CS433/533: Computer Networks

Exam 1: Application and Transport

03/31/2008

7:00-8:30 pm

- This exam is closed book. However, you may refer to a sheet of 8.5"x11" paper of your own design.
- ***Keep your answer concise.***
- Show your reasoning clearly. If your reasoning is correct, but your final answer is wrong, you will receive most of the credit. If you just show the answer without reasoning, and your answer is wrong, you may receive no points at all.

Name: _____

Network Applications and Architecture (35 points)	Reliability (23 points)	Congestion Control (32 points)	Total (90 points)

a) [35 points] **Network Applications**

a) [5 points] If an HTTP server is serving N clients, at least how many sockets are open at the server?

N+1
Typical error: N (forgot the listening socket)

b) [5 points] If a hacker wants to spoof the sender address of an email, where would he place the spoofed address?

MAIL FROM: <spoof>; From: <spoof>

c) [5 points] DNS: Does a DNS server for a domain (e.g., cs.yale.edu) have to be on the same network as the hosts whose names it resolves?

No.

d) [5 points] Consider query flooding in P2P file sharing. Suppose the forwarding policy is that each peer keeps no state about queries, and forwards each received query to all but the incoming neighbor. Someone claims that this may cause infinite forwarding of a query message. Is this possible? If no, what is the maximum number of times a query message is forwarded? If yes, please give a simple example, and then propose how do you fix the issue?

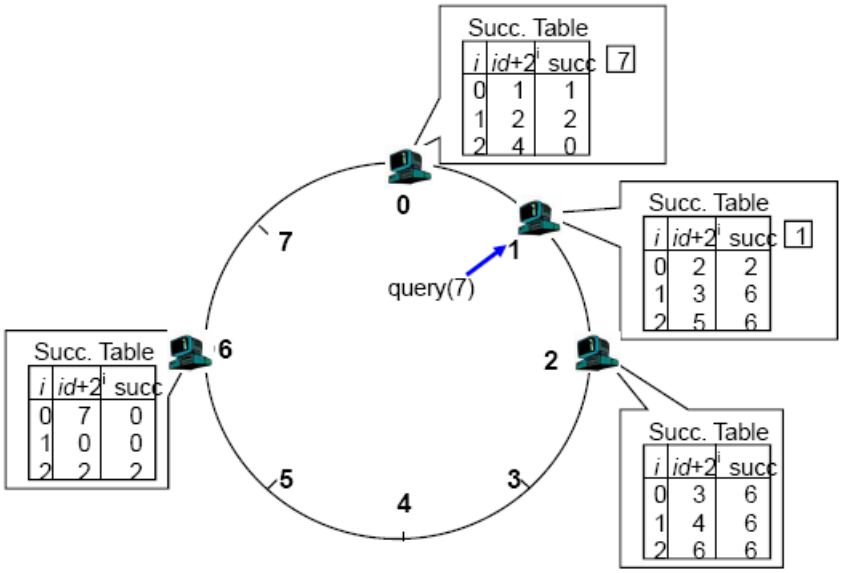
A simple case is 3 nodes forming a circle.
TTL; each peer remembers seen queries; a query carries all peers visited.

e) [5 points] The designers of Freenet claim that their P2P system supports anonymity and is censorship-resistant. Please list two features of Freenet that are related with this claim.

Chang of originator when insertion (anonymous); random TTL (hide distance); caching along the search path to avoid search-and-delete (censorship-resistant).
Relatively weak answer: a distributed system.

f) [5 points] Distributed Hash Tables are being integrated into many P2P applications. An example DHT design we have seen is Chord, which has been used in several P2P applications. In Chord, each node with *id* maintains a successor table for item ids of $id+2^i$, i=0, … For a query of an item with *qid*, the next stop is the largest $id+2^i$ entry that does not exceed *qid*. Consider the Chord network shown in the figure below. Two items are inserted into the network: item 1 at node 1, and item 7 at node 0. Assume that a query for *qid*=7 is issued at node 1. Please label on the figure how the query is forwarded.

Typical error: goes to 5 instead of 6 but there is no node at 5.



g) [5 points] In class, we discussed that when the utilization (ρ) of a link approaches 1, the delay will approach ∞ (i.e., delay is proportional to 1/(1-ρ)). One objective of TCP congestion control is to reach full bandwidth utilization. Does this imply that delay under TCP will reach ∞? Please justify briefly.

The analysis model does not apply to TCP: In the derivation of the delay formula, we assumed that arrivals are random and fixed (not dependent on delay). TCP is self-clocking and therefore arrivals will slow down as delay increases; finite window size/buffer size arguments are also fine.

[23 points] **Reliability and Connection Management**

a) [5 points; NAK or ACK] You are designing a reliable transport protocol from an outer space probe to the ground station. Suppose that the probe sends data only infrequently and irregularly. If you could choose only between a NAK-only or an ACK-only protocol, which one would you choose? Please justify briefly.

ACK is more appropriate. NAK-only is ineffective to detect losses (points depend on justification; if uses BW as reason, gives more points).

b) [6 points; Selective-Repeat/Go-back-n] With Selective-Repeat, is it possible for the sender to receive an ACK for a packet that falls outside of the current window? How about Go-back-n? The network has no out-of-order packets.

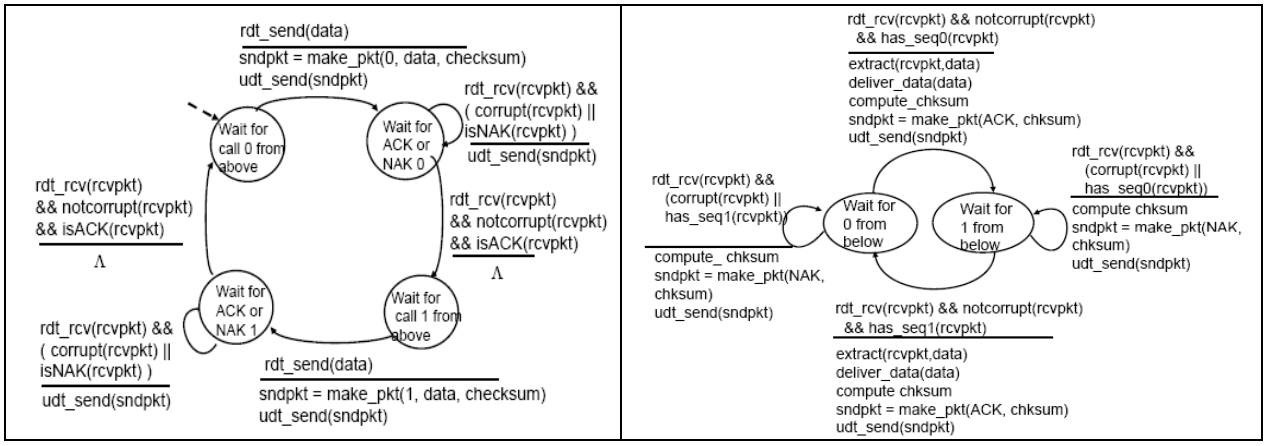
True. Suppose the sender has a window size of 3 and sends packets 1, 2, 3 at t0. At t1 (t1 > t0) the receiver ACKS 1, 2, 3. At t2 (t2 > t1) the sender times out and resends 1, 2, 3. At t3 the receiver receives the duplicates and re-acknowledges 1, 2, 3. At t4 the sender receives the ACKs that the receiver sent at t1 and advances its window to 4,5, 6. At t5 the sender receives the ACKs 1, 2, 3 the receiver sent at t2. These ACKs are outside its window.

c) [6 points; Connection Setup] A common denial-of-service attack on servers is TCP SYN flood attack, where a bad guy sends a large amount of TCP SYN packets with spoofed source addresses to a server. The server allocates resources (e.g., socket data structure, buffers) and sends ACK/SYN, but no reply comes back since there is no host with a spoofed address. SYN cookies are now employed by major operating systems. With SYN cookies, when a server first receives a TCP SYN packet with <src address, src port, dest address, dest port>, it does not keep any state, but only returns an ACK/SYN with a server initial sequence number N that is a one-way function (not reversible function such as MD5) of <src address, src port, dest address, dest port, X>. Then when a server receives the first ACK from a client, the server checks if the ACK is for N. What should the X be in the calculation? Is this scheme effective?

X should be a random, private number of the server that changes periodically. It is private so that cannot spoof the seq number. It should be changed periodically to prevent replaying attack. X is not a per-connection state.

Other answers from the exam: a fixed value; -1; a random value for this connection (accepted); initial SYNC request (-4; -2 if state hash function is private).

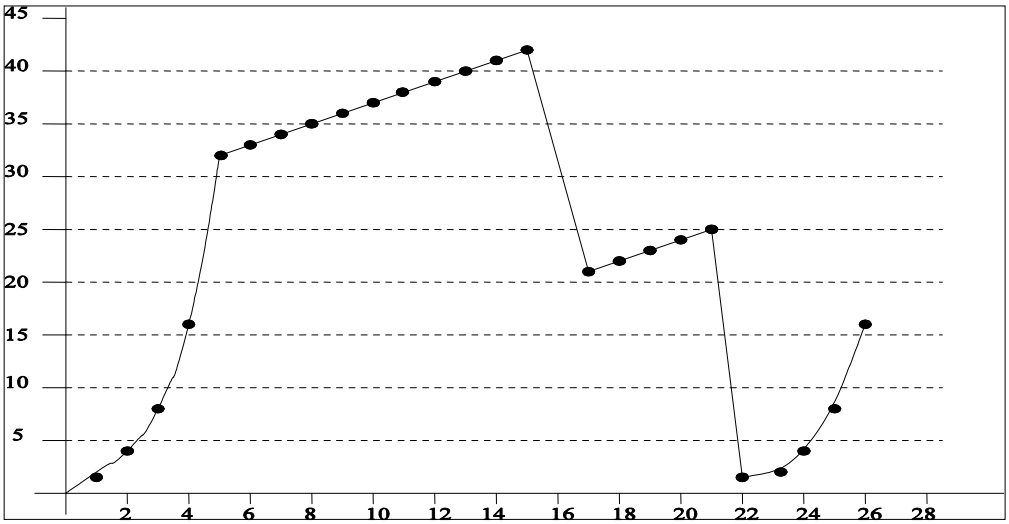
[6 points; State Management] Consider the following alternating-bit protocol (stop-and-wait), where the left is the sender finite state diagram and the right receiver. Assume only corruptions of packets. Someone argues that this design might lead to deadlock---an scenario that sender and receiver cannot make any progress. Is this true or false?



Yes. Suppose the sender is in state “Wait for call 1 from above” and the receiver (the receiver shown in the homework problem) is in state “Wait for 1 from below.” The sender sends a packet with sequence number 1, and transitions to “Wait for ACK or NAK 1,” waiting for an ACK or NAK. Suppose now the receiver receives the packet with sequence number 1 correctly, sends an ACK, and transitions to state “Wait for 0 from below,” waiting for a data packet with sequence number 0. However, the ACK is corrupted. When the rdt2.1 sender gets the corrupted ACK, it resends the packet with sequence number 1. However, the receiver is waiting for a packet with sequence number 0 and always sends a NAK when it doesn't get a packet with sequence number 0. Hence the sender will always be sending a packet with sequence number 1, and the receiver will always be NAKing that packet. Neither will progress forward from that state.

3. **[32 points] Congestion Control and Optimization**

Consider the following plot of TCP/Reno window size as a function of time.



a) [4 points] Identify the intervals of time when TCP slow start is operating (plus or minus 1 is fine; similar below).

`0-5, 22-26`

b) [4 points] Identify the intervals of time when TCP congestion avoidance is operating.

`5-22`

c) [4 points] After 16th transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

`triple duplicate ACK`

d) [4 points] After the 22nd transmission round, is segment loss detected by a triple duplicate ACK or by a timeout?

`timeout`

e) [4 points] What is the initial value of ssthresh at the first transmission round?

~ 32
Answer from exam: infinite (due to ambiguity??).

f) [6 points] There are serious concerns about TCP for high bandwidth links, in particular for wireless. The new 802.11n wireless standard, which is estimated to be released in June 2009, specifies a data rate of 300 Mbit/s. Assume 1000 bits per packet, and 1 ms round-trip time. What is (approximately) the highest wireless packet loss rate so that TCP/Reno can still achieve the aforementioned data rate?

$(1.2*MSS/RTT/Thrupt)^2 \sim 0.002\%$ or you may use 1.4 instead of 1.2.

g) [6 points] Someone states that TCP/Vegas implements Nash Bargaining Solution (NBS). What is NBS and what does the statement mean?

See lecture slides. Several people lost all six points.