Server
Write request
from Client

from Client

DS port#5000   TFTP Server   TFTP service   TFTP Service DS

D1   S1   S2 S3 S4   D2

D3

S5

D4

Client Side
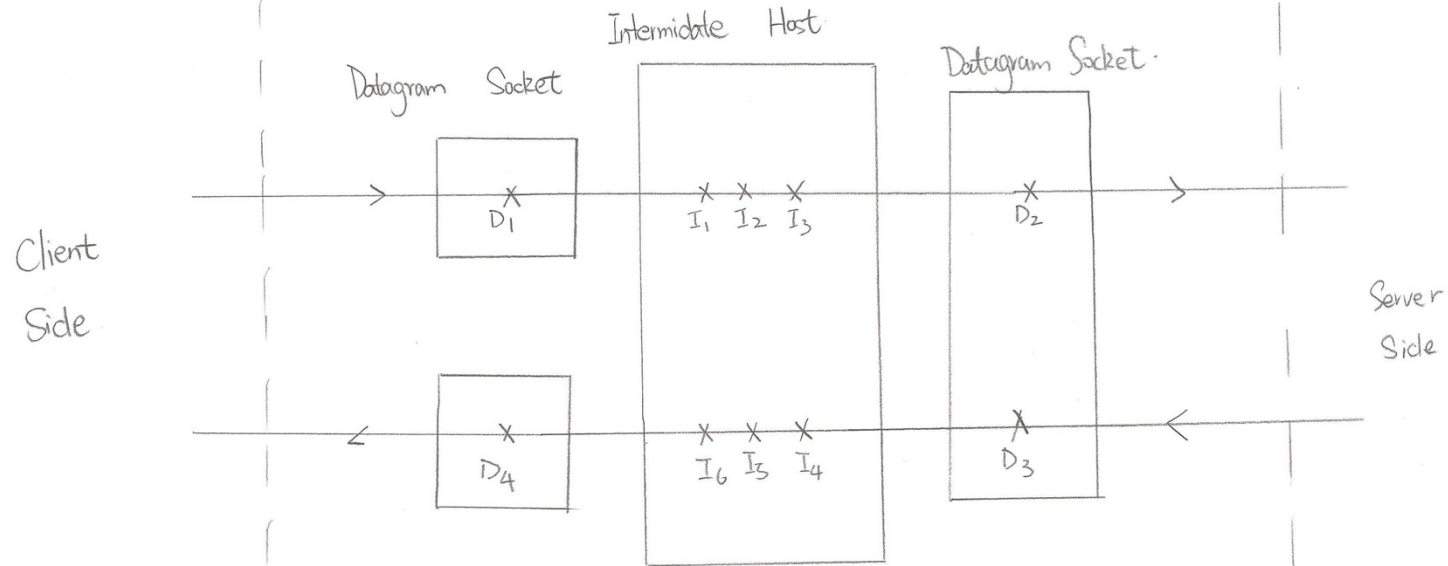
← To Client

from Client

← To Client

NOTE:

* DS: Datagram Socket

* This diagram depicts the UCM when the request at S2 is a write request

* D1 & S1 repeat until the admin control gives a shutdown signal

* D3, S5 and D4 repeat until all data packets from client is received

D1: receives request from Client

D2: send datagram

D3: receive datagram containing data packets to be put in the file

D4: send datagram

S1: creates a service thread

S2: creates ReadWrite Packet Packet Builder and gets its request type

S3: creates Write Packet Builder

S4: create datagram packet with acknowledgment

S5: create datagram containing acknowledgement that data packet is given

DS port #5000

TFTP Server

TFTP Service

TFTPservice DS

from client

D1

S1

S2 S3 S4 S5

D2

D3

To Client

From Client

NOTE!
* DS: Datagram
    socket
*This datagram
depicts the UCM
when the request
at S2 is a read
request.
* D1 & S1 repeat
until the admim
council gives a
shutdown signal

* S4, S5 & D3
  repeat until
  all packets from
  the client are
  acknowledged.

D1: receives request from client
D2: send datagram packet
D3: receive datagram packet
    containing Ack packets from the
    client.

S1: creates a service thread
S2: creates a ReadWritePacket PacketBuilder and
    gets its request type
S3: creates a read packet builder
S4: create data packet from last packet received
S5: create datagram packet containing data packet

Intermidate Host
VCM

(Error Simulator)

Intermidate Host

Datagram Socket

Datagram Socket.

Client
Side

Server
Side

D₁

I₁ I₂ I₃

D₂

D₄

I₆ I₅ I₄

D₃

D_1: Intermidate Host opens a socket listening and get the Packet from client.

D_2: create socket and Send Packet.

D_3: receive datagram

D_4: create a new Socket and send datagram to client.
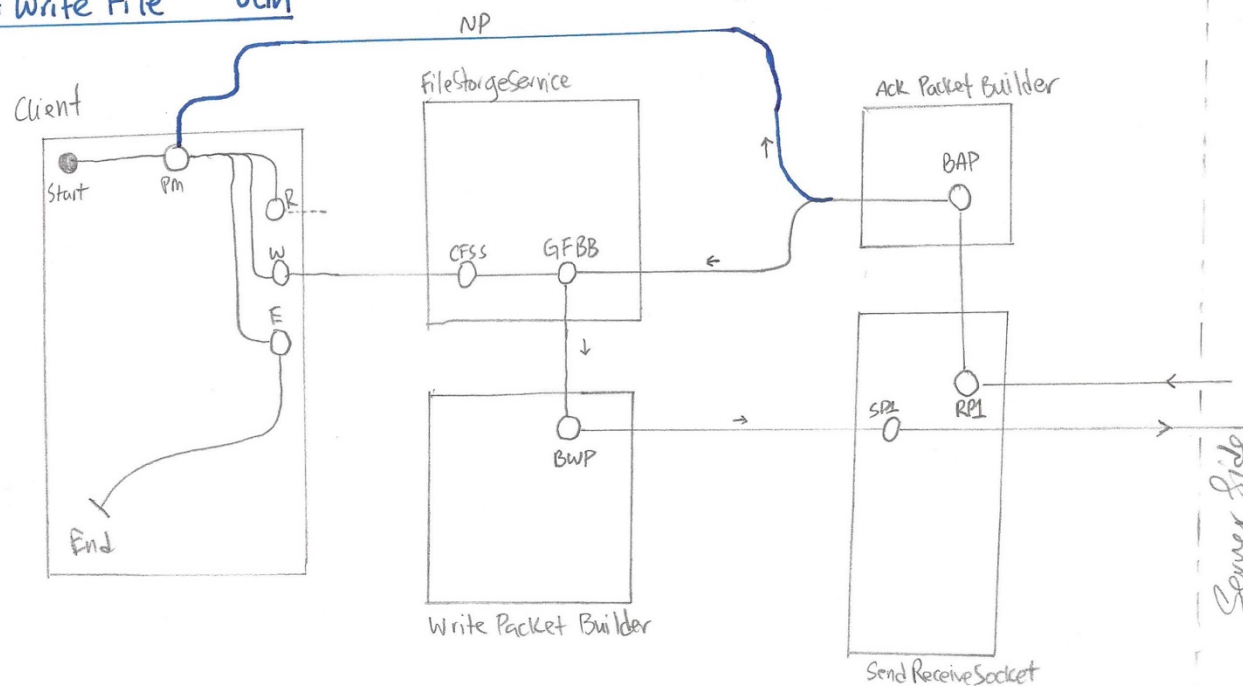
I_1: receive client's Packet

I_2: extract info.

I_3: create a new Packet put info from client's Packet into the new Packet and Send Packet

I_4: receive response from Server

I_5: extract info from received Packet.

I_6: create a new packet and put info into the Packet then send it.

Client Write File   UCM

Client

Start — run Client Main

PM — Print Menu Options

R — client Select Read
W — client Select Write
E — client Select Exit

CFSS — Create File Storage Service
     using file Name provided

GFBB — Get File Byte Buffer

BWP — Build Write Packet using
     The Byte Buffer

SP1 — Send Packet from
     Write Packet Builder
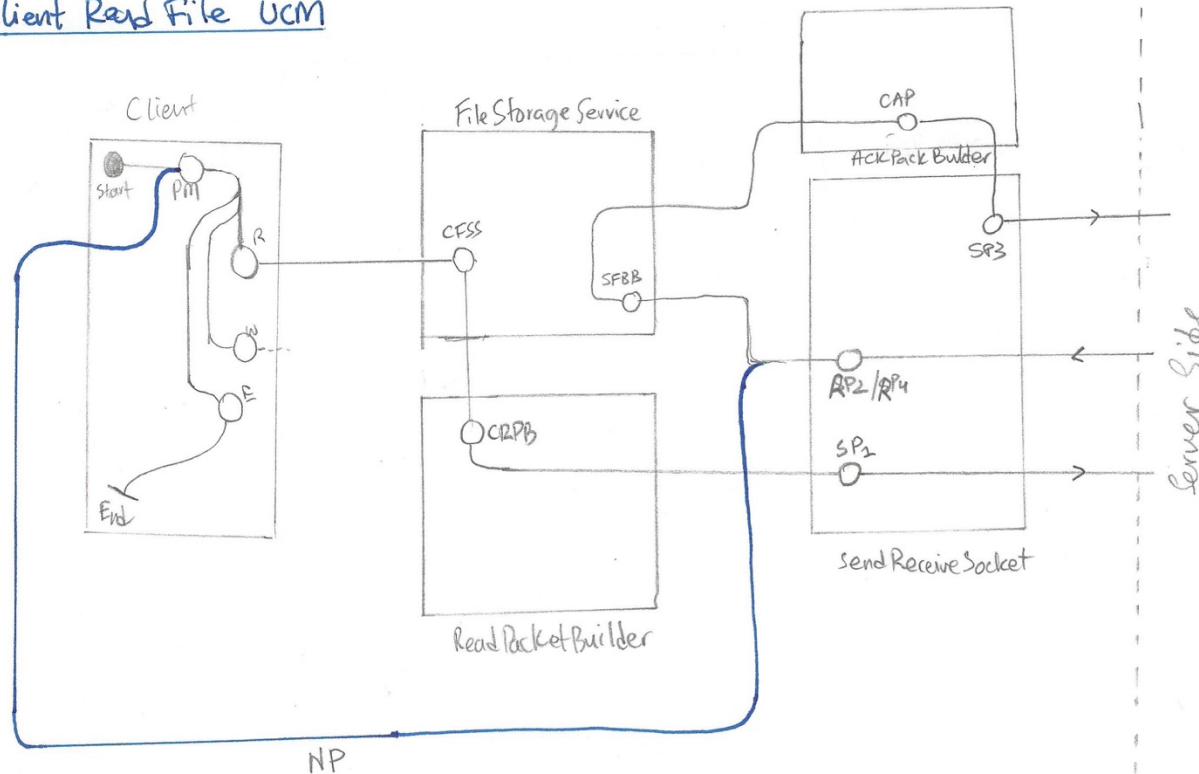
RP1 — receive ACK Packet

BAP — Build ACK Packet
     using ACK Packet Builder

→ loop back to (GFBB)

NP — new path if no more
     new Bytes to write

End

FileStorageService

Ack Packet Builder

Write Packet Builder

Send Receive Socket

Server Side

NP

BAP

CFSS   GFBB

BWP

SP1   RP1

Start   PM   R   W   E   End

Client Read File UCM

Client

Start   PM   R   W   E   End

File Storage Service

CFSS   SF8B   CRPB

Read Packet Builder

CAP   ACK Pack Builder   SP3

RP2/RP4   SP1

Send Receive Socket

Server Side

NP

Start — run client Main

PM — Print Menu Options

R — Client selects Read File & provid FileName

W — Client selects WriteFile

E — Client Ends Program

CFSS — create fileStorage Service Using provided FileName

CRPB — Create Read Packet builder

SP1 — send packet
RP2 — receive Packet
SFBB — Save File Byte Buffer to Disk
CAP — create ACK Packet
SP3 — send ACK Packet
RP4 — receive Packet
NP — New Path (in case File read was complete)

End