**第1章**

R11

At time t0 the sending host begins to transmit. At time *t1 = L/R1*, the sending host completes transmission and the entire packet is received at the router (no propagation delay). Because the router has the entire packet at time *t1*, it can begin to transmit the packet to the receiving host at time *t1*. At time *t2 = t1 + L/R2*, the router completes transmission and the entire packet is received at the receiving host (again, no propagation delay). Thus, the end-to-end delay is *L/R1 + L/R2*.

R19

a) 500 kbps

b) 64 seconds

c) 100kbps; 320 seconds

P2

At time N\*(L/R) the first packet has reached the destination, the second packet is stored in the last router, the third packet is stored in the next-to-last router, etc. At time N\*(L/R) + L/R, the second packet has reached the destination, the third packet is stored in the last router, etc. Continuing with this logic, we see that at time N\*(L/R) + (P-1)\*(L/R) = **(N+P-1)\*(L/R)** all packets have reached the destination.

P5

Tollbooths are 75 km apart, and the cars propagate at 100km/hr. A tollbooth services a car at a rate of one car every 12 seconds.

a) There are ten cars. It takes 120 seconds, or 2 minutes, for the first tollbooth to service the 10 cars. Each of these cars has a propagation delay of 45 minutes (travel 75 km) before arriving at the second tollbooth. Thus, all the cars are lined up before the second tollbooth after 47 minutes. The whole process repeats itself for traveling between the second and third tollbooths. It also takes 2 minutes for the third tollbooth to service the 10 cars. Thus the total delay is 96 minutes.

b) Delay between tollbooths is 8\*12 seconds plus 45 minutes, i.e., 46 minutes and 36 seconds. The total delay is twice this amount plus 8\*12 seconds, i.e., 94 minutes and 48 seconds.

P6

a)  seconds.

b)  seconds.

c)  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

km.

P7

Consider the first bit in a packet. Before this bit can be transmitted, all of the bits in the packet must be generated. This requires

sec=7msec.

The time required to transmit the packet is

sec=sec.

Propagation delay = 10 msec.

The delay until decoding is

7msec +sec + 10msec = 17.224msec

A similar analysis shows that all bits experience a delay of 17.224 msec.

P10

The first end system requires *L/R1* to transmit the packet onto the first link; the packet propagates over the first link in *d1/s1*; the packet switch adds a processing delay of *dproc*; after receiving the entire packet, the packet switch connecting the first and the second link requires *L/R2* to transmit the packet onto the second link; the packet propagates over the second link in *d2/s2*. Similarly, we can find the delay caused by the second switch and the third link: *L/R3*, *dproc*, and *d3/s3*.

Adding these five delays gives

*dend-end = L/R1 + L/R2 + L/R3 + d1/s1 + d2/s2 + d3/s3+ dproc+ dproc*

To answer the second question, we simply plug the values into the equation to get 6 + 6 + 6 + 20+16 + 4 + 3 + 3 = 64 msec.

P11

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

*dend-end = L/R + d1/s1 + d2/s2+ d3/s3*

For the values in Problem 10, we get 6 + 20 + 16 + 4 = 46 msec.

P20

Throughput = *min{Rs, Rc, R/M}*

P21

If only use one path, the max throughput is given by:

.

If use all paths, the max throughput is given by .

P31

1. Time to send message from source host to first packet switch = With store-and-forward switching, the total time to move message from source host to destination host = 
2. Time to send 1st packet from source host to first packet switch = . . Time at which 2nd packet is received at the first switch = time at which 1st packet is received at the second switch = 
3. Time at which 1st packet is received at the destination host = . After this, every 5msec one packet will be received; thus time at which last (800th) packet is received = . It can be seen that delay in using message segmentation is significantly less (almost 1/3rd).
4. Without message segmentation, if bit errors are not tolerated, if there is a single bit error, the whole message has to be retransmitted (rather than a single packet).
5. Without message segmentation, huge packets (containing HD videos, for example) are sent into the network. Routers have to accommodate these huge packets. Smaller packets have to queue behind enormous packets and suffer unfair delays.
6. Packets have to be put in sequence at the destination.
7. 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.

P33

There are *F*/*S* packets. Each packet is S=80 bits. Time at which the last packet is received at the first router is sec. At this time, the first F/S-2 packets are at the destination, and the F/S-1 packet is at the second router. The last packet must then be transmitted by the first router and the second router, with each transmission taking sec. Thus delay in sending the whole file is

To calculate the value of S which leads to the minimum delay,



**第2章**

### Problem 1

a) F

b) T

c) F

d) F

e) F

### Problem 7

The total amount of time to get the IP address is

.

Once the IP address is known,  elapses to set up the TCP connection and another  elapses to request and receive the small object. The total response time is



### Problem 8



.





1. Persistent connection with pipelining. This is the default mode of HTTP.



.

Persistent connection without pipelining, without parallel connections.



.

### Problem 9

1. The time to transmit an object of size L over a link or rate R is L/R. The average time is the average size of the object divided by R:

Δ = (850,000 bits)/(15,000,000 bits/sec) = .0567 sec

The traffic intensity on the link is given by βΔ=(16 requests/sec)(.0567 sec/request) = 0.907. Thus, the average access delay is (.0567 sec)/(1 - .907) ≈ .6 seconds. The total average response time is therefore .6 sec + 3 sec = 3.6 sec.

1. The traffic intensity on the access link is reduced by 60% since the 60% of the requests are satisfied within the institutional network. Thus the average access delay is (.0567 sec)/[1 – (.4)(.907)] = .089 seconds. The response time is approximately zero if the request is satisfied by the cache (which happens with probability .6); the average response time is .089 sec + 3 sec = 3.089 sec for cache misses (which happens 40% of the time). So the average response time is (.6)(0 sec) + (.4)(3.089 sec) = 1.24 seconds. Thus the average response time is reduced from 3.6 sec to 1.24 sec.

### Problem 10

Note that each downloaded object can be completely put into one data packet. Let Tp denote the one-way propagation delay between the client and the server.

First consider parallel downloads using non-persistent connections. Parallel downloads would allow 10 connections to share the 150 bits/sec bandwidth, giving each just 15 bits/sec. Thus, the total time needed to receive all objects is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p )

+ (200/(150/10)+*T*p + 200/(150/10) +*T*p + 200/(150/10)+*T*p + 100,000/(150/10)+ *T*p )

= 7377 + 8\**T*p (seconds)

Now consider a persistent HTTP connection. The total time needed is given by:

(200/150+*T*p + 200/150 +*T*p + 200/150+*T*p + 100,000/150+ *T*p )

+ 10\*(200/150+*T*p + 100,000/150+ *T*p )

=7351 + 24\**T*p (seconds)

Assuming the speed of light is 300\*106 m/sec, then Tp=10/(300\*106)=0.03 microsec. Tp is therefore negligible compared with transmission delay.

Thus, we see that persistent HTTP is not significantly faster (less than 1 percent) than the non-persistent case with parallel download.