Application layer

FTP server
control: map<port number, TTPService>

FTP client

MD5 verified, put the file from temp file into client folder

Mediator between FTP and TTP

TTPService

Re-assemble file, check MD5

TTP layer

Server
[TTP server sender, TTP server receiver]

Client
[TTP client sender, TTP client receiver]

SYN

SYN_ACK

SYN_ACK

Connection established phase

Filename

MD5

ACK

content part 1

content part 2

content part 3

ACK

ACK

ACK

content part 4

content part 5

content part 6

File transfer phase

[details omitted]

EOF

FIN

server close

FIN_ACK

client close

Close phase

Client:
1. First client and server connection established: with 3-way handshake:
   Host A sends a SYN packet to Host B
   Host B receives A's SYN
   Host B sends a SYN-ACK
   Host A receives B's SYN-ACK
   Host A sends SYN-ACK
   Host B receives SYN-ACK.
   740ftp socket connection is ESTABLISHED.
2. Client sent request with filename
3. Client receive the fragments transferred by server
4. Client receive the EOF and send the FIN to server
5. Client close the connection when receive the Fin-Ack
6. Client reassemble the fragment into file
7. Client verified the MD5 file
8. Store the file into local directory of the Client

Server:
1. Server and client connection established with 3-way handshake
2. Server receive filename from Client
3. Server send the file fragments to the Client with go-back N strategy
4. Server send the EOF to the client
5. Server receive the FIN from client and send the FIN-ACK and then close the server

Note:
1. Sequence number is the previous sequence number plus the sending fragment size
2. Time listener:
   Timer is initialized with the user defined time-out value. An action listener is attached and is registered with corresponding sender. When timeout happens, the action listener is triggered and will retransmit the whole window's sent-not-acked segments.
3. Go-back N strategy:
   Every time we send one fragment from the server and start the timer. When we receive an ACK number from the client or time is out, we would enter the go-back-N function. In the go-back-N function, we use two queues. One queue is for buffering the un-sending fragment and one queue is for storing the not acknowledged sending fragment (this is the window). When the Ack-Number is get from the server, we will delete this acknowledged fragment. At the mean time if the window is not full, we will add more fragment from the un-sending buffer to the storing the not acknowledged buffer. Lastly, we re-send all the sub-sequent fragments in a new server thread.

4. Util class has all the function that is used by more than one function: Flag, getSegmentSize, checksum, getMd5.
5. Payload: we create a TTP-segment and in the segment we have its corresponding sequence-number, Ack-Number and checksum.

6. Checksum: when the server send the fragment, it will calculate this fragment checksum and when it is received by the client, we will re-calculate the checksum and verified that the fragment is intact when received.