Ramaiah Institute of Technology

Certificate

This report is submitted for the evaluation of Assignment component for the subject "DATA COMMUNICTION" with the subject code CS44 during the term Jan-May 2020.

Submitted by

1MS18CS066	**Manish M**
1MS18CS083	**Nikhil RV**

Signature of Faculty

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

RAMAIAH INSTITUTE OF TECHNOLOGY
(Autonomous Institute, Affiliated to VTU)
BANGALORE-560054
www.msrit.edu

Jan-May 2020

Problem Statement

Write a C/C++/Java program to implement Selective Repeat ARQ in noisy channel. The sender should send more than one data frames(within window size) and start timer. If it receives an acknowledgement, then it should stop the timer, move the sending window and send the next set of frame/frames. If the sender receives negative acknowledgement or if the timer times out for a particular frame, it should retransmit those frame/frames that it expects. If receiver receives frame out of order, it should buffer them and send a negative acknowledgement for the lost frame/frames.

Introduction

Selective Repeat Automatic Repeat Request/ Selective Repeat ARQ:

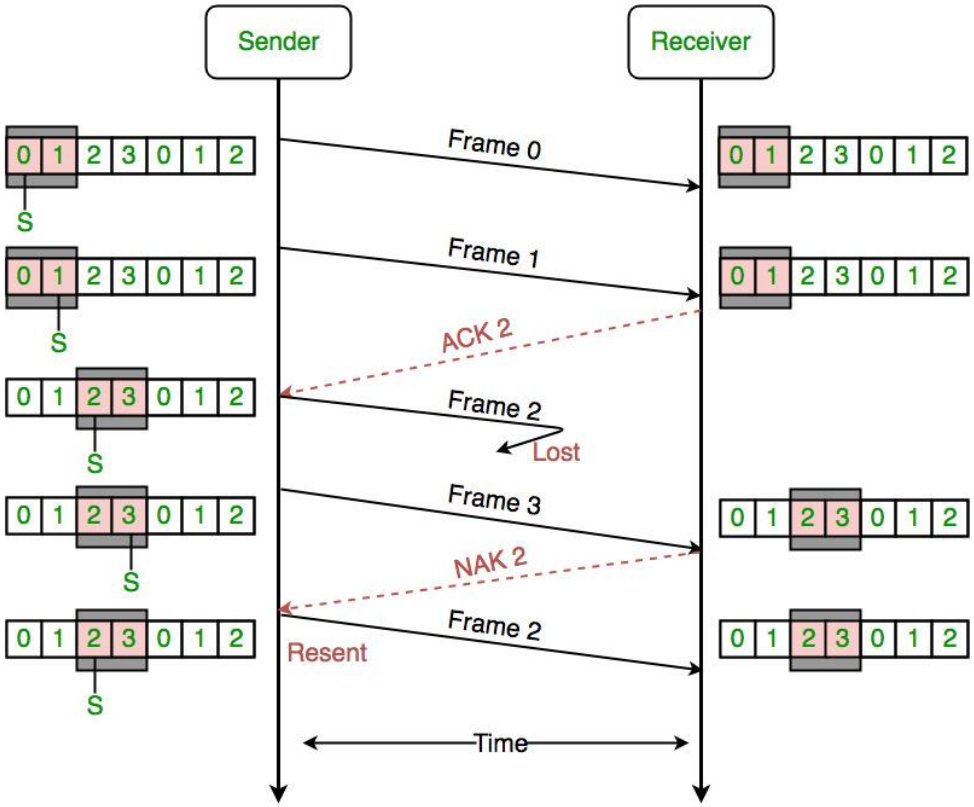
In a noisy link, a frame has a higher probability of damage, which means the resending of multiple frames. This resending uses up the bandwidth and slows down the transmission. For noisy links, there is a mechanism that does not resend N frames when just one frame is damaged; only the damaged frame is resent. This mechanism is called 'Selective Repeat ARQ'. It is more efficient for noisy links but the processing at the receiver is more complex.

Windows:

The Selective Repeat protocol uses two windows: a send window and a receive window. The size of the send window is 2^(m-1). For example if m = 4, then the sequence numbers go from 0 to 15 but the size of the window is just 8. The size of the receive window is the same as the size of the send window.

Sliding Window Protocol:

Selective Repeat ARQ uses 'Sliding window protocol'. A sliding window protocol is a feature of packet-based data transmission protocols. Sliding window protocols are used where reliable in-order delivery of packets is required, such as in the data link layer (OSI layer 2) as well as in the Transmission Control Protocol (TCP). They are also used to improve efficiency when the channel may include high latency. When the receiver receives the packets in order, it sends an 'ACK'(acknowledgement) signal but if it does not receive the packets in order, it sends a 'NAK'(negative acknowledgement).



Example for 'Selective Repeat ARQ' for a window size of 2

Code

```
/* Program in C++ to implement Selective Repeat ARQ in noisy channel
  Done by:
  Manish M - 1MS18CS066
  Nikhil RV - 1MS18CS083
*/
#include<iostream>
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<time.h>
#include<Windows.h>

using namespace::std;

void Send_frames(int,int);
int Receive();
void Resend_lost_frame(int);
void Resend_frames_due_to_timeout(int,int);

int main()
{
   srand(time(0));
   int i,window_size=8,number_of_frames,remaining_number_of_frames;
   bool timeout = false;

   cout<<"Enter the number of frames: ";
   cin>>number_of_frames;
   cout<<"Window size is : "<<window_size;
   cout<<"\n----------------------------------\n\n";
   remaining_number_of_frames = number_of_frames;

   int start_frame = 0;
   while (remaining_number_of_frames!=0)
   {
      timeout = false;
      if(remaining_number_of_frames < window_size) {
         window_size = remaining_number_of_frames;
      }
      cout<<"\nWindow is: "<<start_frame<<" to "<<start_frame+window_size;
      Send_frames(start_frame,window_size);

      receive:
      clock_t time_start = clock();
      int x = Receive();     //Receiver should receive the frames of the window
      clock_t time_end = clock();

      if((time_end - time_start)/CLOCKS_PER_SEC > 5)
      {
         if(!timeout) {
            cout<<"\n\tTimeout...\n"<<"\tResending window again";   //Timeout
```

```
            Resend_frames_due_to_timeout(start_frame,window_size);
          }
          timeout = true;
          goto receive;
       }
       else if(x == 0) {
          int frame_to_be_resent = rand()%(window_size) + start_frame;
          cout<<"\n\tNAK "<<frame_to_be_resent;       //Negative Acknowledgement
          cout<<"\nTime taken = "<<(double)(time_end - time_start)/CLOCKS_PER_SEC<<" secs";
          Resend_lost_frame(frame_to_be_resent);
          timeout = true;
          goto receive;
       }
       else {
          cout<<"\n\tACK";    //Acknowledged
          cout<<"\nTime taken = "<<(double)(time_end - time_start)/CLOCKS_PER_SEC<<" secs";
       }

       start_frame += 8;
       remaining_number_of_frames -= window_size;
    }

    return 0;

}

//Function to send the frames
void Send_frames(int start_frame,int window_size) {
    for(int i=start_frame;i<start_frame+window_size;i++) {
       cout<<"\nSending frame " <<i;
       Sleep(500);
    }
}

// Function that simulates the receiver
int Receive() {
    int n = rand()%6+1;     //Generate a random number between 1 and 6
    Sleep(n*1000);          //Sleep for 1,2,3,4,5 or 6 seconds
    return rand()%2;    //0 or 1 randomly where 1=>ACK and 0=>NAK
}

//Function to resend the frames due to timeout
void Resend_frames_due_to_timeout(int start_frame, int window_size) {
    cout<<"\nResending the frames "<<start_frame<<" to "<<start_frame+window_size;
    for(int i=start_frame;i<start_frame+window_size;i++){
       cout<<"\nSending frame " <<i;
       Sleep(500);
    }
}

//Function to resend the frame lost (NAK)
void Resend_lost_frame(int frame_number) {
```

```
    cout<<"\nResending the frame "<<frame_number;
    Sleep(1000);
}
```

Results Snapshots

1) Number of frames is 5

```
PS C:\Users\Manish.M\Desktop\C++> ./a.exe
Enter the number of frames: 5
Window size is : 8
-----------------------------------


Window is: 0 to 5
Sending frame 0
Sending frame 1
Sending frame 2
Sending frame 3
Sending frame 4
        ACK
Time taken = 4.006 secs
PS C:\Users\Manish.M\Desktop\C++>
```

2) Number of frames is 11

```
PS C:\Users\Manish.M\Desktop\C++> ./a.exe
Enter the number of frames: 11
Window size is : 8
-----------------------------------


Window is: 0 to 8
Sending frame 0
Sending frame 1
Sending frame 2
Sending frame 3
Sending frame 4
Sending frame 5
Sending frame 6
Sending frame 7
        NAK 4
Time taken = 5.001 secs
Resending the frame 4
        NAK 0
Time taken = 2.007 secs
Resending the frame 0
        ACK
Time taken = 5 secs
Window is: 8 to 11
Sending frame 8
Sending frame 9
Sending frame 10
        ACK
Time taken = 1.009 secs
PS C:\Users\Manish.M\Desktop\C++> ./a.exe
```

3) Number of frames is 16

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL                    2: powershell
PS C:\Users\Manish.M\Desktop\C++> ./a.exe
Enter the number of frames: 16
Window size is : 8
-----------------------------------


Window is: 0 to 8
Sending frame 0
Sending frame 1
Sending frame 2
Sending frame 3
Sending frame 4
Sending frame 5
Sending frame 6
Sending frame 7
        NAK 1
Time taken = 5 secs
Resending the frame 1
        ACK
Time taken = 1.002 secs
Window is: 8 to 16
Sending frame 8
Sending frame 9
Sending frame 10
Sending frame 11
Sending frame 12
Sending frame 13
Sending frame 14
Sending frame 15
        Timeout...
        Resending window again
Resending the frames 8 to 16
Sending frame 8
Sending frame 9
Sending frame 10
Sending frame 11
Sending frame 12
Sending frame 13
Sending frame 14
Sending frame 15
        ACK
Time taken = 3.007 secs
PS C:\Users\Manish.M\Desktop\C++>
```