EECS489 Computer Networks, Take-home Makeup Midterm (Winter 2007) due in class Wednesday 3/28 (SOLUTIONS)

Note that this is entirely optional, taking this exam will only improve your grade, and not taking it will not make your existing grade worse.

Instructions: You are allowed to use books or any other reference material; however, you cannot consult anyone else regarding the exam. Read each question carefully, and note all that is required of you. Please keep your answers clear and concise, and state all of your assumptions carefully.
Please write out the details of how you reach your final answer in order to get partial credit.

You are to abide by the University of Michigan/Engineering honor code. Please sign below to signify that you have kept the honor code pledge. Honor code pledge: I have neither given nor received aid on this exam.

Name: ______________________________

Signature: ______________________________

Unique name: ______________________________

Problem 1: Nagle's algorithm (TCP) – 10 points

The Nagle's algorithm, built into most TCP implementations, requires the sender to hold a partial segment's worth of data (even if PUSHed, i.e., with the PUSH flag set) until either a full segment accumulates or the most recent outstanding ACK arrives.

(a) Suppose the letters `abcdefghi` are sent, one per second, over a TCP connection with an RTT of 4.1 seconds. Draw a timeline indicating when each packet is sent and what it contains.

T=0.0, 'a' sent
T=1.0, 'b' collected in buffer
T=2.0, 'c' collected in buffer
T=3.0, 'd' collected in buffer
T=4.0, 'e' collected in buffer
T=4.1, ACK of 'a' arrives, "bcde" sent
T=5.0, 'f' collected in buffer
T=6.0, 'g' collected in buffer
T=7.0, 'h' collected in buffer
T=8.0, 'i' collected in buffer
T=8.2, ACK arrives, '"fghi" sent

(b) If the above were typed over a full-duplex Telnet connection, what would the user see?

The user would type ahead blindly at times. Characters would be echoed between 4 and 8 seconds late, and echoing would come in chunks of four or so. Such behavior is quite common over telnet connections, even those with much more modest RTTs, but the extent to which this is due to Nagle algorithm is unclear.

(c) Suppose that mouse position changes are being sent over the connection. Assuming that multiple position changes are sent each RTT, how would a user perceive the mouse motion with and without the Nagle algorithm?

With the Nagle algorithm, the mouse would appear to skip from one spot to another. Without the Nagle algorithm the mouse cursor would move smoothly, but it would display some inertia: it would keep moving for one RTT after the physical mouse were stopped.

Problem 2: TCP's congestion control– 10 points

Suppose TCP is used over a lossy link that loses on average one segment in four. Assume the bandwidth $\times$ delay window size is considerably larger than four segments, i.e., the link should accommodate more than 4 segments.

(a) What happens when we start a connection? Do we ever get to the linear increase phase of congestion avoidance?

Here is how a connection startup might progress: Send packet 1, Get ACK 1, Send packets 2 and 3, Get ACK 2, Send packet 4, which is lost due to link errors, so congestionWindow is set to 1.

We get lots of coarse-grained timeouts when the window is still too small for fast retransmit. We will never be able to get past the early stages of slow start.

(b) Without using an explicit feedback mechanism from the routers, would TCP have any way to distinguish such link losses from congestion losses, at least over the short term?

Over the short term such link losses cannot be distinguished from congestion losses, unless some router feedback mechanism (e.g., ICMP source quench) were expanded and made more robust. (Over the long term, congestion might be expected to exhibit greater temporal variability, and careful statistical analysis might indicate when congestion was present.)

(c) Suppose TCP senders did reliably get explicit congestion indications from routers. Assuming links as above are common, would it be feasible to support window sizes much larger than four segments? What would TCP have to do?

In the presence of explicit congestion indications, TCP might now be tuned to respond to ordinary timeout losses by simply retransmitting, without reducing the window size. Large windows could now behave normally. We would, however, need some way for keeping the ACK clocking running; coarse-grained timeouts would still necessitate a return to CongestionWindow=1 because ACKs would have drained. Either TCP's existing fast retransmit/fast recovery, or else some form of selective ACKs, might be appropriate. Either might need considerable tuning to handle a 25% loss rate.

Problem 3: Packet switched networks – 10 points

In this problem we consider sending voice from Host A to Host B over a packet-switched network (for example, Internet phone). Host A converts analog voice to a digital 64 kbps bit stream on the fly. Host A then groups the bits into 48-byte packets. There is one link between Host A and B; its transmission rate is 1 Mbps and its propagation delay is 2 msec. As soon as Host B receives an entire packet, it converts the packet's bits to an analog signal. How much time elapses from the time a bit is created (from the original analog signal at Host A) until the bit is decoded (as part of the analog signal at Host B)?

Consider the first bit in the packet. Before this bit can be transmitted, all the bits in the packet must be generated, requiring
$48 \times 8/(64 \times 10^3)$ sec = 6msec. The time required to transit the packet is $48 \times 8/(1 \times 10^6)$ sec = 384μsec.

Propagation delay is 2msec. The delay until decoding is
$6msec + 384\mu sec + 2msec = 8.384msec$.

Problem 4: Routing – 5 points

IP hosts that are not designated routers are *required* to drop packets misaddressed to them, even if they would otherwise be able to forward them correctly. In the absence of this requirement, what would happen if a packets addressed to IP address A were inadvertently broadcast at the link layer? What other justifications for this requirement can you think of?

If an IP packet addressed to a specific host A were inadvertently broadcast and all hosts on the subnet did forwarding, then A would be inundated with multiple copies of the packet.

Other reasons for hosts' not doing routing include the risk that misconfigured hosts could interfere with routing, or might not have up-to-date tables, or might not even participate in the same routing protocol that the real routers were using.

Problem 5: Routing – 10 points

Suppose most of the Internet used some form of geographical addressing (i.e., addresses are tied with particular geographical locations), but that a large international organization has a single IP network address and routes its internal traffic over its own links.

(a) Explain the routing inefficiency for the organization's inbound traffic inherent in this situation.

Inbound traffic takes a single path to the organization's address block, which corresponds to the organization's official location. This means that all traffic enters the organization at a single point even if much shorter alternative routes exist.

(b) Explain how the organization might solve this problem for outbound traffic.

For outbound traffic, the organization could enter into its own tables all the highest-level geographical blocks for the outside world, allowing the organization to route traffic to the exit geographically closest to the destination.

(c) For your method above to work for inbound traffic, what would have to happen?

For an approach such as the preceding to work for inbound traffic as well, the organization would have to be divided internally into geographically based subnets, and the outside world would then have to accept routing entries for each of the subnets. Consolidation of these subnets into a single external entry would be lost.

(d) Suppose the large organization now changes its addressing to separate geographical addresses for each office. What will its internal routing structure have to look like if internal traffic is still to be routed internally?

We now need each internal router to have entries for internal routes to all the other internal IP networks; this suffices to ensure internal traffic never leaves.