**ARQ**

Implemented stop and wait protocol for sender and receiver application. The sender tries to send a packet and waits for certain amount time till it receives ACK and if the ACK is not received, the sender tries to send the packet again. Using timers, we can calibrate the application and currently I have given timer of 2sec, which is very large for an application. But it can be reduced, and data can be retrieved faster at another end.

**Packet Format for ARQ**

1. File name send packet-

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | SRC | DST | PAYLOAD |

1. Data Packet –

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| TYPE | SRC | DST | SEQ\_N | EOF | PAYLOAD |

1. ACK packet-

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | SRC | DST | SEQ\_N |

**Type represents type of packet sent-**

Type- 0-Data,1-ACK,2-NACK,3-Control (sending filename and end of file (eof))- 4 Byte

SRC/DST – Address – 1 Byte

Seq\_n – Sequence number 4 byte

Payload – 1452 Bytes (68 packets for current data)

The efficiency is good with no loss in data transferred at the other end and at least one of the two programs exited successfully. Also, I have used diff command to check for any difference in the data received.

**Usage**:

g++ common.cpp senderv2.cpp -o sender -lpthread

g++ common.cpp receiverv2.cpp -o receiver -lpthread

./receiver 5001 localhost 5000 text2.dat (start first)

./sender localhost 5001 text2.dat

COMMENT: I couldn’t integrate the application with click router, will try to figure it out by this weekend and resubmit it later