|  |
| --- |
|  |
| DEPARTMENT OF COMPUTER ENGINEERING |
| CSL502Computer Network Laboratory |
| Fifth Semester, 2020-2021 (Odd Semester) |

**Name of Student : Neha Mahendra Yadav**

**Roll No. : 67**

**Division :TE-CMPN**

**Batch :B-3**

**Day / Session : Tuesday/Afternoon**

**Venue : Google Meet**

**Experiment No. : 07**

**Title of Experiment : Understanding and Analyzing Concept of Sliding Window Protocol**.

**Date of Conduction :**

**Date of Submission :**

|  |  |  |
| --- | --- | --- |
| Particulars | Max. Marks | Marks Obtained |
| Preparedness and Efforts(PE) | **3** |  |
| Knowledge of tools(KT) | **3** |  |
| Debugging and results(DR) | **3** |  |
| Documentation(DN) | **3** |  |
| Punctuality & Lab Ethics(PL) | **3** |  |
| Total | **15** |  |

**Grades – Meet Expectations (3 Marks), Moderate Expectations (2 Marks), Below Expectations (1 Mark)**

**Checked and Verified by**

**Name of Faculty : Prof. Rajesh Gaikwad**

**Signature :**

**Date : 4/08/2020**

**EXPERIMENT 7**

**Title:** Understanding and Analyzing Concept of Sliding Window Protocol. **Objectives:**

1) All modern communication systems rely on the correct operation of communication protocols that are running in different layers of protocol stacks.

2) Hence, a deep knowledge and understanding of them must be possessed by all communications engineers.

3) Therefore, a basic communication protocols course must be an essential constituent of any higher education communications engineering program.

4) Automatic repeat request (or ARQ for short) protocols that provide for reliable data transfer over noisy and loss channels and are usually used in data-link and transport layers of a protocol stack often form the core of such courses.

**Pre Concepts: 1) 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).

**2) C++ Basic Compilation and execution:** The C++ preprocessor copies the contents of the included header files into the source code file, generates macro code, and replaces symbolic constants defined using #define with their values. The expanded source code file produced by the C++ preprocessor is compiled into the assembly language for the platform. The assembler code generated by the compiler is assembled into the object code for the platform. The object code file generated by the assembler is linked together with the object code files for any library functions used to produce an executable file. After that we execute the program and output box is popped up where we can see output of are compiled Program.

**New Concepts:** The concepts we understood while analyzing Sliding Window Protocol are as follows:

**1)** Both sender and receiver maintains a finite buffer to hold incoming and outgoing packets from other side.

**2)** Every packet send by the sender, must be acknowledged by the receiver. The sender maintains a timer for every packet sent**.**

**3)** The sender may send whole window packets before receiving an acknowledgement for the first packet in the window.

**4)** This results in high transfer rate as the sender can send multiple packets**.**

**5)** The receiver advertises an window size that tells the sender how much data it can recieve , in order that sender do not fill buffer of the receiver.

**Program Code:**

The program for sliding window protocol using C++ is as follows:

#include<iostream>

using namespace std;

int main()

{

int w,i,f,frames[50];

cout<<"Enter window size: ";

cin>>w;

cout<<"\nEnter number of frames to transmit: ";

cin>>f;

cout<<"\nEnter "<<f<<" frames: ";

for(i=1;i<=f;i++)

cin>>frames[i];

cout<<"\nWith sliding window protocol the frames will be sent in the following manner (assuming no corruption of frames)\n\n";

cout<<"After sending "<<w<<" frames at each stage sender waits for acknowledgement sent by the receiver\n\n";

for(i=1;i<=f;i++)

{

if(i%w==0)

{

cout<<frames[i]<<"\n";

cout<<"Acknowledgement of above frames sent is received by sender\n\n"; }

else

cout<<frames[i]<<" ";

}

if(f%w!=0)

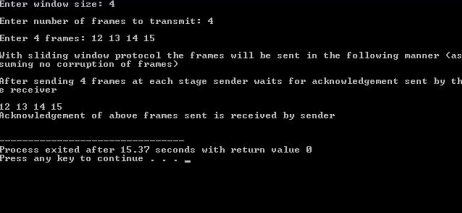
cout<<"\nAcknowledgement of above frames sent is received by sender\n";

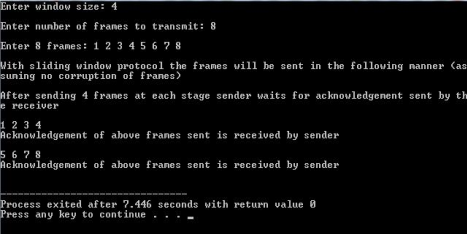
return 0;

}

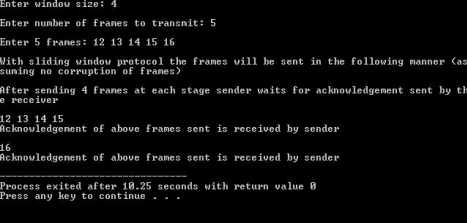
**Output :**

**Below are the output for Sliding Window Protocol using C++ Programing Language : 1)** Here we have implemented this Protocol for 4 Frames.

**2)** Here we have implemented this Protocol for 8 Frames :

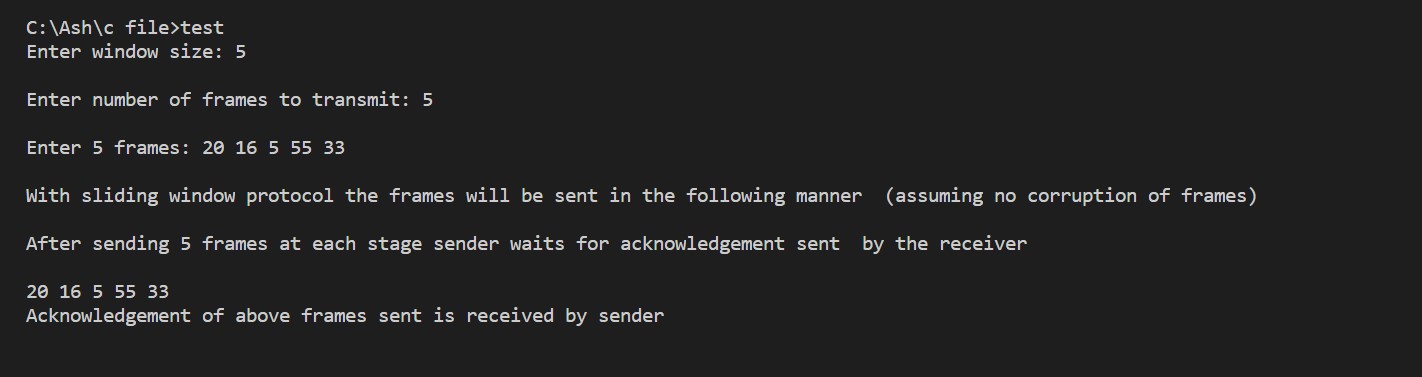


**3)** Here we have implemented this Protocol for 5 Frames:

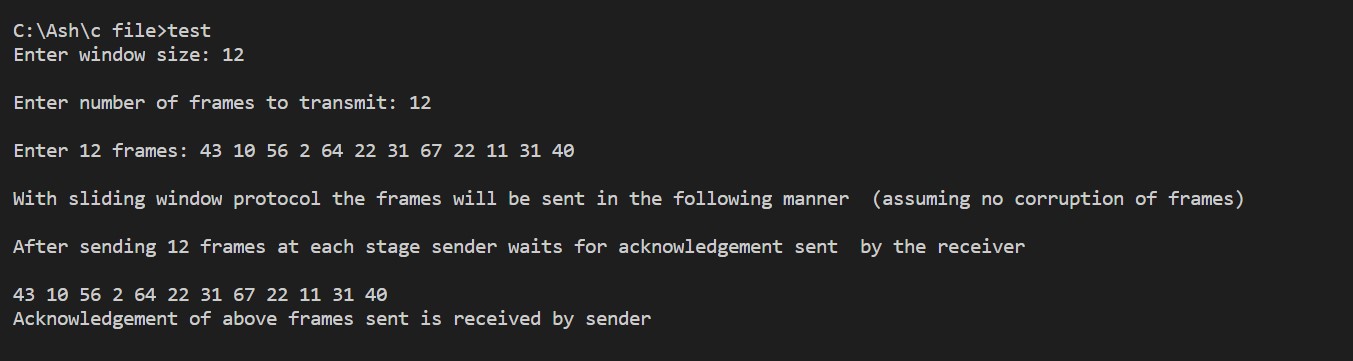


**My output:**

**1)** Here we have implemented this Protocol for 5 Frames:



**2)** Here we have implemented this Protocol for 12 Frames:



**Conclusion**: Thus we have implemented and analyzed concept of Sliding Window Protocol.