




RELIABLE UDP PROTOCOL

PROJECT DELIVERABLE

Computer Networks

Submitted On:

14th December, 2014



Introduction:-

We are using selective repeat pipelined approach as performance is of paramount importance and buffer size is not an issue. In this approach we are sending packets over the network and if the ack for particular packet is not received till timeout then we are retransmitting that packet. So a timer for each packet is maintained and retransmission of only timed out packet is done. On receiver side there is a buffer which stores the packets in their correct position even if they come duplicate or out of order. This is done by using a list to store every packet received at its correct index, hence that list is populated accordingly and in the end that list is written into a file.

Techniques and Methodology used to combat the relevant problems in the network:-

We encountered following problems in network and how we encountered them is also explained.

Packet Drops:

- We used Seq and Ack number to take account of packets received and dropped.
- Timer is for each packet as in selective repeat.
- So when a timer for a particular packet is out only that packet is retransmitted.

Bit Errors:

- It was caused by badnet1 in our file data.
- We used CRC16 to check for errors in packet to care for bit errors.
- We used CRC16 for performance and reliability issues.

Packet Duplication:

- We used Seq and Ack number to assign identity to packets.
- So when a duplicate packet received is identified by seq number.
- The receiver Buffer at receiver only stores those packets which are already not received.

Packet Disordering:

- Sequence number is used to assign identity to packets.
- When an out of order packet is received, it is simply written at its adequate position in the list data structure.
- The ack is still sent for disordered packet as selective repeat is used and out of order packets are acked.

Retransmission:

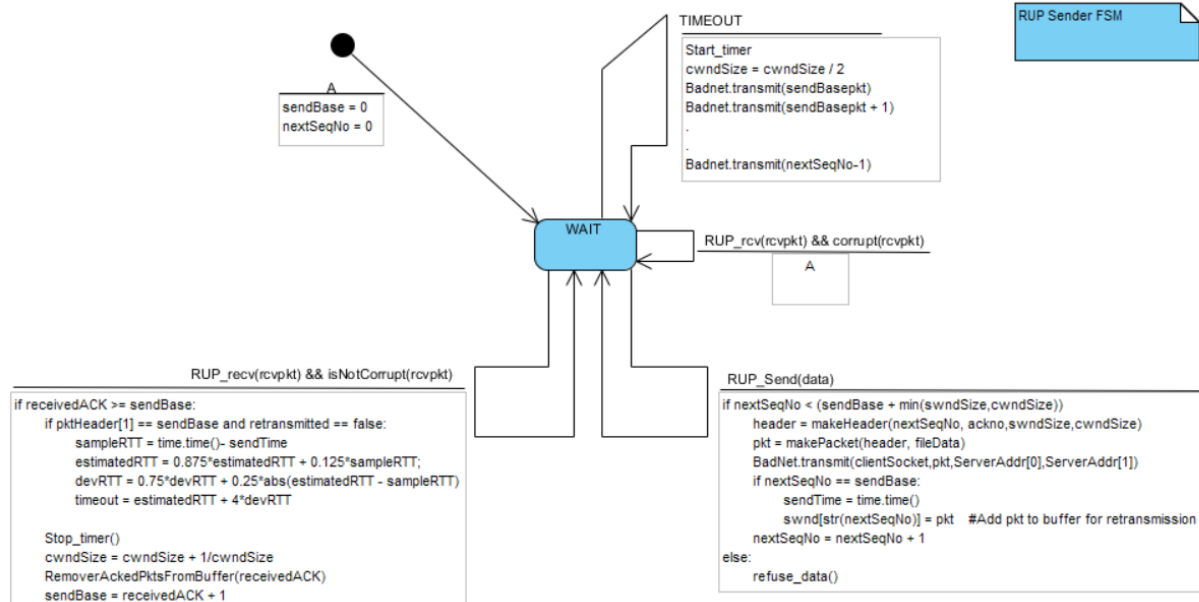
- It was caused when an ack for a packet was not received in time.
- So whenever a timer for the packet is timed out, it is retransmitted.

Performance:

- If the sender keeps waiting for ack of each packet and sends next one, when ack is received it takes too long, hence stop and wait is not used.
- We used pipelining to solve this issue, we used non-blocking receive to achieve this, to wait for ack in non-blocking state.
- If acks for all packets haven't received then waiting for ack is in non-blocking state.
- Use of selective repeat makes sure that packets are sent in a burst and only those for which timer is out are retransmitted.
- The file takes approximately 5 secs to transfer.

Finite State Machine Diagrams:

RUP Sender FSM:



RUP Receiver FSM:

