Stop & Wait Automatic Repeat ReQuest (ARQ) protocol

CSN-361 Networks Lab

Assignment - 1

Team Members:

Gagan Sharma (19114032)

R Chinmay (19114067)

Divyansh Agarwal (19115055)

Jitesh Jain (19114039)

Shlok Goyal (19114078)

Hardik Thami (19114035)

We have prepared the code for the given protocol and divided it into two classes and in their corresponding files i.e. Receiver and Sender classes. Header files are common so they are listed in protocol.h to be included once in both the files.

Protocol Simulation - In our program we are using socket programming functionalities to simulate certain operations happening in the protocol :-

*to_network_layer(packet *p) - the message is printed*

*from_network_layer(packet *p) - input taken from the user*

*from_physical_layer(frame *r) - recv()*

*to_physical_layer(frame *r) - send()*

For example - when the protocol dictates to send the frame to the physical layer for transmission we are using *send()* of sockets to send the frame to the other socket to be received.

Rest of the protocol operations like *inc()* , *start_timer()* , *stop_timer()* etc. are simulated in the program using c++ functionalities.

The protocol is tested to be correct when the Network Layer receives packets in the order in which they are sent without any repetition which is evident by our test cases.

Error Simulation - Our protocol should be robust enough to handle errors creeping into our frames during transmission so we are simulating errors with 10% probability during acknowledgment transmission and 20% probability during data transmission.

Commands to execute the program

1. *g++ Reciever.cpp -o Reciever*
2. *g++ Sender.cpp -o Sender*
3. *./Reciever*
4. *./Sender (in another terminal after the previous command)*

Test Cases images

1. image1 - Frames are sent by the sender and acknowledgements are received correctly until error occurs at the receiver side and the acknowledgment frame gets corrupted. Acknowledgement received by the sender is detected to be improper and the frame is re-sent. But the receiver is expected to receive the next frame so this frame is not sent to the Network Layer (not printed) but acknowledgement for the frame is still sent which is received correctly this time.
2. image2 - Frame is sent and correctly received but acknowledgement is not received which results in time-out at the sender. Sender resends the frame but receiver makes sure to not send the frame to Network Layer (not printed) as it is just a repetition of previous frames. Acknowledgment for this frame is sent but again not received by sender resulting in time-out and resending of the frame. The third time acknowledgement is correctly received.
   Note - Time-out event is simulated by altering the code of receiver to not send acknowledgments back for some trials. Else time-out would have never occurred cause both the programs are running on the same pc.