

HW #7 Solutions

1. A sender S and receiver R use the Selective-Repeat protocol to communicate reliably across a network that drops or corrupts packets in each direction (from S to R and also from R to S) with probability p . Assume that the loss process is Bernoulli and that the sender never times out early (i.e., the sender does not retransmit a packet while a previous copy is heading to the receiver using the Selective-Repeat protocol or an acknowledgment for a previous transmission is on its way back to the sender).

- (a) Compute the expected number of times the sender transmits a particular packet (i.e., packet i for any i).

Answer:

We will introduce two solutions that give the same result but use a different formula to produce the result. Let R denote the number of times that the sender transmits a particular packet until a success. Also let $X_i = 0$ if the i th round trip transmission was a failure or never transmitted and $X_i = 1$ if the i th round trip transmission was a success. We have that $\forall i > 1, P(X_i = 1) = (1 - p)^2$ (no errors) and $P(X_i = 0) = 1 - (1 - p)^2$ (at least one error). Then $P(R = n) = P(X_i = 0, \dots, X_{n-1} = 0, X_n = 1) = P(X_1 = 0) \cdot \dots \cdot P(X_{n-1} = 0) \cdot P(X_n = 1)$ (independence) $\Rightarrow P(R = n) = [1 - (1 - p)^2]^{n-1} \cdot (1 - p)^2$. Next we compute the expected value:

$$E[R] = \sum_{n=1}^{\infty} n \cdot [1 - (1 - p)^2]^{n-1} \cdot (1 - p)^2 \Rightarrow E[R] = \sum_{n=1}^{\infty} n \cdot k^{n-1} \cdot (1 - k), \quad k = 1 - (1 - p)^2 \Rightarrow$$

$$E[R] = \frac{(1 - k)}{(1 - k)^2} = \frac{1}{1 - k} \Rightarrow E[R] = \frac{1}{1 - (1 - (1 - p)^2)} \Rightarrow E[R] = \frac{1}{(1 - p)^2}$$

Second Solution:

We know from the definition of the expected value that:

$$E[R] = \sum_{n=0}^{\infty} P(R > n) \text{ but we know that } P(R > n) = (1 - (1 - p)^2)^n \text{ (we need more than } n \text{ retries)}$$

$$E[R] = \sum_{n=0}^{\infty} (1 - (1 - p)^2)^n = k^n, \quad k = 1 - (1 - p)^2 \Rightarrow E[R] = \frac{1}{1 - k} = \frac{1}{1 - (1 - (1 - p)^2)} \Rightarrow E[R] = \frac{1}{(1 - p)^2}$$

- (b) If the window size used by the protocol is w , and it takes time τ for a packet to be received and acknowledged when no loss or corruptions take place, give an upper bound on the maximum rate of the protocol. Explain why your result is an upper bound (hint: at most w packets can be transmitted at any given time).

Answer:

Using Little's law we can find that the upper bound on the rate of the protocol (R), given that we have no errors in transmission, is: $R_{max} = \left(\frac{w}{\tau}\right)$. We can see this because we are allowed to have at each time t w packets on transition that need time τ (no errors) to be serviced. By applying Little's law we get $w = R_{max} \cdot \tau \Rightarrow R_{max} = \left(\frac{w}{\tau}\right)$

2. Assume S and R communicate in a networking environment similar to that in problem 1. The reliable data transfer protocol used is Go-back-N, where packets that are received out of order are dropped by the receiver (i.e., packet $i + 1$ is accepted only if packet i has already been received). Assume the sender and receiver communicate in rounds where each round, the sender sends the w packets that are currently in its window.

- (a) Assuming ACKs are never lost, compute the expected number of packets accepted by the receiver each round.

Answer:

Let R be a random variable that equals the number of packets accepted by the receiver between the time the sender receives acknowledgments. Then $P(R > n) = (1 - p)^{n+1}$ (p is the probability of failure for a single packet and the packets are independent). Then we have that:

$$E[R] = \sum_{n=0}^{n=w-1} (1-p)^{n+1} = (1-p) \cdot \sum_{n=0}^{n=w-1} (1-p)^n = (1-p) \cdot \frac{1 - (1-p)^w}{1 - (1-p)} = (1-p) \cdot \frac{1 - (1-p)^w}{p} \Rightarrow$$

$$E[R] = (1-p) \cdot \frac{1 - (1-p)^w}{p}$$

- (b) Suppose the receiver responds each round by sending a single ACK (at the end of the round) indicating the largest sequence number it has accepted, and that this ACK is lost with probability p . Let R be a r.v. that equals the number of packets accepted by the receiver between the time the sender receives acknowledgments. Let N be the number of rounds that take place until the sender receives an acknowledgment. Compute $E[R]$ and $E[N]$.

Answer:

We will first compute $E[N]$ using the fact that $P(N > i) = p^i$ so

$$E[N] = \sum_{i=0}^{i=\infty} P(N > i) = \sum_{i=0}^{i=\infty} p^i \Rightarrow E[N] = \frac{1}{1-p}$$

In order to compute $E[R]$ we define an indicator random variable $X_{i,r} = \begin{cases} 1 & \text{we keep packet } i \text{ after } r \text{ rounds} \\ 0 & \text{otherwise} \end{cases}$

Then we have that:

$$E[R] = \sum_{r=1}^{\infty} E[R|N=r]P(N=r) \text{ and } R = \sum_{i=1}^{i=w} X_{i,r} \Rightarrow$$

$$E[R] = \sum_{r=1}^{\infty} \sum_{i=1}^{i=w} E[X_{i,r}|N=r]P(N=r) \quad (1)$$

But we know that $E[X_{i,r}|N=r] = P(R \geq i|N=r) = (1-p)^{i-1}$ and $P(N=r) = p^{r-1} \cdot (1-p)$. Now using (1) we get :

$$E[R] = \sum_{r=1}^{\infty} \sum_{i=1}^{i=w} (1-p)^{i-1} \cdot p^{r-1} \cdot (1-p)$$

- (c) (Extra Credit) The expected number of transmissions per round is $E[R]/E[N]$ (and not $E[R/N]$). Why?

Answer:

Let T be the number of transmission per round. Then because:

$$E[T] = \lim_{n \rightarrow \infty} \frac{R_1 + R_2 + R_3 + \dots + R_n}{N_1 + N_2 + N_3 + \dots + N_n} = \lim_{n \rightarrow \infty} \frac{(R_1 + R_2 + R_3 + \dots + R_n)/n}{(N_1 + N_2 + N_3 + \dots + N_n)/n}$$

All random variables R_i are independent, identically distributed so from the strong law of large number we get:

$$\lim_{n \rightarrow \infty} (R_1 + R_2 + R_3 + \dots + R_n)/n = E[R]$$

Also N_i random variables are independent, identically distributed so from the strong law of large number we get:

$$\lim_{n \rightarrow \infty} (N_1 + N_2 + N_3 + \dots + N_n)/n = E[N]$$

Finally :

$$E[T] = \lim_{n \rightarrow \infty} \frac{(R_1 + R_2 + R_3 + \dots + R_n)/n}{(N_1 + N_2 + N_3 + \dots + N_n)/n} = \frac{E[R]}{E[N]}$$

3. Consider a reliable data transfer protocol that transfers data using a window of size w in an environment in which packets can be lost, but cannot be reordered. The protocol uses a sequence numbering scheme that goes from 0 to $n - 1$, such that packet representing the i th segment of data is assigned sequence number $i \pmod{n}$. If n is too small, the receiver might mistake one data packet for another. For instance, if $n < w$, then the receiver might mistake packet n (which has sequence number 0) as a retransmission of packet 0, which also has sequence number 0, as both packets could be in the window at the same time. Show that a receiver can make such an error when

- (a) The protocol is Selective Repeat and $n = 2w - 1$.

Answer:

The fact that $n = 2w - 1$ restricts us to sequences from 0 to $2w - 2$. We have that the first window will look like: $[0 \ 1 \dots (w - 1)]$ and the second $[w \ (w + 1) \dots (2w - 2) \ 0]$. Because there is a probability that an error can happen in the acknowledgement of packet 0 in the first window the receiver will not be able to distinguish if the second window is a retransmission of packet 0 or of packet $n = 2w - 1$.

- (b) Go-Back-N (assume the receiver discards any packets that cause a gap in the received sequence) and $n = w$.

Answer:

The fact that $n = w$ restricts us to sequences from 0 to $w - 1$. We have that the first window will look like: $[0 \ 1 \dots (w - 1)]$ and the second $[0 \ 1 \dots (w - 1)]$. Because there is a probability that an error can happen in the acknowledgement of the first window the receiver will not be able to distinguish if the second window is a retransmission of the first window or a transmission of the second window.

4. A sender wishes to perform reliable communication with two receivers simultaneously. Upon receiving a packet, a receiver sends an ACK if the packet is received uncorrupted. Otherwise, it sends a NAK if the packet is corrupted. A receiver's acknowledgment can be corrupted as well. Each receiver's acknowledgment arrives at the sender as an uncorrupted ACK independently with probability p (i.e., After sending a packet to a receiver, the sender "knows" with probability p that the receiver successfully received the packet.) Assume that the sender can determine from which receiver an arriving acknowledgment belongs to (e.g., it can make the determination from the source address of the acknowledgment packet.)
- (a) Is it sufficient to have just two sequence numbers (i.e., just using sequence numbers 0 and 1 for packets) as is sufficient in the single-receiver case? Explain how two can be used or give a counter-example showing that two is not sufficient.

Answer:

Yes, because we can distinguish between the two receivers so we can assign variables A and B to them. Let A = 1 if we receive an ACK from receiver 1 and 0 otherwise. Similarly let B = 1 if we receive an ACK from receiver 2 and 0 otherwise. We will know if we received ACK from both if both A=1 and B=1.

- (b) Consider the reliable transmission of a *single* packet over a series of rounds. The following lists three algorithms used by the sender to handle receiver acknowledgments. In each algorithm, at the start of each round, if the sender's state indicates that some receiver has not acknowledged receipt, then the sender retransmits the packet to *both* receivers. The round ends with the sender receiving and processing feedback from the receivers. For each of the following algorithms, indicate whether or not the algorithm is correct (i.e., that it always reliably delivers the packet to both receivers.) If not, give a scenario where it fails. If it is correct, do both of the following: draw the finite state machine for the sender that indicates its state during the transmission of the packet. Also, let R_i be a random variable that equals the number of rounds required to reliably deliver the packet to both receivers. Compute $E[R_i]$.
- i. A single bit is initially set to 0, and is set to 1 when an uncorrupted ACK is received from *either* receiver within a round. The sender stops transmitting the packet once the bit is set to 1.

Answer:

This algorithm is not correct because in case that one of the receivers fails to get the packet and the other succeeds. The receiver that got the packet will send an ACK and the sender will continue to the next packet without retransmitting. The previous scenario causes one of the receivers to miss a packet and thus the algorithm fails.

- ii. A single bit is initially set to 0, and is set to 1 when uncorrupted ACKs are received from both receivers in a single round. The sender stops transmitting once the bit is set to 1.

Answer:

This algorithm is correct because it tracks the status of both receivers. Although correct it is not the most optimal in terms of transmissions as even when something is received by one receiver it gets discarded if a new transmission is done. So the $P(R_i > n) = (1 - p^2)^n$ and so:

$$E[R_i] = \sum_{n=0}^{\infty} (1 - p^2)^n \Rightarrow E[R_i] = \frac{1}{1 - (1 - p^2)} \Rightarrow E[R_i] = \frac{1}{p^2}$$

- iii. 2 bits are used, one for each receiver. Both bits are initially set to 0. When an uncorrupted ACK is received from a receiver, that receiver's bit is set to 1. The sender stops transmitting once both bits are set to 1.

Answer:

In the n th round the probability that we will have a failure for one of the transmitters is $(1 - p)^n$ so the probability of success for one of the transmitters is $1 - (1 - p)^n$. We need that either of them have failed so we have to compute the probability that both succeed and then to subtract it from 1. Thus we have that the probability that both succeed is $[1 - (1 - p)^n]^2$ therefore the probability that either failed (we need more than n rounds) is $1 - [1 - (1 - p)^n]^2$ Then

$$P(R_i > n) = 1 - [1 - (1 - p)^n]^2 \Rightarrow E[R_i] = \sum_{n=0}^{\infty} (1 - [1 - (1 - p)^n]^2) = \sum_{n=0}^{\infty} 2 \cdot (1 - p)^n - (1 - p)^{2n} \Rightarrow$$

$$E[R_i] = \frac{2}{p} - \frac{1}{1 - (1 - p)^2}$$