

This application supports the following rdt features:

- 1) Packet loss detection
- 2) Acknowledgment
- 3) Packet retransmission
- 4) Flow control
- 5) checksum

Brief description of “our udp-ftp”:

- We used multiple threads to send the data to the receiver (server).
- We used worker threads to send data and the main thread to receive the acknowledgment sent by the receiver and the main thread notifies the corresponding worker thread so that the worker thread can proceed further.
- Each thread sends the data assigned to it irrespective of other threads.

- For example,

If t = no. of threads, then

A thread with id “ i ” sends the packets $i, i+t, i+2t, \dots$

And each thread waits for acknowledgment for the packet it has sent and then it will continue to send the other packets assigned to it.

- If the thread did not receive acknowledgment for a particular period of time (timeout), then it will resend the previous packet (packet retransmission).
- If the packet has been lost then the server will not send any acknowledgment, so the worker thread will wait and a timeout happens. So it retransmits the packet to the server. Hence Packet loss is detected.

- We have modified the ack seq no in such a way that the main thread

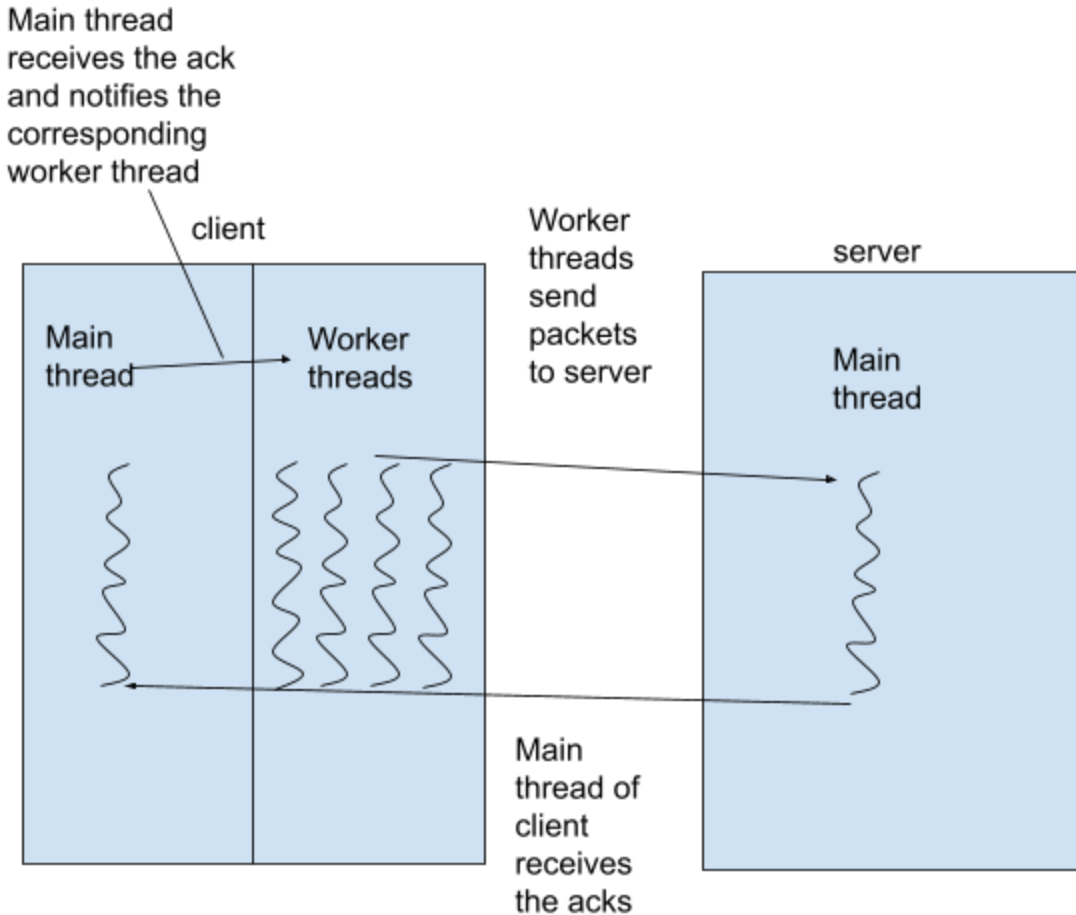
can identify the thread to which ack corresponds to.

$\text{ack_no} = (\text{thread_id} * 10) + (0 \text{ or } 1)$

- checksum: We also added checksum while sending the packet to the server. And at the client-side, it will calculate the checksum for the data received and also checks if both are the same or not.

If yes, it sends the ack, or else it will ignore.

- Flow Control: 1st packet sent by the thread "0" contains the total no. of packets that the client wants to send to the server. After receiving this packet, the server dynamically allocates the memory for the buffer of size total no. of packets. As the server has a buffer of the required size, Flow control has been addressed.



- The receiver receives the data and sends the acknowledgment for the same.
- It also handles duplicate packets (since the packet contains packet no.)
- It stores data of every packet in the buffer until all the packets are received. And finally, it writes the data to the output file.

Application Header:

ack_no	packet_no	checksum	Payload
--------	-----------	----------	---------