Jordan Dube
HW #1
CMSC 440
QUESTIONS 2,3,4,6,8,10,20,23,25,31,33

2.

| | |
|---|---|
| $N * (L / R)$ | for 1 packet |
| $N * (L / R) + L / R$ | for 2 packets |
| $N * (L / R) + 2 * (L / R)$ | for 3 packets |
| $N * (L / R) + (4 - 1) * (L / R)$ | for 4 packets |

General Equation = $N * (L / R) + (P - 1) * (L / R)$
Simplified Gen. Eq. = $(N + P - 1) * (L / R)$

N = number of links
L/R = transmission rate
P = number of packets to transmit

3.

A. In this case, a circuit-switched network is better. This is because data transmission is fixed, so links can be allocated with minimal wasted capacity. The application will also be transmitting data for a relatively long time, so reserving channels would not hinder much in terms of average transmission time. The path is reserved, but since the data is produced regularly for a long time, it is the most efficient method.

B. No. Since the links can transmit faster than the applications can produce data, there should not be any congestion to control. The packets can enter the next link in the path before the previous packets can catch up. In the worst case where all applications send packets simultaneously, the link has sufficient capacity to handle all packets since it is stated that the sum of the application data rates is less than the capacities of each and every link.

4.

A. Assuming that “simultaneous connections” refers to connections between switches, there are 4 connections from any 1 router to another, this network can support up to 16 connections. A<->B = 4; B<->C = 4; C<->D = 4; D<->A = 4

B. There are 2 paths that A can take to C (and vice versa): A<->B<->C OR A<->D<->C. There are 4 channels from A to B, and those can take 4 channels to C, yielding a total of 4 connections for the first path A<->B<->C. Similarly, there are 4 channels from A to D, and that leaves 4 channels available to get from D to C, yielding a total of 4 connections for the second path A<->D<->C. Adding these together, if all channels are taken between A and C, there are 8 total connections that can exist simultaneously between A and C.

C. Yes, we can partition each link to accommodate these 8 connections. We can run 4 paths A<->C and 4 paths B<->D by taking different routes. To ensure that there are connections available, we can have 2 connections A<->C route through B and the other 2 connections A<->C route through D. This satisfies the 4 connections A<->C. We can ensure 4 connections B<->D in the same way as A<->C, with 2 connections B<->A<->D and 2 connections B<->C<->D. This satisfies the 4 connections B<->D. Thus, all 8 connections are assured.

6.

A. d_prop = x seconds = (meters) / (meters/sec) = m/s

B. d_trans = x seconds = (bits) / (bits/sec) = L/R

C. d_end-end = x seconds = d_prop + d_trans = (m / s) + (L / R)

D. The last bit of the packet is leaving the first host at t = d_trans

E. The first bit is in the link traveling to the second host at t = d_trans

F. The first bit is already at the second host at t = d_trans

G. <u>Note</u>: Using significant figures and no rounding intermediate calculations
s = 2.5*10^8 meters/sec; L = 120 bits; R = 56*10^3 bits/sec; m = ?
D_trans = d_prop = L/R = 120/(56*10^3) = 0.0021 sec = m/s
M = 0.0021 * s = 0.0021 * (2.5*10^8) = 540000 meters = 540 kilometers

8.

A. Circuit switching reserves space, so since the link is 3 Megabits (3000 kilobits) per sec. capacity, and each user will need to reserve 150 kilobits per sec., the number of users that can use the link with circuit switching is expressed as 3000 / 150 = 20, so the link can support 20 users.

B. p_active = 0.1 since each connected user spends 10% of their time transmitting

C. binomial dist = $\binom{n}{k} p^{k} (1 - p)^{(n-k)}$
P = 0.1, n = 120, k = num of users (n)

P(n users are active) = $\binom{120}{n} 0.1^{n} (1 - 0.1)^{(120-n)}$

D. Sum all probabilities from k = 21 to k = 120 to determine probability that num of concurrent users $\geq 21$:

P(21 or more users are transmitting) = $\sum_{k=21}^{120} \left( \binom{120}{k} 0.1^{k} (1 - 0.1)^{(120-k)} \right)$

10. A. A (L) -link1-> SWITCH1 -link2-> SWITCH2 -link3-> end
d_link1 = L/R_1 + d_1 / s_1 ; d_switch1 = d_proc ; d_link2 = L/R_2 + d_2 / s_2
d_switch2 = d_proc ; d_link3= L/R_3 + d_3 / s_3
D_end-end = d_link1 + d_switch1 + d_link2 + d_switch2 + d_link3

$$d_{end-end} = \frac{L}{R_1} + \frac{L}{R_2} + \frac{L}{R_3} + \frac{d_1}{s_1} + \frac{d_2}{s_2} + \frac{d_3}{s_3} + 2(d_{proc})$$

B.

L = 1500B * 8b/B = 12000b / 1000b/kb = 12kb
s = 2.5*10^8 m/s * km/1000m = 2.5*10^5 km/s
R = 2Mbps * 1000kbps/Mbps = 2000kbps
d_proc = 3ms;
d_1 = 5000km; d_2 = 4000km; d_3 = 1000km
Using $d_{end-end}$ from part A), we can plug in these values:

$$d_{end-end} = \frac{L}{R} + \frac{L}{R} + \frac{L}{R} + \frac{d_1}{s} + \frac{d_2}{s} + \frac{d_3}{s} + 2(d_{proc}) = \frac{3L}{R} + \frac{d_1+d_2+d_3}{s} + 2(d_{proc})$$

=

$$\frac{3(12kb)}{2000kbps} + \frac{5000km+4000km+1000km}{2.5*10^5 km/s} + 2(3ms) = \frac{36kb}{2000kbps} + \frac{10000km}{2.5*10^5 km/s} + 6ms$$

=

$$\frac{9kb}{500kbps} + \frac{1.0*10^4 km}{2.5*10^5 km/s} + 6ms = \frac{9}{500}s + \frac{1}{25}s + 6ms = 0.018s + 0.04s + 0.006s$$

= $0.064s$ or 64ms

20.

The capacity of the network link is R, and it is shared by M users. The throughput is limited by R split across M users. The throughput is determined by the minimum value of either the R_c, R_s, or R/M, since these are all components that can limit the traffic from hosts. If there is abundant capacity on the network link, then throughput is only limited by either client or server link capacity. Mathematically, throughput = min(R_c, R_s, R/M).

23.

A. Since the first link is the bottleneck link, the first packet gets transmitted into the first link in L/R_s, and the second packet waits in queue at the server until the transmission. Once the Transmission delay of L/R_s is passed by the first and second packet, ther packets both travel at the same speed as each other, the second packet just waited for the first packet to be transmitted. Thus, the time that elapses from when the last bit of the first packet arrives at client and the last bit of the second packet arrives is L/R_s.

B. Yes. If the packets arrive at the input switch before the first packet is transmitted, then the second packet must wait in queue before it gets transmitted into second link. The second packet would have also taken L/R_s to get transmitted from the server, which the first packet would have already done. So the second packet would need to wait up to L/R_c - L/R_s in the queue before it is able to be transmitted. T must be larger than L/R_c - L/R_s in order to avoid any queuing, if the packet arrived before that, then there would be a waiting period, but sending the second packet after L/R_c - L/R_s effectively waits out the queuing time.

25.

A. $R * d_{prop} = R * \frac{d}{m} = 2Mbps * \frac{20000km}{2.5*10^8 m/s} = 2Mbps * \frac{2*10^4 km}{2.5*10^5 km/s} = 2Mbps * \frac{2}{25} s$

$= 0.16Mb * 10^6 b/Mb = 160000$ bits

B. The time for first bit in link to reach host B is d_prop. At this time, link is potentially full. Host A puts data into the link at rate R. The max amount of bits in the link is the product of the rate at which data is put into the link and the time it takes to fill the link: $R * d_{prop}$, which was found in A. to be 160000 bits

C. The bandwidth-delay product is the maximum amount of bits that can be in the link.

D. Link width = $(2 * 10^4 km) * 1000m/km = 2 * 10^7$ meters
Number of bits in link = 160000 bits
W_bit = width of bit (meters/bit); w_link = width of link; n_bits = max number of bits in the link:

$$w_{bit} = \frac{w_{link}}{n_{bits}} = \frac{2*10^7 m}{1.6*10^5 b} = \frac{2*10^2 m}{1.6b} = \frac{10^3 m}{8b} = 125m/b$$

A Football field ≈ 91.4 meters (from Google Search)
125 > 91.4, so in this link, 1 bit is longer than a football field.

E. $w_{bit} = \frac{m}{R*d_{prop}}$

$R * d_{prop} = R * \frac{m}{s}$

$w_{bit} = \frac{m}{R*\frac{m}{s}} = \frac{m}{\frac{Rm}{s}} = \frac{m}{1} * \frac{s}{Rm} = \frac{s}{R}$

31.

A. 1 hop:

$$t_{hop} = \frac{L}{R} = \frac{8Mb}{2Mbps} = 4s$$

All 3 hops:

$$d_{end-end} = 3 * t_{hop} = 3 * 4s = 12s$$

B. 1 hop:

$$t_{hop} = \frac{L}{R} = \frac{10^4 b}{2*10^6 bps} = \frac{1}{200} s = 0.005s = 5ms$$

Second packet fully received in switch 1 when first packet completes 2 hops:

$$t_{2-hop} = 2 * t_{hop} = 2 * 5ms = 10ms$$

C. Using equation from 2. Where n = 3, P = 800, L/R = 5ms

$$t_{total} = (N + P - 1) * \frac{L}{R}$$

$t_{total} = (3 + 800 - 1) * 5ms = 802 * \frac{1}{200}s = \frac{401}{100}s = 4.01s$

Using message segmentation is much faster than sending the whole message at once. The segmented method is almost 67% faster than non-segmentation.

D. It reduces congestion since if a packet arrived while the whole message was transmitting into the link, then that later packet would potentially wait a long time. Also, if the network is not loss-tolerant, then any errors when sending would mean that the whole message needs to be resent, rather than the problem packet.

E. Packets need to be processed at the destination to form cohesive data, which can cause a delay for big messages. A lot of packets can clog queues and prevent other packets from being transmitted/processed at each step in network paths. Each packet needs its own header, which can add a lot of data to the packet sizes, when only 1 header would be needed for a non-segmented file.

33. L = 80 + S; N = 3; P = F/S

$d_{end-end} = (N + P - 1) * \frac{L}{R} = (\frac{F}{S} + 2) * \frac{80+S}{R} \rightarrow \frac{d}{dS}((\frac{F}{S} + 2) * \frac{80+S}{R})$

$= (\frac{F}{S} + 2) * 1 + (\frac{80+S}{R}) * \frac{-F}{S^2} = \frac{F}{S} + 2 - \frac{80F+SF}{S^2} = \frac{SF}{S^2} + \frac{2S^2}{S^2} + \frac{-80F-SF}{S^2}$

$= \frac{2S^2-80F}{S^2}$

Function (delay) has a relative extremum when First Derivative = 0:

$= \frac{2S^2-80F}{S^2} \rightarrow 2S^2 - 80F = 0 \rightarrow S^2 = 40F \rightarrow S = \pm\sqrt{40F}$

Relative extrema at S = $\sqrt{40F}$ (negative value of S does not make sense in this context.)

Looking at Desmos, we can see that this is a minimum value, since delay cannot be negative, S = $\sqrt{40F}$ is the first value of S (for any value F) that yields a non-negative y-value (always 0). Thus, this value of S minimizes the delay of moving a file from host A to host B. The screenshot is included below.


Untitled Graph
desmos
Log In
or
Sign Up
$f(x) = \frac{2x^2 - 80F}{2x^2}$
$F = 1$
-10
10
$y = \frac{d}{dx} f(x)$
$x = \sqrt{40F}$
$x = 6.32455532034$
(6.325, 0)
powered by desmos