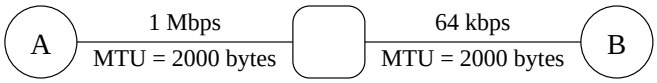
EECS 122: Introduction to Communication Networks

Final Exam Solutions

Problem 1. (6 points) How long does it take for a 3000-byte IP packet to go from host A to host B in the figure below. Assume the overhead of any packet headers is negligible, as is the length of the cables, and assume there is no other traffic. (If you need to make an additional assumption, state it.)



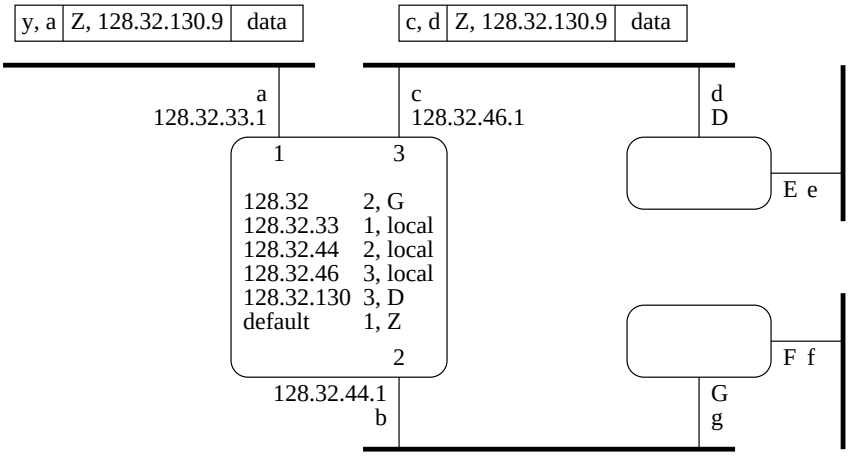
Because the packet is larger than the MTU, host A must fragment it. Any fragmentation is acceptable as long as all fragments are less than the MTU, and their total size is 3000 bytes. Typically, all fragments will equal the MTU except the last, so this solution assumes that host A sends a 2000-byte fragment followed by a 1000-byte fragment.

After the first fragment fully arrives at the router, the router will start sending it to host B. Meanwhile, host A will send the second fragment, which will fully arrive at the router before the router is finished sending the first fragment (because the second link is slower than the first). Therefore the total transfer time is the time for the first fragment to get from A to the router, plus the time for both fragments to get from the router to B:

$$\frac{2000 \text{ bytes}}{10^6 \text{ bits/s}} \frac{8 \text{ bits}}{\text{byte}} + \frac{3000 \text{ bytes}}{64{,}000 \text{ bits/s}} \frac{8 \text{ bits}}{\text{byte}} = 16 \text{ ms} + 375 \text{ ms} = 391 \text{ ms}$$

(Notice that fragments are reassembled at the destination, not at the router. This is stated in the textbook and the lecture notes. Actually, IPv4 permits routers to reassemble packets, but that is generally discouraged and not done. IPv6 specifies that reassembly takes place only at the destination, not the router.)

Problem 2. (6 points) In the figure below, a packet is arriving at an IP router connected to three Ethernets. IP addresses are shown in dotted decimal notation or denoted by upper case letters, while Ethernet addresses are denoted by lower case letters, and internal interface identifiers are integers. Draw the packet near the next Ethernet it will traverse. Include the addresses.



The router finds the longest match for the destination IP address (128.32.130.9), which is the entry mapping 128.32.130 to output port 3. The entry specifies that the next hop router has IP address D. Routers do not alter the IP addresses of the packets they forward, but ARP is used to map D to the corresponding Ethernet address d, which is used as the destination of the Ethernet frame. The source address of the Ethernet frame is c, the Ethernet address of the output interface.

Problem 3. (6 points) Pretend for a moment that IP addresses are 12 bits instead of 32, and suppose you are allocated the block of addresses 1011xxxxxxxx for assigning addresses to hosts in four networks. Network A has 100 hosts, network B has 50, network C has 25, and network D has 20. Choose an address prefix for each network.

Network A needs at least 7 bits for the host part (using up 128 addresses), network B needs at least 6 bits (64 addresses), and networks C and D need at least 5 bits each (32 addresses). That adds up to 256 addresses, which is the entire block. So the number of host bits for each network must be exactly the minimum stated, leaving the remaining bits for the prefix. Each prefix must begin with 1011, and none may be a prefix of another. Here is one possible assignment:

A: 10110

B: 101110

C: 1011110

D: 1011111

Problem 4. (12 points) At the console of a workstation attached to an Ethernet, a user has composed an email message. Between the time the user selects "send" and the time the voltage *first* changes on the Ethernet, describe what happens at each of the following layers. Assume nothing is already cached at any layer. The simplest layer, UDP, has been done for you.

- MTA: Looks up the mail exchanger (MX) for the domain of the destination email address, by sending a DNS query to the local name server (whose IP address is known).
- UDP: Prepends a UDP header (containing port numbers and a checksum) to the data, and passes the result to the IP layer, along with the destination IP address.
- IP: Checks whether the destination address is on the local network by checking whether it begins with the prefix of the local network (in other words, it checks whether the destination AND netmask equals its own address AND netmask). If the destination is local, the packet will be sent directly to the destination, otherwise it will be sent to the default router (whose IP address is known). Either way, an IP address must be mapped to an Ethernet address by sending an ARP request. (The Ethernet destination address of ARP requests is a well-known Ethernet group address, but you weren't expected to know that.)
- Ethernet: Prepends an Ethernet header, then listens for activity on the wire (carrier sense). When the line is clear, begins sending the preamble.

Problem 5. Consider 12 stations (hosts) attached to a 10 Mbps Ethernet. The throughput of the Ethernet is the total rate at which data is delivered to all the hosts. Assume all frames are addressed to individual stations, not to group or broadcast addresses. What is the maximum possible throughput if…

a) (2 points) …each host is connected to a single hub (repeater)?

10 Mbps. This is achievable if, for example, only one host is sending all the time. No greater throughput is possible because the hub forwards the data on all ports, so there can be only one sender at a time.

b) (2 points) …each host is connected via a half-duplex interface to a single Ethernet switch (bridge)?

60 Mbps. This is achievable if, for example, six hosts are sending all the time and the other six are receiving all the time. No greater throughput is possible because every frame needs a place to go, and a half duplex interface cannot send a frame and receive a frame at the same time.

c) (2 points) …each host is connected via a full-duplex interface to a single Ethernet switch (bridge)?

120 Mbps. This is achievable if all twelve hosts are both sending and receiving all the time. Obviously no greater throughput is possible.

Problem 6. (6 points) The routers A, B, C, D, and E are fully connected, and table below gives the symmetric cost of the link between each pair. After the routing protocol has stabilized, a packet is sent from A to B. What path does it take?

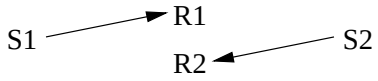
	A	B	C	D
E	2	7	3	6
D	8	2	2	
C	6	3		
B	9			

A shortest-path algorithm, like Dijkstra's or Bellman-Ford, can be used to find the shortest-path tree rooted at A (or B). The packet will take the shortest (least cost) path from A to B, which is A $\rightarrow$ E $\rightarrow$ C $\rightarrow$ B (a cost of 8).

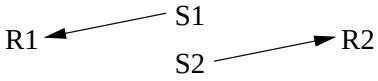
Problem 7. (6 points) Why doesn't collision detection work well in wireless networks? (Describe a scenario.)

It's because the nodes are not fully connected (they cannot all hear each other).

One scenario is the 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, and thus cannot detect the collision.



Another scenario is the exposed terminal problem: Two senders hear each other and think there is a collision, when in fact there is none because each receiver can hear only one sender.



Problem 8. An ATM switch has a single buffer that holds 2 million bits. There is currently one connection flowing through the switch, with traffic characteristics SCR = 4 Mbps, BT = 0.1 seconds, PCR = 7 Mbps, CDVT = 0, and the network has guaranteed that none of its cells will be dropped. Suppose a request comes in for a second connection through the switch and requiring the same guarantee, and it would use the same outgoing link, which is 10 Mbps. Can the switch admit the new connection if its traffic characteristics are:

a) (2 points) SCR = 8 Mbps, BT = 0.01 seconds, PCR = 9 Mbps, CDVT = 0 ?

No. The sustained cell rates of the two connections add up to more than the link rate, so the switch might have to drop cells.

$$8 \text{ Mbps} + 4 \text{ Mbps} = 12 \text{ Mbps} > 10 \text{ Mbps}$$

b) (2 points) SCR = 5 Mbps, BT = 0.3 seconds, PCR = 155 Mbps, CDVT = 0 ?

Yes. The sustained cell rates of the two connections add up to less than the link rate, and the bucket sizes of the two connections (SCR $\times$ BT) add up to less than the buffer size of the switch, so the buffer will never overflow.

$$5 \text{ Mbps} \times 0.3 \text{ s} + 4 \text{ Mbps} \times 0.1 \text{ s} = 1.5 \times 10^6 \text{ bits} + 0.4 \times 10^6 \text{ bits} = 1.9 \times 10^6 \text{ bits} < 2 \times 10^6 \text{ bits}$$

c) (2 points) SCR = 1 Mbps, BT = 3 seconds, PCR = 2 Mbps, CDVT = 0 ?

Yes. The peak cell rates add up to less than the link rate, so the buffer can never fill up.

$$2 \text{ Mbps} + 7 \text{ Mbps} = 9 \text{ Mbps} < 10 \text{ Mbps}$$

Problem 9. Consider an error correction code in which each 2-bit message m is appended with a 3-bit check sequence $f(m)$ to form a 5-bit codeword, as follows:

m:	00	01	10	11
$f(m)$:	000	101	110	011
codeword:	00000	01101	10110	11011

a) (3 points) If the decoder receives the following sequence of codewords from the channel, what sequence of 2-bit messages will it output?

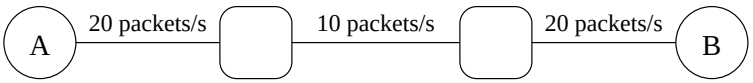
11111 10100 00010 11101 01011 00000

The decoder uses the nearest codeword to each received message, so it uses the codewords:
11011 10110 00000 01101 11011 00000
which correspond to the messages:
11 10 00 01 11 00

b) (3 points) Under what assumption is the output of the decoder equal to the input of the encoder?

The assumption that the channel inverts at most one bit per codeword. (The minimum distance between two codewords is 3, so we can correct less than 3/2 bit inversions. With more inversions than that, the nearest codeword is not necessarily the one that was sent.)

Problem 10. (6 points) Suppose hosts A and B are running a sliding window protocol with congestion control (not necessarily TCP). All packets (both data and acks) are the same size, so the link rates are given in packets per second. Assume there is no propagation delay, no queueing delay, and no other traffic. If the congestion control algorithm is effective (packet drops are very rare), then the average window size (in packets) cannot be more than what? (In other words, what would the average window size be for a hypothetical 100% efficient congestion control algorithm?)



(During the exam it was stated that the links are symmetric and the protocol is transferring data in only one direction.)

The average throughput cannot exceed 10 packets/s, otherwise there would be many packet drops. The throughput is one window per round-trip time. It takes a data packet 1/20 second to traverse the first link, 1/10 second for the next, and 1/20 second for the last, and the ack takes just as long to return, so RTT is 0.4 s.

$$\frac{\text{window}}{0.4\text{ s}} \leq 10\text{ packets/s} \Rightarrow \text{window} \leq 4\text{ packets}$$

Problem 11. (6 points) Consider a protocol intended to prove to Bob that he is communicating with Alice: Alice chooses a symmetric session key K, signs it using Alice's private key, encrypts the result using Bob's public key, and sends the result to Bob. Thereafter Alice and Bob converse, encrypting everything with K. There are two serious problems with this protocol. Describe either one.

One problem is that if Alice is in the habit of using this protocol to talk to Carol as well as Bob, then Carol can use the message she gets from Alice to impersonate Alice to Bob. Alice sends $\text{PubKey}_\text{C}(\text{PrivKey}_\text{A}(K))$ to Carol, who extracts $\text{PrivKey}_\text{A}(K)$ and encrypts it with Bob's public key, sending $\text{PubKey}_\text{B}(\text{PrivKey}_\text{A}(K))$ to Bob. This is exactly what Bob was expecting from Alice, but Carol knows K and can continue the conversation with Bob. (This is almost exactly like homework 10 problem 4.)

Another problem is that the session key K is not really a *session* key, because someone who somehow discovers it (perhaps by spying on Alice or Bob) can replay the message she sends to Bob at any time in the future and impersonate Alice. A session key is supposed to work only for one session. (If the message signed by Alice had included a *nonce* generated by Bob, then it wouldn't be useful in the future because Bob would choose a different nonce, and the signature cannot be generated without Alice's private key.)

Problem 12. Suppose a memoryless source emits symbols A–F with the following probabilities:
A: 32% B: 25% C: 20% D: 10% E: 8% F: 5%

a) (6 points) At most how many symbols per second can be successfully transmitted over a channel with bandwidth 10 kHz and signal-to-noise ratio 20 dB? You need not simplify your answer, but make it unambiguous and don't use dB in the expression.

First, 20 dB $= 10^{20/10} = 100$. The maximum bit rate of the channel is given by Shannon's capacity theorem:

$$C = B\log_2(1+\text{SNR})\text{bits} = 10\text{ kHz}\log_2(1+20\text{ dB})\text{bits} = 10\text{ kbps}\log_2(101) \doteq 66.6\text{ kbps}$$

The minimum bits per symbol is the entropy:

$$\begin{aligned} H &= -\sum_i p_i \log_2 p_i \\ &= -0.32\log_2 0.32 - 0.25\log_2 0.25 - 0.2\log_2 0.2 - 0.1\log_2 0.1 - 0.08\log_2 0.08 - 0.05\log_2 0.05 \\ &\doteq 2.33 \end{aligned}$$

The maximum possible symbol rate is therefore

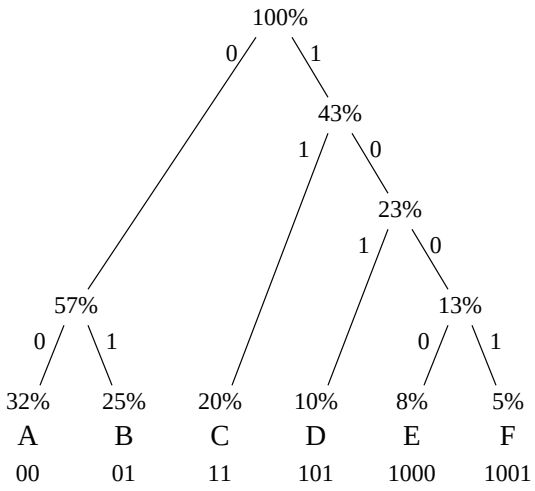
$$\frac{C}{H\text{ bits/symbol}} \doteq 2.33\text{ symbols/s}$$

Although plugging numbers into the Nyquist equation can lead to 66.6 kbps for the bit rate, the Nyquist theorem does not talk about the capacity of an analog channel for carrying digital information. Just the opposite, it talks about how many discrete-time samples you need to represent analog information.

Also note that the Huffman code is nearly optimal, but not optimal. The entropy is the lower bound on bits per symbol.

b) (6 points) Consider the Huffman code for this source that maximizes the occurrence of zeros (whenever 0 and 1 are being assigned to a pair of edges, 0 is assigned to the edge attached to the more probable node). Decode the following bit string into a string of symbols: 10010011001011000

The Huffman code is constructed by repeatedly combining the two least probable nodes:



The bit string can be uniquely decoded as FACADE.