CS425A: Computer Networks
Assignment 2

Manasvi Jain
210581

February 2024

1 Problem 1

Execute the provided code snippet with this report. The program can be modified easily to generate errors for frames of different sizes. The results will be printed out in the terminal.

2 Problem 2

In the Go-Back-N ARQ mechanism with a window size of 2^k, if the sender sends frames in the order $(0, 1, \ldots, 2^k - 1)$ and an acknowledgment is lost, the receiver might incorrectly accept a resent previous window, leading to unintended duplication.

In contrast, with a reduced window size of 2^k-1, such as frames $(0, 1, \ldots, 2^k-2)$, if an acknowledgment for the first window is lost, the receiver, expecting frame $2^k - 1$, correctly identifies the start of a new window when the sender resends frame 0. This avoids unintended duplication and aligns with the proposed hypothesis for a window size limitation.

In conclusion, limiting the window size to $2^k - 1$ in Go-Back-N ARQ mitigates potential errors associated with acknowledgment loss and ensures reliable communication in the protocol.

3 Problem 3

Let's explore two scenarios for $k = 5$, considering window sizes of $2k$ and $2k - 1$.

Scenario 1: Window Size 2^5 (32 frames) - Suppose the sender uses 5-bit sequence numbers, allowing for a maximum of $2^5 = 32$ unique frame numbers. With a window size of 2^5, consecutive windows would overlap. For example, the first window might have frame numbers $f_1 = (0, 1, 2, ..., 31)$, and the next window could have frame numbers $f_2 = (32, 33, ..., 0)$. If an acknowledgment for f_1 is lost, and the sender resends the entire window, ambiguity arises as the receiver may accept the frames into the next window, leading to errors.

Scenario 2: Window Size 2^{5-1} (16 frames) - Now, consider a window size of $2^{5-1} = 16$. With this limitation, the sender can safely resend the first window without ambiguity. For instance, if the first window has frame numbers $f_1 = (0, 1, 2, ..., 15)$ and an acknowledgment is lost, the sender can resend this window without causing confusion. The frames will not overlap with the next window, as it strictly contains frame numbers $f_2 = (16, 17, ..., 31)$, ensuring clear identification.

This reasoning can be extended to any k (e.g., $k = n$). A window size of2^{k-1} guarantees that consecutive windows are disjoint and have a sufficient number of distinct frames. This eliminates the possibility of ambiguity during retransmissions, as frames from the retransmitted window do not interfere with the frames of the current window.

4 Problem 4

Given a channel with a data rate of 4 kbps and a propagation delay of 20 ms, the stop-and-wait efficiency is given by $U = \frac{1}{1+2a}$.

To achieve an efficiency of at least 50%, we solve for a:

$$U \geq 50\% \implies 1 \geq \frac{1}{1+2a} \implies a \leq \frac{1}{2}.$$

Using the definition $a = \frac{t_{\text{prop}}}{t_{\text{frame}}}$ and given $t_{\text{prop}} = 20$ ms, we find:

$$40 \text{ ms} \leq t_{\text{frame}} \implies 40 \text{ ms} \leq \frac{\text{Size of the frame in bits}}{4 \text{ kbps}} \implies 160 \text{ bits} \leq \text{Size of the frame in bits.}$$

Therefore, the frame size needs to be at least 160 bits for the efficiency to be at least 50% in stop-and-wait.

5 Problem 5

Assuming a bit error probability of 10^{-3} (denoted as p) for a frame consisting of one character of 4 bits:

1. Probability of no error in the frame: $(1-p)^4 = (1-10^{-3})^4 \approx 0.996$
   Explanation: Let E_i be the event that the ith bit is flipped. The probability $P(E_i)$ is given by p, and the probability of no error in each bit is $1-p$. The probability of no errors in the entire frame is the product of the individual probabilities.

2. Probability of at least one error in the frame: $1-(1-p)^4 \approx 0.004$
   Explanation: The complement of the probability of no error gives the probability of at least one error in the frame.

3. When a parity bit is added, we receive a 5-bit frame. Errors cannot be detected if there are an even number of bit inversions. We consider cases with exactly 2 or 4 errors (omitting 0 errors).

   - For exactly 2 errors: Considering pairs of 2 bits out of 5 (including the parity bit), there are $\binom{5}{2}$ pairs. The probability of a pair being erroneous is p^2, and the probability of the rest being correct is $(1-p)^3$. The total probability of exactly 2 errors is calculated as $\binom{5}{2} \times p^2 \times (1-p)^3$, which is approximately 9.97×10^{-6}.
   - For exactly 4 errors: Considering combinations of 4 bits out of 5, there are $\binom{5}{4}$ combinations. The probability of exactly 4 errors is calculated as $\binom{5}{4} \times p^4 \times (1-p)$, resulting in approximately 4.995×10^{-12}.
   - The overall probability of error not being detected is the sum of the probabilities for 2 and 4 errors, which is approximately 9.97×10^{-6}. This is because either case contributes to the scenario where errors go undetected.

6 Problem 6

To determine the CRC (Cyclic Redundancy Check), the message M is first extended by appending five zeros (shown in a different color for clarity), making it $M00000$. The extended message is then divided (modulo 2) by the divisor P. The quotient and remainder obtained are 10110110 and 11010, respectively.

CRC Calculation for $P = 110011$ and $M = 11100011$:

$$
\begin{array}{l}
10110110 \\
110011\overline{)\,1110001100000} \\
\underline{110011} \\
010111 \\
\underline{000000} \\
101111 \\
\underline{110011} \\
111000 \\
\underline{110011} \\
010110 \\
\underline{000000} \\
101100 \\
\underline{110011} \\
111110 \\
\underline{110011} \\
011010 \\
\underline{000000} \\
R = 11010
\end{array}
$$

$$
\begin{aligned}
\text{Extended Message} &: \underline{1110001100000} \\
\text{Divisor} &: \underline{110011} \\
\text{Quotient} &: 10110110 \\
\text{Remainder (CRC)} &: 11010
\end{aligned}
$$

7 Problem 7

1. **Part(a):**

   The binary sequence $M = 10010011011$ is translated into its polynomial representation as $M(x) = x^{10} + x^7 + x^4 + x^3 + x + 1$. This step is crucial for the encoding process, which involves multiplying $M(x)$ by x^4 and then dividing the product by the polynomial $P(x) = x^4 + x + 1$, yielding a remainder of $R(x) = x^3 + x^2$. This remainder is then converted into a 4-bit binary string, $R = 1100$. The encoding is completed by appending R to M, resulting in the encoded string $MR = 100100110111100$.

2. **Part(b):**

   The received pattern W is obtained after the first and fifth bits of the encoded string MR are altered, leading to $W = 000110110111100$. This can also be achieved by XOR-ing MR with a specific error pattern, where '1' indicates a flipped bit. To detect errors, W is divided by the polynomial $P(x) = x^4 + x + 1$, with the operation resulting in a remainder $R = 1110$ (or in polynomial form, $R(x) = x^3 + x^2 + x$). A non-zero remainder signals the presence of an error.

3. **Part(b):**

   To ascertain the modified received pattern W, an XOR operation between the original encoded pattern and an error pattern is performed, resulting in $W = 000010110111100$. This is part of the error detection mechanism, where W is then divided by $P(x) = x^4 + x + 1$. The division yields a remainder of $R = 0000$ (or in polynomial notation, $R(x) = 0$), indicating that the error remains undetected in this scenario.

$$
\begin{array}{rl}
 & x^{10} + x^6 + x^4 + x^2 \\
P(x) = x^4 + x + 1 & \big) \; x^{14} + x^{11} + x^8 + x^7 + x^5 + x^4 = x^4.M(x) \\
 & \underline{x^{14} + x^{11} + x^{10}} \\
 & x^{10} + x^8 + x^7 \\
 & \underline{x^{10} + \quad x^7 + x^6} \\
 & x^8 + \quad x^6 + x^5 + x^4 \\
 & \underline{x^8 + \quad x^5 + x^4} \\
 & x^6 \\
 & \underline{x^6 + x^3 + x^2} \\
 & R(x) = x^3 + x^2
\end{array}
$$

Figure 1: Polynomial Division - Part(a)

$$
\begin{array}{rl}
 & x^7 + x^6 + x^3 + x^2 + x \\
P(x) = x^4 + x + 1 & \big) \; x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^3 + x^2 = x^4.M'(x) \\
 & \underline{x^{11} + \quad x^8 + x^7} \\
 & x^{10} \\
 & \underline{x^{10} + \quad x^7 + x^6} \\
 & x^7 + x^6 + x^5 + x^4 + x^3 \\
 & \underline{x^7 + \quad x^4 + x^3} \\
 & x^6 + x^5 + \quad x^2 \\
 & \underline{x^6 + \quad x^3 + x^2} \\
 & x^5 + \quad x^3 \\
 & \underline{x^5 + \quad x^2 + x} \\
 & R(x) = x^3 + x^2 + x
\end{array}
$$

Figure 2: Polynomial Division - Part(b)

$$
\begin{array}{rl}
 & x^6 + x^4 + x^2 \\
P(x) = x^4 + x + 1 & \big) \; x^{10} + x^8 + x^7 + x^5 + x^4 + x^3 + x^2 \\
 & \underline{x^{10} + \quad x^7 + x^6} \\
 & x^8 + \quad x^6 + x^5 + x^4 \\
 & \underline{x^8 + \quad x^5 + x^4} \\
 & x^6 + x^3 + x^2 \\
 & \underline{x^6 + x^3 + x^2} \\
 & R(x) = 0
\end{array}
$$

Figure 3: Polynomial Division - Part(c)