# CSC458 Problem Set #1

Filip Tomin

1001329984

1. (a) The minimum RTT would be the RTT for a single bit, which would just be $(3.85 \times 10^8_m)/(3 \times 10^8_{m/s}) = 1.283_s \times 2 = 2.57_s$.

   (b) Bandwidth-delay product is $(1.0 \times 10^9) \times 2.57 = 2.57 \times 10^9$ or 2.57Gb.

   (c) The bandwidth-delay product is extremely large. Assuming the wire is being used fully, there is a lot of data that the wire can hold unacknowledged by the destination. Slow response time is also a problem for protocols such as TCP.

   (d) Assuming no errors and the image is sent as one packet, the delivery time will be the transmission time + propagation delay.
   Transmission time: $(2.5 \times 10^7_B) \times 8/(1.0 \times 10^9_{b/s}) = 0.2_s$
   Propagation delay is just $1.28_s$, half of the RTT.
   If we assume the request to be extremely small, such as 1 bit, then the total time it would take would be $0.2 + 1.28 + 1.28 = 2.76$ seconds.

2. (a) Latency $= \sum_i (Prop_i + M/R_i)$
   Latency $= 2((1.0 \times 10^{-5}) + (1.2 \times 10^4)/(1.0 \times 10^8)) = 0.00026s$ or $260\mu s$.

   (b) Latency $= 4((1.0 \times 10^{-5}) + (1.2 \times 10^4)/(1.0 \times 10^8)) = 0.00052s$ or $520\mu s$.

   (c) Latency $= M/R_{min} + \sum_i Prop_i$
   Latency $= (1.2 \times 10^4)/(1.0 \times 10^8) + 2(1.0 \times 10^{-5}) = 0.00014s$ or $140\mu s$.

3. (a) RTT is the one-way delay times 2, so RTT $= 2 \times 10\mu s = 20\mu s$
   Bandwidth-delay product is $(1.0 \times 10^8) \times (2.0 \times 10^{-5}) = 2000$b or 2kb.

   (b) RTT $= 520\mu s$ from #16.
   Bandwidth-delay product $= (1.0 \times 10^8) \times (5.2 \times 10^{-4}) = 52000$b or 52kb.

   (c) RTT $= 2 \times 50$ms$=$ 100ms. Bandwidth-delay product $= (1.5 \times 10^6) \times (0.1) = 1.5 \times 10^5$ or 150kb.

   (d) RTT $= 4 \times ((3.59 \times 10^7)/(3 \times 10^8)) = 479$ms. Bandwidth-delay product $= (1.5 \times 10^6) \times (0.479) = 7.185 \times 10^5$ or 718kb.

4. (a) Total bytes per second $= 640 \times 480 \times 3 \times 30 = 27.7$MB.
   Bandwidth must be $\geq 27.7$MB/s or 221.6Mbps.

(b) Total bytes $= 160 \times 120 \times 1 \times 5 = 96$kB.
Bandwidth must be $\geq 96$kB/s or 768kbps.

(c) Total bytes $= (6.5 \times 10^8)/4500 = 144$kB.
Bandwidth must be $\geq 144$kB/s or 1.16Mbps.

(d) Since it's black and white, assume each pixel requires only one bit.
Total bits $= 8 \times 72 \times 10 \times 72 = 414720$ bits.
$414720/14400 = 28.8$s.

5. The Internet checksum is usually calculated by taking the ones complement sum in 16-bit units. This method is equivalent since the 32-bit sum is the same as two side-by-side 16-bit sums, even maintaining the overflow rules. The 32-bit sum is converted into the 16-bit sum next. The only problem is this 16-bit value has been ones complemented 2 times, so it is inverted from what it should be. Thus, we take the ones complement of the result again, to produce the checksum value.
32-bit:

$$
\begin{array}{rr}
0001101000010101 & 0000000100110000 \\
+\ 0000000000011101 & 0000001001010010 \\
\hline
0001101000110010 & 0000001110000010 \\
\hline
\text{1's comp - } 1110010111001101 & 1111110001111101
\end{array}
$$

$$
\begin{array}{r}
1110010111001101 \\
+\ 1111110001111101 \\
\hline
1110001001001011 \\
\hline
\text{1's comp - } 0001110110110100 \\
\hline
\text{1's comp - } \mathbf{1110001001001011}
\end{array}
$$

16-bit:

$$
\begin{array}{r}
0001101000010101 \\
0000000100110000 \\
0000000000011101 \\
+\ 0000001001010010 \\
\hline
0001110110110100 \\
\hline
\text{1's comp - } \mathbf{1110001001001011}
\end{array}
$$

As shown above, both methods ended up with the same 16-bit checksum value.

6. (a) $M(x) = 11100011$, $G(x) = 1001$, $x^r M(x) = 11100011000$

```
           -------------
    1001 | 11100011000
             1001
               1100
               1001
                 1011
                 1001
                   1000
                   1001
                      10 = Remainder
```

$T(x) = 11100011010$, this is the message that should be transmitted.

(b) The received bits would be 01100011010. Then:

```
           -------------
    1001 | 01100011010
             1001
              1110
              1001
                1101
                1001
                  1010
                  1001
                     110 = Remainder
```

Since the remainder does not equal 0, the receiver knows there has been an error.

7. In this scenario, the successful transmissions occur in the order of C, B, A, the attempted transmissions occur in the order of A, B, C, and there are at least 4 collisions.

```
D----------| C----------| B----------| A----------|
  ^    ^     ^     ^      ^     ^      ^
  A    B     C     A      B     A      A
  |    |           |            |
  \____\___________\____________\___-- Collisions.
```

8. Based on the graph:

A:

| Destination | Next |
|---|---|
| B | C |
| C | C |
| D | C |
| E | C |
| F | C |

B:

| Destination | Next |
|---|---|
| A | E |
| C | E |
| D | E |
| E | E |
| F | E |

C:

| Destination | Next |
|---|---|
| A | A |
| B | E |
| D | E |
| E | E |
| F | F |

D:

| Destination | Next |
|---|---|
| A | E |
| B | E |
| C | E |
| E | E |
| F | E |

E:

| Destination | Next |
|---|---|
| A | C |
| B | B |
| C | C |
| D | D |
| F | C |

F:

| Destination | Next |
|---|---|
| A | C |
| B | C |
| C | C |
| D | C |
| E | C |

9. For this answer, A->B means that the bridge has knowledge that to get to A, it must send the packet to B.

A sends to C

| B1 | B2 | B3 | B4 |
|---|---|---|---|
| A->A | A->B1 | A->B2 | A->B2 |

C sends to A

| B1 | B2 | B3 | B4 |
|---|---|---|---|
| A->A | A->B1 | A->B2 | A->B2 |
| C->B2 | C->B3 | C->C | |

D sends to C

| B1 | B2 | B3 | B4 |
|---|---|---|---|
| A->A | A->B1 | A->B2 | A->B2 |
| C->B2 | C->B3 | C->C | |
| | D->B4 | D->B2 | D->D |

10. (a) The packet will go to both B1 and B2, which will record M->M. Then they will send the packet to L and the other bridge. Since neither of the bridges has an entry for L in their table, they will continue to send the packet to each other in a loop, while also giving L the packet multiple times. This will also replace the M->M records with M->B1 or B2.

    (b) Let's assume that the packet from (a) is already circling clockwise around the bridges. When L sends its packet, B1 has just obtained the packet from M. B1 believes M is at the bottom port and B2 believes M is at the top port. The packet from L which hits B1 is sent to B2 from the bottom and then B1 again from the top, a counter-clockwise loop. Now, a packet from M is moving clockwise around the loop and a packet from L is moving counter-clockwise around the loop.