CS 164 Fall 1994

Final Examination

S O L U T I O N S

3 Hours, Closed Book
Aids allowed: Calculator

I. Briefly explain the following terms:

(i) isarithmic flow control; *a technique for limiting congestion by restricting the total number of packets in the subnet. each packet needs to acquire a "permit" before it can enter the subnet, and the total number of "permits" is fixed.*

(ii) choke packet; *a control packet generated within the subnet at a congested node and sent back to a host that is responsible for (some of) the traffic at the node. when the host receives the choke packet, it reduces its transmission rate for a while.*

(iii) idle token; *a control symbol in token ring networks that is used to grant transmission rights to a host. A host wishing to transmit must wait for the arrival of the idle token, convert it to a busy token and send its packet, and then generate a new idle token.*

(iv) Hamming distance; *A method of evaluating the robustness of digital codes subject to bit errors. The Hamming distance for 2 codewords is the number of bit positions where they differ, and the Hamming distance for a code is the minimum across all pairs of codewords.*

(v) idempotency; *The property that an operation can be repeated without changing the result. If A==0 initially, then "A=A+1;" and "A=1;" have the same effect, but only the second form is idempotent.*

(vi) quarantining. *a technique for ensuring all-or-nothing execution of remote operations. The information about the desired transaction is gathered first (ie., kept in quarantine) and is then executed when all is available. Otherwise, you could get stuck part way through if you start the operation before all the data are present.*

II. Name three methods for controlling congestion in packet-switched networks. Describe each one using a few sentences. For each method, state whether or not it is equally good for virtual circuit and datagram networks.

Tanenbaum, section 5.3, lists the following five methods, any 3 of which would be ok: (1) preallocation of buffers; (2) packet discarding; (3) isarithmic; (4) flow control; (5) choke packets. I would accept any sensible argument about the applicability. Generally speaking, I think (1) is best left for VCs (since there are too many places a datagram could go to preallocate things); (2) is best for datagrams, since VCs normally assume in order delivery so it would mess things up; (3) would be best for datagrams because it is ineffective in VC systems because it won't stop local hot spots, but there are fewer obviously better alternatives with datagrams; (4) works OK with both, but VCs generally use small windows on a hop-by-hop basis, whereas datagrams use a big end-end window; and (5) choke packets are really only intended for datagram nets, although I suppose they could work in a VC environment if *you really wanted.*

III.

a. For a binary message `1001011101001111`, compute its CRC code using the polynomial $x^8 + x^7 + x^2 + 1$. Your should use long division method to solve it.

```
          1110011111011010
         -------------------------
110000101)1001011101001111100000000
          110000101        ^^^^^^^^  CRC  needs  8  bits  of
padding to be added,
          ---------                 and  not  just  the  plain
long division...
          0101010111
           110000101
           ---------
           0110100100
            110000101
            ---------
            000100001011
               110000101
               ---------
               0100011101
                110000101
                ---------
                0100110001
                 110000101
                 ---------
                 0101101000
                  110000101
                  ---------
                  0111011010
                   110000101
                   ---------
                   00101111100
                     110000101
                     ---------
                     0111110010
                      110000101
                      ---------
                      00111011100
                        110000101
                        ---------
                        0010110010
                         ^^^^^^^^^  these 8 digits are the
CRC.
```

b. For CRC encoding, The long division method is not suitable for programming. Describe a simpler algorithm using a 8 bit shift register to solve this problem. You don't need to re-do the computation, just describe the algorithm in a high level language.

The feedback shift register algorithm is shown in Stevens' text, on page 454.

IV. Alice is trying to send a long sequence of fixed-length frames to Bob over a data link that uses Go-Back-N. Assume that errors occur *deterministically*, once every ten frame transmissions from Alice to Bob, and that there are never any errors on the acknowledgements. Also assume that Alice receives the ACK or NACK for the frame she sent at time T just before she must decide which frame to send at time $T + N$.

a. What transmit window size should Alice use? Your answer should be a function of N.
Alice should use a window of size N, since smaller than that will get her stalled, waiting for an ACK to come back, and Alice will never let her xmit window of unacknowledged frames grow larger than N because she gets the feedback after the Nth frame, and there is no point in continuig to send after an error occurs because Bob uses go-back-N and won't accept it.

b. Trace what Alice does during the first 25 time steps, assuming that $N = 5$.
At time 1, Alice starts transmitting frames 1,2,3,.... At time 6, Alice receives ACK for 1 and transmits 6, etc. At time 15, Alice receives a NACK for 10 and starts over from 10,11,12.... At time 25, Alice receives a NACK for 15 and starts over from 15,16,17.... etc.

c. What is the channel efficiency as a function of N?
Assuming N is smaller than 10, we see that after the first error has been reached, there is a repeating pattern where the first N frames after each error get damaged or thrown away, and the next 10-n frames get accepted, so the efficiency is (10-n)/n

d. What happens when $N = 10$?
Deadlock, after the first 9 frames have been sent, since frame 10 will have an error, but you won't detect it (and retry) until time 20 -- which will be another error, and so on.

e. What happens when $N > 10$?
The first 9 frames are delivered successfully, then frame 10 is in error but we don't find out about it until frame 10+N for N>10, and all of those N frames are going to be lost. However, the retry for frame 10 will take place in slot 10+N, which will be successful if N is not divisible by 10. In general, Alice can sent a total of 9-(N mod 10) new frames before hitting the next error. Therefore the efficiency is [9-(N mod 10)] / [N+9-(N mod 10)]

V. You have been asked to set up a three-node network connecting some host computers located in Los Angeles, Riverside and San Diego. Assume that all packet lengths are exponentially distributed with a mean of 1200 bits, and that two configurations of equivalent cost are possible. First, three 4800 bps lines could be used, allowing direct connections between each pair of hosts. In this case, all traffic will be sent over the direct line. Second, two 9600 bps lines could be used, one connecting Los Angeles with Riverside, and the other connecting Riverside with San Diego. In this case, traffic between Los Angeles and San Diego must be sent via Riverside. The traffic matrix that this network must support is shown below.

	L	R	S
L	-	2	3
R	3	-	2
S	1	3.5	-

a. What is γ for this system?

$$\gamma = 14.5$$

b. Draw both networks and indicate the flow each channel in each case.
Sorry, it's too hard to try to draw these with the available tools... Config #1 is fully connected with link capacities of 4 packets/sec. everywhere and flows given in the above table. Config #2 only has R-L and R-S links, each with capacities of 8 packets/sec, and flows given in the table below.

	L	*R*	*S*
L	-	*5*	-
R	*4*	-	*5*
S	-	*4.5*	-

c. Find the average path length.
Obviously, $\bar{n} = 1$ *for Config #1. For Config #2 we sum up the flows in the table above and divide by* γ *to get* $18.5/14.5 = 1.27$.

d. Find the overall mean packet delay in each case.
Config #1: $T = 1.06$, Config #2: $T = 0.39$.

e. [Bonus question — the answer is short but it's tricky]
You should have found that three-line design had a higher delay than the two-line design. Since alternate routing is possible with three lines, and since there was no effort to optimize the routing, your boss asks whether your answer to part (a) is valid. Give a convincing argument to show him that no amount of rerouting of traffic would make the overall mean packet delay for the three-line design as small as the delay for the two-line design?
Optimal routing tries to balance the traffic in each link (think of the "square root rule"). That traffic would cover at least one hop. Therefore, in the best case, the flow in each direction over each link would be at least 14.5/6 packets/sec, which gives a delay of
$$\frac{1}{4 - 2.4} = 0.63$$

VI. The next generation of Metropolitan Area Networks (MANs) is expected to run at data rates in excess of 1 Gbit/sec (i.e., 10^9 bps) and span distances on the order of 100 kilometres.

a. For CSMA/CD to work properly, the transmission time for a packet needs to be at least twice the end-to-end propagation delay across the network. Assuming a propagation speed of 2×10^8 metres/sec. through optical fibre, what would be the minimum packet size for CSMA/CD on such a Gigabit MAN?
The worst case one-way propagation delay is 10^5 *m. /* $2 * 10^8$ *m/sec.* $= 5 * 10^{-4}$ *sec. Minimum packet transmission time is twice this, or* 10^{-3} *sec, which at* 10^9 *bits/sec represents* 10^6 *bits.*

b. What is the maximum throughput that a single station could achieve using 10,000 bit packets on a 1 Gbit/sec. token ring, assuming ordinary service (i.e., one packet per visit of the token) and a total ring circumference of 200 kilometres?
Max throughput by one station happens when nobody else gets in the way to delay its attempts. In each "cycle", the station sends 10^4 *bits, and the time to do it is* $10^4/10^9 = 10^{-5}$ *for transmission and* 10^{-3} *sec. for the idle token to go around a 200 km ring before the next turn. Therefore the throughput is*

$$\frac{10^4 bits}{1.01 * 10^{-5} sec} = 10^7 bits/sec$$

c. The most likely architecture for this type of Gigabit MANs uses small packets (roughly 500 bits, for compatibility with ISDN switches), and a medium access control protocol vaguely similar to the slotted ring. How many slots would there be on a 200 kilometre ring? How large a window size would be required to allow the sender to keep going without stopping? How practical would the Selective-Repeat ARQ be in this application?

From above, a 200 km ring can store 10^6 *bits, which is 2000 slots. But since the ACK comes at the END of the slot, the minimum window size is 2001. This isn't very practical for selective repeat.*

VII. Consider an error recovery technique where, in the event of an error, you don't repeat the damaged packet. Instead you send blocks of extra error correction information that can be combined with the damaged packet to repair the damage. Under this scheme, assume that the initial transmission will contain an error with probability 1/2. If there is an error, the sender will send a first block of extra error correction information that will allow the receiver to correct the damaged packet with probability 3/4. And, if there is still an error, the sender will send a second block of extra error correction information that is guaranteed to correct the damaged packet.

a. What is the average service time for a packet, assuming it takes 1 second to transmit each packet or block of error correction information, and that the time until the receiver returns an acknowledgement is negligible?

*1/2 of the time, you get through in 1 second; 1/2*3/4 = 3/8 of the time you have a first error but get through on the second try using 2 seconds; 1/2*1/4 = 1/8 of the time you have a first error AND a second error, and get through in 3 seconds. Therefore*

$$\bar{X} = \frac{1}{2} + 2 \cdot \frac{3}{8} + 3 \cdot \frac{1}{8} = \frac{13}{8} = 1.625$$

b. If the arrival rate of new packets is 0.4 packets/second, what is the average packet delay in this system?

We need to apply the M/G/1 formula, since the service time has this funny 3-valued distribution instead of an exponential. First we need the second moment of the service time:

$$\overline{X^2} = \frac{1}{2} + 2^2 \cdot \frac{3}{8} + 3^2 \cdot \frac{1}{8} = 25/8 = 3.125$$

Now we plug $\lambda = 0.4$, $\bar{X} = 1/\mu = 1.625$, $\rho = 0.4 * \bar{X} = 0.65$ *and* $X^2 = 3.125$ *into the PK equation to get*

$$\bar{T} = \frac{1}{\mu} + \frac{\lambda \overline{X^2}}{2(1-\rho)} = 1.625 + \frac{0.4 * 3.125}{2 * (1 - 0.65)} = 3.41$$

FORMULAS: $\rho = \frac{\lambda}{\mu}$; $\sum_{j=0}^{\infty} x^j = \frac{1}{1-x}$ if $|x| < 1$.

M/M/1: $\pi_k = (1-\rho)\,\rho^k$; $\bar{N} = \frac{\rho}{1-\rho}$; $\bar{T} = \frac{1/\mu}{1-\rho}$;

Mean Residual Life: $\frac{\overline{X^2}}{2\,\bar{X}}$; M/G/1: $\bar{T} = \frac{1}{\mu} + \frac{\lambda \overline{X^2}}{2(1-\rho)}$;

END OF EXAMINATION

Network Delay: $T = \frac{1}{\gamma} \sum_{l} \frac{\lambda_l}{C_l - \lambda_l}; \quad \frac{\partial T}{\partial \lambda_l} = \frac{C_l}{(C_l - \lambda_l)^2};$