

Lab 4 Documentation

Christopher Jensen (chjense@ucsc.edu)

How To Use

1. Cd into folder lab04
2. There are 4 folders and 2 files. The two files are Makefile and README.txt. The four folders are doc, bin, build, and src. Doc is where the documentation is located, build is for the .o files, bin is for the binaries, and src is where the source code for this lab is located. In src there is myclient.cpp and myserver.cpp.
3. Run a quick “make” in the top directory where the Makefile is
4. Cd into bin
5. Firstly you need to set up the server. You can do this by running the command:
 - a. `./myserver port_number droppc root_folder_path`
 - b. Where *port_number* is a number between 1024 and 65536
 - c. *droppc* is a percentage value between 0-100. This is the percentage of packets/ACKSs that will be dropped on the server side.
 - d. *root_folder_path* is the root directory under which the transferred files will be saved
6. Once the server is up and running, create a new instance of the terminal and cd back into the bin folder
7. Now you can send requests to the server. You can do this by running the command;
 - a. `./myclient servn servaddr.conf mtu winsz in_file_path out_file_path`
 - b. Where *servn* is the number of servers you’d like to replicate to
 - c. *servaddr.conf* is a configuration file that has the IP addresses and port numbers of each available server. Formatted like:

```
# comment line starting with #
# IP address  port
10.1.0.2 9090
10.1.0.3 9091
i. 192.71.62.5 9092
```
 - d. *Mtu* is the maximum transmission unit, and for this client protocol, the smallest MTU you can enter is 61. These bytes are what makes up the header, and the rest is made up of payload bytes.
 - e. *Winsz* is the size of the window used for the “go back n” protocol implemented in this client.
 - f. *In_file_path* is where the file that you want to send is located
 - g. *Out_file_path* is where you want to put the new copied file on the server's system. It will be placed under the root directory (*root_folder_path*) of the server that you just started. If this directory does not exist it will create it for you.

Internal Design

- **Myserver.cpp:**

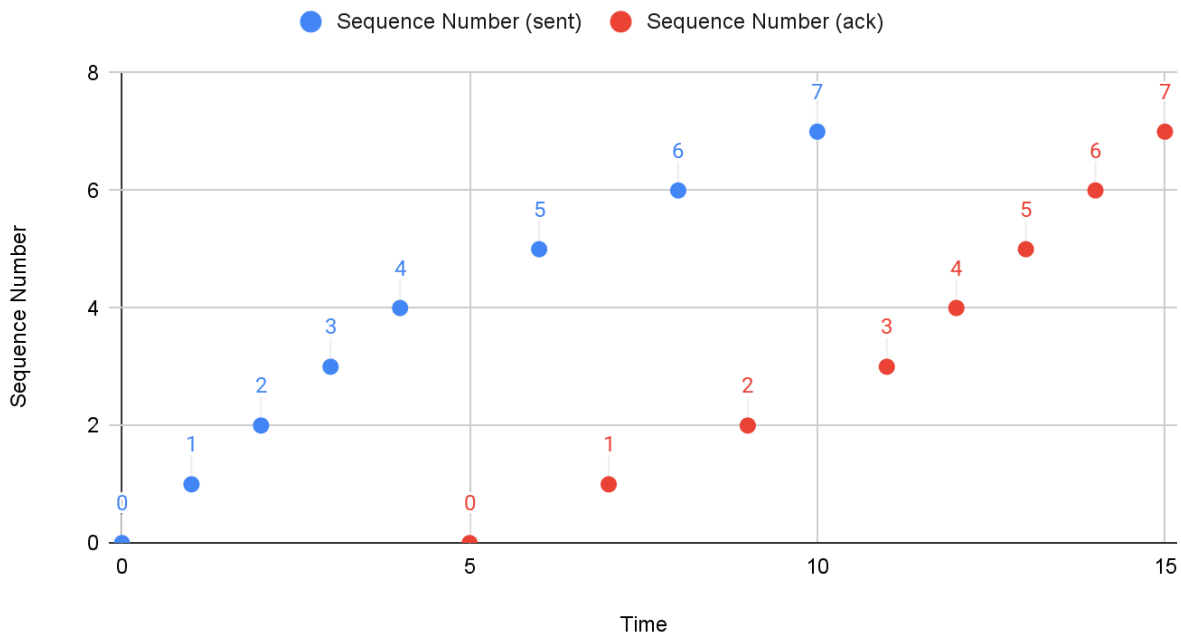
- This server can support multiple clients by storing them each in a vector of struct `client_info`, which includes information that can identify each client every time a packet is sent. The server listens for 3 types of packets. Firstly a header packet, which contains information about the client and the file it is preparing to send. It also listens for an ender packet, which tells the server that we're done sending it information and it's time to close the file and remove the client from the client vector. Finally, it listens for data packets, which is where the bulk of the work is done. If the server receives a data packet, it first determines which client sent the packet, then processes the payload and writes the data into the correct file. In accordance with the go back n protocol, the server keeps track of the sequence numbers and drops any out of order packets as they come. The server is also given an input `droppc`, which drops the given percentage of packets sent to the server, used for debugging. I used a random number generator to do this, every time through the loop generating a new random number between 0 and 1. Then, this number is compared to the percentage value `droppc/100`, and if the random number is less than or equal to that value, the packet is dropped.
- The server's `recvfrom()` buffer extended from the default setting to `BUFFERLENGTH*30`, which allows the server to buffer up to 30 packets for processing.
- Upon starting the server, a directory is created specified by the *root_folder_path*.
- **Myclient.cpp:**
 - After starting the client, the main thread gets the command line arguments and parses the configuration file for the multiple server's IP addresses and port numbers. This information is stored in an array.
 - After collecting the basic information, the main thread creates copies of global variables to later give to the threads taking care of replicating the file to the servers. The main thread then calls `pthread_create servn` times and passes in the copies of the global variables. From here, each thread enters the thread function and creates its own file pointer for the socket and the in file. Each thread proceeds to execute the go back n protocol until an error is thrown and it needs to exit, or it completes successfully.
 - This client is designed using a go back n protocol. I kept track of the resend window using a list data structure. Firstly, `winsz` packets are sent to the client, the pointer `nextsn` is moved forward and the sent packets are pushed to the resend window. After sending the packets, the client calls `recvfrom()` with a timeout, waiting for either an ack from the server or a timeout. If we receive an ack, the pointer `basesn` is moved forward and the acked packets are popped off of the list. If we receive a timeout, the client resends every packet in the window. If the client times out on the same packet too many times, it exits with "reached max re-transmission limit". This process is repeated until the file descriptor is empty and the client has received the final ack. The client sends an ender packet to the server, telling it to close the file. The client then closes with success.

Shortcomings

- For this lab, I increased the timeout variable a little bit to counteract how many times the client loses a packet due to timeout. This slowed it down only slightly, but is now more reliable than it was before. Other than that, I don't think there are any more shortcomings to mention.

Client Sequence/ACKs Over Time

Packets Sent And ACKs Received By Client Over Time



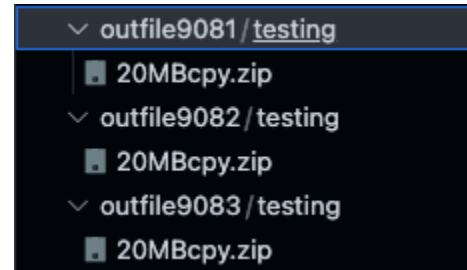
- To make this chart simple and readable, I ran a small file through my program called basic.txt. This file was just replicated to one server. The client log output is shown below:

```
l-bash-4.2$ ./myclient 1 ~/cse156/lab04/serv.conf 512 5 ~/cse156/basic.txt testing/basic.txt
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 0, 0, 0, 5
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 1, 0, 1, 5
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 2, 0, 2, 5
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 3, 0, 3, 5
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 4, 0, 4, 5
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 0, 1, 5, 6
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 5, 1, 5, 6
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 1, 2, 6, 7
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 6, 2, 6, 7
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 2, 3, 7, 8
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, DATA, 7, 3, 7, 8
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 3, 4, 8, 9
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 4, 5, 8, 10
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 5, 6, 8, 11
2023-03-03T00:12:47.696Z, 51010, 192.168.122.1, 9080, ACK, 6, 7, 8, 12
2023-03-03T00:12:47.697Z, 51010, 192.168.122.1, 9080, ACK, 7, 8, 8, 13
Server 9080 Completed
All servers completed successfully, exiting with success
```

Testing

3 Concurrent 20MB File Replication Test With 5% Packet Loss (window size 100, mtu 6000):

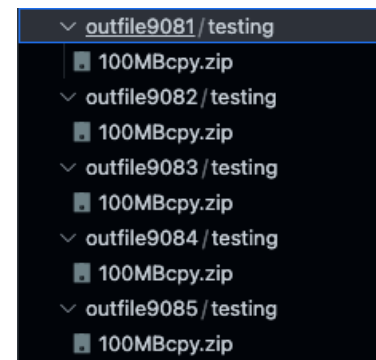
- `./myclient 3 ~/cse156/lab04/serv.conf 6000 100 ~/cse156/20MB.zip testing/20MBcpy.zip`
- `./myserver 9081 5 ~/cse156/lab04/outfile9081`
- `./myserver 9082 5 ~/cse156/lab04/outfile9082`
- `./myserver 9083 5 ~/cse156/lab04/outfile9083`



```
Server 9082 Completed
Server 9081 Completed
Server 9083 Completed
All servers completed successfully, exiting with success
-bash-4.2$ diff -s ~/cse156/20MB.zip ~/cse156/lab04/outfile9081/testing/20MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/20MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9081/testing/20MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/20MB.zip ~/cse156/lab04/outfile9082/testing/20MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/20MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9082/testing/20MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/20MB.zip ~/cse156/lab04/outfile9083/testing/20MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/20MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9083/testing/20MBcpy.zip are identical
-bash-4.2$
```

5 Concurrent 100MB File Replication Test With 5% Packet Loss (window size 100, mtu 15000):

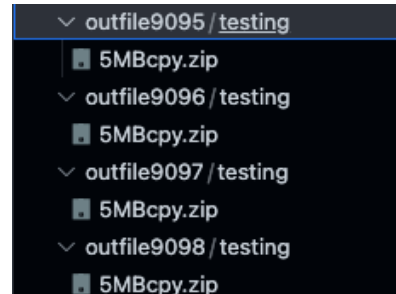
- `./myclient 5 ~/cse156/lab04/serv.conf 15000 100`
`~/cse156/100MB.zip testing/100MBcpy.zip`
- `./myserver 9081 5 ~/cse156/lab04/outfile9081`
- `./myserver 9082 5 ~/cse156/lab04/outfile9082`
- `./myserver 9083 5 ~/cse156/lab04/outfile9083`
- `./myserver 9084 5 ~/cse156/lab04/outfile9084`
- `./myserver 9085 5 ~/cse156/lab04/outfile9085`



```
Server 9085 Completed
Server 9081 Completed
Server 9082 Completed
Server 9083 Completed
Server 9084 Completed
All servers completed successfully, exiting with success
-bash-4.2$ diff -s ~/cse156/100MB.zip ~/cse156/lab04/outfile9081/testing/100MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/100MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9081/testing/100MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/100MB.zip ~/cse156/lab04/outfile9082/testing/100MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/100MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9082/testing/100MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/100MB.zip ~/cse156/lab04/outfile9083/testing/100MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/100MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9083/testing/100MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/100MB.zip ~/cse156/lab04/outfile9084/testing/100MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/100MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9084/testing/100MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/100MB.zip ~/cse156/lab04/outfile9085/testing/100MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/100MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9085/testing/100MBcpy.zip are identical
-bash-4.2$
```

4 Concurrent 5MB File Replication Test With 5% Packet Loss (window size 50, mtu 3000):

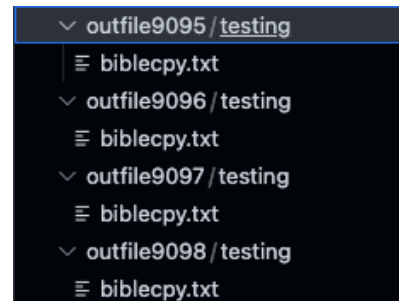
- ./myclient 4 ~/cse156/lab04/serv.conf 3000 50 ~/cse156/5MB.zip testing/5MBcpy.zip
- ./myserver 9096 10 ~/cse156/lab04/outfile9096
- ./myserver 9097 10 ~/cse156/lab04/outfile9097
- ./myserver 9098 10 ~/cse156/lab04/outfile9098
- ./myserver 9095 10 ~/cse156/lab04/outfile9095



```
Server 9097 Completed
Server 9095 Completed
Server 9098 Completed
Server 9096 Completed
All servers completed successfully, exiting with success
-bash-4.2$ diff -s ~/cse156/5MB.zip ~/cse156/lab04/outfile9095/testing/5MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/5MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9095/testing/5MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/5MB.zip ~/cse156/lab04/outfile9096/testing/5MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/5MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9096/testing/5MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/5MB.zip ~/cse156/lab04/outfile9097/testing/5MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/5MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9097/testing/5MBcpy.zip are identical
-bash-4.2$ diff -s ~/cse156/5MB.zip ~/cse156/lab04/outfile9098/testing/5MBcpy.zip
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/5MB.zip and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9098/testing/5MBcpy.zip are identical
-bash-4.2$
```

5 Concurrent Large Text File Replication Test With 10% Packet Loss (window size 100, mtu 2000):

- ./myclient 4 ~/cse156/lab04/serv.conf 3000 50
~/cse156/bible.txt testing/biblecpy.txt
- ./myserver 9096 10 ~/cse156/lab04/outfile9096
- ./myserver 9097 10 ~/cse156/lab04/outfile9097
- ./myserver 9098 10 ~/cse156/lab04/outfile9098
- ./myserver 9095 10 ~/cse156/lab04/outfile9095



```
Server 9096 Completed
Server 9095 Completed
Server 9098 Completed
Server 9097 Completed
All servers completed successfully, exiting with success
-bash-4.2$ diff -s ~/cse156/bible.txt ~/cse156/lab04/outfile9095/testing/biblecpy.txt
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/bible.txt and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9095/testing/biblecpy.txt are identical
-bash-4.2$ diff -s ~/cse156/bible.txt ~/cse156/lab04/outfile9096/testing/biblecpy.txt
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/bible.txt and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9096/testing/biblecpy.txt are identical
-bash-4.2$ diff -s ~/cse156/bible.txt ~/cse156/lab04/outfile9097/testing/biblecpy.txt
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/bible.txt and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9097/testing/biblecpy.txt are identical
-bash-4.2$ diff -s ~/cse156/bible.txt ~/cse156/lab04/outfile9098/testing/biblecpy.txt
Files /afs/cats.ucsc.edu/users/u/chjjense/cse156/bible.txt and /afs/cats.ucsc.edu/users/u/chjjense/cse156/lab04/outfile9098/testing/biblecpy.txt are identical
-bash-4.2$
```

If All Servers Fail, Exit With Error Test:

- For this test, I started a replication to 4 servers, and in the middle of all of those transfers I ended all of the servers.

```
2023-05-05T05:55:40.577Z, 45552, 192.168.122.11, 9090, DATA, 551, 502, 552, 552
Server 9095 Failed
Server 9096 Failed
Server 9097 Failed
Server 9098 Failed
Reached max re-transmission limit
-bash-4.2$
```

Too Many Packets Lost Test:

- For this test, I set one of the server's droppc to 90%, and kept the rest at 5%.
- `./myclient 4 ~/cse156/lab04/serv.conf 3000 50 ~/cse156/bible.txt testing/biblecpy.txt`
- `./myserver 9013 5 ~/cse156/lab04/outfile9013`
- `./myserver 9012 5 ~/cse156/lab04/outfile9012`
- `./myserver 9011 5 ~/cse156/lab04/outfile9011`
- `./myserver 9010 90 ~/cse156/lab04/outfile9010`

```
Server 9010 Failed
Server 9011 Completed
Server 9012 Completed
Server 9013 Completed
Some servers failed, exiting with failure
-bash-4.2$
```