Homework 1

Adrian Darian

9/15/2020

Chapter 1

3 Calculate the total time required to transfer a $1000Kb$ file in the following cases, assuming an RTT of $50byte$, a packet size of $1KB$ data, and an initial $2 \times RTT$ of "handshaking" before data is sent:

(a) The bandwidth is $1.5Mbps$, and data packets can be sent continuously.
Answer: $100ms+1000Kb/1.5KBps = 0.125+8Mbit/15Mbps = 0.125+5.333sec = 5.458sec$. A mega is $10^6 vs. 2^20$, we get $8,192,000bits/1,500,000bps = 5.461sec = 5.586sec$

(b) The bandwidth is $1.5Mbps$, but after we finish sending each data packet we must wait one RTT before sending the next.
Answer: 999 RTTs, for a total of $5.586 + 49.95 = 55.536$.

(c) The bandwidth is "infinite," meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT.
Answer: This is $49.5RTTs$, plus the initial 2, for $2.575seconds$

(d) The bandwidth is infinite, and during the first RTT we cansend one packet (21-1), during the second RTT we can send two packets (22-1), during the third we can send four (23-1),and so on. (A justification for such an exponential increase will be given in Chapter 6.)
Answer: $1 + 2 + 4 + ... + 2^n = 2^{n+1} - 1packets$. At $n = 9$ we have thus been able to send all $1,000packets$; the last batch arrives $0.5RTT$ later. Total time is $2 + 9.5RTTs$, or $.575sec$

5 Consider a point-to-point link $4km$ in length. At what bandwidth would propagation delay (at a speed of $2 \times 10^8 m/s$) equal transmit delay for $100byte$ packets? What about $512byte$ packets?
Answer: $4 \times 10^3 m/2 \times 10^8 m/s = 2x10^{-5} sec = 20\mu s$. $100bytes/20\mu s$ is $5bytes/\mu s$. For $512byte = 204.8Mbps$

11 How "wide" is a bit on a $10 - Gbps$ link? How long is a bit in copperwire, where the speed of propagation is $2.3x10^8 m/s$?
Answer: width is $10^{-10} sec(0.1ns)$. length is $.1nsx2.3x10^8 m/sec = 0.023mor23mm$

13 Suppose a $1 - Gbps$ point-to-point link is being set up between theEarth and a new lunar colony. The distance from the moon to theEarth is approximately $385,000km$, and data travels over the linkat the speed of light—$3x10^8 m/s$.

(a) Calculate the minimum RTT for the link.
Answer: $2 \times 385,000,000m/3 \times 10^8 m/s = 2.57seconds$

(b) Using the RTT as the delay, calculate the $delay * bandwidth$ product for the link.
Answer: $2.57byte \times 1Gbps = 2.57Gb = 321Mb$

(c) What is the significance of the $delay * bandwidth$ product computed in (b)?
Answer: amount of data that can be sent prior to response

(d) A camera on the lunar base takes pictures of the Earth andsaves them in digital format to disk. Suppose Mission Controlon Earth wishes to download the most current image, which is $25MB$. What is the minimum amount of time that willelapse between when the request for the data goes out andthe transfer is finished?
Answer: $200Mb/1000Mbps = 0.2seconds$. total time of $0.2 + 2.57 = 2.77sec$

18 Calculate the effective bandwidth for the following cases. For (a) and (b) assume there is a steady supply of data to send; for(c) simply calculate the average over $12hours$.

(a) $100 - Mbps$ Ethernet through three store-and-forward switches as in Exercise 16(b). Switches can send on one link while receiving on the other.
Answer: The effective bandwidth is $100Mbps$; the sender can send data steadily at this rate and the switches simply stream it along the pipeline. We are assuming here that no ACKs are sent, and that the switches can keep up and can buffer at least one packet

(b) Same as (a) but with the sender having to wait for a 50-byteacknowledgment packet after sending each 12,000-bit datapacket.
Answer: The data packet takes $520\mu s$; the $400bit$ ACKs take $4\mu s$/link to be sent back, plus propagation, for a total of $4 \times 4\mu s + 4 \times 10\mu s = 56\mu s$; thus the total RTT is $576\mu s$. $12000bits$ in $576\mu s$ is about $20.8Mbps$.

(c) Overnight ($12hour$) shipment of $100DVDs$ that hold $4.7GB$ each
Answer: $100 \times 4.7 \times 10^9 bytes/12hours = 4.7 \times 10^11 bytes/(12 \times 3600s) = 10.9MBps = 87Mbps$

19 Calculate the $delay * bandwidth$ product for the following links. Use one-way delay, measured from first bit sent to first bit received

(a) $100 - Mbps$ Ethernet with a delay of $10\mu s$.
Answer: $100 \times 10^6 bps \times 10 \times 10^{-6} sec = 1000bits = 125bytes$

(b) $100 - Mbps$ Ethernet with a single store-and-forward switchlike that of Exercise 16(b), packet size of $12,000bits$, and $10\mu s$ per link propagation delay.
Answer: $100 \times 10^6 bps \times 520 \times 10^{-6} sec = 52000bits = 650bytes$.

(c) $1.5Mbps$ T1 link, with a transcontinental one-way delay of $50ms$.
Answer: $1.5 \times 10^6 bps \times 50 \times 10^{-3} sec = 75,000bits = 9375bytes$

(d) $1.5Mbps$ T1 link between two ground stations communicating via a satellite in geosynchronous orbit, $35,900km$ high. The only delay is speed-of-light propagation delay from Earth to the satellite and back.
Answer: $2 \times 35,900,000meters$. With a propagation speed of $c = 3 \times 10^8 meters/sec$, the one-way propagation delay is thus $2 \times 35,900,000/c = 0.24sec$. Bandwidth x delay is thus $1.5 \times 10^6 bps \times 0.24sec = 360,000bits = 45KBytes$

Chapter 2

2 Show the $4B/5B$ encoding, and the resulting NRZI signal, for the following bit sequence: 1110010100000011

4 In the $4B/5B$ encoding (Table 2.2), only two of the $5bit$ codesused end in two 0s. How many possible $5bit$ sequences are there (used by the existing code or not) that meet the stronger restriction of having at most one leading and at most one trailing 0? Could all $4bit$ sequences be mapped to such $5bit$ sequences?
Answer: One can list all $5bit$ sequences and count, but here is another approach: there are 2^3 sequences that start with 00, and 2^3 that end with 00. There are two sequences, 00000 and 00100, that do both. Thus, the number that do either is $8 + 8 - 2 = 14$, and finally the number that do neither is $32 - 14 = 18$. Thus there would have been enough $5bit$ codes meeting the stronger requirement; however, additional codes are needed for control sequences

11 Show that two-dimensional parity allows detection of all $3bit$ errors
Answer: Suppose an undetectable three-bit error occurs. The three bad bits must be spread among one, two, or three rows. If these bits occupy two or three rows, then some row must have exactly one bad bit, which would be detected by the parity bit for that row. But if the three bits are all in one row, then that row must again have a parity error (as must each of the three columns containing the bad bits)

12 Give an example of a $4bit$ error that would not be detected by two-dimensional parity, as illustrated in Figure 2.14. What is thegeneral set of circumstances under which $4bit$ errors will be undetected?
Answer: If we flip the bits corresponding to the corners of a rectangle in the $2 - D$ layout of the data, then all parity bits will still be correct. Furthermore, if four bits change and no error is detected, then the bad bits must form a rectangle: in order for the error to go undetected, each row and column must have no errors or exactly two errors

18 Suppose we want to transmit the message 11100011 and protect it from errors using the CRC polynomial $x^3 + 1$.

(a) Use polynomial long division to determine the message that should be transmitted.
Answer: If we flip the bits corresponding to the corners of a rectangle in the $2 - D$ layout of the data, then all parity bits will still be correct. Furthermore, if four bits change and no error is detected, then the bad bits must form a rectangle: in order for the error to go undetected, each row and column must have no errors or exactly two errors

(b) Suppose the leftmost bit of the message is inverted due tonoise on the transmission link. What is the result of the receiver's CRC calculation? How does the receiver know thatan error has occurred?
Answer: Inverting the first bit of the transmission gives 01100011100; dividing by $1001(x^3+1)$ gives a remainder of 10; the fact that the remainder is non zero tells us a bit error occurred

23 Consider an ARQ algorithm running over a $40km$ point-to-pointfiber link.

(a) Compute the one-way propagation delay for this link, assuming that the speed of light is $2 \times 10^8 m/s$ in the fiber.
Answer: $40 \times 103m/(2 \times 108m/s) = 200\mu s$

(b) Suggest a suitable timeout value for the ARQ algorithm to use.
Answer: The roundtrip time would be about $400\mu s$. A plausible timeout time would be twice this, or $0.8ms$. Smaller values (but larger than $0.4ms$!) might be reasonable, depending on the amount of variation in actual RTTs.

(c) Why might it still be possible for the ARQ algorithm to timeout and retransmit a frame, given this timeout value?
Answer: The propagation-delay calculation does not consider processing delays that may be introduced by the remote node; it may not be able to answer immediately

24 Suppose you are designing a sliding window protocol for a $1Mbps$ point-to-point link to the moon, which has a one-way latency of $1.25seconds$. Assuming that each frame carries $1KB$ of data, what is the minimum number of bits you need for the sequence number?
Answer: $125KBps \times 2.5s = 312KB = 624.10bits$

Extra Credit:

Let's consider the cost model for flooding to be directly proportional to the total number of times nodes have to process a packet (either send or receive). What is the order of the worst and best case cost of sending a message from any node to any other node (not including reply messages) within this model? What are the topologies that produces these two cases? Assume there are N nodes in the network; the answers should be in terms of O(f(N)). Trivial cases such as node A sends to node B, which is its only neighbor, will not be accepted, because they cannot be generalized!