CS118
Spring 2010 Midterm Exam Solution
4/29/2010

1 hour 50 minutes
OPEN book and OPEN notes

- This exam has 7 pages, including this cover page. Do all your work on these exam sheets, use the back side if needed.
- If you use the back pages for *scratch space*, cross out your scratch work before you submit the exam.
- Be *specific* and *clear* in your answers, and *explain* all your answers.
- When the answer to a problem is not immediately clear, think about it carefully instead of simply dumping anything and everything, relevant or irrelevant, on the paper. Entirely irrelevant answers may lead to point deduction as they show lack of understanding to the problem.

Your name: ______________________

ID number: ______________________

	Points	Your score
Problem 1	20	
Problem 2	20	
Problem 3	20	
Problem 4	20	
Problem 5	20	
Total	100	

Problem 1 (20 points) Short and quick questions

(1) Popular web servers, such as cnn.com, may handle web requests from millions of users at the same time. How does a web server distinguish TCP the connections with different browsers?

(IP address, Port number) tuple can be used to distinguish a TCP connection of a different browser from another TCP connection.

(2) Are both the HTTP status code and corresponding reason string in HTTP response messages (see slides 13 in week2 slide deck) necessary for the protocol to function correctly? Please explain your answer.

No, only the status code is needed for HTTP to operate correctly. The reason string is optional and used by the browser to show human readers to understand the error.

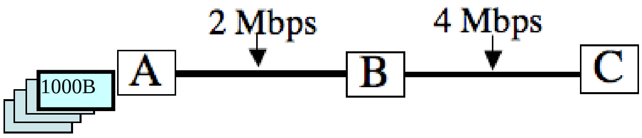
(3) HTTP is called a stateless protocol, that is, it handles each HTTP request independently from any previous requests. However when one looks up certain websites, such as www.amazon.com, that one has visited before, the websites can recognize the user instantly. With this fact, do you believe that HTTP is indeed a stateless protocol, or is that a false statement?

It's a false statement, and HTTP is indeed a stateless protocol. Each HTTP connection is independent from any other previous HTTP connection and does not carry any state information. This makes HTTP a stateless protocol. What makes the above phenomenon happen is through the use of HTTP cookie, which can be used to record states by the end host, and not by the protocol.

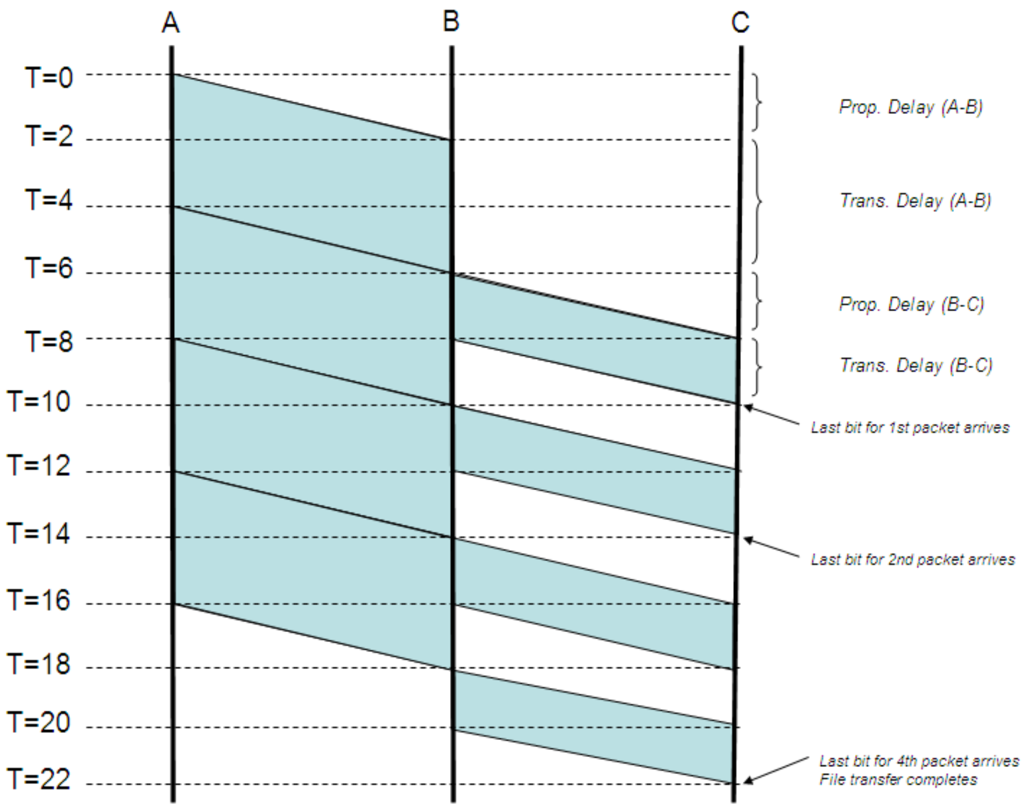
(4) The email system uses 2 types of protocols. The first is made of a single protocol, SMTP. The second type includes POP3, IMAP, etc. which are used for accessing email. Why do we need both types? Please state the basic difference in their functionality?

Both protocols are needed to build an email system because the functionality provided by SMTP and POP3 (or IMAP) is different. SMTP is used to send emails whereas POP3 is used to fetch emails. Without both, one will only be able to send or receive, but not both.

Problem 2 **(20 points)**
Consider sending 4 packets from Node A to Node C via Node B (see the figure below). The packet length is 1000 bytes each. The propagation delay of both Link A-B and link B-C is 2 msec (0.002 second). Link A-B's bandwidth is 2 Mbps (2x10^6 bits per second), and link B-C's bandwidth is 4 Mbps.



Assume A starts transmitting the first packet at time t = 0,
a) What is the time gap between the first and second packets when they arrive at C? (the time gap between receiving the last bit of the first packet and the last bit of 2nd packet)



The last bit for the 1st packet will be received by C at T=10ms. The last bit for the 2nd packet will be received by C at T=14ms. So, the time gap between the two is 4ms.

b) When will C receive all the 4 packets?

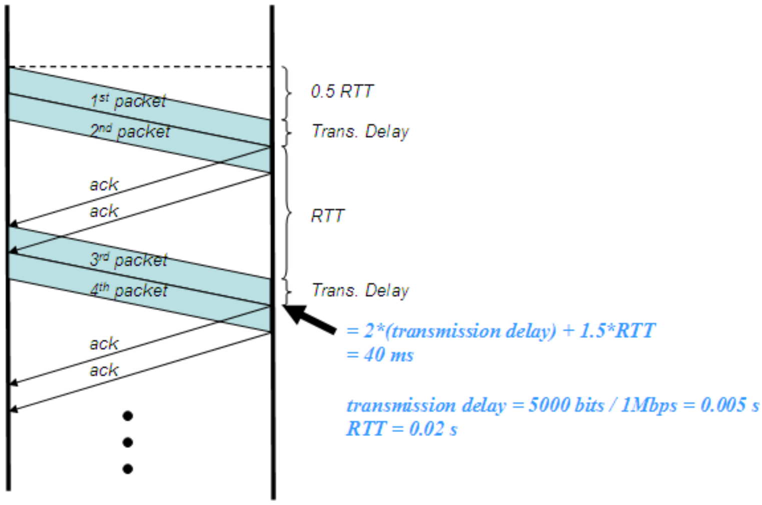
Time it takes for A to send all 4 packets to B:
Propagation delay (A-B) + Transmission delay of sending 4 packets = 2 ms + 4ms * 4 = 18ms

Within the above 18ms, B can send the first 3 packets to C. It takes B an additional propagation delay (B-C) + transmission delay (B-C) to send the final packet to C.

Total time: 18 ms + 4 ms = 22 ms

Problem 3 (20 points) Two hosts A and B are connected by a link with bandwidth of 1 Mbps (10^6 bits-per-second) and propagation delay of 10 msec (the round trip propagation delay will be 0.02 seconds). Host A has a 10,000 byte file to send to host B. A uses GoBackN reliable transport protocol and divides the file into 5000-bit packets. The GoBackN protocol uses a fixed window size of 2 packets. Assume the *transmission time* of ACK packets is negligible (zero) and no data or ACK packet ever gets lost.

(1) How long will it take before the 3rd packet has completely arrived at Host B? (drawing a diagram may help here).

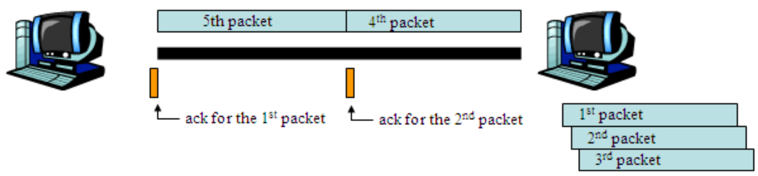


(2) Can the file be delivered to host B faster by changing the GoBackN window size? If so, what is the minimal window size that would allow the file be received at B with shortest possible time (assume no other setting is changed)?

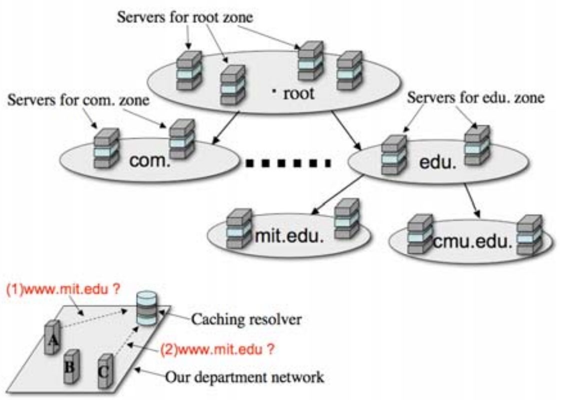
The formula to calculate the minimum window size is

Minimum window size = ceiling (BDP / packet_size) + 1 = ceiling (1Mbps*0.02s / 5000 bits) + 1 = 4 + 1 = 5
The minimum windows size is 5 packets.

The intuition behind the formula is that the sender must keep sending the data without waiting for the receiver's windows to open. By being able to send 5 packets (ie. window size = 5) without waiting for any acknowledgement, the sender will be able to fill up the link until the acknowledgement for the first packet arrives to the sender.



Problem 4 (20 points)



Consider the following DNS resolution process: assuming that at time T=0, the caching resolver in the figure has an empty cache, and Host-A sends a query to resolve the DNS name of *www.mit.edu*. After Host-A received the answer from the caching resolver, Host-C sends a query for the same name. 10 seconds after Host-C receives its answer, Host-B sends a query for DNS name *www.cmu.edu* (not shown in the figure). Assuming that it takes 0 (zero) second for all packet exchanges between the local hosts and the caching resolver, and it takes 100 msec for the caching resolver to get a reply for all the DNS queries it sends, and all the DNS data has a TTL of 24 hours.

(1) How long does it take for Host-A to get the answer back for the IP address of *www.mit.edu*?

The caching resolver first has to query the root since it does not have any cache. Then, it has to make an additional query to edu. and mit.edu. servers to finally find the IP address of www.mit.edu. There were 3 queries that the cache resolver has to make in this case. So the latency will be 100ms * 3 = 300 ms.

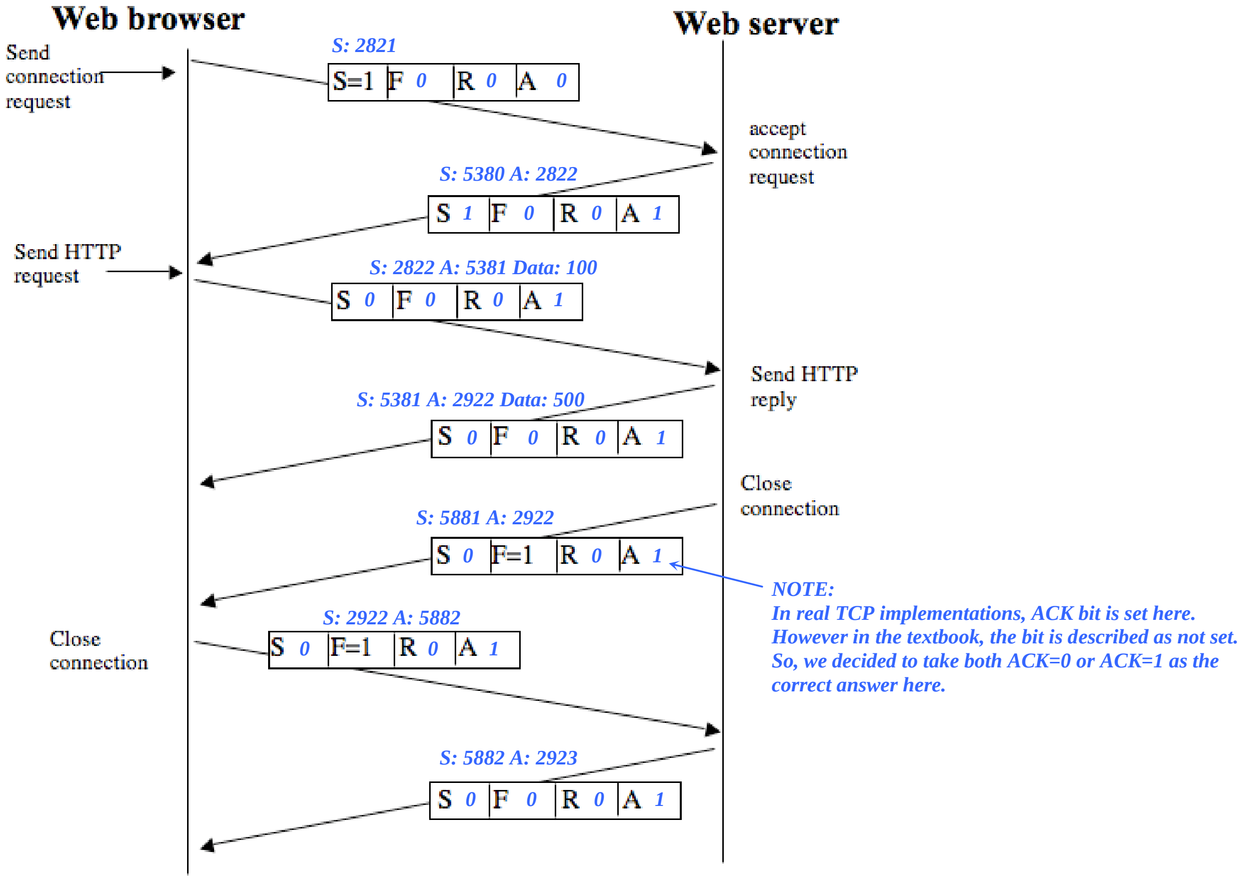
(2)How long does it take for Host-C to get the answer back for the IP address of *www.mit.edu*?

Since the name is already in the cache, the cache resolver will not have to query the DNS system again. The latency in this case is 0ms.

(3)How long does it take for Host-B to get the answer back for the IP address of *www.cmu.edu*?

Because cache resolver knows the IP address of edu. server, it can start querying from there. The cache resolver needs to ask .edu server and cmu.edu. server to find the IP addresses for DNS server for cmu.edu. zone and the IP address for www.cmu.edu respectively. The total number of queries to the DNS system from the cache resolver in this case was 2, and the latency will be 100ms * 2 = 200 ms.

Problem 5 (20 points) The following diagram shows a sequence of TCP packets for a session between a web browser and a web server. The HTTP in use is version 1.0 (non-persistent HTTP).
(1) Fill in all the missing flag values for the SYN, FIN, RST, and ACK flags in the TCP headers (when the flag is set, the value is 1, otherwise is 0).



(2) The web browser starts its TCP connection with the initial sequence number 2821, the web server's initial sequence number is 5380. Assuming the HTTP request size is 100 bytes, and the HTTP reply is just one packet with 500 byte data. What is the data sequence number in the *last* packet (with F=1) sent by the browser? Explain your answer.

2922
In TCP, accumulative sequence number and acknowledgement is used.

(3) What is the value in the sequence number, and acknowledgement number in the TCP header of the last packet sent by the Web server?

SEQ: 5882, ACK: 2923 (Please refer to the figure above how these numbers are obtained)