Name_______________ Student ID____________________ Department/Year__________

Midterm Examination

Introduction to Computer Networks
Class#: 901 E31110
Fall 2006

9:20-11:00 Tuesday
November 14, 2006

Prohibited

1. You are not allowed to write down the answers using pencils. Use only black- or blue-inked pens.
2. You are not allowed to read books or any references not on the question sheets.
3. You are not allowed to use calculators or electronic devices in any form.
4. You are not allowed to use extra sheets of papers.
5. You are not allowed to have any oral, visual, gesture exchange about the exam questions or answers during the exam.

Cautions

1. Check if you get 14 pages (including this title page), 6 questions.
2. Write your **name in Chinese**, student ID, and department/year down on top of the first page.
3. There are in total 150 points to earn. You have 100 minutes to answer the questions. Skim through all questions and start from the questions you are more confident with.
4. Use only English to answer the questions. Misspelling and grammar errors will be tolerated, but you want to make sure with those errors your answers will still make sense.
5. If you have any extra-exam emergency or problem regarding the exam questions, raise your hand quietly. The exam administrator will approach you and deal with the problem.

1. (Overview) Packet switching and circuit switching are two ways of architecting data transfer in the core of a network. Suppose we have a simple 2-node network with a link in the middle connecting the 2 nodes. L users send data from one node to another through the link in the middle. The bandwidth of the link is M bps. Each user is active sending data P% of the time. Let p be P/100. When the user is active, it sends data in a constant rate R bps. Try if you can answer the following questions.

   (1) How do packet switching and circuit switching networks differ in general? (10%)
   (2) Give formula for the number of users N such a circuit-switching network can serve (in terms of M and R). (5%)
   (3) Give formula for the probability Pr of less than or equal to N users active simultaneously in such a packet-switching network (in terms of N, L, and p). (5%)

Sample Solution:

(1) Packet switching:

data sent in discrete chunks,
no call setup,
multiplexing,
congestion,
need additional mechanism to provide quality of service

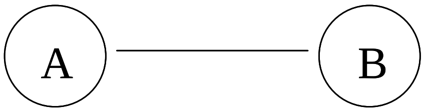
Circuit switching:

one dedicated circuit per call,
call setup,
reserved resource (idle when there're no data within the call),
no congestion,
easier to provide quality of service

(2) $N = \frac{M}{R}$

(3) $Pr = \sum_{i=0}^{N} \binom{L}{i} \mathrm{p}^{\mathrm{i}} (1-\mathrm{p})^{\mathrm{L-i}}$

2. (Overview) Consider a simple network as follows. The bandwidth of the link is R bits per second and the packets always come in the size of P bytes. The link speed is S meter per second and the length of the link is D meters.

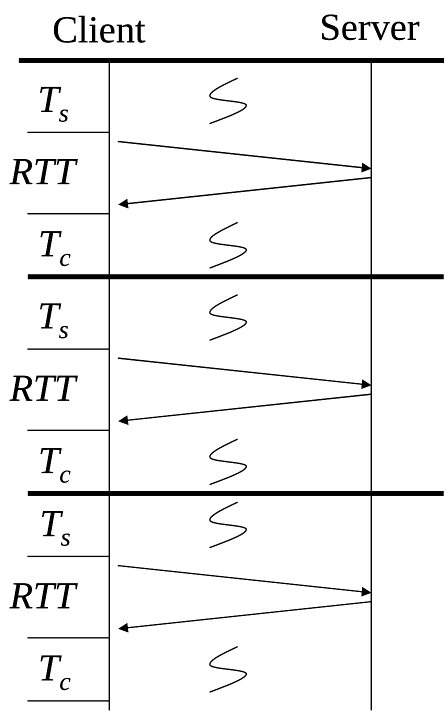


(1) Give the transmission delay of a packet (in terms of R, P, S, or D). (5%)
(2) Give the propagation delay of the link (in terms of R, P, S, or D). (5%)
(3) Suppose node A has infinite amount of buffer space and the data come into node A in a constant rate, 2R. Give the amount of data waiting to be transmitted in node A at time T seconds (in terms of T, R, P, S, or D). (5%)
(4) Suppose the processing delay is negligible. Give the nodal delay of the packet arriving at node A at time T (in terms of T, R, P, S, or D). (5%)

Sample Solution:

(1) 8P/R seconds
(2) D/S seconds
(3) RT bits
(4) T + 8P/R + D/S seconds

3. (Application) Web clients and servers may send their HTTP requests and replies using the (1) non-persistent, (2) persistent without pipelining, or (3) persistent with pipelining mode. Suppose we have a web page which contains 3 objects, a .html main file, a .jpg image, and a .mp3 background music. The 3 objects are all so small that (1) the transmission time is negligible and (2) each object can be completely transferred in one TCP packet. Assume there are no packet losses, the round-trip time between the server and client is *RTT*, the time to set up a TCP connection are T_s, and the time to close down a TCP connection is T_c. In the non-persistent HTTP mode, the time required to complete the transfer of the web page is $3T_s + 3RTT + 3T_c$ and the communication between the client and server is depicted on the right. Based on the same assumptions, try if you can address the following questions.



(1) What will be the time it takes to complete transferring the web page running in the persistent without pipelining mode? (5%)

(2) If the web client learns the URL of the 2nd and 3rd objects in the page after fully downloading the main html object, what will the entire page download time be using the persistent with pipelining mode? (5%)

Sample Solution:

(1) $T_s + 3RTT + T_c$

(2) $T_s + 2RTT + T_c$

4. (Application) Compare the client-side cache of HTTP and DNS.
   (1) How does HTTP client-side cache work? (5%)
   (2) How does DNS client-side cache work? (5%)
   (3) Compare the two caching schemes in terms of delay, bandwidth and server load they save? In other words, discuss how effective they are in reducing the delay, bandwidth, and server load. (10%)
   (4) Following (3), state the possible reason(s) of the design decision. Hint: think how frequent an HTTP or DNS server might need to reply to the clients and how large the replies are to be sent to the client. (10%)

Sample Solution:

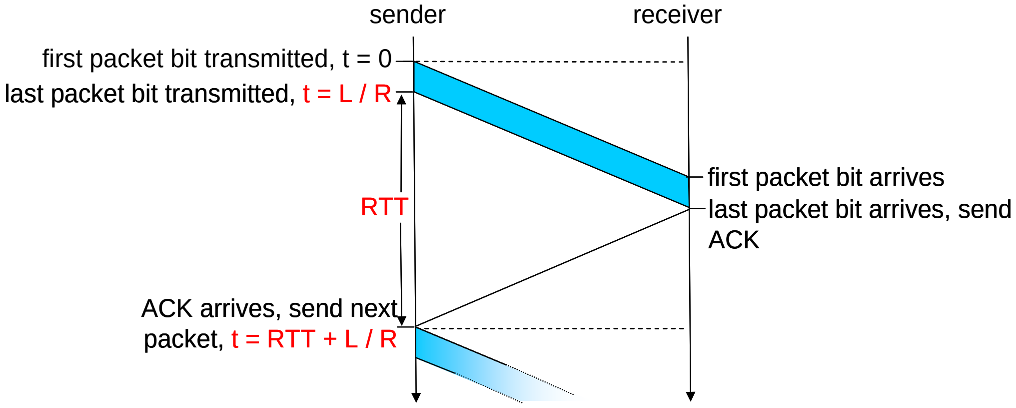
(1) An HTTP web page is stored on the client with the date of its last update. When the same page is requested, HTTP sends the HTTP REQUEST message with the if-modified-since field set to the date of last update of the cached copy. In case the date of last update of the server copy is not newer, an HTTP RESPONSE with code 304 Not Modified will be sent back to the client without the content of the web page. Otherwise, an HTTP RESPONSE with the latest content of the web page is sent back to the client and the client caches the page with the new date of last update.

(2) A DNS mapping entry is stored on the client for a certain period of time. It times out (or disappears) after the period of time. When the same mapping is requested and a cache entry exists, the client does not send a DNS QUERY to its DNS server. Otherwise, it does.

(3) The client-side cache of HTTP reduces the amount of bandwidth consumption due to transmission of already up-to-date web page. The client-side cache of DNS reduces the server load in addition to the bandwidth and delay.

(4) There are usually a small number of DNS servers per administrative domains. Each of the servers needs to answer a large amount of queries everyday. The DNS QUERY and REPLY messages are, on the other hand, small containing usually a small number of hostname-IP mappings. These make the server load the performance bottleneck for DNS servers. The HTTP servers are quite the opposite. The HTTP RESPONSE messages are large (Web pages are relatively larger files to transmit). Besides, only the clients looking for specific web pages hosted by the servers will request to the servers. In terms of the frequency of receiving requests, the HTTP servers have much less a problem than the DNS servers do. Instead, the performance bottleneck for HTTP servers is more the bandwidth consumption. Trying to tackle their performance bottlenecks, each comes up with its own suitable client-side caching scheme. DNS opts for the scheme that reduces the server load effectively, and HTTP uses one that saves the bandwidth.

5. (Transport) Consider rdt 3.0. To ensure reliable transfer, the sender sends a packet and waits for the acknowledgement to come back. Assume there is no error. The total time to complete reliable transfer of a packet is $L/R+RTT$, where L is the packet size in bits, R is the link rate in bits per second, RTT is the round-trip time between the sender and receiver in seconds. During the $L/R+RTT$ seconds of time, the fraction used really to transmit data is L/R. Therefore, the system utilization of rdt 3.0 is $\frac{L/R}{RTT+L/R}$.



Consider rdt3.0 with pipelining, where N packets are sent back to back within a round. Note that $N * L/R$ is significantly smaller than RTT. Derive the following terms step by step to come to the utilization of rdt3.0 with N pipelined packets for long data transfers.

(1) The time used really to transmit these N packets (5%)
(2) The total time to complete reliable transfer of these N back to back packets (5%)
(3) The utilization of rdt3.0 with N pipelined packets in one round (5%)
(4) The utilization of rdt 3.0 with N pipelined packets in K rounds (10%)
(5) The utilization of rdt 3.0 with N pipelined packets in K rounds when $K \to \infty$ (5%)

Sample Solution:

(1) $N * L/R$

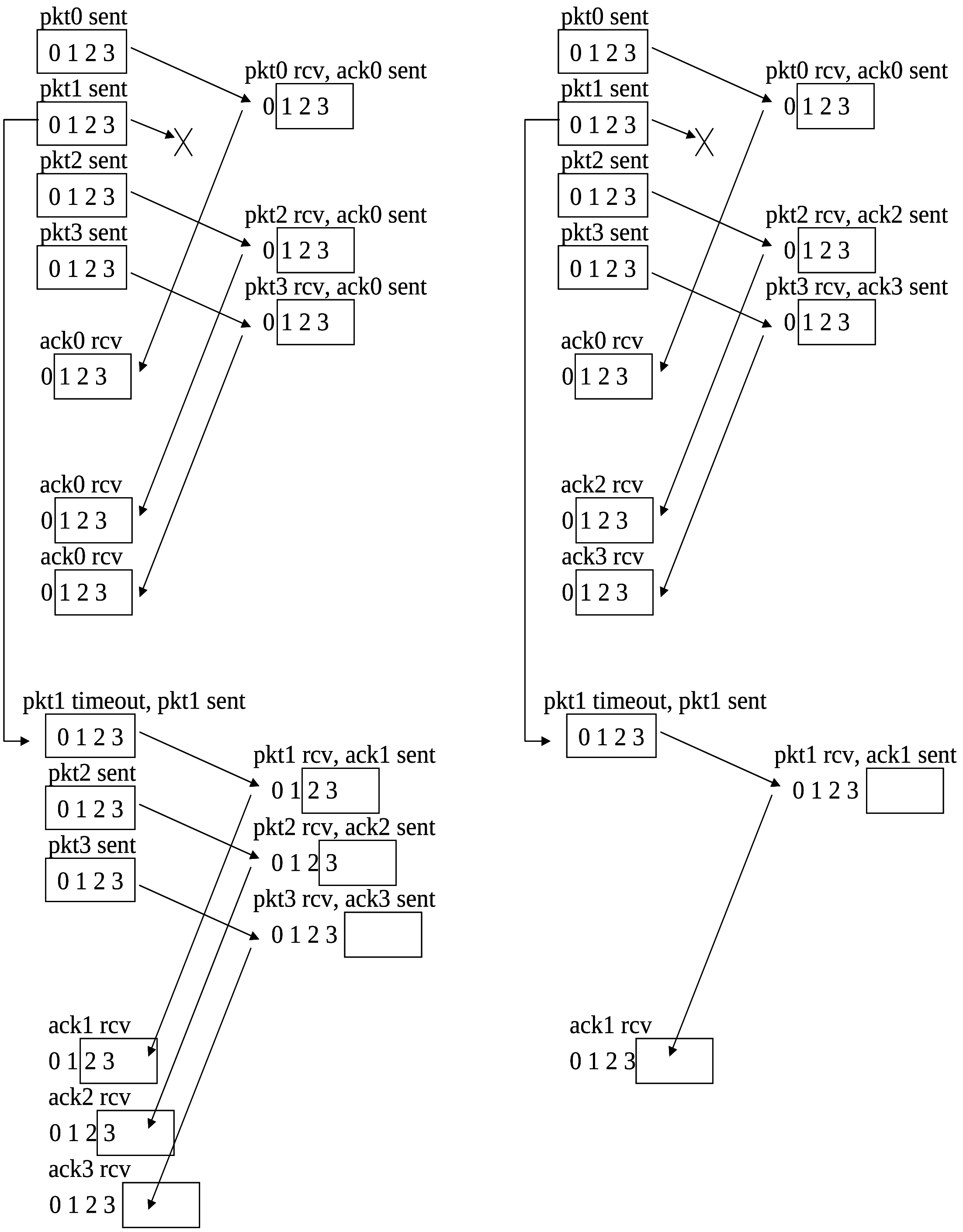
(2) $N * L/R + RTT$

(3) $\frac{N * L/R}{L/R + RTT + (N-1) * L/R}$

(4) $\frac{K * N * L/R}{K * (L/R + RTT) + (N-1) * L/R}$

(5) $\frac{N * L/R}{L/R + RTT}$

6. (Transport) Let's compare and contrast Go-Back-N (GBN) and Selective Repeat (SR) using the following scenario. There are in total 4 packets to send and the window sizes of GBN and SR are both 4. Let the transmission time of a packet (*TX*) be significantly smaller than the round-trip time (*RTT*) and the *RTT* be significantly smaller than the retransmission timeout duration (*TO*). The packet transfer diagrams for GBN (left) and SR (right) when packet 1 is lost are depicted as follows.



The total amount of time required to complete transferring 4 packets for GBN is *4TX + RTT + TO* and for SR is *2TX+RTT+TO*. Although the transfer time for GBN is longer, there is no need of buffer space at the receiver and only one timer is required at the sender.

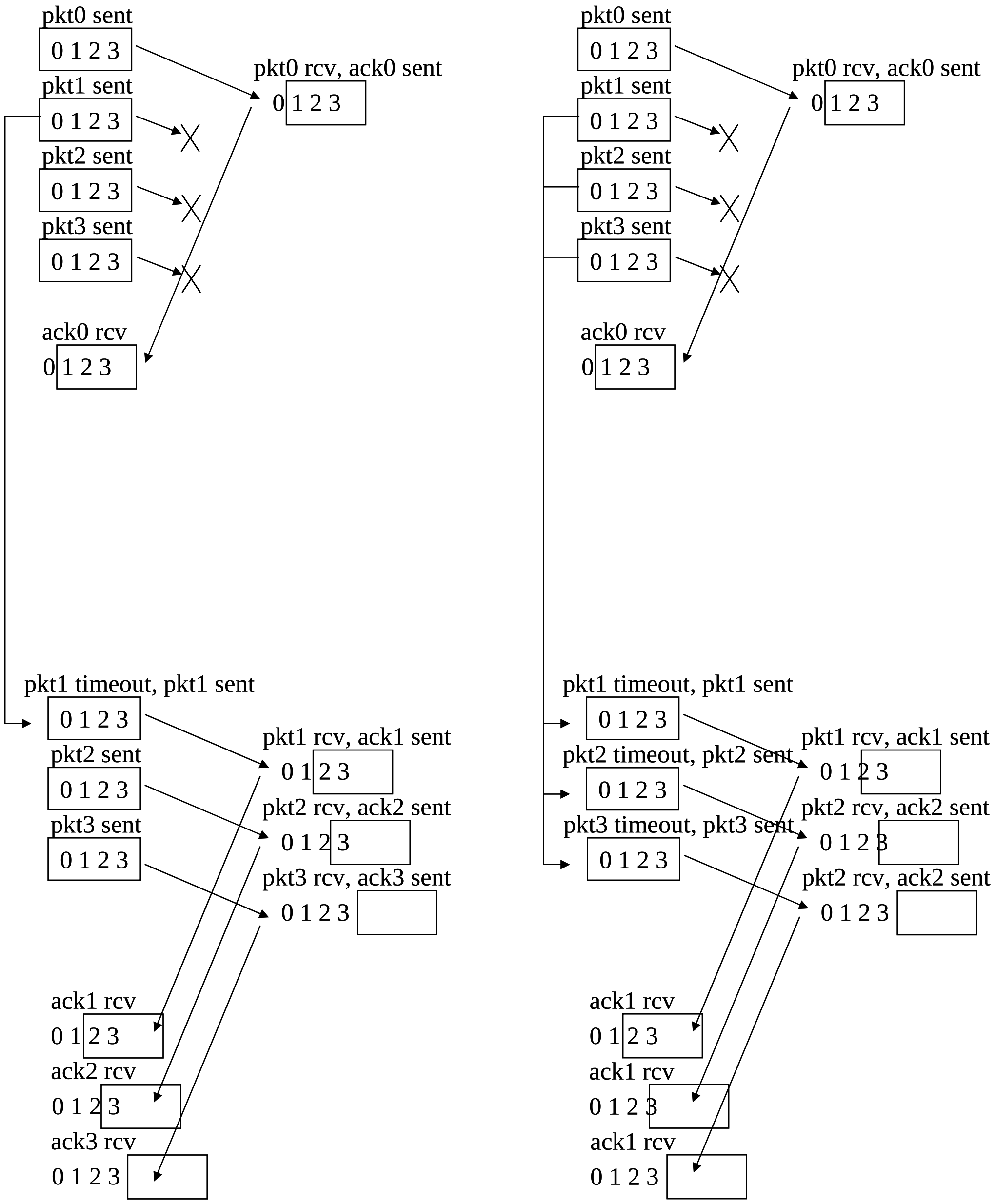
(1) Draw the packet transfer diagram GBN and SR when packet 1, 2, 3 are all lost. (20%)

(2) Derive the total amount of time required to complete transferring 4 packets when packet 1, 2, 3 are all lost. (10%)

(3) If the packet losses within a window are often consecutive, which pipelining reliable transfer mechanism, GBN or SR, will be better? And why? (10%)

Sample Solution:

(1)



(2) Both *4TX + RTT + TO*

(3) Make your choice here and give arguments. For example: I choose GBN. When losses are often consecutive, GBN is not worse than SR in terms of delay and retransmission overhead. Yet, GBN is better in terms of the number of timers required at the sender and the amount of buffer space required at the receiver.

Actually, there a major flaw in the description of problem 6. The packet exchange diagram for GBN when packet 1 is lost should be as follows and the time to finish the reliable transfer should be *4TX +2RTT +TO*. The packet exchange and transfer time for GBN when packet 1, 2, and 3 are all lost are the same as the case when only packet 1 is lost. Given that we might have mislead students towards thinking the timer for pkt 1 starts at the time pkt 1 is sent, we will give away the points in (1) and (2) about GBN packet exchange and transfer time.