Answers to the October 9, 2015 test of Pearls of Computer Science

P.T. de Boer, UT/EWI/DACS, 2015-10-23

---

1. Operating Systems

(a) Input/output deals with controlling peripheral equipment, such as screen, keyboard, mouse, printer, etc.

(b) Yes, because also different programs from a single user need to be protected from each other, against e.g. accidentally overwriting each other's memory.

(c) Yes: not all parts of the operating system are continuously in use (e.g., the printer driver if you're not printing continuously), and those parts could be swapped to disk.

---

2. Networks – protocols

(a) Layering means dividing the system into layers, where each layer use the services offered by a lower layer, to provide a better service to the higher layer. This makes it much simpler to add a new application: the new application can simply use the reliable transport service offered by the existing transport layer, instead of having to implement all that complexity itself.

(b) No, the TCP connection will not survive this. This is because the laptop gets a different IP address when it is connected to the home network. The server does not know this, so it will keep sending its packets to the laptop's old address, and those packets will be routed to the UT campus, not to the home, and thus never reach the laptop.

(c) Either the first packet, or its acknowledgement (with `Ack=280`) is lost, since we don't see such an acknowledgement in the trace. Note that we can't distinguish between these two possibilities.

Furthermore, the 4th packet is lost. It's an acknowledgement sent by "our" side (i.e., the computer on which Wireshark is run) for the 3rd packet. However, you can see that the 5th packet is identical to the 3rd, so apparently it was retransmitted by the other side, so apparently the other side did not get the 4th packet.

> Purely judging by the trace, it could also be that the other side's retransmit timeout could was too short, so the 4th packet wasn't really lost but simply arrived too late; but in the question it was given that 2 packets are lost.
>
> Note that the 3rd packet cannot have been lost: it is a packet destined for 130.89.13.213, and it was given that Wireshark is run on that host. So the fact that we see the packet, proves that it has reached its destination.
>
> Also, some students said that what is missing is a packet acknowledging the last packet shown here. However, that last packet is a pure acknowledgement; it does not contain any application-layer data (`Len=0`), so there's no need to acknowledge it.

Continued on next page...

3. Networks – delay

Remark: the text and the picture disagree about the speed of the C-D link; I intended 2 Mb/s, but calculating consistently with 1 Mb/s is of course also correct.

(a)

- Processing delay is the time the routers need for calculations, and this was negligible as given in the text.
- Propagation delay was negligible on A-B and C-D because of small geographical distance; on B-C it was 5 ms.
- Transmission delay was 1 ms on A-B, 2 ms on B-C, and 0.5 ms on C-D.

(b)

- Not in A because packets are generated just as slowly as they can be sent.
- Not in C because the outgoing link is faster than the incoming link.
- But there is queueing in B, because packets arrive here at 1 ms intervals, but need 2 ms to be transmitted.

  The fact that the incoming link of B is faster than its outgoing link by itself is not enough to guarantee that a queue builds up: that only happens if packets indeed arrive at a sufficiently high rate.

(c) Let's start at $t = 0$. Then at $t = 1$ ms, the first packet has reached B, and B can start transmitting it to C, which will finish at $t = 3$ ms.
The second packet was generated at $t = 1$, and arrives at B at $t = 2$, but it has to wait until $t = 3$ before its transmission to C can start; at $t = 5$ its last bit has been sent by B.
In the mean time, the third packet was generated at $t = 2$ and arrived at B at $t = 3$; its transmission starts at $t = 5$, and is completed at $t = 7$.
So, at $t = 7$ the last bit has left node B; it still needs 5 ms to travel to node C (propagation time), and then another 0.5 ms for the transmission from C to D (there's no queue at C, as we noticed before). So the last bit arrives in D at $t = 7 + 5 + 0.5 = 12.5$ ms.

Some frequently made errors:

- letting the packets in B not just wait for the transmission time, but also for the propagation time (that is like not sending the next car onto a road until the previous one has left the highway at the other end)
- completely forgetting one of the delay contributions;
- simply adding all transmission times of all packets, as if the links cannot be used simultaneously;
- writing an answer with so little or so unclear explanation that I can't follow your reasoning.