Directions: Start early! We will pick 1 to grade out of 4. We will however, give you 25% completion credit for a good faith attempt at all 4 problems.

- 1, **HDLC Framing**: The HDLC protocol uses a flag 01111110 at the start and end of frames. In order to prevent data bits from being confused with flags, the sender stuffs a zero after every 5 consecutive ones in the data. We want to understand further that not all flags work but perhaps some others do work so the HDLC flag is not the only possible one.

  - Consider the flag 11111111 and a similar stuffing rule to HDLC (stuff a 0 after 5 consecutive 1's). Show a counterexample to show this does not work. (5 points)
  - Consider the flag 11111111. Find a stuffing rule that works and argue that it is correct. What is the worst case efficiency of this rule? (Recall HDLC had a worst case efficiency of 1 in 5 bits, or 20%) (5 points)
  - Hugh Hopeful has invented another new flag for HDLC (Hopeful Data Link Control) protocol. He uses the flag 00111100. In order to prevent data bits from being confused with flags, the sender stuffs a one after receiving 001111. Does this work? Justify your answer with a short proof or counterexample (8 points)
  - To reduce the overhead, Hugh tries to stuff a 1 after receiving 0011110. Will this work? Justify your answer with a short proof or counterexample. (7 points)
  -

2, **CRCs Polynomial View**: Consider the 5-bit CRC generator $x^5 + x^3 + x + 1$.

  - Consider the message 100. Calculate the CRC for this message using the 5th degree polynomial generator above (5 points)

  - The simplest way to create an undetected error is to add an error polynomial equal to generator to the message + CRC. What is the resulting message that is *received? (2 points)*

  - Does this generator detect all 1 bit errors? Why? (Hint: review arguments in Notes) (5 points)

  - Does this generator detect all odd bit errors. Why? (5 points, see Notes)

  - . How many undetected burst errors (starting at Offset 0) can there be of burst length 11? To help you do this, note that a burst error of length 11 is a polynomial that starts with x^{10} and ends with 1 with optional terms for the remaining powers between 9 and 1. For an undetected error, the resulting burst error must be divisible by the generator $x^5 + x^4 + x + 1$. Instead of

seeing which bursts are divisible by the generator, find how many bursts of length 11 are *multiples* of the generator? For example, if we multiply $x^5 + x^3 + x + 1$ with $x^5 + 1$ we get a length 11 burst. To do this notice that the only way to get x^{10} as the highest power and 1 as the lowest power is to multiply by polynomials whose highest power is x^5 and lowest power is 1. The remaining powers (4, 3, 2, 1) may or not be present leading to several possibilities. Can you write down all such distinct polynomials after the multiplication. How many such polynomials are there? (10 points)

- (Extra Credit, 5 points) Based on the last two answers, can you argue briefly why the probability of a detecting a burst of arbitrary length is $1/2^k$ where k is the degree of the CRC (5 in this case). Hint: consider the ratio of the number of undetected bursts to the total number of possible bursts of any given length. What does this tell you about the power of CRC 32?

-

3. **Data Link Protocols on Synchronous Links**: So far in all our Data Link protocols we have assumed the links to be asynchronous in that the delay of a frame or ack could be arbitrary. Now we consider the case that the time taken for a message or ack is 0.5 time units. Further senders send frames only at integer times like 0,1,2. When a receiver gets an error-free frame (sent at time n) at time $n + 0.5$, the receiver sends an ack back that arrives (if successful) just before time $n + 1$. Suppose we use the standard alternating bit protocol except that the sender allowed to send at integer times.

- Does the sender need to number the data frames? If your answer is yes give a counterexample to show what goes wrong when it does not. (10 points)
- Does the receiver need to number the ack frames? If your answer is yes give a counterexample to show what goes wrong when it does not. For this question, assume that an alternating bit (0, 1) counts as a number (10 points)
- Describe a simple protocol for the sender to initialize the receiver state after a crash? (5 points)

-

4. **Error Recovery:** Peter Protocol has been consulting with an Internet Service Provider and finds that they use an unusual error recovery protocol shown in the Figure below (next page). The protocol is very similar to Go-back-N with numbered data packets; the key difference is that the sender does not normally send ACKs. The receiver sends a message called a NACK only if it detects an error in the received sequence or if it receives a so-called STATUS packet. In the figure below, the sender sends off the first three data packets. The second one is lost; thus when the receiver gets the third packet it detects an error and sends a NAK which contains the highest number the receiver has received in sequence. When NAK 1 gets to the sender, the sender retransmits data packets 2 and 3.

Periodically, based on a timer, the sender transmits a STATUS packet. The receiver always replies to a STATUS packet using a NAK.

- Why is the STATUS packet needed? What can go wrong if the sender does not send STATUS packets? (5 points)
- A STATUS packet is sent when a STATUS timer expires. The sender maintains the following property: ``While there remains unacknowledged data, the STATUS timer is running.'' Why does this property guarantee that any data packet given to the sender will evntually reach the receiver (as long as the link delivers most packets without errors)? (5 points)
- Under what conditions must the timer be stopped and started so as to maintain the property (5 points)
- Consider sending a single data packet D that is lost. After that no packets get lost. What is the worst-case latency before the receiver receives D and the sender knows the receiver has got D? (10 points)

Figure for Problem 4

Sender   **Receiver**

D,1

D,2

D,3   REJECT

Nack 1

D, 2

D,3

Status

Nak 3

1.

https://powcoder.com
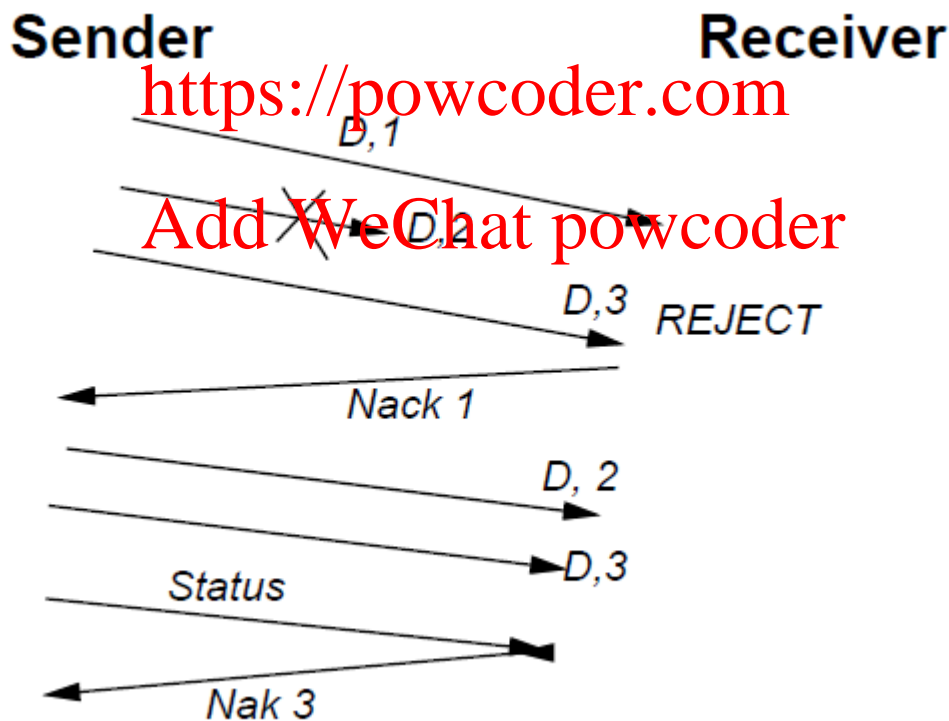
Assignment Project Exam Help

Assignment Project Exam Help

Add WeChat powcoder

https://powcoder.com

Add WeChat powcoder