Ch1

Problem 5

a) $d_{prop} = m/s$ seconds.

b) $d_{trans} = L/R$ seconds.

c) $d_{end-to-end} = (m/s + L/R)$ seconds.

d) The bit is just leaving Host A.

e) The first bit is in the link and has not reached Host B.

f) The first bit has reached Host B.

g) Want

$$m = \frac{L}{R}S = \frac{100}{28 \times 10^3}\left(2.5 \times 10^8\right) = 893\,\text{km}.$$

Problem 8

a) 10,000

b) $\sum_{n=N+1}^{M} \binom{M}{n} p^n (1-p)^{M-n}$

Problem 9

Because bits are immediately transmitted, the packet switch does not introduce any delay; in particular, it does not introduce a transmission delay. Thus,

$$d_{end\text{-}end} = L/R + d_1/s_1 + d_2/s_2$$

For the values in Problem 9, we get 8 + 16 + 4 = 28 msec.

Problem 10

It takes LN/R seconds to transmit the N packets. Thus, the buffer is empty when a batch of N packets arrive.

The first of the N packets has no queuing delay. The 2nd packet has a queuing delay of L/R seconds. The n^{th} packet has a delay of $(n-1)L/R$ seconds.

The average delay is

$$\frac{1}{N}\sum_{n=1}^{N}(n-1)L/R = \frac{L}{R}\frac{1}{N}\sum_{n=0}^{N-1} n = \frac{L}{R}\frac{1}{N}\frac{(N-1)N}{2} = \frac{L}{R}\frac{(N-1)}{2}.$$

Problem 11

The queuing delay is 0 for the first transmitted packet, *L/R* for the second transmitted packet, and generally, *(n-1)L/R* for the n^{th} transmitted packet. Thus, the average delay for the *N* packets is

$(L/R + 2L/R + ... + (N-1)L/R)/N = L/RN(1 + 2 + ... + (N-1)) = LN(N-1)/(2RN)$
$= (N-1)L/(2R)$

Note that here we used the well-known fact that

$1 + 2 + ... + N = N(N+1)/2$

Problem 13

a) b)
The transmission delay is L/R. The total delay is

b) $\frac{IL}{R(1-I)} + \frac{L}{R} = \frac{L/R}{1-I}$

Let $x = L/R$.

Total delay = $\frac{x}{1-ax}$

Problem 22

a) 150 msec
b) 1,500,000 bits
c) 600,000,000 bits

Problem 24

a) Time to send message from source host to first packet switch = $\frac{7.5\times10^6}{1.5\times10^6}\sec = 5\sec$. With store-and-forward switching, the total time to move message from source host to destination host = $5\sec\times 3\ hops = 15\sec$
b) Time to send 1st packet from source host to first packet switch = . $\frac{1.5\times10^3}{1.5\times10^6}\sec = 1\ m\sec$. Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = $2\times1\ m\sec = 2\ m\sec$
c) Time at which 1st packet is received at the destination host = . $1\ m\sec\times 3\ hops = 3\ m\sec$. After this, every 1msec one packet will be received; thus time at which last (5000th) packet is received = $3\ m\sec + 4999 * 1m\sec = 5.002\sec$. It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).
d) Drawbacks:
    i. Packets have to be put in sequence at the destination.
    ii. Message segmentation results in many smaller packets. Since header size is usually the same for all packets regardless of their size, with message segmentation the total amount of header bytes is more.

Ch2

Problem 1

a) F
b) F
c) T
d) F

Problem 12

a) Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of a rate of u_s/N. Note that this rate is less than each of the client's download rate, since by assumption $u_s/N \leq d_{\min}$. Thus each client can also receive at rate u_s/N. Since each client receives at rate u_s/N, the time for each client to receive the entire file is $F/(u_s/N) = NF/u_s$. Since all the clients receive the file in NF/u_s, the overall distribution time is also NF/u_s.

b) Consider a distribution scheme in which the server sends the file to each client, in parallel, at a rate of $d_{\min}$. Note that the aggregate rate, $N\,d_{\min}$, is less than the server's link rate u_s, since by assumption $u_s/N \geq d_{\min}$. Since each client receives at rate $d_{\min}$, the time

for each client to receive the entire file is $F/d_{\min}$. Since all the clients receive the file in this time, the overall distribution time is also $F/d_{\min}$.

c) From Section 2.6 we know that

$D_{CS} \geq \max\{NF/u_s, F/d_{\min}\}$ (Equation 1)

Suppose that $u_s/N \leq d_{\min}$. Then from Equation 1 we have $D_{CS} \geq NF/u_s$. But from (a) we have $D_{CS} \leq NF/u_s$. Combining these two gives:

$D_{CS} = NF/u_s$ when $u_s/N \leq d_{\min}$. (Equation 2)

We can similarly show that:

$D_{CS} = F/d_{\min}$ when $u_s/N \geq d_{\min}$ (Equation 3).

Combining Equation 2 and Equation 3 gives the desired result.

Problem 16

For calculating the minimum distribution time for client-server distribution, we use the following formula:

$$D_{cs} = max\ \{NF/u_s,\ F/d_{min}\}$$

Similarly, for calculating the minimum distribution time for P2P distribution, we use the following formula:

$$D_{P2P} = max\{F/u_s,\ F/d_{min},\ NF/(u_s + \sum_{i=1}^{N} u_i)\}$$

Where, F = 10 Gbits = 10 * 1024 Mbits
u_s = 20 Mbps
$d_{min} = d_i$ = 1 Mbps

Client Server

		N		
		10	**100**	**1000**
	200 Kbps	10240	51200	512000
u	**600 Kbps**	10240	51200	512000
	1 Mbps	10240	51200	512000

Peer to Peer

		N		
		10	**100**	**1000**
	200 Kbps	10240	25904.3	47559.33
u	**600 Kbps**	10240	13029.6	16899.64
	1 Mbps	10240	10240	10240

Problem 18

a) Define $u = u_1 + u_2 + \ldots.. + u_N$. By assumption

$u_s <= (u_s + u)/N$ Equation 1

Divide the file into N parts, with the i^{th} part having size $(u_i/u)F$. The server transmits the i^{th} part to peer i at rate $r_i = (u_i/u)u_s$. Note that $r_1 + r_2 + \ldots.. + r_N = u_s$, so that the aggregate server rate does not exceed the link rate of the server. Also have each peer i forward the bits it receives to each of the N-1 peers at rate r_i. The aggregate forwarding rate by peer i is $(N-1)r_i$. We have

$(N-1)r_i = (N-1)(u_s u_i)/u <= u_i$,

where the last inequality follows from Equation 1. Thus the aggregate forwarding rate of peer i is less than its link rate u_i.

In this distribution scheme, peer i receives bits at an aggregate rate of

$$r_i + \sum_{j<>i} r_j = u_s$$

Thus each peer receives the file in F/u_s.

b) Again define $u = u_1 + u_2 + \ldots.. + u_N$. By assumption

$u_s >= (u_s + u)/N$ Equation 2

Let $r_i = u_i/(N-1)$ and
$r_{N+1} = (u_s - u/(N-1))/N$

In this distribution scheme, the file is broken into N+1 parts. The server sends bits from the i^{th} part to the i^{th} peer (i = 1, …., N) at rate r_i. Each peer i forwards the bits arriving at rate r_i to each of the other N-1 peers. Additionally, the server sends bits from the $(N+1)^{st}$ part at rate r_{N+1} to each of the N peers. The peers do not forward the bits from the $(N+1)^{st}$ part.

The aggregate send rate of the server is

$r_1 + \ldots. + r_N + N\, r_{N+1} = u/(N-1) + u_s - u/(N-1) = u_s$

Thus, the server's send rate does not exceed its link rate. The aggregate send rate of peer i is

$(N-1)r_i = u_i$

Thus, each peer's send rate does not exceed its link rate.
In this distribution scheme, peer i receives bits at an aggregate rate of

$$r_i + r_{N+1} + \sum_{j<>i} r_j = u/(N-1) + (u_s - u/(N-1))/N = (u_s + u)/N$$

Thus each peer receives the file in $NF/(u_s+u)$.
(For simplicity, we neglected to specify the size of the file part for i = 1, …., N+1. We now provide that here. Let $\Delta = (u_s+u)/N$ be the distribution time. For i = 1, …, N, the i^{th} file part is $F_i = r_i \Delta$ bits. The $(N+1)^{st}$ file part is $F_{N+1} = r_{N+1} \Delta$ bits. It is straightforward to show that $F_1 + ….. + F_{N+1} = F$.)

c) The solution to this part is similar to that of 17 (c). We know from section 2.6 that

$$D_{P2P} >= max\{F/u_s, NF/(u_s + u)\}$$

Combining this with (a) and (b) gives the desired result.

Problem 25

There are *N* nodes in the overlay network. There are *N(N-1)/2* edges.

Ch3

Problem 7

The sender side of protocol rdt3.0 differs from the sender side of protocol 2.2 in that timeouts have been added. We have seen that the introduction of timeouts adds the possibility of duplicate packets into the sender-to-receiver data stream. However, the receiver in protocol rdt.2.2 can already handle duplicate packets. (Receiver-side duplicates in rdt 2.2 would arise if the receiver sent an ACK that was lost, and the sender then retransmitted the old data). Hence the receiver in protocol rdt2.2 will also work as the receiver in protocol rdt 3.0.