have to be large enough to ensure that not only the packet but also its acknowledgements have vanished?

Answer:

Look at the second duplicate packet in Fig. 6-11(c). When the packet arrives, it would be a disaster if acknowledgements to *y* were still floating around.

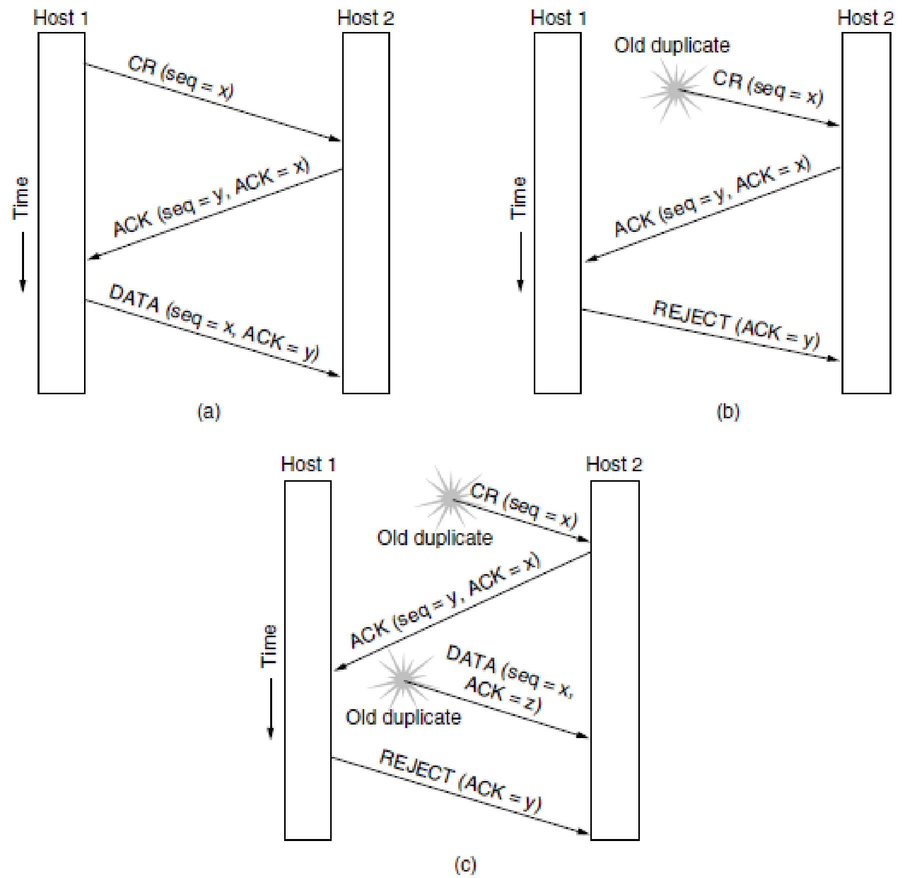

Figure 6-11. Three protocol scenarios for establishing a connection using a three-way handshake. CR denotes CONNECTION REQUEST. (a) Normal operation. (b) Old duplicate CONNECTION REQUEST appearing out of nowhere. (c) Duplicate CONNECTION REQUEST and duplicate ACK.

Question 2

Imagine that a two-way handshake, rather than a three-way handshake were used to set up connections. In other words, the third message was not required. Are deadlocks now possible? Give an example or show that none exist.

Answer:

Deadlocks are possible.

For example, a packet arrives at A out of the blue, and A acknowledges it. The acknowledgement gets lost, but A is now open while B knows nothing at all about what has happened. Now the same thing happens to B, and both are open, but expecting different sequence numbers. Timeouts have to be introduced to avoid the deadlocks.

Question 3

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

Answer:

No. IP packets contain IP addresses, which specify a destination machine. Once such a packet arrived, how would the network handler know which process to give it to? UDP packets contain a destination port. This information is essential so they can be delivered to the correct process.

Question 4

A client sends a 128 byte request to a server located 100 km away over a 1 gigabit optical fibre. What is the efficiency of the line during the remote procedure call?

Answer:

- Sending 1024 bits over a 1 Gbps line takes about 1 µs (one way and not RTT)
- Assuming the speed of information propagation in fibre is 2/3 the speed of light, i.e. 2×10^8 m/s, then the round trip time will be: $2 \times 100 \times 10^3$ m / 2×10^8 m/s = 1 ms
- Efficiency = 1 µs / 1 ms = 0.1%

Question 5

Both UDP and TCP use port numbers to identify the destination entity when delivering a message. Give two reasons for why these protocols invented a new abstract ID (port numbers), instead of using process IDs, which already existed when these protocols were designed?

Answer:

Here are three reasons.

- First, process IDs are OS-specific. Using process IDs would have made these protocols OS-dependent.
- Second, a single process may establish multiple channels of communications. A single process ID (per process) as the destination identifier cannot be used to distinguish between these channels.
- Third, having processes listen on well known ports is easy, but well-known process IDs are impossible.

Question 6

Datagram fragmentation and reassembly are handled by IP and are invisible to TCP. Does this mean the TCP does not have to worry about data arriving in the wrong order?

Answer:

Even though *each datagram* arrives intact, it is possible that the datagrams arrive in the wrong order, so TCP has to be prepared to reassemble the parts of the message properly.

Question 7

A process on host 1 has been assigned port *p*, and a process on host 2 has been assigned port *q*. Is it possible for there to be two or more TCP connections between these two ports at the same time?

Answer:
No. A connection is identified only by its sockets. Thus, (1, p) → (2, q) is the only possible connection between those two ports.

Question 8

The maximum payload of a TCP segment is 65,495 bytes. Why was such a strange number chosen?

Answer:

- The maximum IP packet size is 65,535
- Since the header is 20 bytes, the entire TCP segment must fit in the 65,515 byte payload field of an IP packet.
- Since the TCP header is a minimum of 20 bytes, only 65,495 bytes are left for TCP data.

Question 9

To get around the problem of sequence numbers wrapping around while old packets still exist, one could use 64 bit sequence numbers. However, theoretically, an optical fibre can run at 75 Tbps.

What maximum packet lifetime is required to make sure that future 75 Tbps networks do not have wraparound problems even with 64 bit sequence numbers? Assume that each byte has its own sequence number, as TCP does.

Answer:

- The sequence space is 2^{64}. Each byte is assigned a unique sequence number, so the sequence numbers can represent 2^{64} unique bytes.
- A 75 Tbps transmitter can output 9.375×10^{12} bytes/s.
- $t = 2^{64} / 9.375 \times 10^{12} = 1.96 \times 10^{6}$ s = 22.77 days
- It takes 22.77 days, which is over three weeks, to wrap around. A maximum packet lifetime of less than 22.77 days will prevent the problem. In short, going to 64 bits is going to work for quite a while.