HW #3 Solutions

ELEN E6761 - Computer Communication Networks
Fall 2000 Prof. Rubenstein

Please turn your homework into Vassilis Stachtos' mailbox, E2 (oustide of the Electrical Engineering office, fax it by 5pm, or make some other arrangement with Vassilis. On-campus students can fax to Professor Rubenstein at (212) 932-9421 (remember to include your name and that the fax is for Professor Rubenstein). CVN students can fax as above, or fax to the CVN fax number. LATE ASSIGNMENTS WILL HAVE THEIR GRADE REDUCED BY 10% EACH BUSINESS DAY IN WHICH THEY ARE LATE!!

1. Assume a set of n receivers, labelled $r_1, \cdots, r_n$, reside on an unbridged LAN. Each receiver transmits a fraction of time p (e.g., if $p = .2$, then the receiver transmits 20% of the time.) A transmission of a frame takes time T. Let $X_A(i)$ be a random variable that equals 1 if, using algorithm A, r_1's ith transmission is successful (i.e., does not collide with any other receiver transmissions.) Compute $P(X_A(i) = 1)$ for each of the following algorithms (Note: solutions should have no dependence on i)

   (a) $A = 1$: A is Slotted Aloha

   Answer: Let $R_j(i)$ be a random variable that equals 1 if r_j transmits during the ith slot and equal 0 otherwise. Then

$$\begin{aligned} P(X_1(i) = 1) &= P(R_1(i) = 1, R_2(i) = 0, \cdots, R_n(i) = 0 | R_1(i) = 1) \\ &= P(R_1(i) = 1, R_2(i) = 0, \cdots, R_n(i) = 0)/P(R_1(i) = 1) \\ &= p(1-p)^{n-1}/p = (1-p)^{n-1} \end{aligned}$$

   (b) $A = 2$: A is Aloha, and each receiver transmits periodically with every T/p seconds, where the start time of the transmission is chosen uniformly by each receiver. Note: this is a dumb protocol, since two receivers that collide will collide every time, but it makes the problem much easier to solve. Hint: Draw a circle...

   Answer: Let a circle with circumference T/p represent the timeline within a period in which the various receivers can transmit. Each receiver transmits for a time T. r_1's transmission is successful if no other receiver's transmission overlaps with r_1's. If r_1 starts transmitting at time x, then any transmission by another receiver that starts in the interval $[x - T, x + T]$ will overlap with r_1's transmission. The probability that some receiver r_j's transmission overlaps with r_1's is $2T/(T/p) = 2p$ for $p < .5$ and 1 otherwise. Hence, $P(X_2(i) = 1) = (1 - 2p)^{n-1}$

   (c) $A = 3$: A is Aloha, where each receiver waits a random amount of time that is exponentially distributed with rate $\frac{p}{T(1-p)}$. Hint: Apply PASTA when r_1 decides to transmit, then use the memoryless nature of the exponential distribution used by the other receivers. (Note: when $p = 1/n$, then $P(X_3 = 1) = P(X_1 = 1)/e$.)

   Answer: Let $N_j(i)$ be a random variable that equals 1 if r_j is transmitting when r_1 starts its transmission, and 0 otherwise. Let $M_j(i)$ a random variable that equals the time until r_j begins next transmits after r_1's starts its ith transmission ($M_j(i) = 0$ if r_j is transmitting when r_1 starts its ith transmission.) Note that $P(X_3 = 1) = \prod_{j=2}^{n} P(N_j > T)$.
   When r_1 chooses to start transmitting, by PASTA, the probability that some other receiver r_j is already transmitting is simply the steady state probability that r_j is transmitting, which is p. Hence, $P(N_j(i) = 0) = p$. Now consider any r_j that is not transmitting when r_1 starts its ith transmission.

Since the exponential distribution is memoryless, the time from the start of r_1's transmission to the time of the start of r_j's next transmission is independent of how much time had elapsed between the time of r_j's previous transmission and the time of the start of r_1's ith transmission. More formally, $P(N_j(i) > t | N_j(i) > 0) = e^{\frac{p}{T(1-p)}t}$.

$$
\begin{aligned}
P(N_j(i) > T) &= P(N_j(i) > T, N_j(i) > 0) \\
&= P(N_j(i) > T | N_j(i) > 0) P(N_j(i) > 0) \\
&= e^{\frac{p}{(1-p)}}(1-p)
\end{aligned}
$$

Hence, $P(X_3(i) = 1) = e^{-\frac{(n-1)p}{(1-p)}}(1-p)^{n-1}$.
Setting p to $1/n$ yields

$$
\begin{aligned}
P(X_3(i) = 1) &= e^{-\frac{(n-1)/n}{1-1/n}}(\frac{n-1}{n})^{n-1} \\
&= e^{-1}(\frac{n-1}{n})^{n-1}.
\end{aligned}
$$

When $p = 1/n$, then $P(X_1(i) = 1) = (\frac{n-1}{n})^{n-1}$.

2. Assume that the sender's and receiver's clocks are skewed (running at different rates and unsynchronized). Let $R(t) = At + B$ be the time returned by the receiver's clock when the sender's clock has time t, where A and B are constants. For jitter calculations to be accurate, what restrictions (if any) must there be on the values of A and B?

   Answer: Let x be the actual jitter between two packet transmissions, p_1 and p_2. Let x' be the computed jitter, using the skewed clocks. Define $s(p_i)$ and $r(p_i)$ to be the time according to the *sender's clock* that packet p_i is respectively sent by the sender and received by the receiver. Hence, using the definition of jitter, $x = s(p_1) - r(p_1) - (s(p_2) - r(p_2))$. The computed jitter (using the receiver's clock for receiving times of packets) yields

$$
\begin{aligned}
x' &= s(p_1) - (Ar(p_1) + B) - (s(p_2) - (Ar(p_2) + B)) \\
&= s(p_1) - Ar(p_1) - (s(p_2) - Ar(p_2))
\end{aligned}
$$

   It follows that for $x = x'$, A must be restricted to 1, but B can take on any value.

3. Prove by induction that a single-bit parity check can always detect an odd number of bit-flips

   Answer: First, note that whenever the sum of j bits mod 2 is 0, then flipping a single bit changes the sum of the bits mod 2 to 1. Note that whenever the sum of j bits mod 2 is 1, then flipping a single bit changes the sum of the bits mod 2 to 0.

   Let n be the number of bits that are flipped. If $n = 0$, then the sum of the data bits plus the parity mod 2 equals 0, so if 1 bit is flipped ($n = 1$), the sum mod 2 must be 1, and 1 bit-flip will be detected. Now assume that $n - 2$ bit flips can be detected(i.e., the sum of the bits mod 2 is 1), where $n - 2$ is odd. Then $n - 1$ bit flips mod 2 must be 0, and the sum of n bit flips mod 2 is 1.

1	0	1	1	1
0	1	0	1	0
1	1	1	0	1
0	1	0	0	1
0	1	0	0	0

4. Consider the 2-D parity check above. Assume bits are incorrect with probability $p < .5$. What is the most likely set of bits that are flipped (give row, column coordinates)?

   Answer: Consider any one scenario in which j bits are flipped. The probability of this specific scenario occuring is $p^j(1-p)^{25-j}$, which gets smaller as j gets larger (since $p < .5$). Hence, the most likely scenario is the one in which the fewest number of bits are flipped. Note that it cannot be the case that 0 bits are flipped because the sum of the 5th row mod 2 equals 1. Also note that if the bit in the bottom right corner is the one that is flipped, then the parity 2-D parity check does not indicate error. This is the only possible scenario in which one bit is flipped, hence in the most likely scenario, the set of bits flipped is only the bottom right bit.

5. What is the minimum number of bits that need to be flipped in a 2-D parity check such that a corrupt packet can be accepted? Prove your answer.

   Answer: The answer is 4 (e.g., $b_{1,1}, b_{1,2}, b_{2,1}$, and $b_{2,2}$ being flipped would be accepted, where $b_{i,j}$ is the bit in the ith row, jth column.) To prove that no smaller number of bits is sufficient, note that any row or column must have an even number of bits flipped, else a bit-flip will be detected. Hence, if any bit b_{i_1,j_1} is flipped, then there must exist $i_2 \neq i_1$ and $j_2 \neq j_1$ where bits b_{i_2,j_1} and b_{i_1,j_2} are flipped. Since a bit is flipped in row i_2 and in column j_2, having only these three bits is not sufficient.

6. Consider a packet-level forward error correction scheme that uses only XOR operations between sets of packets to generate repair packets. Show (by induction on the number of packets used in the encoding) that if n arbitrary packets are used in the encoding, then it is impossible to retrieve all original data packets if less than n packets are receieved.

   Answer: Let the number of packets used to generate the XOR repairs be n. If $n = 1$, clearly at least 1 packet must be received in order to deduce the original packet. Assume that for $n-1$ packets, at least $n-1$ packets must be received at the receiver before all packets can be decoded.

   Let F_n describe an encoding scheme for n packets that uses XOR operations. Since F_n must work on an arbitrary set of packets, it must work for the case where the last (nth) data packet is all 0's. But note that XOR'ing a packet of all 0's to a bit-string has no effect on the result. In effect, one can omit applying the XOR of the all-0 packet during both encoding and decoding without effect. When the nth packet is all 0's, the encoding scheme F_n maps directly to an encoding scheme, F_{n-1} that applies XOR operations to the first $n-1$ packets.

   If F_n can decode with only the $n-1$ packets in general, then clearly it can do so when one of the packets is all-0. Consider now $n-1$ packets arriving at the receiver, one of them being the all-0 packet. Then the other $n-2$ packets must be used to retrieve $n-1$ data packets, since the all-0 packet essentially has no effect on XOR operations. But this contradicts the assumption that there is no scheme F_{n-1} that can be used to decode using any combination of $n-2$ packets. By induction, no XOR-based scheme F_n can always provide recovery for a group of n packets with only $n-1$ packets received.