CNL ASSIGNMENT 06

SELECTIVE ARQ

ANTRIKSH SHARMA **28 / 08 / 2022**
20070122021
CS-A1

Objective: Main objective of this lab task to let student understand the basics off error-recovery using window-based control and simulate **Selective Repeat** protocols**.**

Theory:

Both Go-Back-N Protocol and **Selective Repeat Protocol** are the types of sliding window protocols. The main difference between these two protocols is that after finding the suspect or damage in sent frames go-back-n protocol re-transmits all the frames whereas selective repeat protocol re-transmits only that frame which is damaged. Now, we shall see the difference between them:

S.NO	Go-Back-N Protocol	Selective Repeat Protocol
1.	In Go-Back-N Protocol, if the sent frame are find suspected then all the frames are re-transmitted from the lost packet to the last packet transmitted.	In selective Repeat protocol, only those frames are re-transmitted which are found suspected.
2.	Sender window size of Go-Back-N Protocol is N.	Sender window size of selective Repeat protocol is also N.
3.	Receiver window size of Go-Back-N Protocol is 1.	Receiver window size of selective Repeat protocol is N.
4.	Go-Back-N Protocol is less complex.	Selective Repeat protocol is more complex.
5.	In Go-Back-N Protocol, neither sender nor at receiver need sorting.	In selective Repeat protocol, receiver side needs sorting to sort the frames.
6.	In Go-Back-N Protocol, type of Acknowledgement is cumulative.	In selective Repeat protocol, type of Acknowledgement is individual.
7.	In Go-Back-N Protocol, Out-of-Order packets are NOT Accepted (discarded) and the entire window is re-transmitted.	In selective Repeat protocol, Out-of-Order packets are Accepted.
8.	In Go-Back-N Protocol, if Receives a corrupt packet, then also, the entire window is re-transmitted.	In selective Repeat protocol, if receives a corrupt packet, it immediately sends a negative acknowledgement and hence only the selective packet is retransmitted.
9.	Efficiency of Go-Back-N Protocol is `N/(1+2*a)`	Efficiency of selective Repeat protocol is also `N/(1+2*a)`

LAB TASK 2: Implement the Selective Repeat Protocol

```cpp
#include <bits/stdc++.h>
using namespace std;
int buffer[32];
void generate_data()
{
    for (int i = 0; i < 32; i++)
    {
        buffer[i] = rand() % 100;
    }
}
int main()
{
    int i, p, w, k = 0, p2;
    int timer[32];
    for (i = 0; i < 32; i++)
    {
        timer[i] = 5;
    }
    cout << "Enter window size: ";
    cin >> w;
    generate_data();
    i = 0;
    while (k < 32)
    {
        for (int j = 0; j < w; j++)
        {

            if ((i + j) <= 32)
                cout << "Packet " << (i + j) << " sent." << endl;
            p = rand() % 2;
            timer[k] -= 1;
            if (p == 0 || timer[k] == 0)
            {
                p2 = rand() % 2;
                if (p2 == 0)
                {
                    if (timer[k] == 0)
                    {
                        cout << "Timeout occured for packet " << k << ".\n";
                    }
                    else
                    {
                        cout << "Packet " << k << " not received.\n";
                    }
                    cout << "Packet " << k << " sent.\n";
                    timer[k] = 5;
                }
                else
                {
                    cout << "Packet " << k << " received.\n";
                    k += 1;
                }
            }
        }
        i += w;
    }
    return 0;
}
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS F:\SIT\5. SEM 5\CNL> cd "f:\SIT\5. SEM 5\CNL\Go Back to N\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Enter window size: 5
Packet 0 sent.
Packet 1 sent.
Packet 0 received.
Packet 2 sent.
Packet 1 received.
Packet 3 sent.
Packet 4 sent.
Packet 5 sent.
Packet 2 received.
Packet 6 sent.
Packet 7 sent.
Packet 3 received.
Packet 8 sent.
Packet 9 sent.
Packet 4 received.
Packet 10 sent.
Packet 11 sent.
Packet 12 sent.
Packet 5 received.
Packet 13 sent.
Packet 6 not received.
Packet 6 sent.
Packet 14 sent.
Packet 15 sent.
Packet 16 sent.
Packet 17 sent.
Packet 18 sent.
Packet 6 received.
Packet 19 sent.
Packet 7 not received.
Packet 7 sent.
Packet 20 sent.
Packet 21 sent.
Packet 7 not received.
Packet 7 sent.
Packet 22 sent.
Packet 7 not received.
Packet 7 sent.
Packet 23 sent.
```

PROBLEMS OUTPUT TERMINAL JUPYTER COMMENTS DEBUG CONSOLE

```
Packet 7 not received.
Packet 7 sent.
Packet 24 sent.
Packet 7 not received.
Packet 7 sent.
Packet 25 sent.
Packet 7 received.
Packet 26 sent.
Packet 8 received.
Packet 27 sent.
Packet 9 not received.
Packet 9 sent.
Packet 28 sent.
Packet 9 received.
Packet 29 sent.
Packet 30 sent.
Packet 10 received.
Packet 31 sent.
Packet 32 sent.
Packet 11 not received.
Packet 11 sent.
Packet 11 not received.
Packet 11 sent.
Packet 11 not received.
Packet 11 sent.
Packet 11 not received.
Packet 11 sent.
Packet 11 received.
Packet 12 not received.
Packet 12 sent.
Packet 12 received.
Timeout occured for packet 13.
Packet 13 sent.
Packet 13 received.
Packet 14 received.
Packet 15 received.
Packet 16 not received.
Packet 16 sent.
Packet 16 received.
Packet 17 not received.
Packet 17 sent.
Packet 17 received.
Packet 18 received.
Packet 19 received.
Packet 20 not received.
Packet 20 sent.
```

PROBLEMS OUTPUT TERMINAL JUPYTER COMMENTS DEBUG CONSOLE

```
Packet 15 received.
Packet 16 not received.
Packet 16 sent.
Packet 16 received.
Packet 17 not received.
Packet 17 sent.
Packet 17 received.
Packet 18 received.
Packet 19 received.
Packet 20 not received.
Packet 20 sent.
Packet 20 received.
Packet 21 not received.
Packet 21 sent.
Packet 21 received.
Packet 22 received.
Packet 23 received.
Packet 24 not received.
Packet 24 sent.
Packet 24 not received.
Packet 24 sent.
Packet 24 received.
Packet 25 not received.
Packet 25 sent.
Packet 25 received.
Packet 26 received.
Packet 27 received.
Packet 28 received.
Packet 29 not received.
Packet 29 sent.
Packet 29 not received.
Packet 29 sent.
Packet 29 not received.
Packet 29 sent.
Packet 29 received.
Packet 30 not received.
Packet 30 sent.
Packet 30 received.
Packet 31 not received.
Packet 31 sent.
Packet 31 not received.
Packet 31 sent.
Packet 31 received.
Packet 32 received.
Packet 33 received.
PS F:\SIT\5. SEM 5\CNL\Go Back to N>
```

Q. What are the key functions of error control techniques?

A. Ways of doing Error Control :

There are basically two ways of doing. Error control as given below :

Error Detection :

Error detection, as name suggests, simply means detection or identification of errors. These errors may cause due to noise or any other impairments during transmission from transmitter to the receiver, in communication system. It is class of technique for detecting garbled i.e. unclear and distorted data or message.

Error Correction :

Error correction, as name suggests, simply means correction or solving or fixing of errors. It simply means reconstruction and rehabilitation of original data that is error-free. But error correction method is very costly and is very hard.