Dan Bonnett

CMPSCI 381 Lab 4

Due Date: 2/25/2016

Honor Code: *This work is my own unless otherwise cited.*

1. Checksum Example:

```
1)  1010  1110  0110  1000
2)  1001  0100  1101  0010
3)  1011  0101  0100  1010
```

Add the first 2:

```
       111   1   1  1
      1010  1110  0110  1000
    + 1001  0100  1101  0010
    ------------------------
    1 0100  0011  0011  1010
```

Wrap the leftmost 1 around: 0100 0011 0011 1011

now add 3rd value:

```
             1 11  1111   1
      0100  0011  0011  1011
    + 1011  0101  0100  1010
    ------------------------
      1111  1000  1000  0101
```

flip the bits: $\boxed{0000\ 0111\ 0111\ 1010}$ = checksum

#2) Part A: $L = 2000$ bits $R = 1x10^6$ bps must be transmitted over 2 links

Solution: $t = 2(L/R)$ since we will need the transmission delay over 2 links.
$= 2*(2000 / (1x10^6))$
$= 2*(.002)$
$= .004$ seconds

Part B: packet is split into two, 1000 bit packets:

Solution: $t = 3*(1000 / (1x10^6))$
$= 3*(.001)$
$= .003$ seconds

Part C: packet is split into four, 500 bit packets:

Solution: $t = 5*(500 / (1x10^6))$
$= 5*(.0005)$
$= .0025$ seconds

#3) Part A: $L = 2000$ bits $R = 1x10^6$ bps $d = 10km$ $s = 2x10^8$ meters/sec
1 packet transmitted over 2 links

Solution: $t = 2*((2000 / 1x10^6) + (1x10^4 / 2x10^8))$
$= 2*(.002 + .00005)$
$= 2*(.00205)$
$= .0041$ seconds

Part B: $L = 1000$ bits, everything else the same
2 packets transmitted over 2 links

Solution: $t = 3 * ((1000 / 1x10^6) + (10,000 / 2x10^8))$
$= 3 * (.001 + .00005)$
$= 3 * (.00105)$
$= .00315$ seconds

#3) Part C: Packet is split up into four, 500 bit packets. Everything else remains the same. Four packets over 2 links

Solution: t = 5 * ((500 / 1x10^6) + (10,000 / 2x10^8))
= 5 * (.0005 + .00005)
= 5 * (.00055)
= .00275 seconds

#4) Part A:

Solution: This will take 2 RTTs because with a non-persistent connection, 2 Round-Trip-Times are needed per object. The 3-way handshake happens twice.

Part B:

Solution: This will take 10 RTTs. Two are needed for the fetching of the webpage, and two are needed for each of the four images. 2 + 8 = 10.

#5) For connection A-D: Throughput = 100 Mbps
Bottleneck link = link 4

For connection B-E: Throughput = 50 Mbps
Bottleneck link = link 6

For connection C-F: Throughput = 100 Mbps
Bottleneck link = link 4

#6) Part A: L = 1000 bits   R = 250 Kbps   a = 500 packets/sec
Host machine is receiving packets from 4 hosts

Solution: Traffic Intensity = 4 * (1000 * 500) / 250,000
= 2,000,000 / 250,000
= 8

Guaranteed packet loss, infinite queueing delay.

Part B: R = 512 Kbps everything else the same

Solution: Traffic Intensity = 4 * (1000 * 500) / 512,000
= 2,000,000 / 512,000
= 3.90625

Guaranteed packet loss, infinite queueing delay.

Part C: R = 1 Mbps everything else the same

Solution: Traffic Intensity = 4 * (1000 * 500) / 1,000,000
= 2,000,000 / 1,000,000
= 2

Guaranteed packet loss, infinite queueing delay.

#7) Part A: A fully iterative query:

Solution: The requesting host will send a message to the local server. Then the local server will will send a message to the root server, which will send a message back to the local server, telling the local server to send a message to the TLD. The local server then sends a message to the TLD, which then sends a message back to the local server, telling the local server to send a message to the authoritative server. The local server then messages the authoritative server. The authoritative server then sends the correct address/answer back to the local server, which then sends a message with the answer to the requesting host.

Part B: A fully recursive query:

Solution: The requesting host sends a message to the local server, which then sends a message to the root server. The root server then sends a message to the TLD, which then sends a message to the authoritative server. The authoritative server then messages the correct answer/address back to the TLD. The TLD messages it back to the root server, which messages it back to the local server. The local server finally messages the answer back to the requesting host.

Part C: The root server is recursive and the TLD is iterative

Solution: The requesting host will send a message to the local server, which will send a message to the root server. Since the root server executes recursively, the root server sends a message to the TLD. Since the TLD executes iteratively, the TLD sends a message back to the root server, telling the root server to message the authoritative server. The root server then messages the authoritative server, which sends the answer back the root server. The root server then sends a message with the answer back to the local server, which then messages it back to the requesting host.