EECS 461 Final Exam
March 1998

Your Name:

This is a **closed book** examination; you are permitted one 8"x11" page of notes. You have 120 minutes to answer as many questions as possible. The number in parentheses at the beginning of each question indicates the number of points given to the question; there are 80 points in all. We expect the exam to take roughly one point per minute; we included the extra time to help ensure that everyone has enough time to complete the exam.

Write all of your answers directly on this paper. Make your answers as concise as possible (you needn't cover every available nano-acre with writing). If there is something in a question that you believe is open to interpretation, then please go ahead and interpret **but** state your assumptions in your answer.

Problem	Points Scored	Points Possible
1		10
2		20
3		15
4		10
5		15
6		10
Total		90

Problem 1: (10 points)
For each of the following statements, indicate in one sentence whether the statement is true or false, and **why**.

(a) Always trust your bank to correctly compute your balance.

(b) A properly designed congestion control or avoidance algorithm will always prevent network congestion no matter what load is placed on the network.

(c) If an IP multicast packet is dropped somewhere in the network, then either every member of the group receives the packet or no member receives it.

(d) Delay adaptation can be used in an IP network to ensure that real-time audio traffic receives adequate throughput for continuous playback.

(e) On the same workload (the client machines perform the same actions), callback-based cache coherence never sends more messages than timeout-based cache coherence.

Problem 2: End-to-end Argument (20 points)

Almost all network protocols are designed to be robust to failures. For each of the following, explain (i) what would happen if the information was lost, and (ii) how the system would recover, if at all. For example, if one side loses TCP connection state (for example, because it crashes), the TCP connection would be terminated, but the hosts could reestablish communication by making a new connection.

(a) A TCP acknowledgment

(b) A fragment of an IP packet

(c) A distance vector routing table update

(d) A packet carrying link state routing information

(e) An ATM virtual circuit entry

(f) A DNS resource record for an mit.edu host cached on a Washington name server

(g) A DNS resource record for an mit.edu host maintained on its home server at MIT. (Hint: all DNS servers are replicated.)

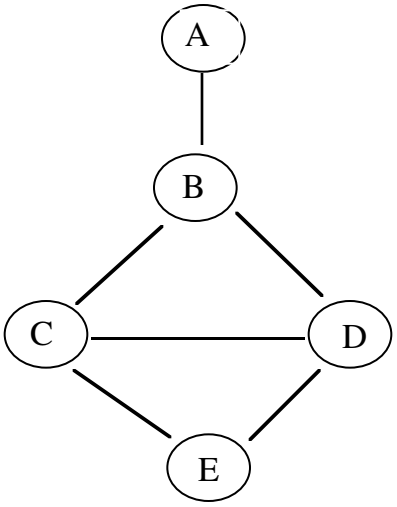
(h) A name/address entry in a host's ARP cache

(i) Prune-state in a multicast routing tree

(j) A user's private key in a public key encryption system

Problem 3: Routing (15 points)

In this problem, we consider how distance vector update messages interact with multicast. Consider the following topology:



(a) Assuming that each node initially starts with itself in its routing table (with hop count 0), and at each step a single message is sent, what is the state of the routing tables at each node after the following sequence of seven updates is completed: E->D, D->C, C->B, B->A, A->B, B->D, D->E.

(b) Show the sequence of packets that that will be sent for a multicast packet originating at node C, using reverse path flooding for the network and routing tables from (a). What happens if shortly after C sends out a multicast packet (but before the multicast has reached all nodes), C sends a routing table update to E?

(c) Suppose we send a few more distance vector messages, after the C->E message in part (b): D->B, C->D, B->C, E->C. Show the reverse path broadcast tree for multicasts originating at node B.

Problem 4: Security (10 points)

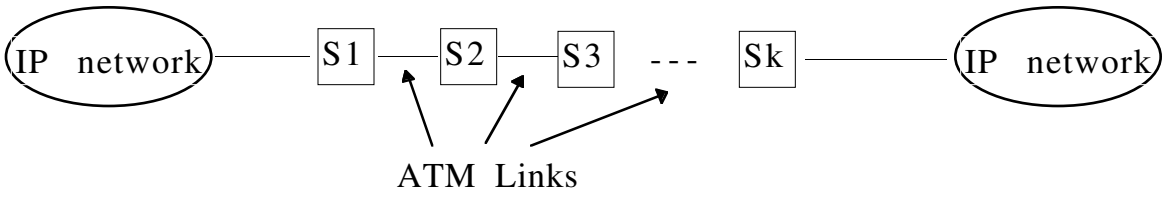
Which is more powerful, public key or secret ("private") key encryption? One way to answer this question is to ask, can we simulate one with the other? Assume we have three parties, A, B, and C, and either each pair shares a secret key (K[A,B], K[B,C], or K[A,C]), or each node has a public key pair, K[A-priv], K[A-pub], K[B-priv], K[B-pub], K[C-priv], K[C-pub], where the private key is known only to the owner, and the public key is known to everyone.

(a) Can we simulate secret key encryption with public key encryption? If so, show how we would encrypt a message from A to B using public key encryption that provides the same security as the message encrypted in K[A,B] -- in other words, the message could only come from A and be read by B. If this is not possible, explain why not.

(b) Can we simulate public key encryption with private key encryption? If so, show how we can use secret key encryption to provide (i) the same security as a message encrypted only in K[A-priv], and (ii) the same security as a message encrypted only in K[A-pub]. If either of these is not possible, explain why not. You can assume C is trusted by everyone and has a secure disk; for example, it can store messages securely for later retrieval.

Problem 5: IP switching (15 points)

Suppose we are using ATM links to carry packets between Internet routers. Here's the scenario. We have **k** routers connected by k-1 ATM links; each link runs at **b** bits per second, the IP packet size is **p** bytes, the ATM cell size is 53 bytes consisting of 5 bytes of header and 48 bytes of useful data, and the propagation delay on each link is **d** seconds. Assume that the average queue length at each router is **q** and that router table lookup takes negligible time. Sample values are k = 4, b = 155 Mb/s, p = 2400 bytes (including the IP header), d = 1 millisecond (0.001 second), and q = 2.



(a) S1 - Sk are IP routers but the links carry ATM cells. When an IP packet arrives that is routed over an ATM link, the packet is fragmented by the router into ATM cells which are then sent over the link. When the cells arrive at the next router, they are reassembled into an IP packet, and the router looks up a route in its IP forwarding table to find the next link. If the packet is routed over another ATM link, it is refragmented back into cells, and transmitted to the next router. Write an expression in terms of **k**, **b**, **p**, **d**, and **q** for the average time for the packet to traverse the k routers, from the time it completely arrives at S1 until it is ready to be sent out the link from Sk into the rest of the Internet. Calculate the result for the sample values above.

(b) A number of router vendors have recently added support, called IP switching, to optimize for this case where a number of ATM links are used in sequence. The routers can "bind" an IP flow (sequence of packets) to a virtual circuit that spans multiple routers, in this case, from S1 to Sk. The IP flow is identified by the (IP source address, IP destination address) of the IP header. Once the binding is complete, when a packet arrives at S1, S1 notices that a virtual circuit (such as VC7) has already been set up for that flow, and S1 fragments the packet into cells sent on VC7. When they arrive at S2, the router pretends it is an ATM switch, and forwards the cells as they arrive to S3, and so forth, until they arrive at Sk, where they are reassembled into an IP packet. Assuming the virtual circuit has already been set up, write an expression as in part (a) for the time for a packet to traverse the k routers. Calculate the result for the sample values above.

(c) Although binding IP flows to ATM connections reduces forwarding delay, the binding process initially incurs a delay at each router to set up the connection. Accordingly, it is not advisable to set up a virtual circuit for every packet that shows up at the edge of an IP. Suppose it takes **T** seconds to set up a virtual circuit at a router/IP switch, write an expression for how big a packet must be for it to make sense to set up a virtual circuit just for that packet. Suppose T = 1 millisecond; would it make sense for the sample values given above? (Of course, the overhead of setting up the virtual circuit can usually be amortized across multiple packets between the same hosts.)

Problem 6: TCP over wireless (10 points)

Recently, a number of wireless networks for carrying digital Internet traffic have been deployed; one example is the Metricom packet radio system which can be used for wireless communication on the UW campus. Wireless communication links are typically much more likely to suffer bit errors than traditional "wired" networks, with drop rates approaching 1 in 4 packets. These networks are also usually just for the final "hop"; Internet packets traverse some number of wired links and routers before being sent over the wireless link, which delivers packets directly to the receiving host (for example, a portable PC with a cellular phone and modem).

(a) When these wireless networks were first deployed, observed TCP bandwidth over the wireless link was usually an order of magnitude worse than the peak bandwidth the hardware could deliver. Explain why.

(b) One proposed solution to this problem is to cache packets at the upstream router just prior to being sent over a wireless link. If the packet is garbled, the packet can be resent without having to wait for a retransmission from the source host. Of course, this solution is easier to deploy if it doesn't require changing the protocol stack at either the sender or the receiver. Consider the case where the packets are being sent to a wireless host and all the traffic is TCP. Can an intelligent router detect when retransmissions are needed without modifying the receiver? If so, how? If not, why not? Hint: the router can observe TCP acknowledgments.