Tutorial 1 Solutions (Week 5)

Introduction

Suppose two hosts, A and B are separated by 10,000 kms and are connected by a direct link of R = 1Mbps. Suppose the propagation speed over the link is 2.5 x 10^8 m/s.

(a) Calculate the bandwidth-delay product, R.t_{prop}
Answer: The propagation delay, tprop = 10,000 x 10^3 /2.5 x 10^8 = 40 msec. Hence, the bandwidth delay product = 40 msec x 1 Mbps = 40,000 bits.

(b) Consider sending a file of 400,000 bits from A to B. Suppose the file is sent continuously as one big message. What is the maximum number of bits that will be in the link at any given time?
Answer: This is same as (a), i.e., 40,000 bits.

(c) How long does it take to send the above file if it is sent continuously?
Answer: $t_{trans} + t_{prop}$ = 400 msec + 40 msec = 440 msec

(d) Suppose now that the file is broken up into 10 packets with each packet containing 40,000 bits. Suppose that each packet is acknowledged by the receiver and the transmission time of an acknowledgement packet is negligible. Finally, assume that the sender cannot send a packet until the preceding packet is acknowledged. How long does it take to send the file?
Answer: The delay to send each packet will be $t_{trans} + t_{prop}$. However there is an additional t_{prop} delay for the acknowledgement packet to arrive at the sender, prior to which the next packet can be transmitted. Hence the total delay to transmit 10 packet is = 10 * (t_{trans} + 2 t_{prop}) = 10*(40 msec + 80 msec) = 1.2 sec

Application Layer

Problem Set 2, Question 2 Consider the figure below, for which there is an institutional network connected to the Internet. Suppose that the average object size is 900,000 bits and that that the average request rate from the institution's browsers to the origin server is 1.5 requests per second. Also suppose that the amount of time it takes from when the router on the Internet side of the access link forwards a HTTP request until it receives the response in two seconds on average. Model the total average response time as the sum of the average access delay and the average Internet delay. For the average access delay, use A/(1-AB) where A is the average time required to send an object over the access link and B is the arrival rate of objects to the access link. You can assume that the HTTP request messages are negligibly small and thus create no traffic on the network or the access link.

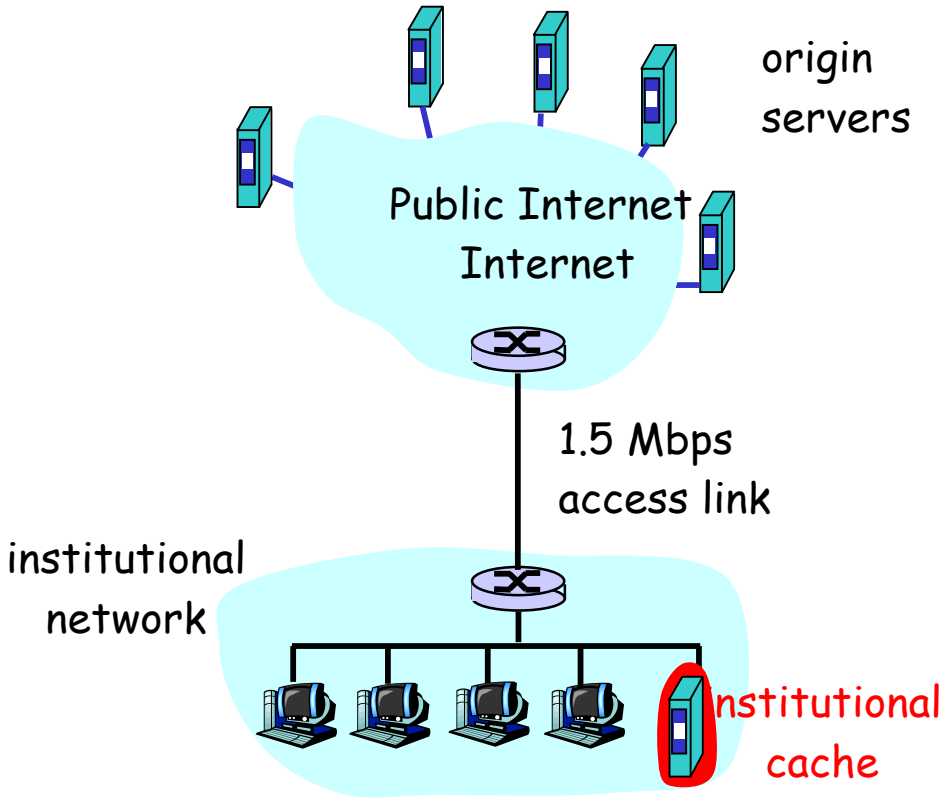


Figure 1: Figure for web cache problem

(a) Find the total average response time.

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

A= (900,000 bits)/(1,500,000 bits/sec) = .6 sec

The traffic intensity on the link is = AB= (.6 msec/request) (1.5 requests/sec) = .9. Thus, the average access delay is (.6 sec)/(1 - .9) = 6 seconds. The total average response time is therefore 6 sec + 2 sec = 8 sec.

(b) Now suppose a cache is installed in the institutional LAN. Suppose the hit rate is 0.4. Find the total response time.

Answer: The traffic intensity on the access link is reduced by 40% since the 40% of the requests are satisfied within the institutional network. Thus, the arrival rate of the objects to the link also changes since only 60% of the objects need to be fetched from the origin servers (the rest are obtained from the cache). As a result, B=1.5 x 0.6 = 0.9 requests/sec.

Thus the average access delay is (.6 sec)/[1 – (.6)(.9)] = 1.3 seconds. The response time is approximately zero if the request is satisfied by the cache (which happens with probability .4); the average response time is 1.3 sec + 2 sec = 3.3 sec for cache misses

(which happens 60% of the time). So the average response time is (.4)(0 sec) + (.6)(3.3 sec) = 1.98 seconds. Thus the average response time is reduced from 8 sec to 1.98 sec.

Question: Consider an HTTP client that wants to retrieve a web document at a given URL. The IP address of the HTTP server is initially unknown. The web document at the URL has one embedded GIF image that resides at the same server as the original document. What transport and application layer protocols besides HTTP are needed in this scenario?

Answer: Before the HTTP GET request can be sent for the web document, the HTTP client needs to obtain the IP address of the HTTP server hosting the document. So a DNS request is sent out to obtain the hostname to IP address mapping. Recall that DNS runs over UDP. Once the mapping is obtained, the HTTP client first establishes a TCP connection with the server (Recall that HTTP runs over TCP). Following the TCP connection establishment a GET request for the web document is sent over this connection. In summary, the following are the protocols that are used:
Application layer protocols: DNS and HTTP
Transport layer protocols: UDP for DNS; TCP for HTTP

Transport Layer

Problem Set 3, Question 6 Consider the GBN and SR protocols. Suppose the sequence number space of size *k*. What is the largest allowable sender size window that will avoid the occurrence of the problems such as that in Figure 3.27 (of the textbook or as discussed towards the end of Week 4 lecture slides).

Answer: In order to avoid the scenario of Figure 3.27, we want to avoid having the leading edge of the receiver's window (i.e., the one with the “highest” sequence number) wrap around in the sequence number space and overlap with the trailing edge (the one with the "lowest" sequence number in the sender's window). That is, the sequence number space must be large enough to fit the entire receiver window and the entire sender window without this overlap condition. So we need to determine how large a range of sequence numbers can be covered at any given time by the receiver and sender windows.

Suppose that the lowest-sequence number that the receiver is waiting for is packet *m*. In this case, its window is *[m, m+w-1]* and it has received (and ACKed) packet *m-1* and the *w-1* packets before that, where *w* is the size of the window. If the sender has yet received none of those *w* ACKs, then ACK messages with values of *[m-w, m-1]* may still be propagating back. If the sender has received no ACKs with these ACK numbers, then the sender's window would be *[m-w, m-1]*.

Thus, the lower edge of the sender's window is *m-w*, and the leading edge of the receiver’s window is *m+w-1*. In order for the leading edge of the receiver's window to

not overlap with the trailing edge of the sender's window, the sequence number space must thus be big enough to accommodate *2w* sequence numbers. That is, the sequence number space must be at least twice as large as the window size, $k \geq 2w$.

Question: UDP and TCP use 1s complement for their checksums. Suppose you have the following three 8-bit bytes: 01010101, 01110000, 01001100. What is the 1's complement of the sum of these 8-bit bytes? (Note although TCP and UDP use 16-bit words in computing the checksum, for this problem we will only consider 8-bit summands). Show all work. Is it possible that a 1-bit error will go undetected by checksum? How about a two-bit error?

Answer:

$$\begin{array}{rcccccccc} & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 1 \\ + & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ \hline & 1 & 1 & 0 & 0 & 0 & 1 & 0 & 1 \end{array}$$

$$\begin{array}{r} 11000101 \\ +01001100 \\ \hline 00010001 \end{array}$$

The last addition has an overflow so we add it to the above sum resulting in a final result of 00010010.

One's complement of this is 11101101 which is the final checksum.

All one-bit errors will be detected by checksum, but two-bit errors can be undetected (e.g., if the last digit of the first word is converted to a 0 and the last digit of the second word is converted to a 1).

Question: Consider a situation where you want to transfer a very large file of *L* bytes from Host A to Host B. Assume an MSS of 1,460 bytes.

(a) What is the maximum value of *L* such that TCP sequence numbers are not exhausted? Recall that the TCP sequence number field has 4 bytes.

Answer: There are $2^{32} = 4{,}294{,}967{,}296$ possible sequence numbers. 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) For the value of L computed in part (a), find how long it takes to transmit the file. Assume that a total of 66 bytes of transport, network and data-link header are added to each segment before the resulting packet is sent out over a 10 Mbps link. Ignore flow control and congestion control so that A can pump the segments back to back continuously.

Answer: The number of segments is $\left\lceil \frac{2^{32}}{1460} \right\rceil = 2{,}941{,}758$. 66 bytes of header get added to each segment giving a total of 194,156,028 bytes of header. The total number of bytes transmitted is $2^{32} + 194{,}156{,}028 = 3{,}591 \times 10^7$ bits. Thus it would take 3,591 seconds = 59 minutes to transmit the file over a 10~Mbps link.

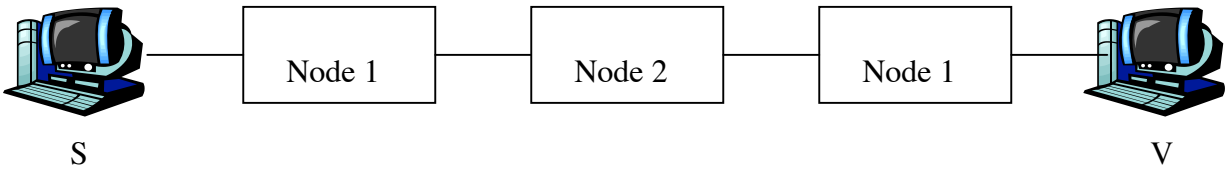
Question: Consider a channel that can lose packets but has a maximum RTT that is known. Modify protocol rdt2.1 to include sender timeout and retransmit. Informally argue why your protocol can communicate correctly over this channel.

Answer: Here, we add a timer, whose value is greater than the known round-trip propagation delay. We add a timeout event to the "Wait for ACK or NAK0" and "Wait for ACK or NAK1" states. If the timeout event occurs, the most recently transmitted packet is retransmitted. Let us see why this protocol will still work with the rdt2.1 receiver.

- Suppose the timeout is caused by a lost data packet, i.e., a packet on the sender-to-receiver channel. In this case, the receiver never received the previous transmission and, from the receiver's viewpoint, if the timeout retransmission is received, it look *exactly* the same as if the original transmission is being received.
- Suppose now that an ACK is lost. The receiver will eventually retransmit the packet on a timeout. But a retransmission is exactly the same action that is take if an ACK is garbled. Thus the sender's reaction is the same with a loss, as with a garbled ACK. The rdt 2.1 receiver can already handle the case of a garbled ACK.

FROM PAST EXAMS

Question: In this problem we will compare the performance of circuit switching with datagram packet switching. Figure shows a source host S and a destination host D connected by a switching network consisting of three nodes. We will make the following assumptions:



a. The transmission rate of all the links is 9600 bits per second.
b. The propagation delay in each hop is 0.001 seconds.
c. The size of the message that needs to be sent from S to D is 3200 bits.
d. The end-to-end connection setup time for circuit switching is 0.2 seconds.
e. For packet switching the packet size is 824 (this includes 24 bits of header).
f. Assume that for packet switching there is no queuing and processing delay at the intermediate nodes.
g. Assume that no acknowledgements are required.

Find the time to send the message from the source to the destination assuming that the network uses (i) circuit switching and (ii) packet switching. Show all calculations.

Answer:

Circuit Switching:
Total delay = setup time + propagation delay + transmission delay
= 0.2 + 4 (0.001) + 3200/9600 = 0.5373 seconds

Datagram Packet Switching:
Total number of packets = 3200/800 = 4
The first packet will take 4 * (824/9600) + 4 (0.001) seconds to reach the destination.
The second packet will arrive (824/9600) seconds after the first one.
In this way, the last packet will arrive 3 * (824/9600) after the first packet.
Hence the total delay = 4 * (824/9600) + 4 (0.001) + 3 * (824/9600) = 0.60483 seconds.

Question: Assume that Bob uses an e-mail client (mail reader) such as Outlook to send an e-mail to Alice who uses a Web-based e-mail account (such as Hotmail). The IP address of Alice's mail server is initially unknown to Bob's mail server. List all the transport and application layer protocols that are involved from the time when Bob sends the e-mail to the time when Alice reads it. Clearly indicate in which part of the transfer of the e-mail these transport/application layer protocols are used

Answer:
SMTP over TCP is used to transfer the mail from Bob's e-mail client to his SMTP server.

DNS over UDP is used by Bob's mail server to get the IP address for Alice's mail server.

SMTP over TCP is used to transfer the mail from Bob's mail server to Alice's mail server.

Finally HTTP over TCP is used by Alice to read this e-mail.

Question: Consider a server that is employing TCP SYN cookies. Why is it necessary for the server to use a special initial sequence number? (Answer in one sentence only)

Answer: The server uses special initial sequence number (that is obtained from the hash of source and destination IPs and ports) in order to defend itself against SYN FLOOD attack.

Question: How do you think TCP would handle the problem if an acknowledgement were lost, so that the sender retransmitted the unacknowledged TCP segment and, therefore, the receiving transport process received the same segment twice? (answer in one sentence only)

Answer: The receiver will simply discard the packet since it can detect that it is a duplicate copy (using sequence numbers).

Question: If a laptop is connected to Ethernet and sends a DNS request, what is the sequence of packet headers on this request packet as it leaves the laptop (starting from the outermost header)? (We are using the term header loosely when it comes to DNS.) Choose from one below.

(a) Ethernet, IP, UDP, DNS

(b) DNS, UDP, IP, Ethernet

(c) IP, TCP

(d) Ethernet, TCP, IP, DNS

(e) None of the above.

Answer: (a)

Question: You are trying to transfer the contents of a 1.25terabyte disk drive between here and New York, and have at your disposal two methods: (a) sending the data over a

100mbps link or (b) sending the drive by Federal Express (with a guarantee that it will arrive in 24 hours). Assume the network charges 10^{-10} cents per bit transmitted, and Federal Express charges $30 for the package. Which option is faster? Which option is cheaper?

Answer:

Time to transmit the data over the link = 1.25terabytes/100mbps = $1.25 \times 8 \times 10^6 / 100$ = 100000 seconds = 1.15 days

Note: approximating 1 terabit = 10^6 megabit & assuming all other network delays (propagation, queueing, etc) are negligible.

Time to transmit data using FedEx = 1 day

Hence, the FedEx option is faster.

Cost to transmit the data over the link = $1.25 \times 8 \times 10^{12} \times 10^{-10}$ = 1000 cents = $10

Cost to transmit the data using FedEx = $30

Hence, the network option is cheaper.