**Computer Networks**

**HW#5 – Download on Demand System**

2014313366 Hong Giwon(홍기원)

* **Introduce the application**

Download on Demand system is the system which has DoD server application. Similar to proxy server, DoD server gets request from client and serves client if it has the requested file in DoD server. If the file does not exist in DoD server, DoD server download that file from file servers.

The purpose of this DoD system is to reduce number of connections to file server when transferring files and makes it easier for client to download files. With DoD system, if the file is in DoD system, there is no need to connect file server. So if the file was demanded earlier and stored in DoD server, there is no need to connect file server again when clients demand that file. And also, clients don’t have to care about where the file is since DoD server will handle it.

I implement this system by using socket programming and multithreading. Through multithreading, multiple clients could be served concurrently. Therefore, there is no need for time scheduling. (Since all requests from clients will be handled right away)

* **Development environment**

**OS: Ubuntu 14.04.3 LTS**

**Complier: gcc 4.8.4**

Complie/build

**gcc –o fileserver fileserver.c**

**gcc –o fileserver2 fileserver2.c**

**gcc –o dodserver dodserver.c –lpthread**

**gcc –o client client.c**

RUN

**-Create and fill up files that are listed in source code. (They are included in each file server folder in executable files folder)**

(File list: "file01.dat", "file02.dat", "file03.dat", "txt01.txt","code01.txt","file01.dat","file02.dat",

"file03.dat", "txt01.txt", "code02.txt")

**./fileserver**

**./fileserver2**

**./dodserver**

**-Enter IP addresses of file servers.**

**./client (repeat 3 times)**

**-Enter IP address of DoD server.**

* **Structure**

There are four source codes, which are fileserver.c, fileserver2.c, dodserver.c, and client.c. fileserver.c and fileserver2.c are the source codes for file servers. The only difference between them is port number. dodserver.c is the source code for DoD server, and client.c is the source code for client.

There are also 5 files in each fileserver folder. These are the files stored by file server. (I assumed that there are maximal 5 files in each file server.)

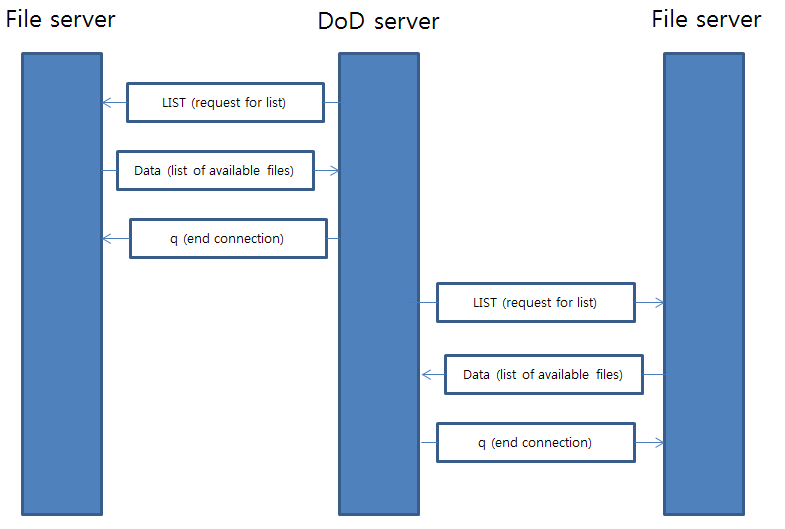
* **Extra non-standard applications/components**

To implement this application, socket programming and file operations are used. Therefore several header files and variables related to socket programming and file operations are used. I will skip details since they were explained in previous assignments.

Also, time.h header file is used to check current time. When current time is midnight, dod server will perform refreshing list.

And pthread.h header file is used to implement multithreading. I use functions and variables of pthread.h such as pthread\_t, pthread\_create, pthread\_cancel, and pthread\_join for threads. And I also use variable pthread\_mutext\_t and functions thread\_mutex\_init, pthread\_mutex\_lock and pthread\_mutex\_unlock for synchronization between threads.

* **Present the designed protocol**

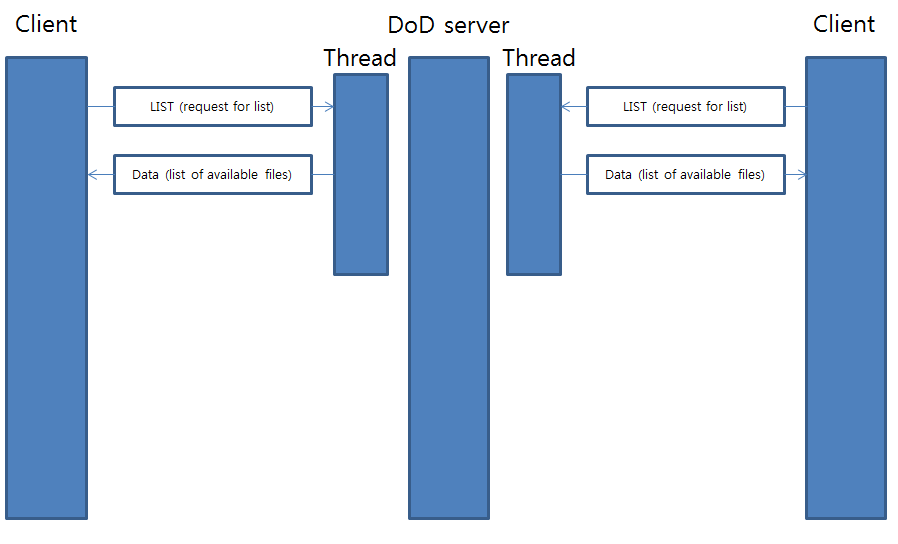


(Initialize/refresh list)

When DoD server is executed, DoD server require to enter the IP addresses of file servers. If IP addresses are entered, DoD server connects all of those file servers and send LIST command to get the list of available files. (This also happens when refreshing list)

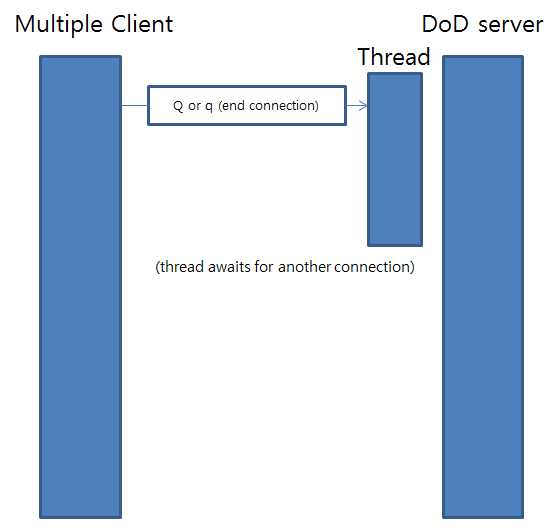
After that, DoD server awaits for clients to connect.

When clients are connected, DoD server sends available commands (LIST, GET, q or Q) to clients.



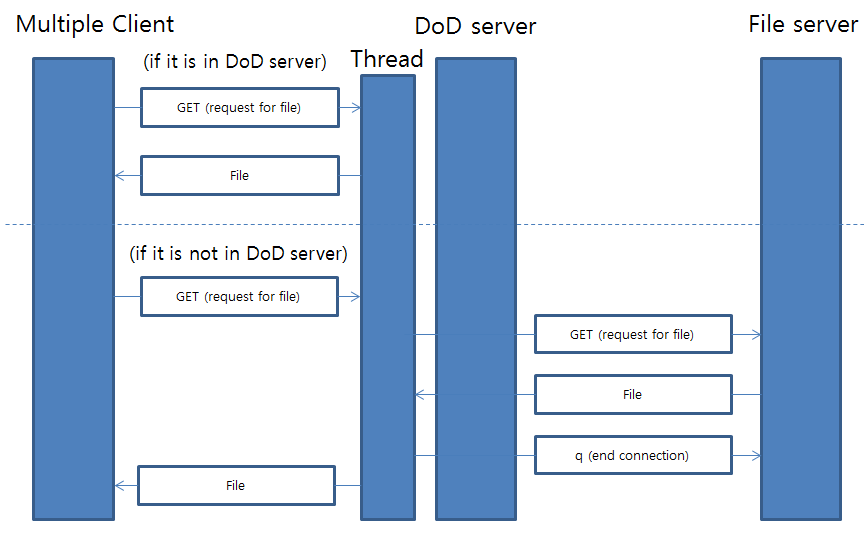
(Client sends LIST command)

1) If clients send LIST command, DoD server sends list of available files to clients and awaits for next command.



(Client sends q or Q command)

2) If clients send q or Q command, DoD server ends connection with client.



(Client sends GET command)

3) If clients send GET command, DoD server first checks whether the requested file is in DoD server or not.

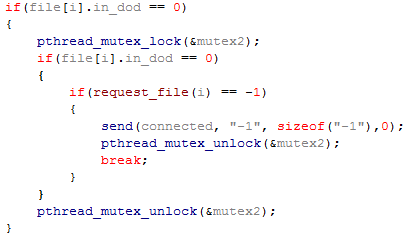
3-1) If the file is in DoD server, DoD server sends that file to client and awaits for next command.

3-2) If there is no such file, DoD server sends request to file server with using GET command and download that file to store in DoD server. And then DoD server sends that file to client and awaits for next command.

* **Implementation of functionalities**

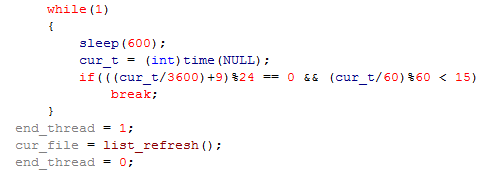
File server and client source codes are similar to those in previous assignment.

DoD server source code has three functions. First is list\_refresh(), which initialize or refresh files list. Second is request\_file(). This function is used when the requested file is not in DoD server. This function sends request for that file to file server and store in DoD server. Third is serverthread(), which implement threads for connecting clients.



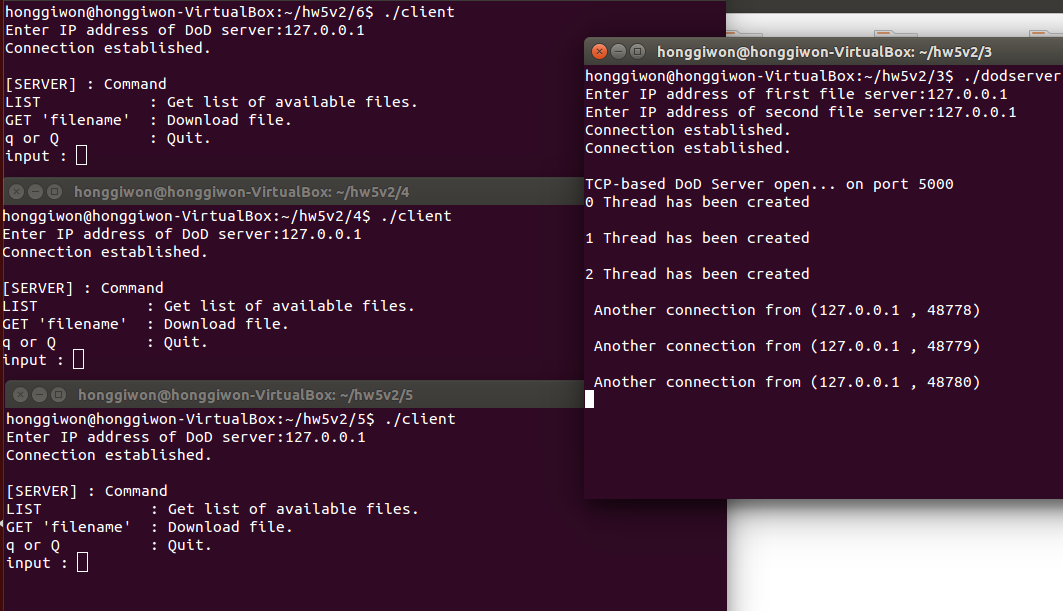
Above code is important piece of serverthread() function. This section of code is about when the requested file is not in DoD server. When file’s in\_dod is 0 (which means file is not in DoD server), thread creates lock using pthread\_mutex\_lock. The reason for this is so that when a file is written by request\_file() function, multiple threads do not write to the same file at the same time. And in such a situation (when multiple thread try to write to the same file), if one thread finishes writing the file and unlocks, other threads don’t have to request and write that file again. Therefore we should check it again whether file’s in\_dod is 0 or not. (Second if statement)

Also, I implement the refreshing functionality.

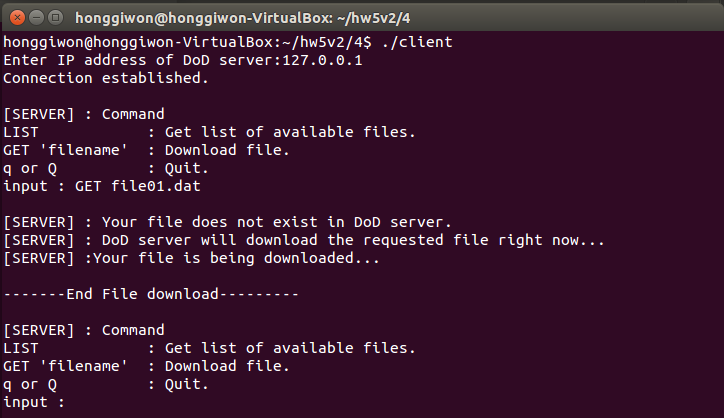


Current time can be gain using time.h header file. If current time is about midnight, DoD server starts to refresh available file list. Since this will change cur\_file, sections of code in thread function that use cur\_file must be stopped. To do that, I use another variable end\_thread that stop some sections of code. (If end\_thread is 1, those sections are stopped by while statement)

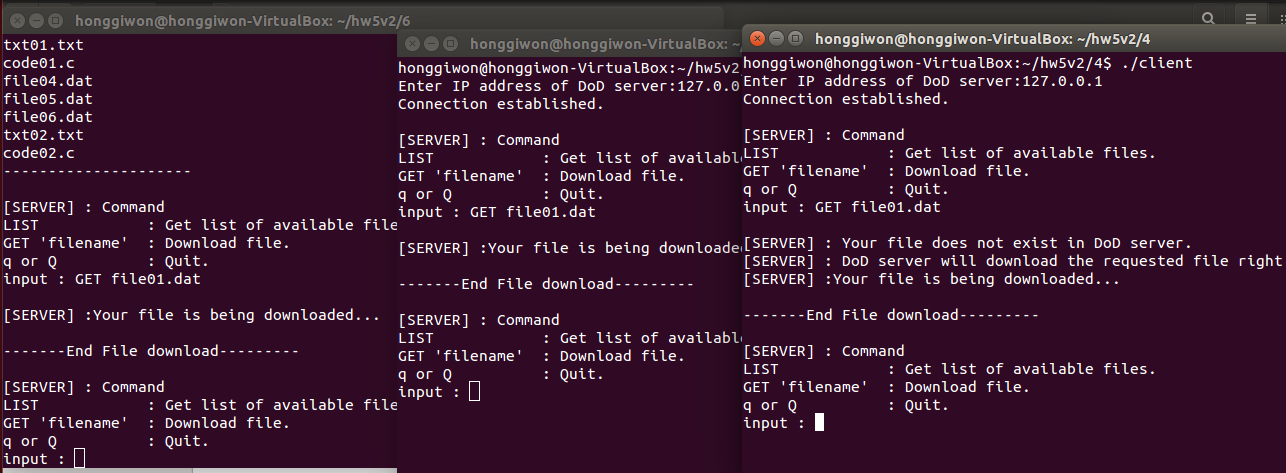
* **Execute result (screenshots)**



After dod server initialize, it create 3 threads. Each client will connect with one thread; therefore multiple clients could be served concurrently. We can see in the screenshot that 3 clients are served at once.

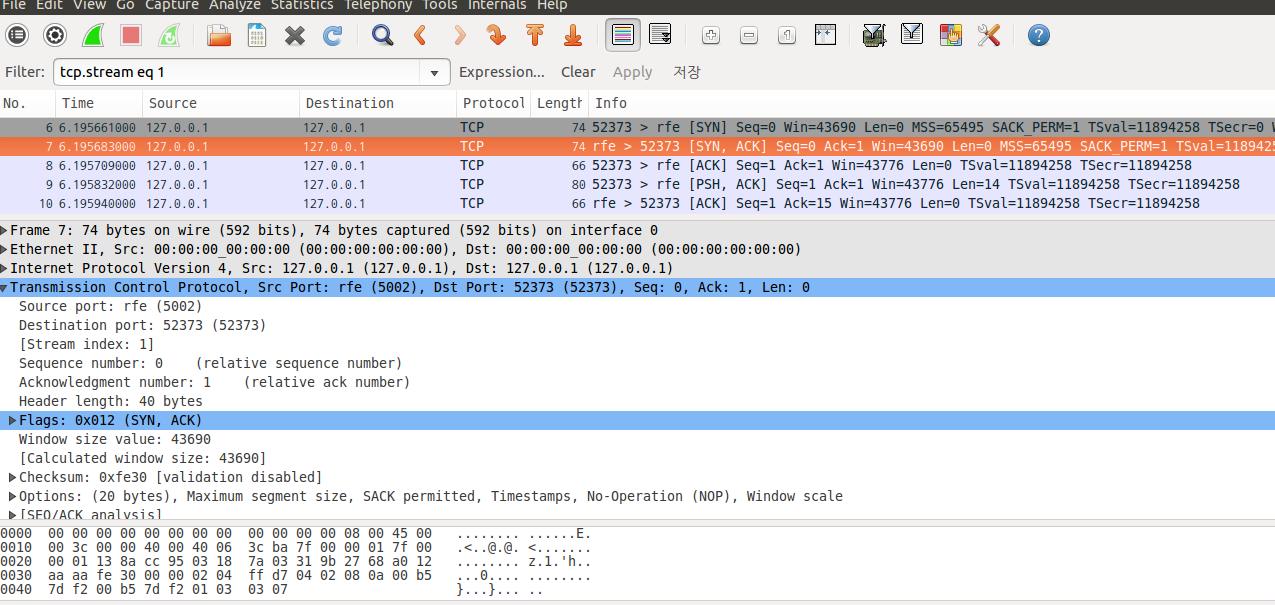


If the requested file is not in DoD server, DoD server informs to client that the requested file does not exist in DoD server. And after downloading from file server is finished, DoD server also informs to client that the requested file is transferring to client.

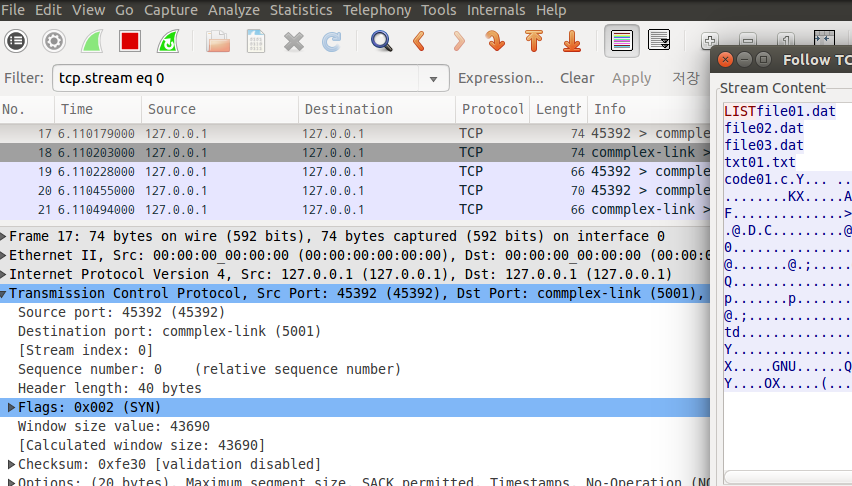


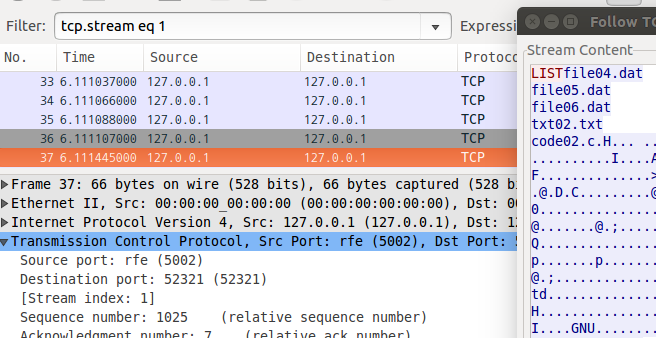
After a file is downloaded and stored in DoD server, if any clients demand that file, DoD server will serve it without connecting to file server. In screenshot, we can see that after file01.dat is downloaded and stored in DoD server (rightmost terminal), other clients could download it without connecting to file server.

* **Screenshots of Wireshark**



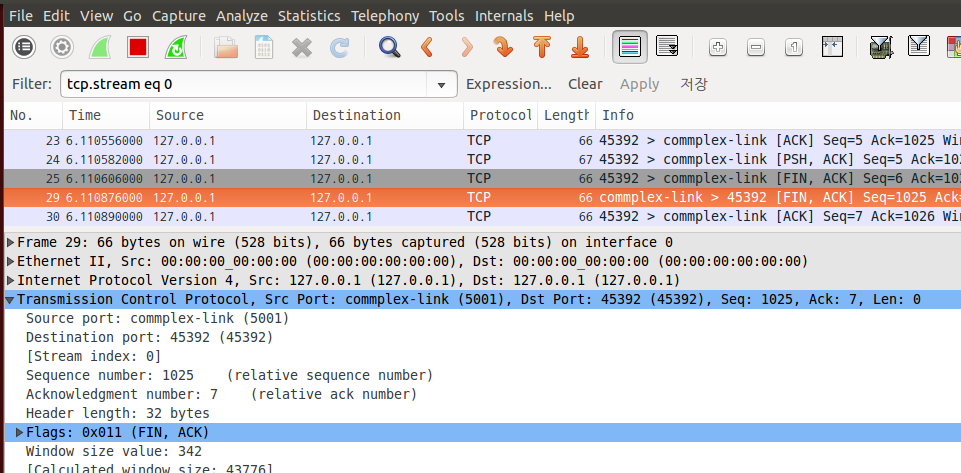
Above screenshot is about SYN packet between file server2 (port number 5002) and DoD server. This packet is sent when connection between them is established.



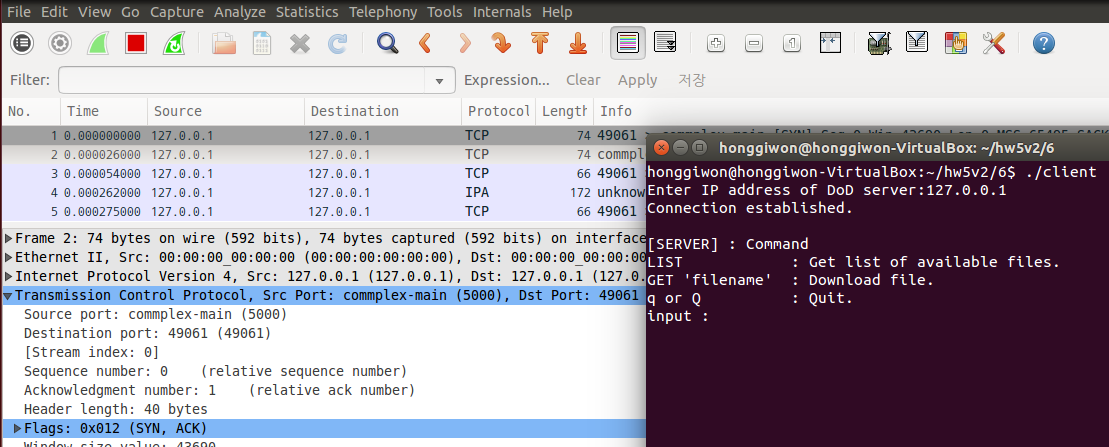


When initializing DoD server, it connects to all file servers and send LIST command. And File server sends available file list to DoD server.

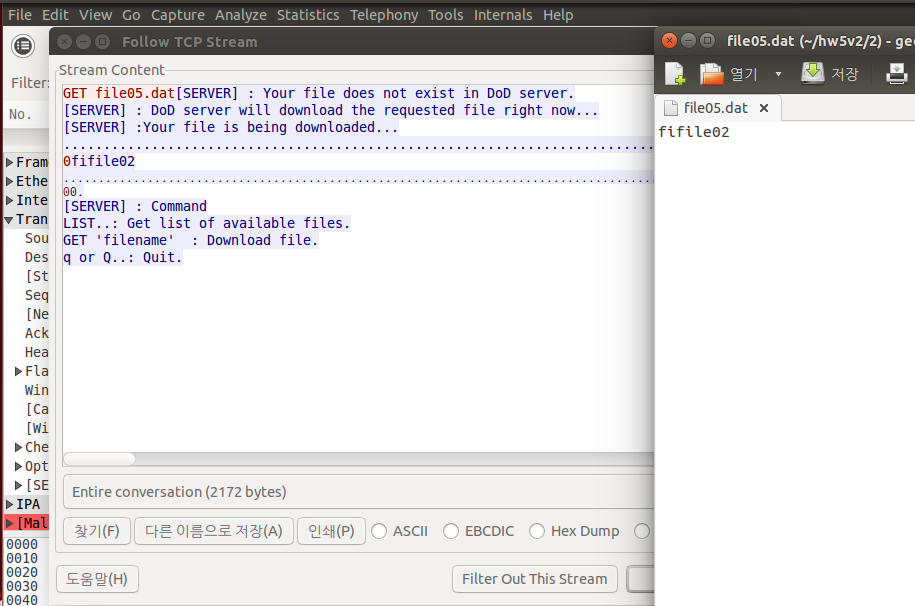
Above screenshots show TCP stream when initializing DOD server. We can see that DoD server sends LIST command (red) and file servers sends available file list (blue).



After initializing, connection between file servers and DoD server is terminated. Above screenshot shows FIN packet in that situation.

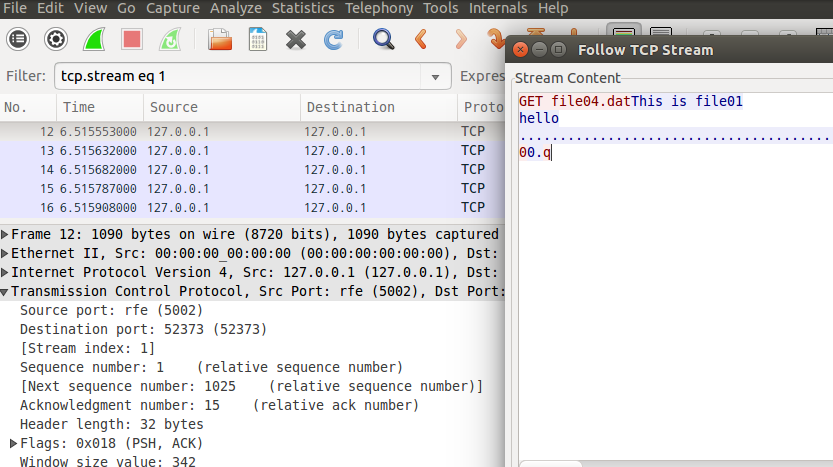


Above screenshot shows SYN packet when a client connect to DoD server. (Thread)



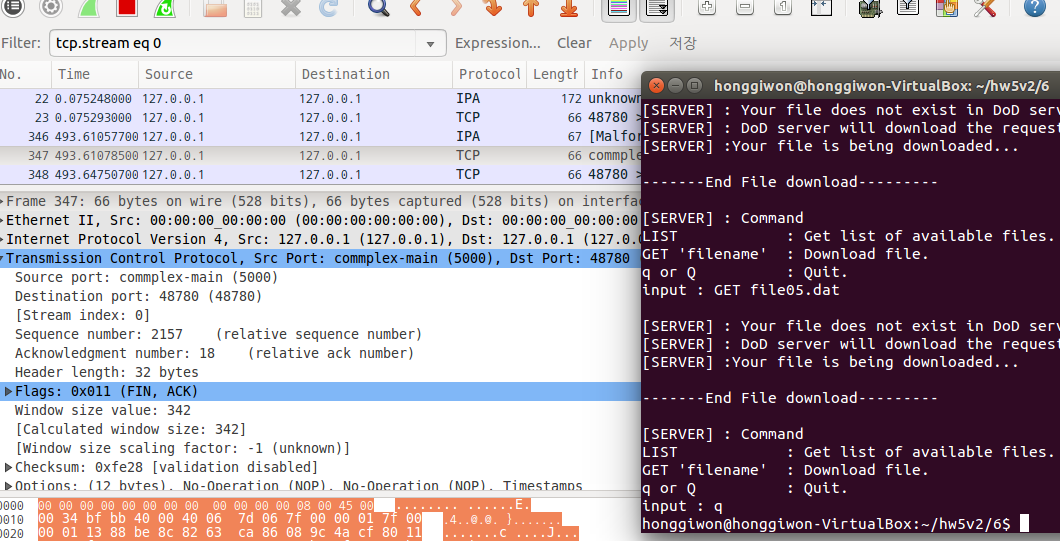
If a client connects to DoD server, DoD server sends list of available commands, and a client sends one of those command to DoD server. Then DoD server handles that command, sends list of available commands again and awaits for next command and so on. (When command is not q or Q)

In above screenshot, we can see that client (red) send GET command to DoD server (blue) and DoD server sends information of download (like “your file is being downloaded”). And then DoD server sends data of file “fifile02” which is exactly match the contents of the file shown on the right.



If the requested file does not exist on the DoD server, the DoD server sends a request to the file server to download and store the file.

In above screenshot shows TCP stream of that situation. DoD server (red) send GET command and file server (blue) sends file. And then DoD server send q command to end the connection.



When the client wants to finish connection, the client sends q or Q command to the DoD server. Above screenshot is about FIN packet which indicates connection between the client and the DoD server ends.