Network Systems (201300179), Test 1

February 14, 2014, 15:45–17:15

Brief answers

1. Performance and reliable data transfer

4 pt (a)

propagation delay dominates; 1000 seconds

4 pt (b)

dtrans=80000 s dominates; total 81000 s

4 pt (c)

RTT=2000s; 1 packet takes 0.1 s, so SWS=20000 sequence numbers is needed to be able to keep the pipe full

2 pt (d)

RWS=SWS; otherwise, lost packets could cause later packets to be retransmitted even if they were received correctly at first, which especially with such a large delay bandwidth product would be inefficient

4 pt (e)

Many aspects can be mentioned here, see also the study material. The main disadvantage of circuit switching to be mentioned here, is the lack of statistical multiplexing: when one rover is not transmitting anything, the link from the orbiter to the earth cannot be fully used.

2. Information theory

4 pt (a)

0.2866 bits

4 pt (b)

The capacity of this channel is $(1 + .4 * \log_2(.4)/\log_2(2) + .6 * \log_2(.6)/\log_2(2)) * 10 = 0.29$ bits/s. This is more than needed for the reports (as calculated in the previous question), so the error probability can be made as close to 0 as desired.

4 pt (c)

0 for Dry, 10 for Water, 11 for Life; takes $1 \cdot .95 + 2 \cdot .05 = 1.05$ bits on average.

4 pt (d)

This question requires some ingenuity. The only way to achieve this, is to not try to make codes for single messages, but make codes for groups of e.g. four messages. One possibility is: let a single '1' mean four times Dry; and let '0' mean that what follows is four messages coded according to the previous question.

Proof (but note that the question didn't ask for this): with probability $0.95^4 = 0.8145$, the three messages are all Dry, and thus together take 1 bit. In the remaining cases, at least one message is not Dry and thus takes 2 bits; the other three messages on average again take 1.05 bits each, making the total 1 (for the '0' bit) + 2 (for the non-Dry message) + $3 \cdot 1.05 = 6.15$ bits. So the total average number of bits is $1 \cdot 0.8145 + 6.15 \cdot 0.1855 = 1.955$ bits, for *four* messages; that's less than 0.5 bits per message.

Many students answered something like "send nothing for Dry, '0' for Water, '1' for Life". This indeed reduces the average message length to about 0.05 bits, but it does not work; suppose the receiver receives "00100", how does it know how many times Dry was sent between e.g. the first two Water messages?

---

3. Peer-to-peer applications

2 pt (a)

Scalability.

5 pt (b)

See book, where it is explained why each of the terms is a lower bound on the distribution time.

---

4. HTTP

2 pt (a)

On the client.

3 pt (b) If HTTP 1.0 (without persistent connections and without parallel connections) is used to fetch this homepage, how many Round Trip Times (RTT) are needed before the complete page can be displayed in the browser? Explain your answer.

1 to open connection, 1 to fetch html page, 1 to open connection for 1st image, 1 to fetch first image, 1 to open connection for 2nd image, 1 to fetch second image. Total: 6 RTTs.

2 pt (c)

1 to open connection, 1 to fetch html page, 1 to fetch 2 images in parallel, total: 3 RTTs.

3 pt (d)

When using pipelining, multiple HTTP requests can already be sent in parallel (before receiving the response). So, using parallel connections does not improve performance. The only case where using parallel connections could improve performance is if the size of the (receive) window is limiting the performance. In this case, parallel connections may allow the server to have more data oustanding (unacknowledged).
