EECS489 Computer Networks, Final Exam Solution (Winter 2007)

Instructions: You are allowed to use two sheets 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

In network communications, there are several desirable security properties. For example, *confidentiality* is the property that the original plaintext message cannot be determined by an attacker who intercepts the cyphertext-encryption of the original plaintext message.
Another important property is *message integrity*. This means that the receiver can detect whether the message sent (regardless if it was encrypted) was altered in transit.

[3 pts] (a) For the two properties of confidentiality and message integrity, can you have one without the other? Justify your answer. (If your answer is yes, show an example of where one can exist without the other. If you answer is no, explain why one would imply the other.)

Answer: Yes, you can have one without the other. An encrypted message that is altered in transit may still be confidential (the attacker cannot determine the original text) but will not have message integrity if the error is undetected. Similarly, a message that is altered in transit (and detected) could have been sent in plaintext and thus would not be confidential.

[3pts] (b) *Digital signatures* is an electronic signature used to authenticate the identity of the sender of a message. Assume that you want to send a message M to your lawyer and give him/her the assurance that it was unchanged from what you sent (message integrity) and it is really from you (message authenticity). Describe how you can achieve this using public key encryption.

Answer: You obtain a message hash using some one-way hash function (e.g., Sha-1): H(M). You use your private key to encrypt the hash. You send M and encrypted hash to your lawyer, i.e., $M||Kpr_{you}(H(M))$, who subsequently computes the hash of the message. And it then uses your public key (obtained from a certificate authority) to decrypt H(M) part of the message. If the computed hash of the message matches the decrypted hash, i.e., $H(M) = Kpub_{you(}Kpr_{you}(H(M)))$, then the message integrity and authenticity are assured.

[3pts] (c) You have achieved message integrity and authenticity in (b). Describe how you can achieve confidentiality by slightly modifying your solution in (b).

Answer: Before M was sent in clear text, instead of sending M directly, encrypt M using your lawyer's public key: $Kpub_{lawyer}(M)||Kpr_{you}(H(M))$, that way no-one except your lawyer can see the message.

[3pts] (d) To prevent replay attacks (you do not want an attacker to re-send the same message to your lawyer), describe a simple modification to your solution in (c).

Answer: A nonce can be used. For example, a time-stamp is used as a nonce. You send to your lawyer $Kpub_{lawyer}(M, time{-}stamp)||Kpr_{you}(H(M))$. Basically the receiver never accepts the message with the same nonce to defeat replay attacks.

[4pts] (e) Suppose we have a very short secret s (e.g., a single bit or even a Social Security number), and we wish to send someone else a message m now that will not reveal s but that can be used later to verify that we did know s. Explain why $m = MD5(s)$ or even $m = E(s)$ with RSA encryption would not be secure choices and suggest a better choice.

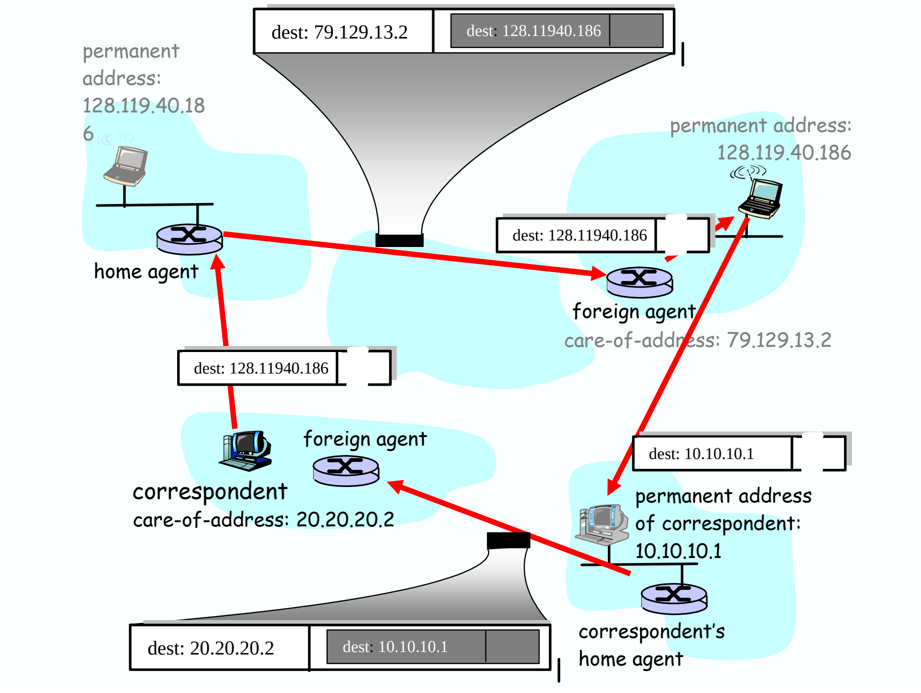
Answer: Because s is short, an exhaustive search conducted by generating all possible s and comparing the MD5 checksums with m would be straightforward. Sending MD5(s^r), for some random or time-dependent r, would suffice to defeat this search strategy, but note that now we would have to remember r and be able to present it later to show we knew s.

[4pts] (f) suppose two people want to play poker over the network. To "deal" the cards they need a mechanism for fairly choosing a random number x between them; each party stands to lose if the other party can unfairly influence the choice of x. Describe such a mechanism. (Hint: You may assume that if either of two bit strings x_1 and x_2 are random, then the exclusive-OR $x = x_1 \oplus x_2$ is also random. Another hint: try to use your solution from (e))

Answer: Each side chooses x_i privately. They exchange signatures of their respective choices as in the part (f). Then they exchange the actual x_i's (and r_i's); because of the signatures, whoever reveals their x_i last cannot change their choice based on knowing the other x_i. Then let $x = x_1 \oplus x_2$; as long as either party chooses their x_i randomly then x is random.

Problem 2: Network Mobility Support [10 pts]

[8pts] (a) Mobile IP support allows mobile hosts to be reached via their original IP addresses associated with their home networks. This is accomplished mainly with the help of the home agent which tunnels the traffic to the foreign agent before traffic finally reaches the mobile host. **Now consider the correspondent also to be mobile.**
Sketch below the additional network-layer infrastructure that would be needed to route the datagram from the original mobile use to the (now mobile) correspondent. Show the structure of the datagrams (in terms of src and dst IP address and any packet encapsulation) between the original mobile user and the (now mobile) correspondent using the figure below. (You need to show at least the packet from the mobile user to the correspondent and the reply packet from the mobile to the correspondent.)


dest: 79.129.13.2
dest: 128.11940.186
permanent address: 128.119.40.18 6
permanent address: 128.119.40.186
dest: 128.11940.186
home agent
foreign agent
care-of-address: 79.129.13.2
dest: 128.11940.186
foreign agent
correspondent
care-of-address: 20.20.20.2
dest: 10.10.10.1
permanent address of correspondent: 10.10.10.1
correspondent's home agent
dest: 20.20.20.2
dest: 10.10.10.1

[2pts] (b) How would mobility support for both hosts (the correspondent and the mobile) affect end-to-end delays of datagrams between the source and the destination?

Answer: This will impose additional delay due to the indirect routing from the correspondent's home agent to the mobile correspondent.

Problem 3: TCP [20 pts]

[3 pts] (a) TCP waits until it has received three duplicate ACKs before performing a fast retransmit. Why do you think the TCP designers chose not to perform a fast retransmit after the first duplicate ACK for a segment for a segment is received?

Answer: Suppose packets n, n+1, and n+2 are sent, and that packet n is received and ACKed. If packets n+1 and n+2 are reordered along the end-to-end-path (i.e., are received in the order n+2, n+1) then the receipt of packet n+2 will generate a duplicate ack for n and would trigger a retransmission under a policy of waiting only for second duplicate ACK for retransmission. By waiting for a triple duplicate ACK, it must be the case that two packets after packet are correctly received, while n+1 was not received. The designers of the triple duplicate ACK scheme probably felt that waiting for two packets (rather than 1) was the right tradeoff between triggering a quick retransmission when needed, but not retransmitting prematurely in the face of packet reordering.

[3 pts] (b) Host A is sending an enormous file to Host B over a TCP connection. Over this connection there is never any packet loss and the timers never expire. Denote the transmission rate of the link connecting Host A and the Internet by Rbps. Suppose that the process in Host A is capable of sending data into its TCP socket at a rate Sbps, where $S = 10 \times R$. Further suppose that the TCP receive buffer is large enough to hold the entire file, and the send buffer can hold only 1% of the file. What prevents the process in Host A from continuously passing data to its TCP socket at rate Sbps?

Answer: In this problem, there is no danger in overflowing the receiver since the receiver's receive buffer can hold the entire file. Also, because there is no loss and acknowledgments are returned before timers expire, TCP congestion control does not throttle the sender. However, the process in host A will not continuously pass data to the socket because the send buffer will quickly fill up. Once the send buffer becomes full, the process will pass data at an average rate of $R << S$.

[4pts] (c) Consider a datagram network using 8-bit host addresses. Suppose a router uses longest prefix matching with the following forwarding

table:

Prefi x match	Interface
00000000	0
01000000	1
10000000	2
11000000	3

For each of the four interfaces, give the associated range of destination host addresses and the number of addresses in the range.

Answer:
Number of addresses in each range should be $2^6 = 64$, as the last 6 bits can be either 0 or 1 for each prefix.

[3pts] (d) Suppose two TCP connections share a path through a router R. The router's queue size is 6 segments; each connection has a stable congestion window of 3 segments. No congestion control is used by these connections. A third TCP connection now is attempted, also through R. The third connection does not use congestion control either. Describe a scenario in which, for at least a while, the third connection gets none of the available bandwidth, and the first two connections proceed with 50% each. Does it matter if the third connection uses slow start? How does congestion avoidance by the first two connections help solve this?

Answer: If the packets from the third connection happenalways to arrive when the queue is full, it will never gain any available bandwidth. This is independent of whether slow start is used. Congestion avoidance will cause the two TCP connections eventually try a window size of 4, and fall back to 2, and give the third connection some chance to occupy router's queue. Slow start by the third connection will allow the sender to increase window more quickly.

[7pts] (e) You are an Internet Service Provider; your client hosts connect directly to your routers. You know some hosts are using experimental TCPs and suspect some may be using "greedy" TCP with no congestion control. What measurements might you make at your router to establish that a client was not using slow start at all? If a client used slow start on startup but not after a timeout, could you detect that?

Answer:

The router can determine the actual number of bytes outstanding by examining sequence and ACK numbers. The host is complying with slow start at startup if only one more packet is outstanding than the number of ACKs received. After any packet is retransmitted due to timeout, we should see the congestion window fall at least in half. We can rule out three duplicate ACKs to identify fast retransmit.

Problem 4: Multimedia Networking [10 pts]

[4 pts] (1) What is the difference between end-to-end delay and packet jitter? What are the causes of packet jitter?
Answer:
End-to-end delay is the time it takes a packet to travel across the network from source to destination. Packet jitter is the difference in delay from one packet to next packet, it measures the first order derivative of delay, i.e., the delay variations. Causes of packet jitter can be due to variations in queuing delay, cross traffic, bursty behavior of the end-user or end-host, etc.

[2 pts] (2) Why is a packet that is received after its scheduled playout time considered lost?
Answer:
A packet that arrives after it scheduled playout time can not be played out due to the real-time constraint of the applications. Therefore, from application's perspective, the packet has been lost.

[2 pts] (3) Suppose we send into the Internet two IP datagrams, each carrying a different UDP segment. The first datagram has source IP address A1, destination IP address B, source port P1, and destination port T. The second datagram has source IP address A2, destination IP address B, source port P2, and destination port T. Suppose that A1 is different from A2 and P1 is different from P2. Assuming that both datagrams reach their final destination, will the two UDP datagrams be received by the same socket? Why or why not?
Answer:
Yes they will pass through the same socket, because UDP sockets are identified by two tuples: destination IP and port.

[2 pts] (4) Given the answer to (3), how would a host communicate with two separate clients at the same UDP port? (Hint: in CS, "a level of indirection" tends to be very powerful).
Answer:
Use another protocol on top of it to establish the connection and distinguish the packets, e.g., RTP+SIP.

Problem 5: [20 pts] Wireless Networks

[2 pts] (1) Ethernet protocol uses CSMA/CD for multiple access control. When a collision is detected, how does Ethernet back off to avoid future collisions? What is the advantage of this scheme compared to random backoff?

Answer:
Exponential backoff, chooses a value for K at random from $\{0, 1, 2, ...2^{m-1}\}$, where $m = min(n, 10)$. n is the nth collision in a row for a given frame. exponential backoff helps infer how many adapters are involved in the collision and chooses K from a larger and more dispersed set of values if there are multiple adapters involved.

[2 pts] (2) Why doesn't collision detection work well in wireless networks? (describe a scenario where this doesn't work.)

Answer:
Because nodes are not fully connected, they cannot all hear each other. One scenario is hidden terminal problem: frames from two senders collide because the receivers are near each other but the senders are far apart and cannot hear each other. Another scenario is exposed terminal problem: two senders hear each other and tink ther is a collision when in fact there is none because each receiver can hear only one sender.)

[9 pts] (3) Suppose nodes A and B are on the same 10Mbps Ethernet segment and the propagation delay between the two nodes is 225 bit times. Suppose A and B send frames at the same time, the frames collide, and then A and B choose different values of K in the CSMA/CD algorithm. Assuming no other nodes are active, can the retransmissions from A and B collide?

Assuming A chooses K=0, B chooses K=1, basically we need to find out whether A's retransmission will reach B before B's scheduled retransmission time.

- t=0: A and B begin transmission
- t=225: A and B detect collision
- t=225+48=273: A and B finish sending Jam signal
- 273+225=498: B's last bit arrives at A; A detects an idle channel

- 498+96=594: A starts transmitting
- 273+512=785: B returns to step2, B must sense idle channel for 96 bit times: 785+96=881
- 594+225=819: A's transmission reaches B
- 819$<$881: A's retransmission reaches B before B can send.

[2 pts] (4) Why are acknowledgments used in 802.11 but not in wired Ethernet?

Answer:
Wireless channels have higher bit error rates. In wired ethernet, there is no hidden node problem, i.e., all collisions can be detected in wired ethernet.

[2 pts] (5) In the CSMA/CA protocol, a station that successfully transmits a frame begins the protocol for a second frame at step 2 where its adapter has to sense the channel idle for 96 bit times before it can start transmitting again. What is the rationale behind this? Why not allow the station transmit the second frame immediately, if the channel is sensed idle?

Answer:
Fairness considerations.

[3 pts] (6) The idea of "snooping TCP" (shown below) is to buffer all unacknowledged packets to the destination mobile host and perform local retransmission in case of packet loss on the wireless link. Why is this beneficial? Would this work for IPsec traffic, what about SSL traffic?

Answer:
Avoid end-to-end TCP retransmission time-out, faster retransmissions. It works for SSL (encryption above transport layer), but not for IPsec where TCP header is encrypted.

Problem 6: [20 pts] P2P and NAT

[4 pts] (1) Give two reasons why native IP multicast is not widely deployed today.

Answer:

1. Need to change routers, there is no incentives from router vendors or ISPs to deploy. 2. Difficult to charge. 3. Too easy to be abused for launching dos attacks. 4. difficult to build congestion control, error recovery due to reasons such as NACK implosion.

[6 pts] (2) Give an advantage and a disadvantage of application-layer multicast (ALM). Describe two metrics for evaluating the overhead of application-layer multicast.

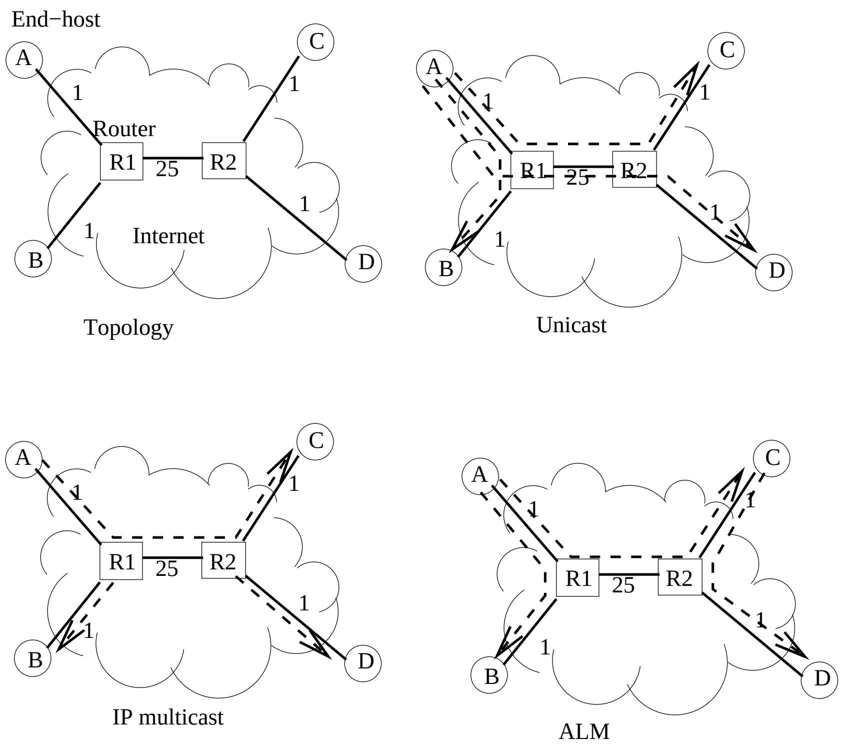
Answer:

Advantage: ease of deployment, application-specific customization, no need to change IP or routers, no need for ISP cooperation, easy to implement reliability, can get around performance bottlenecks on the Internet .

Disadvantage: Generally less efficient compared to native IP multicast.

The two metrics are Stretch (ratio of latency in the overlay to latency in the underlying network) and Stress (number of duplicate packets sent over the same physical link).

[10 pts] (3) Given the following topology, with delay marked at each link. Assuming node A wishes to send data to all other nodes. *Mark* on each of the other three topologies how native unicast, native IP multicast, and ALM would deliver the content from A to B,C,D. For ALM, there are multiple options, pick the one that has the lowest overhead (based on the metrics in (2).



Topology

Unicast

IP multicast

ALM

Extra Credit [10 pts]

[5 pts] (a) *Available bandwidth probing*: Explain whether available bandwidth estimation can be accurately validated using bulk TCP throughput. If so, explain why such estimation would be accurate. Otherwise, explain why such estimation would be inadequate.

Answer:

No, available bandwidth estimation cannot be accurately validated using bulk TCP throughput. The throughput of a bulk TCP transfer depends on a number of factors such as socket buffer sizes at the sender and the receiver, available bandwidth, amount of buffering in the bottleneck link, type of competing cross traffic, round-trip time, loss rate and others. Although available bandwidth is one of the factors, it is not the dominating factor.

[5 pts] (b) *Packet-pair based probing*: Given an H-hop path defined by the sequence of capacities $P = \{C_0, C_1, ..., C_H\}$. Two packets of size L are sent back-to-back from the source to the sink. These packets are

the *packet pair*. The *dispersion* of the packet pair is the interval from the instant the last bit of the first packet is received at a certain path point to the instant of the last bit of the second packet is received at that point. The dispersion is $\Delta_0 = \tau_0 = L/C_0$ after the source, and let it be $Delta_i$ after link i. When the packet pair reaches the sink, the dispersion is $Delta_H$ and the receiver computes a bandwidth estimate $b = L/Delta_H$.

Assuming no cross traffic is present, what does b measure, i.e., what kind of bandwidth does b represent?

Answer:

It measures the smallest link capacity value of the end to end path.

Under what condition can we claim that dispersion Δ_i cannot be lower than the dispersion at the previous hop Δ_{i-1}. What kind of delay would cause the dispersion to decrease at a later hop?

Answer:

When there is no cross traffic. The delay is queuing delay.