7005 Asn4 - Testing and Usage

Make and test all the things!

John Agapeyev; A00928238
Isaac Morneau; A00958405

Building and Running

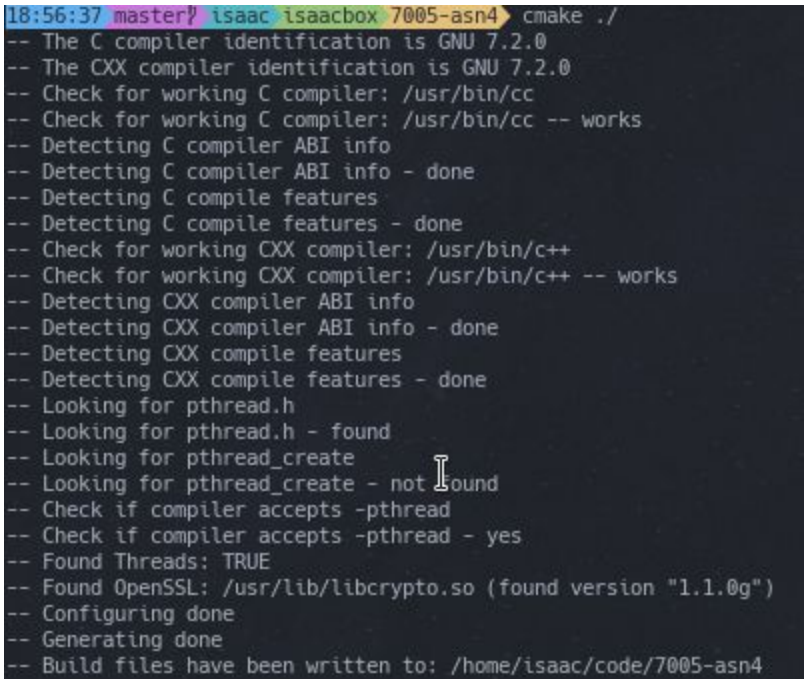
Prerequisites:

CMake is installed
C11 compliant compiler is installed
Linux Kernel is at 4.5 or greater
OpenSSL 1.1 is installed
Note to install OpenSSL run `install.sh`

Step 1

Generate makefile by running `cmake ./`

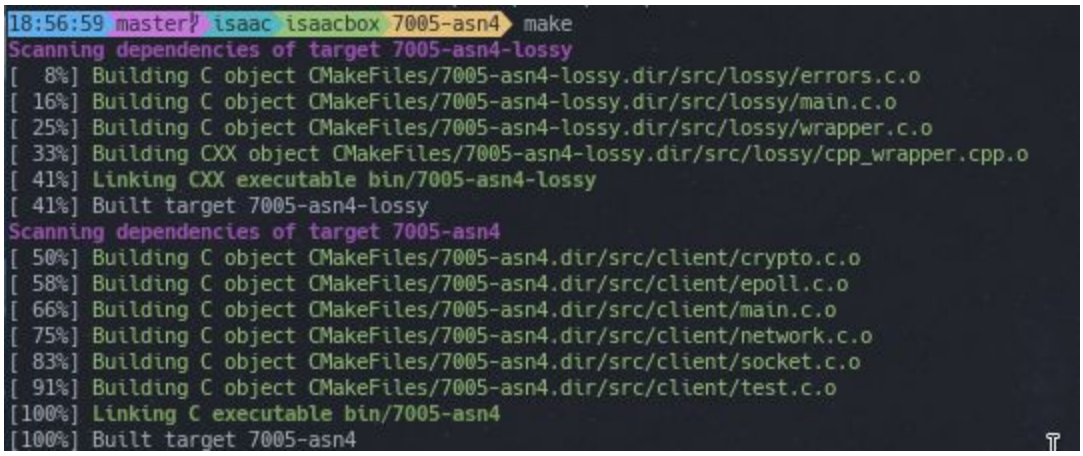
Expected outcome



Step 2

Build project running `make`

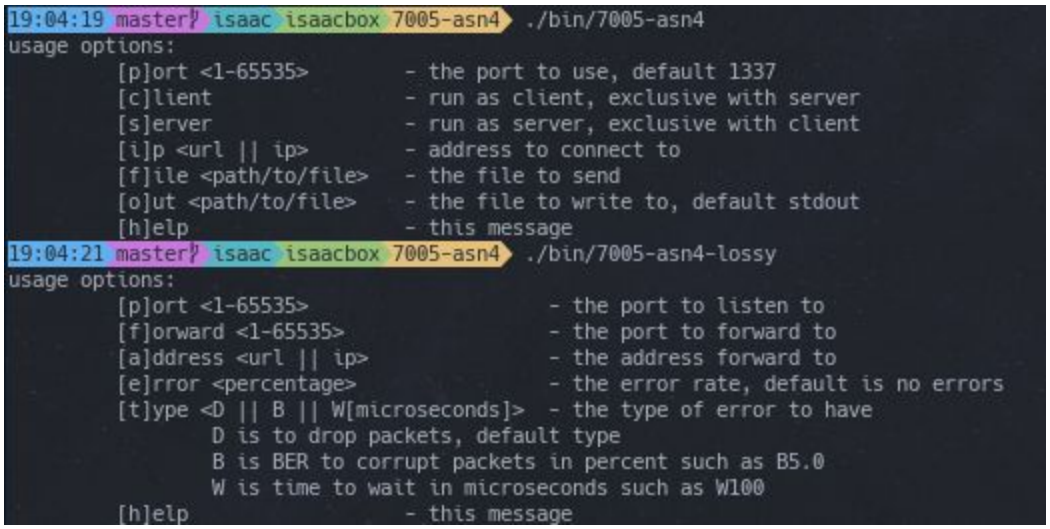
Expected Outcome



Step 3

Verify program built correctly by running `./bin/7005-asn4` and `./bin/7005-asn4-lossy`

Expected Result



Test of direct transfer

Purpose:

Show basic function

Step 1

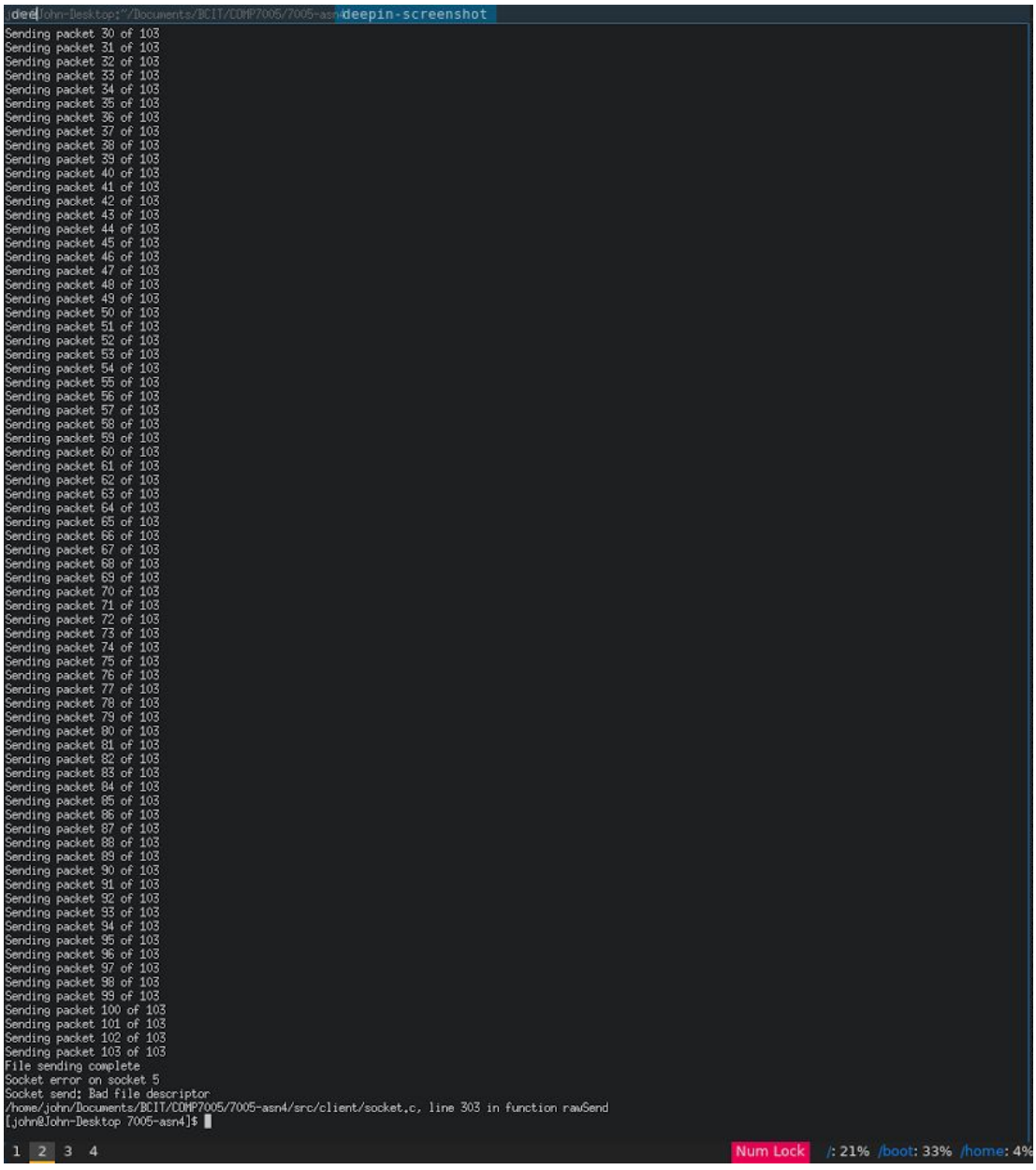
Run server with `./bin/7005-asn-4 -s -p 7001 -f testFile -o testServer`

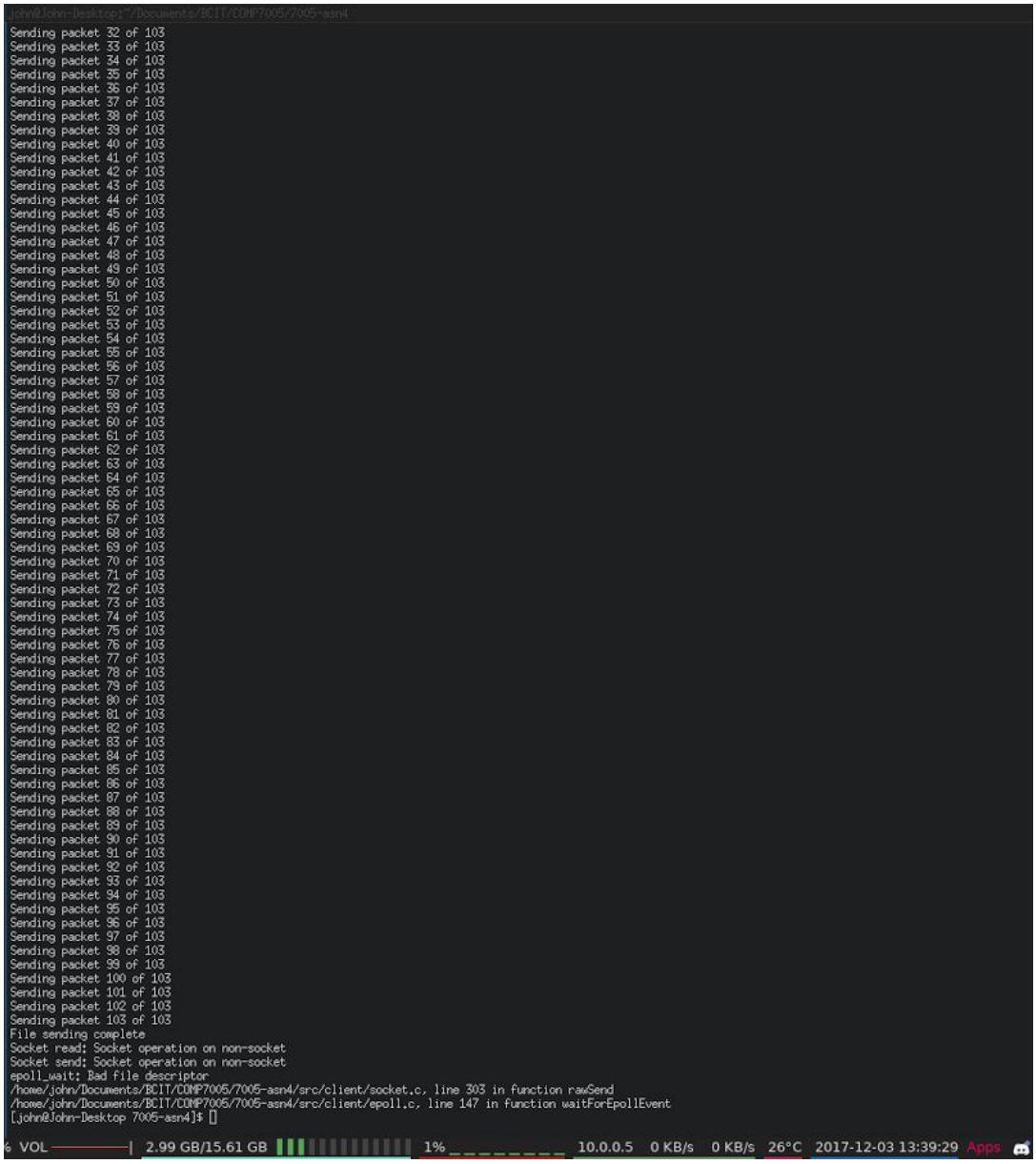
Step 2

Run client with `./bin/7005-asn-4 -c -i 0.0.0.0 -p 7001 -f testFile -o testServer`

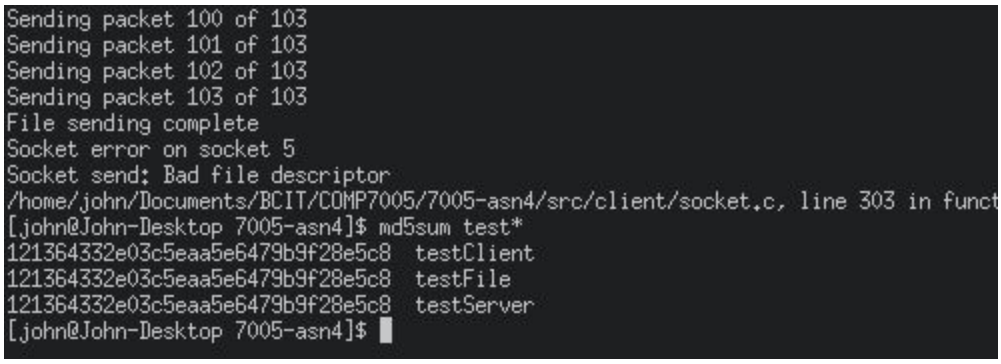
Expected Outcome

The client will connect to the server and transfer the file there as well as the server will transfer its file back as shown below.





After sending is complete the files can be verified by checking their MD5 checksum as shown below.



Test with middleman with no errors transfer

Purpose:

Show the middleman functionality

Step 1

Run server with `./bin/7005-asn-4 -s -p 7000 -f testFile -o testServer`

Step 2

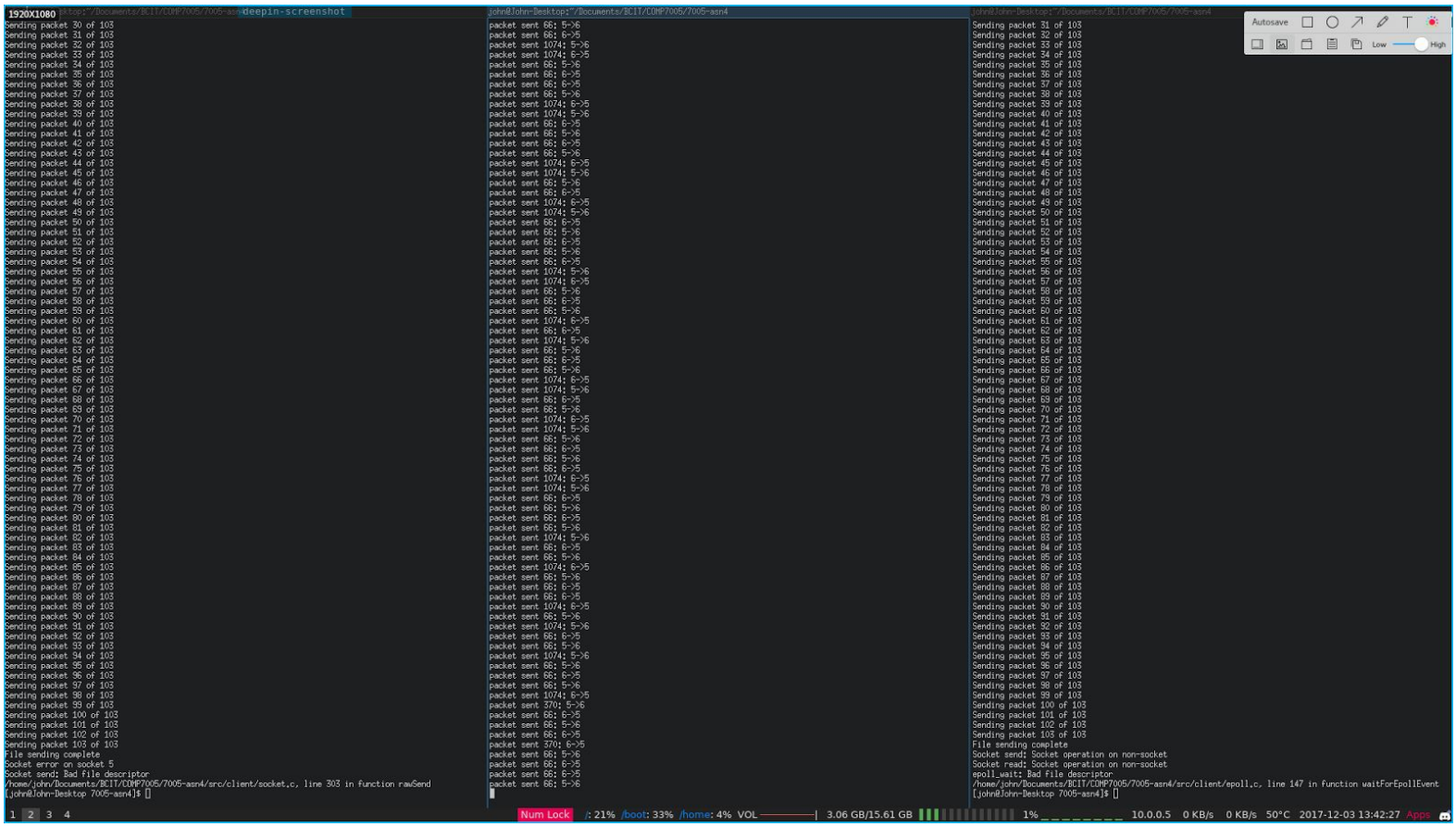
Run middleman with `./bin/7005-asn4-lossy -p 7000 -f 7001 -a localhost`

Step 3

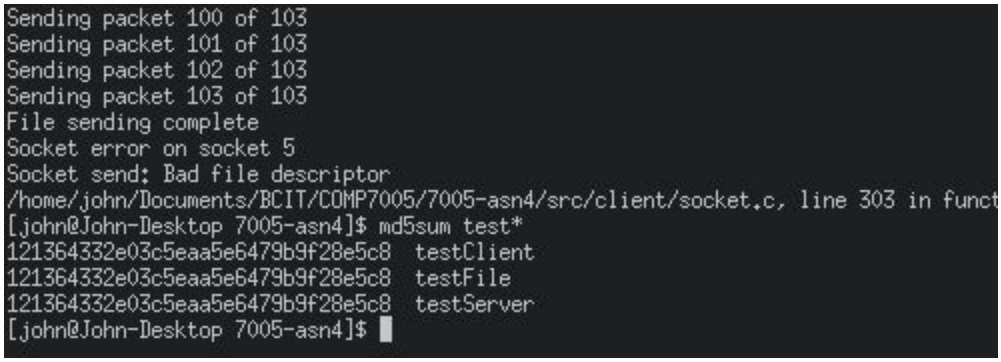
Run client with `./bin/7005-asn-4 -c -i 0.0.0.0 -p 7001 -f testFile -o testServer`

Expected Outcome

As shown with the middleman below the packets are all forwarding back and forth



After sending is complete the files can be verified by checking their MD5 checksum as shown below.



Test with middleman with drop errors

Purpose:

Check that resends and timeouts function correctly

Step 1

Run server with `./bin/7005-asn-4 -s -p 7000 -f testFile -o testServer`

Step 2

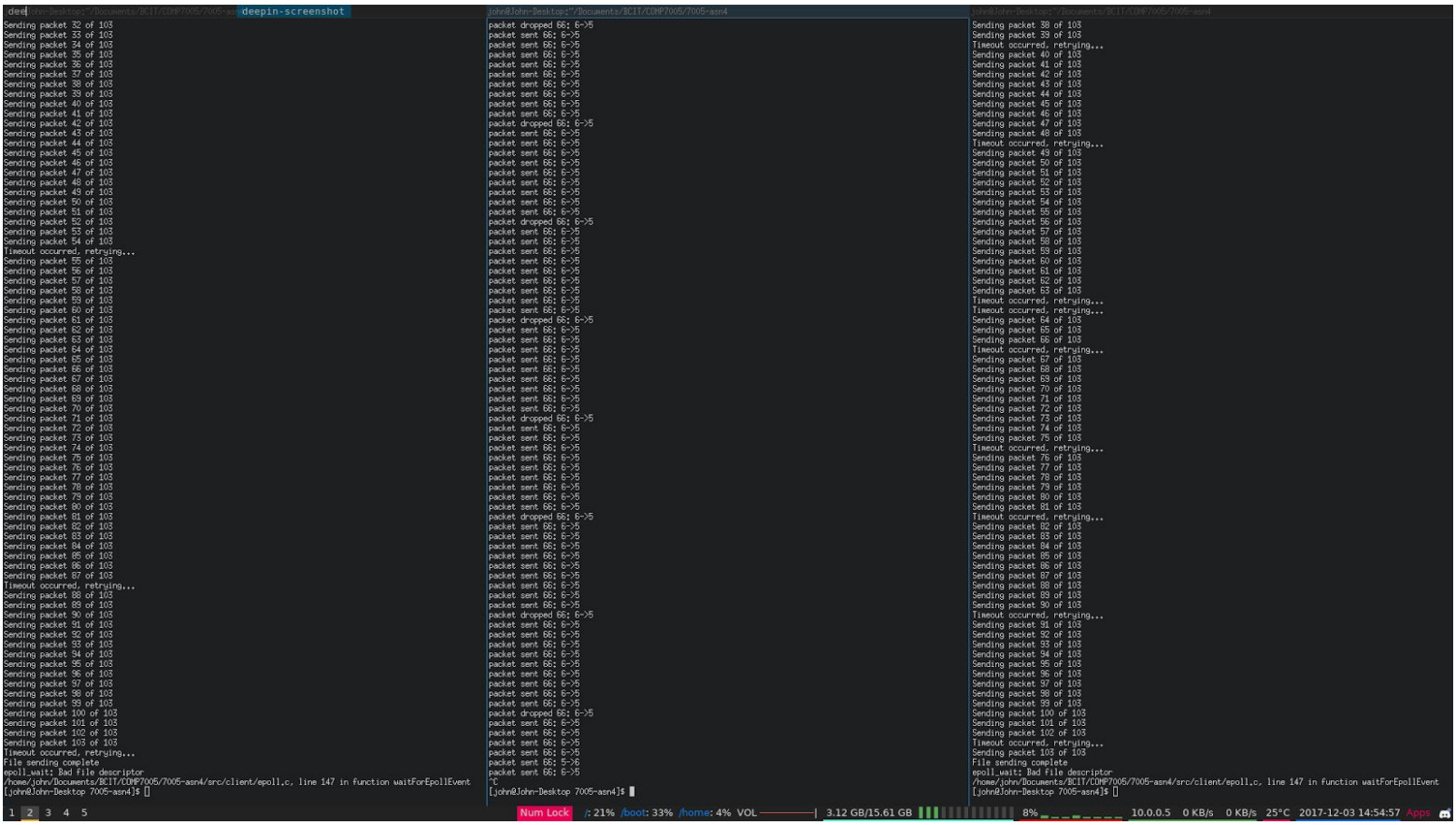
Run middleman with `./bin/7005-asn4-lossy -p 7000 -f 7001 -a localhost -t D -e 10`

Step 3

Run client with `./bin/7005-asn-4 -c -i 0.0.0.0 -p 7001 -f testFile -o testServer`

Expected Outcome

As can be seen below the lossy middle man prints the drops that occur when forwarding the traffic.



The error rate can be verified by checking the logged number of drops compared to the total packets forwarded.

```
[john@John-Desktop 7005-asn4]$ cat logging.txt | grep dropped | wc -l
14881
[john@John-Desktop 7005-asn4]$ cat logging.txt | wc -l
148810
[john@John-Desktop 7005-asn4]$
```

After sending is complete the files can be verified by checking their MD5 checksum as shown below.

```
Sending packet 100 of 103
Sending packet 101 of 103
Sending packet 102 of 103
Sending packet 103 of 103
File sending complete
Socket error on socket 5
Socket send: Bad file descriptor
/home/john/Documents/BCIT/COMP7005/7005-asn4/src/client/socket.c, line 303 in funct
[john@John-Desktop 7005-asn4]$ md5sum test*
121364332e03c5eaa5e6479b9f28e5c8  testClient
121364332e03c5eaa5e6479b9f28e5c8  testFile
121364332e03c5eaa5e6479b9f28e5c8  testServer
[john@John-Desktop 7005-asn4]$
```

Test with middleman with corruption errors

Purpose:

Check that HMAC failures result in packet drops

Step 1

Run server with `./bin/7005-asn-4 -s -p 7000 -f testFile -o testServer`

Step 2

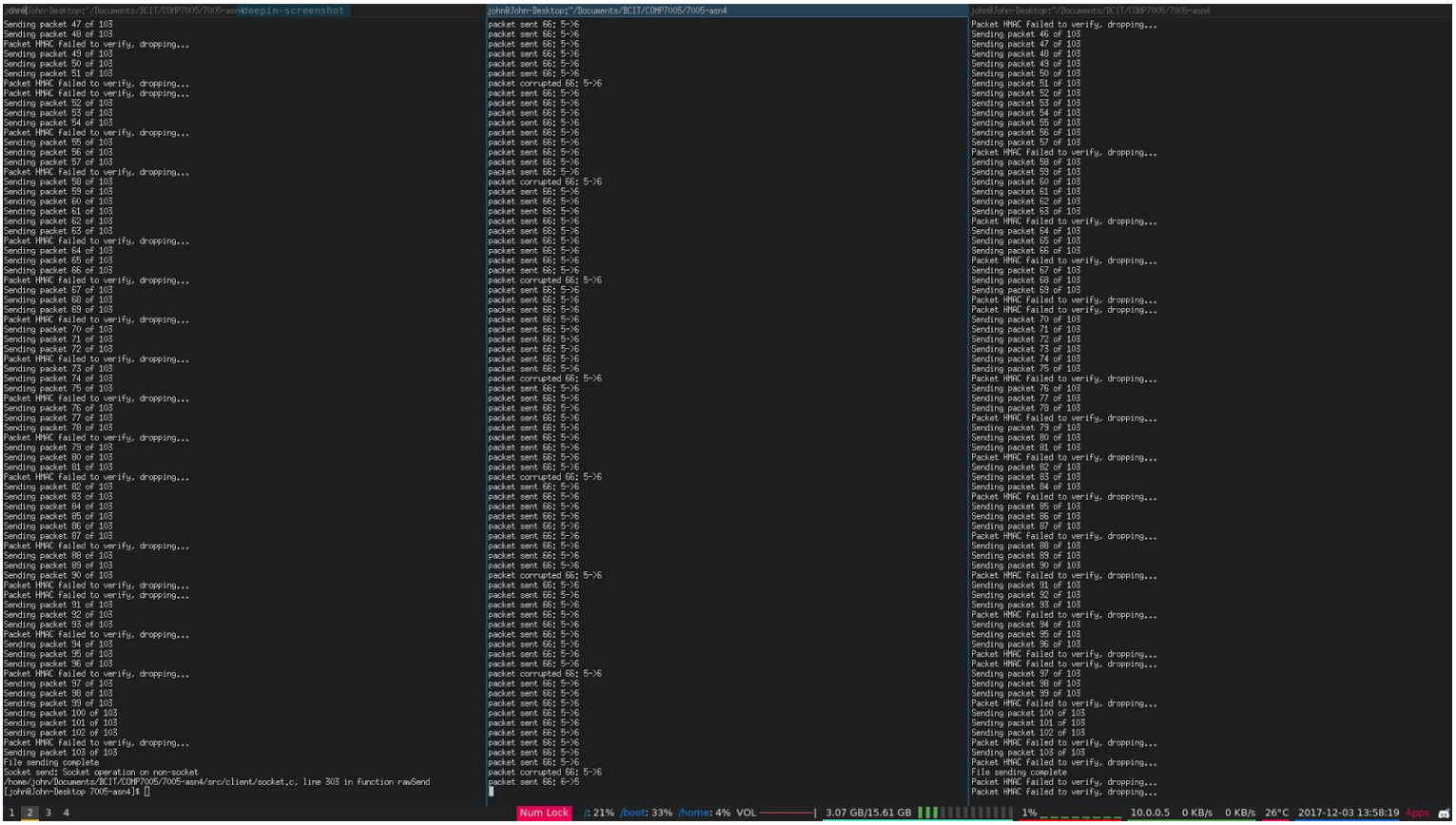
Run middleman with `./bin/7005-asn4-lossy -p 7000 -f 7001 -a localhost -t B5 -e 10`

Step 3

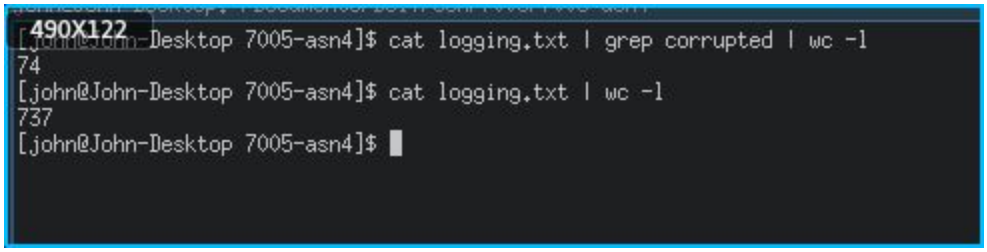
Run client with `./bin/7005-asn-4 -c -i 0.0.0.0 -p 7001 -f testFile -o testServer`

Expected Outcome

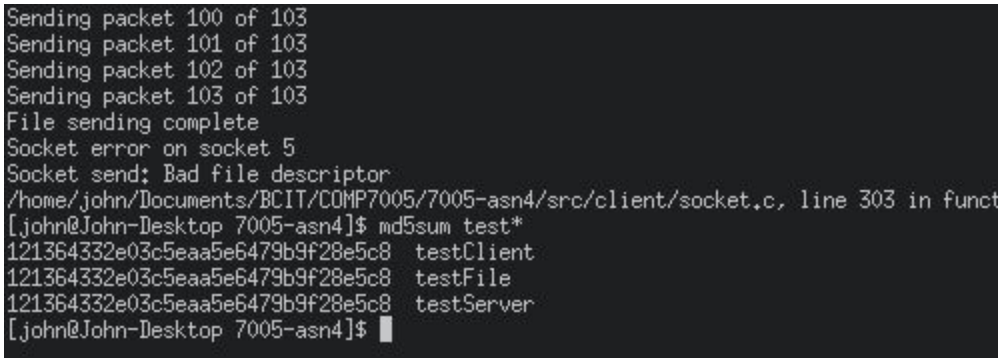
When bit corruption occurs this will invalidate the HMAC and cause the packet to be dropped as you can see below.



The error rate can again be checked by looking at the corrupted prints compared to the total sent to see that the error rate is maintained.



After sending is complete the files can be verified by checking their MD5 checksum as shown below.



Test with middleman with delayed packet errors

Purpose:

Check resends and timeouts function with delays

Step 1

Run server with `./bin/7005-asn-4 -s -p 7000 -f testFile -o testServer`

Step 2

Run middleman with `./bin/7005-asn4-lossy -p 7000 -f 7001 -a localhost -t W400 -e 10`

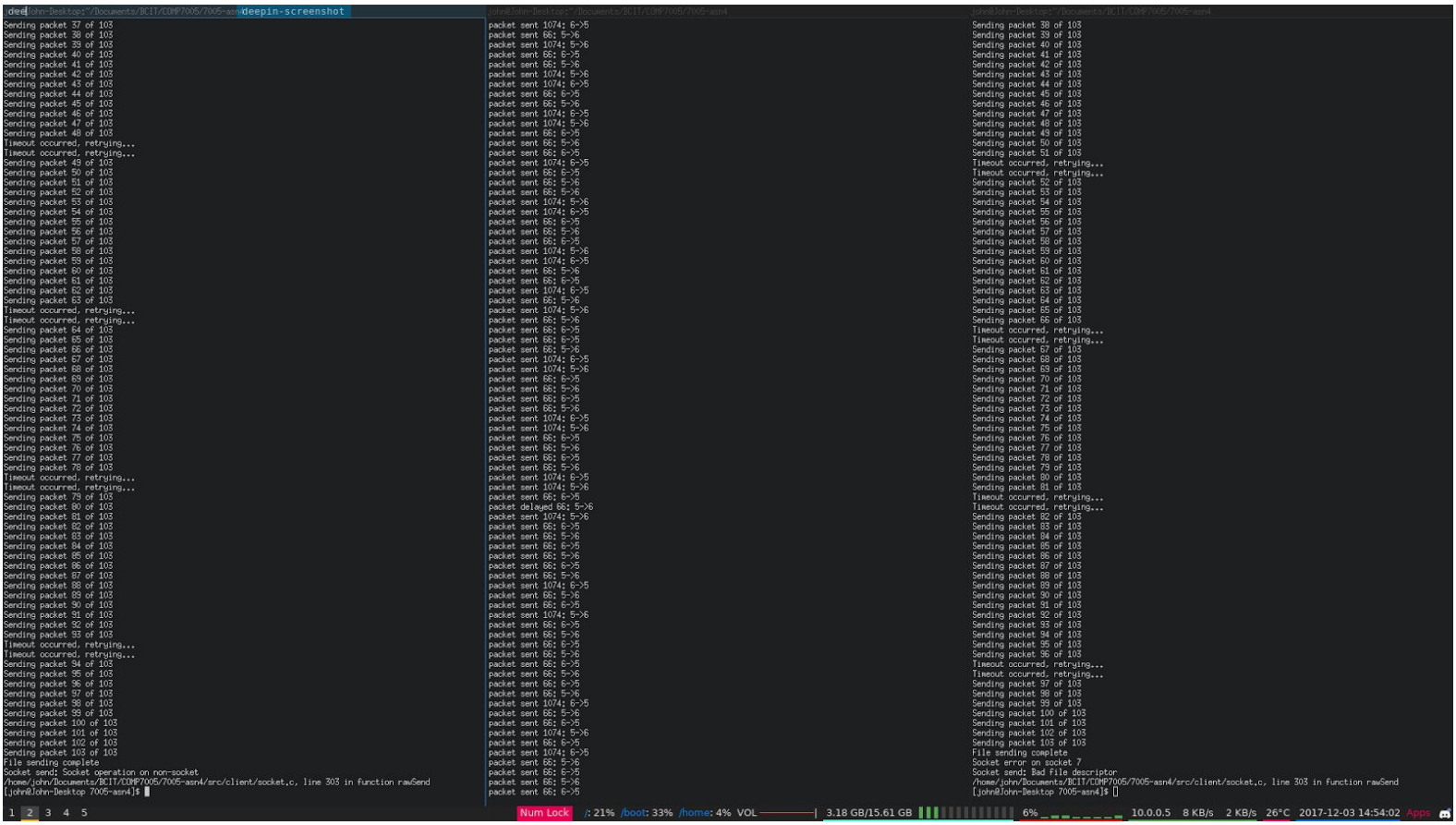
Step 3

Run client with `./bin/7005-asn-4 -c -i 0.0.0.0 -p 7001 -f testFile -o testServer`

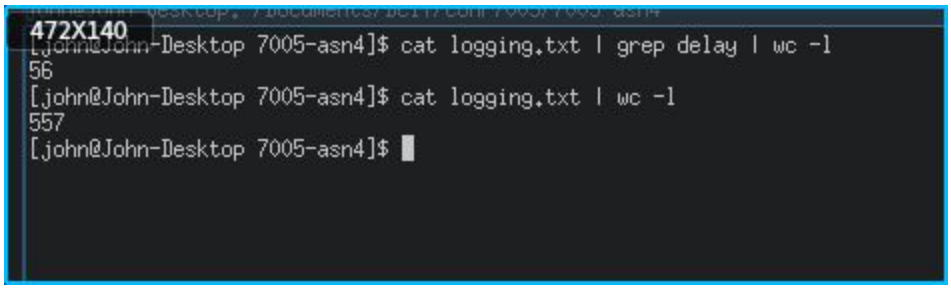
Expected Outcome

When delays occur that exceed the timeout the client and server will detect this and will resend the packet in question as can be seen below.

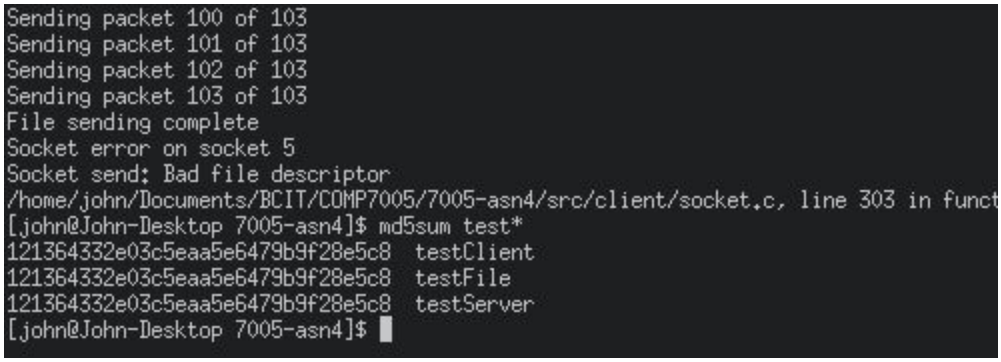
The error rate can again be checked by looking at the delayed prints compared to the total sent



to see that the error rate is maintained.



After sending is complete the files can be verified by checking their MD5 checksum as shown below.


Sending packet 100 of 103
Sending packet 101 of 103
Sending packet 102 of 103
Sending packet 103 of 103
File sending complete
Socket error on socket 5
Socket send: Bad file descriptor
/home/john/Documents/BCIT/COMP7005/7005-asn4/src/client/socket.c, line 303 in funct
[john@John-Desktop 7005-asn4]$ md5sum test*
121364332e03c5eaa5e6479b9f28e5c8 testClient
121364332e03c5eaa5e6479b9f28e5c8 testFile
121364332e03c5eaa5e6479b9f28e5c8 testServer
[john@John-Desktop 7005-asn4]$