Point-to-Multipoint File Transfer Protocol (P2MP-FTP)

Arunkumar Krishnamoorthy (akrish12 – 200194445)

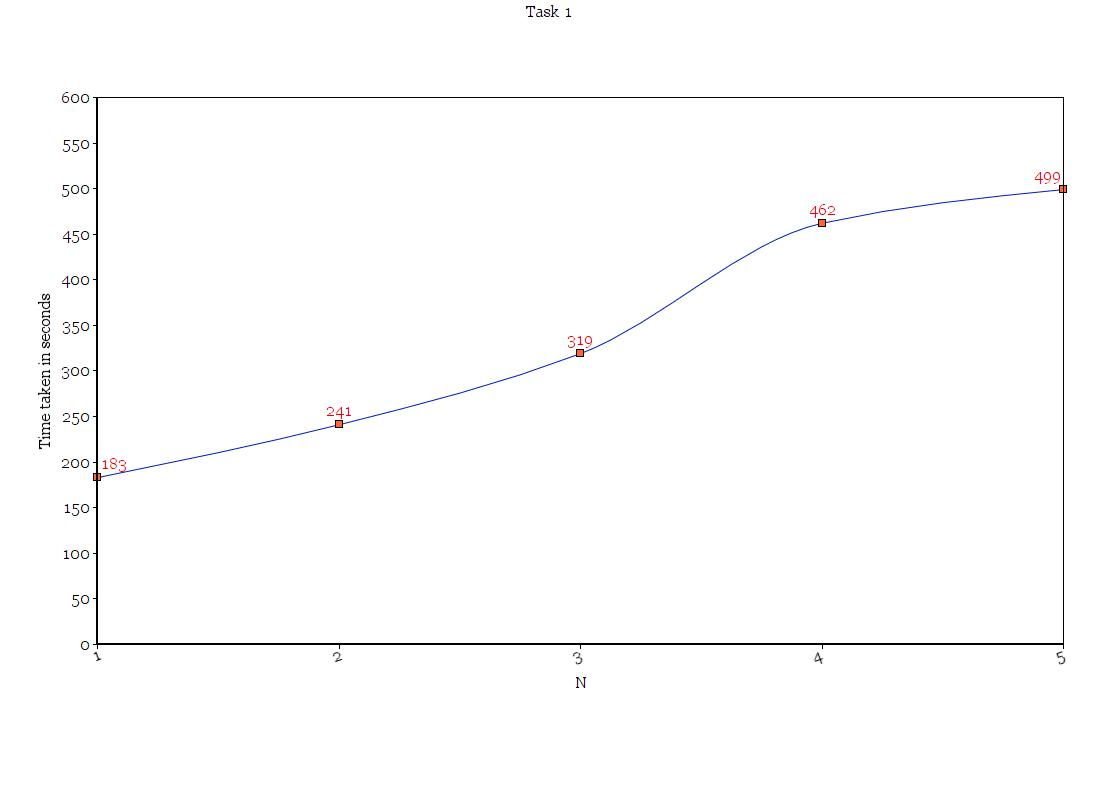
Harish Annamalai Palaniappan (hpalani – 200200275)

Reliable data transfer over unreliable data transfer protocol (User Datagram Protocol) is the main feature of the project. Point to multipoint reliable data transfer is done using the Stop-and-Wait automatic repeat request (ARQ) scheme.

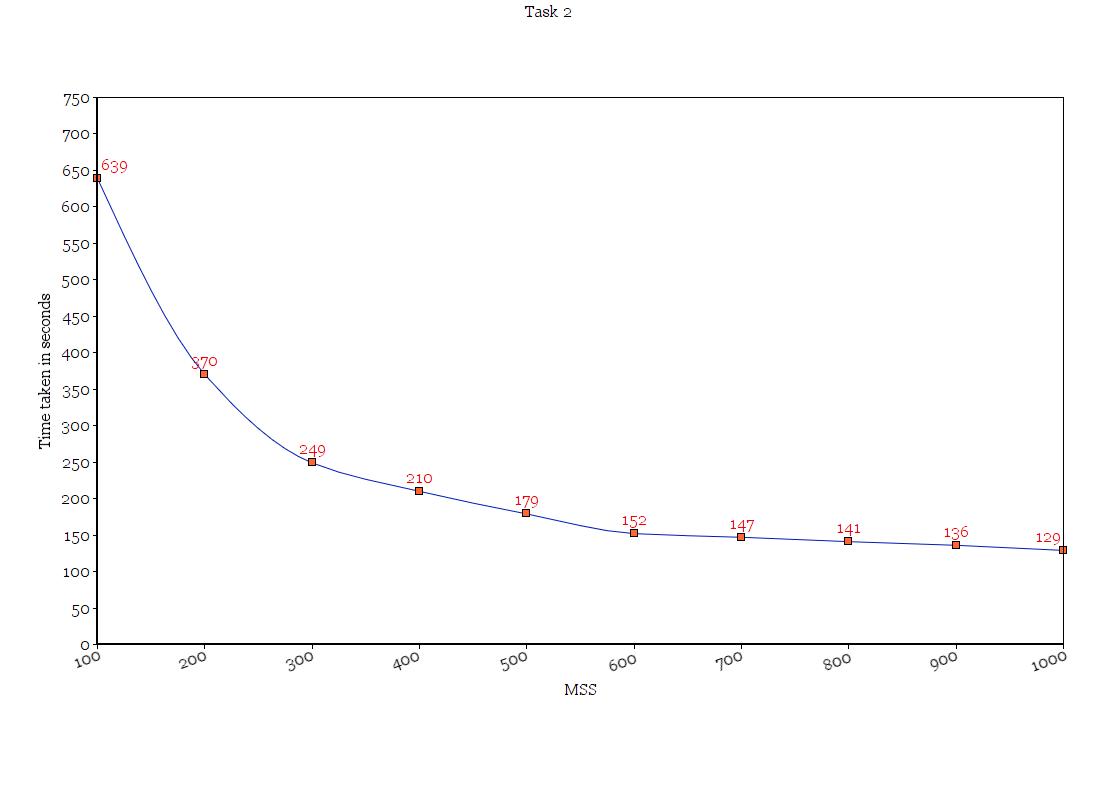
The client sends the file to multiple servers. The client buffers the data that it reads from the file until it reaches the MSS value, and then it’s header field (32-bit sequence number, 16-bit checksum and 16-bit field indicating that it’s a data packet) is appended to the data and sent to the servers using UDP. Since UDP is unreliable, to make it reliable, we’re starting a timer, which waits for the ACK from all the servers. If the timer expires before receiving ACK from all the servers, then client retransmits the segment only to the servers from which it has not received the ACK. Once all the ACKs are received, the client increases the segment number and transmits the next data segment and this process gets repeated. When a server receives a packet, it checks if the client has sent the segment which the server is expecting. If it has received the expected segment, then it computes the checksum on the received data and compares this value to the checksum value in the header field. If there is a mismatch in the sequence number, then the server sends an ACK for the last in order segment. If there are no errors, it sends an ACK to the client (if the randomly generated number > error probability p). If there are errors, it does nothing. The RTT value is determined by performing traceroute to one of the servers ( which was 10ms, on average), hence the timeout value was set to 5\*RTT=0.05 seconds.

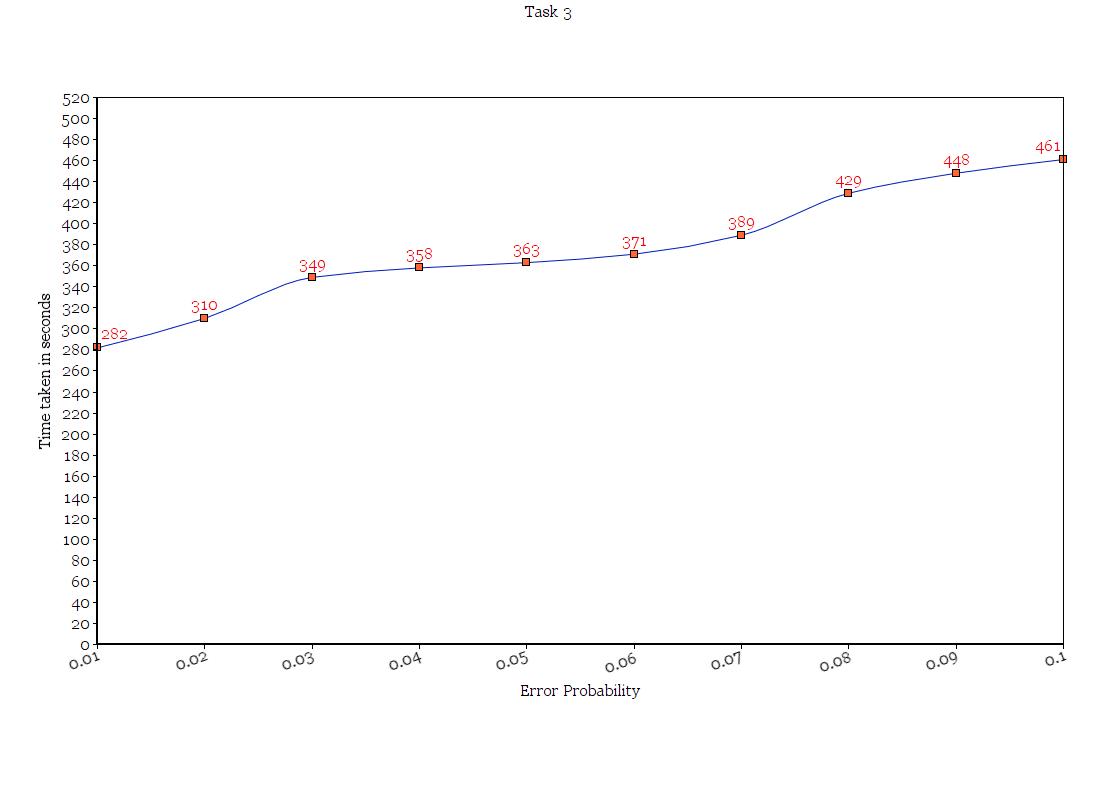
**TASK 1:**

The maximum segment size is set to 500 bytes and the error probability p is set as 0.05. This is done to find the effect of the different number of receivers. Timeout value is 0.05 seconds. When the number of receivers increase, the average time taken also increases, this is because the client waits for an ACK from all the receivers before transmitting the next segment.



**TASK 2:** The number of receivers is set to 3 and the loss probability p is set to 0.05. The effect of MSS is studied in this task. Each file is transmitted 5 times and the average value is calculated. When the MSS increases, the average time take decreases but for higher order MSS, the time taken remains almost constant because the time taken to process the packet increases with MSS.

**TASK 3:** The MSS value is set to 500 bytes and the number of receivers is set to 3. The effect of loss probability p is studied varying the value from 0.01 to 0.10. For each value of p, the file is transmitted 5 times. As the error probability increases, the time taken also increases because higher error probability means there are more chances of an error occurring and hence more retransmissions, which takes more time.



**Conclusion:**

Though we have added reliability over UDP, this project is not efficient because it has to wait for ACKs from each of the servers. Servers that send ACKs properly will have to wait for more time until the client receives the ACKs from all the servers.