Kurose & Ross

Chapter 3

Problem 1

The well-known port for HTTP is 80.

- Source port: 33000, destination port: 80.
- Source port: 80, destination port: 33000 (they are inverted).
- No, the HTTP protocol runs over a TCP connection.
- Yes, if supported by the server (almost all web servers do).

Problem 15

It takes 12 microseconds (or 0.012 milliseconds) to send a packet, as $1500*8/10^9=12$ microseconds. In order for the sender to be busy 98 percent of the time, we must have $util = 0.98 = (0.012n) / 30.012$

or n approximately 2451 packets.

Problem 26

There are $2^{32} = 4{,}294{,}967{,}296$ possible sequence numbers.
a) The sequence number does not increment by one with each segment. Rather, it increments by the number of bytes of data sent. So the size of the MSS is irrelevant -- the maximum size file that can be sent from A to B is simply the number of bytes representable by $2^{32} \approx 4.19$ Gbytes.
b) The number of segments is 232/536 = 8.012.999 . 66 bytes of header get added to each segment giving a total of 528,857,934 bytes of header. The total number of bytes transmitted is

$2^{32} + 528{,}857{,}934 = 4.824 \times 10^9$

Thus it would take 249 seconds to transmit the file over a 155~Mbps link.

Problem 32

a)

Denote $EstimatedRTT^{(n)}$ for the estimate after the nth sample.

$$EstimatedRTT^{(4)} = xSampleRTT_1 + (1-x)[xSampleRTT_2 + (1-x)[xSampleRTT_3 + (1-x)SampleRTT_4]]$$

$$= xSampleRTT_1 + (1-x)xSampleRTT_2$$

$$+ (1-x)^2 xSampleRTT_3 + (1-x)^3 SampleRTT_4$$

b)

$$EstimatedRTT^{(n)} = x\sum_{j=1}^{n-1}(1-x)^{j-1}SampleRTT_j$$

$$+ (1-x)^{n-1} SampleRTT_n$$

c)

$$EstimatedRTT^{(\infty)} = \frac{x}{1-x}\sum_{j=1}^{\infty}(1-x)^j SampleRTT_j$$

$$= \frac{1}{9}\sum_{j=1}^{\infty}.9^j SampleRTT_j$$

The weight given to past samples decays exponentially.

Problem 46

- a) Let W denote the max window size measured in segments. Then, W*MSS/RTT = 10Mbps, as packets will be dropped if the maximum sending rate exceeds link capacity. Thus, we have W*1500*8/0.15=10*10^6, then W is about 125 segments.

- b) As congestion window size varies from W/2 to W, then the average window size is 0.75W=94 (ceiling of 93.75) segments. Average throughput is 94*1500*8/0.15 =7.52Mbps.

- c) 94/2 *0.15 = 7.05 seconds, as the number of RTTs (that this TCP connections needs in order to increase its window size from W/2 to W) is given by W/2. Recall the window size increases by one in each RTT.

Problem 53

Let's assume 1500-byte packets and a 100 ms round-trip time. From the TCP throughput equation $B = \frac{1.22 \cdot MSS}{RTT \cdot \sqrt{L}}$, we have

10 Gbps = 1.22 * (1500*8 bits) / (.1 sec * srqt(L)), or

sqrt(L) = 14640 bits / (10^9 bits) = 0.00001464, or

L = 2.14 * 10^(-10)

Chapter 4

Problem 10

In a P2P application, any participating Peer A should be able to initiate a TCP connection to any other participating Peer B. The essence of the problem is that if Peer B is behind a NAT, it cannot act as a server and accept TCP connections. This NAT problem can be circumvented if Peer A is not behind a NAT. In this case, Peer A can first contact Peer B through an intermediate Peer C, which is not behind a NAT and to which B has established an ongoing TCP connection. Peer A can then ask Peer B, via Peer C, to initiate a TCP connection directly back to Peer A. Once the direct P2P TCP connection is established between Peers A and B, the two peers can exchange messages or files. This strategy is called *connection reversal*.

Problem 14

The maximum size of data field in each fragment = 480 (because there are 20 bytes IP header). Thus the number of required fragments $= \left\lceil \frac{1600-20}{480} \right\rceil = 4$

Each fragment will have Identification number 291. Each fragment except the last one will be of size 500 bytes (including IP header). The last datagram will be of size 160 bytes (including IP header). The offsets of the 4 fragments will be 0, 60, 120, 180. Each of the first 3 fragments will have flag=1; the last fragment will have flag=0.

Extra Problems

3
10.10.0.0/21
10.10.8.0/22
192.168.1.0/25
192.168.1.192/26
192.168.2.0/24

4

Network id	Net mask
10.0.0.0	255.255.252.0
10.0.4.0	255.255.254.0
10.0.6.0	255.255.255.0
10.0.8.0	255.255.254.0
10.0.10.0	255.255.255.0

10
Each global IP address is assigned 65535 ports. That gives simultaneous access to the outside interface to 65535 x 2 = 131070 hosts that are connected to the inside interface. However, a remote host will only have access to 65535 ports at its local interface. Therefore no more than 65535 hosts can connect to a remote host at the same time.

11
Let's assume that the servers are assigned IP addresses 10.0.0.1 and 10.0.0.2 respectively. Also let's assume that the outside interface is assigned the global IP address 12.13.14.15. A NAT port forwarding table could then look similar to this:
10.0.0.1:80 → 12.13.14.15:80
10.0.0.1:22 → 12.13.14.15:2201
10.0.0.2:25 → 12.13.14.15:25
10.0.0.2:22 → 12.13.14.15:2202