Design Document – LAB2 - 14740

FTP Server & Client:

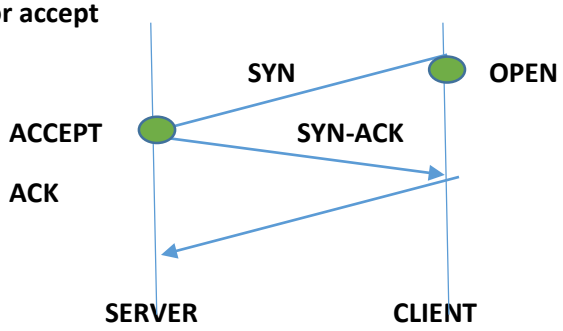
FTP Server takes the following arguments

ServerIP ServerPort Windowsize and timeout interval
The server uses the receiver method provided by tttpService to listen to the incoming Packets

The Client uses the following parameters
<FileName> <ClientIP> <ClientPort> <ServerIP> <ServerPort> <WindowSize> <Interval>

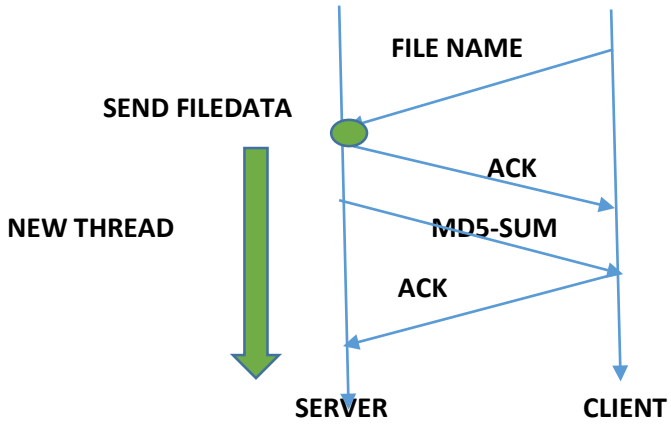
TTP Hand Shake:

Server would be listening on its port for accept



The Client then creates a new datagram service to send the Filename to the FTP Server. Each Data Transfer is handled in a separate Thread by the FTP Server.

FILE- Request and Transfer



The FTP Server sends the MD5-SUM if the file exits.

The FTP Server, sends the data to Client in DATA Packets each with a max payload of 1250 bytes. It sets the first byte(of the entire file content. Not every packet) to the Transferred data to 1, if the data is MD5 sum, otherwise, the first byte is set to 2, and these bytes are removed by the client during reassembly.

The FTP Client receives the MD5-sum and then the actual file data, it performs the reassembly and computes the MD5-SUM and writes the file data to an output file.

TTP Service:

The TTP Service provides the following APIs to the Server and Client

Receiver : Recieves the incoming client connections, sends the server a String int the following format

IP:PORT:FILENAME

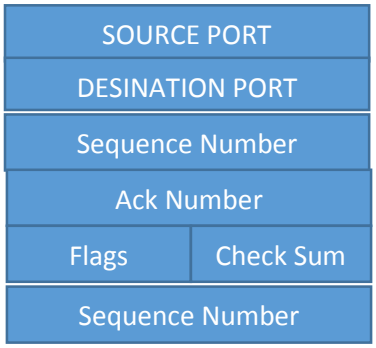
Sender: Sends the File data to the Client, specified by the Server

Each TTP Connection is handled by a unique Connection Object called TTP Connection which is used for communication between the server and the client. Once a new Client is accepted, A unique connection object is created by the TTP Service and added to the Hash map of existing Connections

It maintains the following parameters

- **Source: Source IP, Source Port,**
- **Destination: Destination IP, Destination Port**
- **expectedAckNumber ,currentSequenceNumber, nextSequenceNumber, expectedSequenceNumber, baseSeqNumber;**
- **Maximum Payload ; timer; datagramService;**
- **Data Gram Lists:**
- **sentDataGramList;**
- **unAckDataGramList;**
- **sendQueue;**

TTP Segment:



Design Document – LAB2 - 14740

Flags:

The following flags are used by the TTP Service in TTP segment

SYN = 1 ACK = 2;SYNACK = 3;FIN = 4;DATA = 5;EOF = 6;FILE = 7;FILENAME = 8; SYNACKACK = 9;FINACK = 10;FINACKACK = 11;

Go –Back – N:

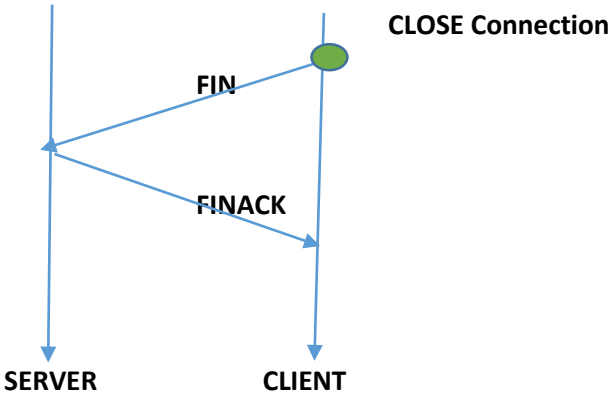
1. **The server maintains a base sequence number for the packets being sent, which is essentially the start of the window.**
2. **SIZE of Window is the default window size received.**
3. **If(Seq Num of Next Packet < (base Sequence + SIZE of WINDOW)) -> send the next packet(s)**
4. **Start the timer with connection timeout**
5. **As soon the in-odered acknowledgement is received, remove the acknowledged packet from the un-ack list. Increase your base sequence number (advance the window). Send the the new packets.**
6. **If a timeout occurs, resend the un-acknowledged packets.**
7. **If there is no more space in the window, add the packets to send buffer and once a new packet is acknowledged, send them.**
8. **The packets are discarded, if they arrive out of order or with incorrect Checksum.**

Reassembly:

The TTP Service, using the TTP conn object, the segment with EOF, indicates the End of file data. Once the data is received in total, it is send back to the TTP Client and it computes and md5 –sum on it.

If the md5-sum is wrong, the file is discarded.

Connection Termination:



Once the Client receives the file with correct MD5-sum, it sends a FIN. The server responds with a FINACK and removes the client connection object for that client. Once the client receives a file, it terminates. Even if the Client does not receive the FINACK from the server after the specified timeout.