CS425 Assignment 2

Naman Singla
#200619

2. **(10 points)** In the Go-back-N ARQ mechanism using k-bit sequence numbers, why is the window size limited to $2^k - 1$ and not 2^k?

Solution
In the Go-Back-N ARQ (Automatic Repeat Request) mechanism, a sender transmits a sequence of packets to a receiver. Each packet is assigned a unique k-bit sequence number, which is used to identify the packet and to ensure that the receiver can distinguish between packets that are in sequence and those that are out of sequence. The sender maintains a sliding window of sequence numbers that it can transmit, and the receiver acknowledges the packets it receives by sending back an acknowledgment packet with the sequence number of the next expected packet. If the sender does not receive an acknowledgment for a transmitted packet within a certain time period, it retransmits all the packets in the current window.
As a case of 3-bit sequence number (i.e. sequence number space is 8). Suppose sender sends frame 0 and gets back an RR 1. Then sends frames 1, 2, 3, 4, 5, 6, 7, 0 and gets another RR 1. This could mean that all eight frames were received correctly and the RR 1 is a cumulative acknowledgment. It could also mean that all eight frames were damaged or lost in transit, and the receiving station is repeating its previous RR 1. The problem is avoided if the maximum window size is limited to 7, i.e. 2^3 - 1
Hence, Max window size = 2^k – 1 (for a k-bit sequence number)

3. **(10 points)** What is the maximum window size that can be used in the Selective-Reject ARQ mechanism that uses k-bit sequence numbers? Explain your answer.

Solution
Window size is more restrictive for selective-reject than for go-back-N
Consider the case of a 3-bit sequence number size for selective-reject. Sender sends frames 0 through 6. Receiver receives all seven frames and cumulatively acknowledges with RR 7. Because of a noise burst, the RR 7 is lost. Sender times out and retransmits frame 0. Receiver has already advanced its receive window to accept frames 7, 0, 1, 2, 3, 4, and 5. It assumes that frame 7 has been lost and that this is a new frame 0, which it accepts. The problem is that there is an overlap in between the sending and receiving window. To overcome the problem, the maximum window size should be no more than half the range of sequence numbers
Hence, Max window size = 2^(k-1) (for a k-bit sequence number)

4. **(10 points)** A channel has a data rate of 4 kbps and a propagation delay of 20 ms. For what range of frame sizes does stop-and-wait give an efficiency of at least 50%?

Solution

$$a = \frac{\text{Propagation time}}{\text{Transmission time}} = \frac{t_{prop}}{t_{frame}}$$

$$\boxed{U = \frac{1}{1 + 2a}}$$

$$t_{frame} = \frac{\text{frame size}}{\text{Bandwidth}}$$

Lets assume frame size to be s, and Bandwidth as B

$$\begin{aligned} U = \frac{s}{s + P \cdot B} &> 0.5 \\ \implies s &> P \cdot B \\ \implies s &> 20ms \cdot 4kbps \\ \implies s &> 80 \text{ bits} \end{aligned}$$

5. **(20 points)** Consider a frame consists of one character of 4 bits. Assume that the probability of bit error is 10^{-3} and that it is independent in each bit.

(a) What is the probability that the received frame contains no errors?

Solution

Since the each bit is independent we can directly multiply the non-occurance of bit error fo each bit

$$\left(1-10^{-3}\right).\left(1-10^{-3}\right).\left(1-10^{-3}\right).\left(1-10^{-3}\right) = \left(1-10^{-3}\right)^4 \\ = 0.996$$

(b) What is the probability that the received frame contains at least one error?

Solution

This is simply the negation of part (a)

$$1 - 0.996 \; = \; 0.004$$

(c) Now assume that one parity bit is added. What is the probability that the frame is received with errors that are not detected?

Solution

Let E denotes the event where parity bit is reversed.

$P(E) = 0.001$

<u>Case I : where parity bit is not reversed and error not detected</u>

This implies that number of flips in first 4 bits should be even(call this event as F)

(a) 2 flips : $\binom{4}{2}\left(10^{-3}\right)^2\left(1-10^{-3}\right)^2$

(b) 4 flips : $\left(10^{-3}\right)^4$

$$P(F) = \binom{4}{2}\left(10^{-3}\right)^2\left(1-10^{-3}\right)^2 + \left(10^{-3}\right)^4 \\ = 5.99 \cdot 10^{-6}$$

Total Probability

$$= P\left(\overline{E}\right) \cdot P(F) \\ = 0.999 \cdot 5.99 \cdot 10^{-6} \\ = 5.98 \cdot 10^{-6}$$

<u>Case II : where parity bit is reversed and error is detected</u>

This implies that number of flips in first 4 bits should be odd(call this as event G)

(a) 1 flip : $\binom{4}{1}\left(10^{-3}\right)\left(1-10^{-3}\right)^3$

(b) 3 flips : $\binom{4}{3}\left(10^{-3}\right)^3\left(1-10^{-3}\right)$

$$P(G) = \binom{4}{1}\left(10^{-3}\right)\left(1-10^{-3}\right)^3 + \binom{4}{3}\left(10^{-3}\right)^3\left(1-10^{-3}\right) \\ = 3.99 \cdot 10^{-3}$$

Total Probability

$$= P(E) \cdot P(G) \\ = 0.001 \cdot 3.99 \cdot 10^{-3} \\ = 3.99 \cdot 10^{-6}$$

Case I + Case II = $9.97 \cdot 10^{-6}$

6. **(10 points)** For P = 110011 and M = 11100011, find the CRC

Solution

Length of P = 6, so length of CRC will be 5. Lets append 5 zeros at end of M and divide it by P.

$$
\begin{array}{r}
110011\overline{)1110001100000} \\
110011 \\
\overline{0010111100000} \\
110011 \\
\overline{01110000000} \\
110011 \\
\overline{0010110000} \\
110011 \\
\overline{01111100} \\
110011 \\
\overline{0011010}
\end{array}
$$

Hence, CRC = 11010

7. **(20 points)** (a). In a CRC error-detecting scheme, choose $P(x) = X^4 + X + 1$. Encode the bits 10010011011.

Solution

This implies P=10011, M=10010011011

Length of P =5, so length of CRC will be 4. Lets append 4 zeros at end of M and divide it by P.

$$
\begin{array}{r}
10011\overline{)100100110110000} \\
10011 \\
\overline{000010110110000} \\
10011 \\
\overline{00101110000} \\
10011 \\
\overline{001000000} \\
10011 \\
\overline{0001100}
\end{array}
$$

CRC = 1100

So, Transmission message = 100100110111100 (b). Suppose the channel introduces an error pattern 100010000000000 (i.e., a flip from 1 to 0 or from 0 to 1 in position 1 and 5). What is received? Can the error be detected?

Solution

After adding error (reversing bits in transmission message at positions 1,5)

Received message = 000110110111100

To detect error, division of received message by P should yield non-zero remainder

```
                11001110
            -------------
10011 ) 110110111100
        10011
        -----
         10000111100
         10011
         -----
          0011111100
             10011
             -----
              1100100
              10011
              -----
               101000
               10011
               -----
                 1110
```

As we can see, remainder is non-zero. Hence, error is detected

(c). Repeat part (b) with error pattern 100110000000000.

Solution

After adding error (reversing bits in transmission message at positions 1,4,5)

Received message = 000010110111100

To detect error, division of received message by P should yield non-zero remainder

```
                1010100
            -------------
10011 ) 10110111100
        10011
        -----
         0101111100
          10011
          -----
           01001100
            10011
            -----
              0000
```

As we can see, remainder is zero. Hence, error is not detected