**Fall 2018 CS120 Homework Assignment #1**

**Handout Date: Sep. 20, 2018 Due Date: Oct. 21, 2018**

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ ID: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Please read the following instructions carefully before answering the questions:**

* This assignment is to be completed by each student **individually**.
* There are a total of **6** questions.
* When you write your answers, please try to be precise and concise.
* Fill your name and student ID at the first page.
* Please typeset the file name and format of your submission to the following one: YourID\_CS120\_HW1.pdf (Replace “YourID” with your student ID). Submissions with wrong file name or format will **NOT** be graded.
* Submit your homework through Blackboard.

1. (10 points) How “wide” is a bit on a 10-Gbps link? (5 points) How long is a bit in copper wire, where the speed of propagation is m/s? (5 points)
2. (20 points) Calculate the total time required to transfer a 1000-KB file in the following cases, assuming an RTT of 50 ms, a packet size of 1 KB data, and an initial 2 RTT of “handshaking” before data is sent:

(a) The bandwidth is 1.5 Mbps, and data packets can be sent continuously. (5 points)

(b) The bandwidth is 1.5 Mbps, but after we finish sending each data packet we must wait one RTT before sending the next. (5 points)

(c) The bandwidth is “infinite”, meaning that we take transmit time to be zero, and up to 20 packets can be sent per RTT. (5 points)

(d) The bandwidth is infinite, and during the first RTT we can send one packet (), during the second RTT we can send two packets (), during the third we can send four (), and so on. (5 points)

1. (5 points) Suppose that a certain communications protocol involves a per-packet overhead of 50 bytes for headers and framing. We send 1 million bytes of data using this protocol; however, one data byte is corrupted and the entire packet containing it is thus lost. Give the total number of overhead + loss bytes for packet data sizes of 1000, 10,000, and 20,000 bytes. Which size is optimal?
2. (25 points) The CRC algorithm as presented in this chapter requires lots of bit manipulations. It is, however, possible to do polynomial long division taking multiple bits at a time, via a table-driven method, that enables efficient software implementations of CRC. We outline the strategy here for long division 3 bits at a time (see Table 1); in practice, we would divide 8 bits at a time, and the table would have 256 entries.

Let the divisor polynomial be , or 1101. To build the table for , we take each 3-bit sequence, , append three trailing 0s, and then find the quotient , ignoring the remainder. The third column is the product , the first 3 bits of which should equal .

|  |  |  |
| --- | --- | --- |
| Table 1 Table-Driven CRC Calculation | | |
| p |  |  |
| 000 | 000 | 000 000 |
| 001 | 001 | 001 101 |
| 010 | 011 | 010 \_\_\_ |
| 011 | 0\_\_ | 011 \_\_\_ |
| 100 | 111 | 100 011 |
| 101 | 110 | 101 110 |
| 110 | 100 | 110 \_\_\_ |
| 111 | \_\_\_ | 111 \_\_\_ |

(a) Verify, for , that the quotients and are the same; that is, it doesn’t matter what the trailing bits are. (5 points)

(b) Fill in the missing entries in the table. (10 points)

(c) Use the table to divide 101 001 011 001 100 by C. Hint: The first 3 bits of the dividend are , so from the table the corresponding first 3 bits of the quotient are . Write the 110 above the second 3 bits of the dividend, and subtract = 101 110, again from the table, from the first 6 bits of the dividend. Keep going in groups of 3 bits. There should be no remainder. (10 points)

1. (20 points) Draw a time line diagram for the sliding window algorithm with SWS = RWS = 3 frames, for the following two situations. Use a timeout interval of about 2 × RTT.

(a) Frame 4 is lost. (10 points)

(b) Frames 4 to 6 are lost. (10 points)

1. (20 points) Let A and B be two stations attempting to transmit on an Ethernet. Each has a steady queue of frames ready to send; A’s frames will be numbered , and so on, and B’s similarly. Let μs be the exponential backoff base unit.

Suppose A and B simultaneously attempt to send frame 1, collide, and happen to choose backoff times of and , respectively, meaning A wins the race and transmits while B waits. At the end of this transmission, B will attempt to retransmit while A will attempt to transmit . These first attempts will collide, but now A backs off for either or , while B backs off for time equal to one of .

(a) Give the probability that A wins this second backoff race immediately after this first collision; that is, A’s first choice of backoff time is less than B’s. (5 points)

(b) Suppose A wins this second backoff race. A transmits , and when it is finished, A and B collide again as A tries to transmit and B tries once more to transmit . Give the probability that A wins this third backoff race immediately after the first collision. (5 points)

(c) Give a reasonable lower bound for the probability that A wins all the remaining backoff races. (5 points)

(d) What then happens to the frame ? (5 points)

This scenario is known as the Ethernet *capture effect*.