EECS489 Computer Networks, Final Exam (Fall 2005)

Instructions: You are allowed to use one sheet of notes (front and back). This exam is closed book, no computers are allowed. You can use a calculator. Read the entire exam through before you begin working. Work on those problems you find easiest first. 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: _____

Uniqname: _____

Problem 1 [20 pts]: Security/Encryption

Suppose two computers wish to communicate securely with each other over the Internet. Both computers are personal computers with IP addresses 64.123.45.6 and 64.123.45.7 respectively, and are protected by a firewall. These computers would like to communicate using UDP messages over port 8000. Both computers, call them A(64.123.45.6) and B(64.123.45.7) have public/private key pairs Pub_A, $Priv_A$, Pub_B, $Priv_B$ respectively. Assume that an attacker could potential modify, delete, view, and/or replay any messages sent between computer A and computer B over the Internet.

[6 pts] (a) **Firewall Rules**
Assuming that A and B are also running Web servers on TCP port 80 that needs to be accessed from the Internet, but are not running any other services. Also assume that A and B does not know ahead of time which IP addresses the others will be sending from. What rules would you want to add to the firewalls protecting computer A and computer B to make the firewalls *as restrictive as possible* while still allowing connections to the Web server and for the secure communication program? (Format the rules in the form of ALLOW/DENY UDP/TCP [source ip range] [destination port range]). Note, the ordering of the rules matter: earlier rules take higher precedence than later rules.

ALLOW UDP [source ip: 0.0.0.0-255.255.255.255] [destination port 8000]
DENY UDP [source ip: 0.0.0.0-255.255.255.255] [0:7999]
DENY UDP [source ip: 0.0.0.0-255.255.255.255] [8001:64k]
ALLOW TCP [source ip: 0.0.0.0-255.255.255.255] [destination port 80]
DENY TCP [source ip: 0.0.0.0-255.255.255.255] [0:79]
DENY TCP [source ip: 0.0.0.0-255.255.255.255] [81:64k]

[6pts] (b) **Authentication**
If A wants to initiate a secure communication session with B by sending B a newly created symmetric AES key, what else should A do to the new key, call it $K_{session}$, in order to make sure that a man-in-the-middle cannot replay the message, cannot pretend to be A, cannot read the key, and cannot modify the key without B detecting the modification?
(Hint: A and B have each other's public key. Denote a hash function with H(), a nonce with N, and encrypting with E(key,message))

E(Pub_B, $<K_{session}$, N, E($Priv_A$, H($K_{session}$, N))$>$)
(Note this is one of many potential correct answers)

[2pts] (c) **Nonrepudiation**
After computer A has sent a key to computer B, computer B receives a message that is encrypted with the key Ksession. These messages are in fact orders for expensive EECS 489 textbooks. Computer B wants to be able to prove in court if necessary that computer A actually sent the message to B containing the order for textbooks. What does A need to do/add to the message, call it M, in order for B to be able to prove that A is the one who sent it originally.

Sign a hash of the message: E($Priv_A$, H(M)) and add it to the end of the message M.

[4pts] (d) **Attacks**
Assume that a man-in-the-middle cannot decrypt any of the message as it is being sent. Also assume that the message consists entirely of alphanumeric characters, and B will know that the message is invalid if it sees an illegal character. What attacks would a man-in-the-middle be able to conduct on the communication channel to compromise the integrity of the message? (Hint: The whole message is sent from A to B as E(Ksession, M). Since AES is a block cipher, the message is broken up into small blocks that are encrypted independently of one another, i.e. E($K_{session}$, m1), E($K_{session}$, m2), etc.) where the little m's are parts of the whole message M.)

A man-in-the-middle probably would not be able to modify the packets without causing a non-alphanumeric character to appear most of the

time, but the probability of modifying a block and having it still be legitimate is high enough for this to be a problem. A man-in-the middle could also replay blocks of the message without B being able to tell that the blocks were not sent twice by A. The man-in-the-middle can also purposely cause blocks to be dropped from the message.

[2pts] (d) **Fixing the protocol**
Assuming a man in the middle cannot obtain the session key, would it be sufficient for A to add a checksum to the end of the message before encrypting it in order to defeat the attacks listed in part (e), or would it be necessary for A to add a cryptographic hash value? Why?

A checksum would be sufficient because the attacker cannot flip bits in the encrypted string to have a known effect on the decrypted stream. Replaying packets would also mess up the checksum causing B to realize the message has been tampered with.

Problem 2: Network Management [10 pts]

[6 pts] (a) Consider the two ways in which communication occurs between a managing entity and a managed device: *request-response mode* and *trapping*. What are the pros and cons of these two approaches, in terms of (1) overhead, (2) notification time when exceptional events occur, and (3) robustness with respect to lost messages between the managing entity and the device? (Please describe pros and cons in all these three areas: 2 pts each).

(1) Overhead:
Request-response has more overhead because each piece of information received by the manager requires two messages: the poll and the response. Trapping generates only a single message to the sender.

(2) Notification time:
Trapping will immediately notify the manager when an event occurs. With polling, the manager needs to wait for half a polling cycle on average between when the event occurs and the manager discovers that the event has occurred.

(3) Robustness:
Trapping is less robust, as the managed device will not retransmit it when it is lost. Polling can always be done again as needed.

[4 pts] SNMP protocol commonly uses UDP as the transport rather than TCP. Give one reason why UDP is preferred. Describe a problem associated with UDP in the context of network management.

Answer: UDP is used because TCP requires reliability and this imposes additional overhead on the network devices. Precisely when the network devices are overloaded, we do not want to impose additional overhead on those devices. The problem with UDP is that when network congestion occurs, UDP packets are likely to be lost and it is difficult to diagnose network problems when this occurs.

Problem 3: TCP [20 pts]

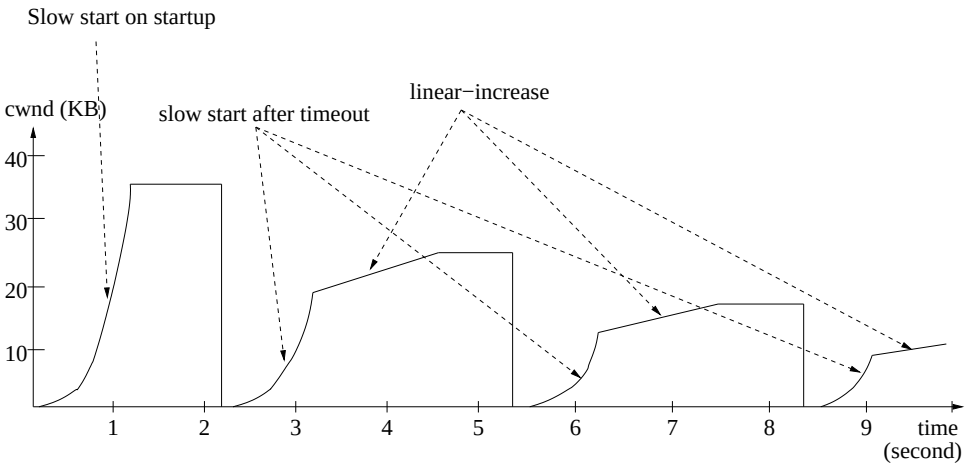

Figure 1: TCP congestion window variation

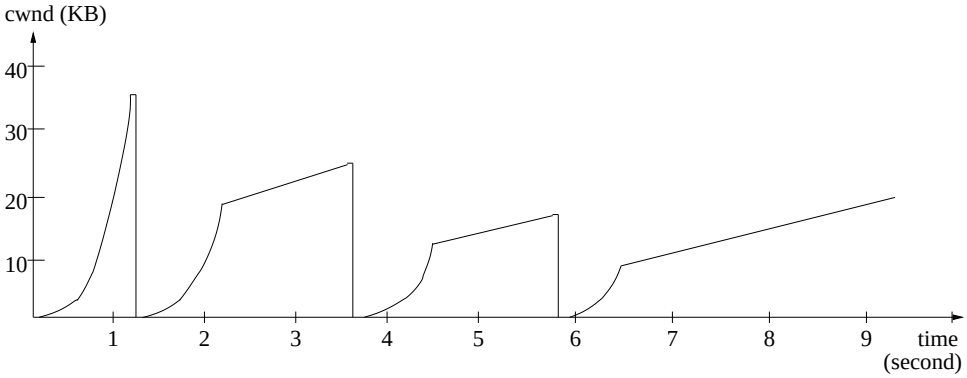

Figure 2: TCP congestion window variation

Consider the TCP congestion window variation in Figure 1 and Figure 2.

[5 pts] (a) In Figure 1, mark the time intervals representing slow start on startup, slow start after timeout, and linear-increase congestion avoidance. One for each phase is enough.

Answer: Slow start on startup is 0 to 1.3 second. Slow start after timeout is 2.3 to 3.2 second. Linear-increase is 3.2 to 4.6 second.

[5 pts] (b) What is the difference between slow start and linear-increase in terms of congestion window variation? How does TCP switch from slow start to linear-increase?

Answer: Congestion window exponentially increases in slow start. However, it linearly increases in linear-increase. When congestion window is less than slow start threshold, TCP is in slow start. Otherwise, it is in linear-increase.

[5 pts] (c) The TCP version that generated the trace in Figure 2 includes a feature absent from the TCP that generated Figure 1. What is this feature? How does it work? What is the main advantage of this feature?

Answer: It is fast retransmit. TCP enters fast retransmit when the sender receives 3 duplicate ACKs. It can avoid timeouts.

[5 pts] (d) Which feature of TCP can help eliminate some of the slow start phases in Figure 1 and 2? Briefly describe how congestion window is set in this feature.

Answer: Fast recovery. Instead of resetting the congestion window to 1, it sets the congestion window to half of the previous congestion window size.

Problem 4: Multimedia Networking [20 pts]

Consider an Internet phone application which generates an audio signal consisting of talk spurts (at a rate of 8000 bytes per second) and silent periods. UDP packets are generated only during talk spurts at the rate of every 20 msec.

[2 pts] Ignoring the header, what is the size of the UDP packet generated every 20 msec? Why isn't TCP used instead of UDP?

Answer: $20msec * 8000bytes/sec = 160bytes$. TCP is not used because we care about real-time delivery, reliability is not as important. Also, TCP doesn't let you control the sending rate very well.

If each UDP packet arrives at the receiver and has a small constant end-to-end delay, then packets arrive at the receiver periodically every 20 msec during a talk spurt. However, some packets can be lost and most of the packets will not have the same end-to-end delay. There are two types of FEC (Forward Error Correction) that help recover packet losses. (1) Redundancy encoding: a redundant encoded packet is sent after every n packets by XOR-ing the n original packets. (2) Low-bit redundant encoding: a lower-resolution stream is sent.

Suppose the first scheme generates a redundant packet for every four original packets. Suppose the second scheme uses a low-bit rate encoding whose transmission rate is 25 percent of the transmission rate of the nominal stream.

[6 pts] (a) How much additional bandwidth does each scheme require? How much playback delay (in terms of number of packets) does each scheme add?

Answer: Both schemes require 25% more bandwidth. The first scheme has a playback delay of 5 packets, the second has a delay of 2 packets.

[4 pts] (b) How do the two schemes perform if the first packet is lost in every group of five packets? Which scheme will have better audio quality?

Answer: The first scheme will be able to reconstruct the original high-quality audio encoding. The second scheme will use the low quality

audio encoding for the lost packets and will have lower quality.

[4 pts] (c) How do the two schemes perform if the first packet is lost in some groups of two packets? Which scheme will have better audio quality?

Answer: The first scheme will have many of the original packets lost and suffer from degraded audio quality. For the second scheme, every audio packet will be available at the receiver, although some of which will be lower quality, but audio quality will be better than the first scheme.

[4 pts] (d) When packets are received, they are usually not immediately played. The playout delay imposed aims to have most of the packets received before their scheduled playout times. Discuss the benefit and disadvantage of having a long playout delay vs. a short playout delay. Ideally the playout delay should be adaptive, what dynamic network property should it be based on? In TCP, there is a parameter that also depends on this dynamic network property. What is it?

Answer: Long playout delay guarantees that most packets will arive and can tolerate large jitter; however the performance will suffer, as the delay will be very long. Adaptive playout shoul be based on network delay variation or jitter. In TCP, congestion timeout also depends on network delay.

Problem 5: [20pts] Wireless Networks

The de facto standard for wireless LAN media access is the IEEE 802.11 standard.

[4 pts] (a) Why is CSMA/CD not practical for wireless networks. List two reasons.

Answer: hidden terminal problem and near far problem: nodes cannot always detect collisions. it is difficult to send and receive at the same time. This means that it is difficult to avoid collisions.

[6 pts] (b) The IEEE 802.3 standard (Ethernet) uses CSMA/CD for media access control, whereas the 802.11 uses CSMA/CA. It is possible to use CSMA/CA on wired medium also. If you use your LAN mostly for transferring streaming media such as video, audio, and real-time data such as multi-player game data, which media access control protocol would you use? Why?

Answer: I would use CSMA/CD. Because for streaming media, the most critical thing is to minimize the transfer delay. CSMA/CA will incur the delay of RTS/CTS frames in additional to the delay of transferring data frames.

[4 pts] (c) Continuing the above question: if you use your LAN mostly for transferring bulk transfer data, such as file transfer and email, which media access control protocol would you use? Why?

Answer: I would use CSMA/CA. For transferring bulk transfer data, the channel will be highly utilized. Therefore, collisions are likely to occur. Instead of wasting time on sending data frames simultaneously, it is better to avoid collision using CTS/RTS frames in advance.

[3 pts] (d) Why are link-layer acknowledgments used in 802.11 but not in wired Ethernet?

Answer: Wireless channel bit error rates are high, also a station cannot detect collision due to hidden terminal problem.

[3 pts] (e) We have discussed 802.11 mobility where a wireless station can move from one BSS to another within the same subnet. When the APs are interconnected with a switch, an AP may need to send a frame with spoofed MAC address to get the switch to forward frames properly, why?

Answer: Force the switch to update its forwarding table's outdated MAC address to IP address mapping.

Problem 6: [10 pts] P2P and NAT

In this problem we study the impact of NAT on peer-to-peer applications. Suppose a peer with user name Alice discovers through querying a DHT that a peer with user name Bob has a file it wants to download. Also suppose that Bob is behind a NAT whereas Alice is not. Let 138.76.29.7 be the WAN-side address of the NAT and let 10.0.0.4 be the internal IP address for Bob. Assume that the NAT is not specifically configured for this P2P application.

[4 pts] (a) Can Alice's peer initiate a TCP connection to Bob's peer assuming Alice knows the WAN-side address of the NAT as well as Bob's internal IP address? If so, how? If not, why not?

Answer: No, it can't. Alice can send TCP SYN packet with destination address to be the WAN address, port number p, but the NAT doesn't know to which internal host it should direct the packet. Also, there is usually no process listening at port p.

[3 pts] (b) Now suppose that Bob has established an ongoing TCP connection to another peer, Cindy, who is not behind a NAT. Also suppose that Alice learned from Cindy that Bob has the desired file and that Alice can establish a TCP connection with Cindy. Describe how Alice can use these two TCP connections to instruct Bob to initiate a direct TCP connection back to Alice. This is called *connection reversal*.

Answer: There is an existing TCP connection between Alice and Cindy and between Cindy and Bob. Via these two TCP connections, Alice can send a message to Bob. Alice can ask Bob to initiate a direct TCP connection from Bob to Alice. Since Bob is initiating this TCP connection, it can be established through Bob's NAT. Once this direct TCP connection is established between Alice and Bob, Alice can ask Bob to send the file over this direct connection.

[3 pts] (c) Now suppose that both Bob and Alice are behind NATs. Is it possible to design a scheme where Alice establishes a TCP connection with Bob without application-specific NAT configuration? If so, describe how. If not, describe why not.

Answer: It is not possible to devise such a technique. To establish a direct TCP connection between Alice and Bob, either of them must initiate a connection to the other. But the NATs covering Alice and Bob drop TCP SYN packets arriving from the WAN side. Thus neither Alice nor Bob can initiate a TCP connection to the other if both are behind NATs.