

Homework 2 Documentation

Author

William Montgomery

Copyright

None

University of Illinois at Chicago

CS 450 - Introduction to Networking

Spring 2014

About the project

The file transfer project is composed of two programs, myClient and myServer. The general idea is to allow the client to transfer files and headers over UDP, and the server to process those messages. I chose to implement RDT 2.2.

Documentation

The project is documented with Doxygen tags. These tags allow HTML and RTF documentation to be generated. The HTML documentation can be found in the docs/html/index.html document. It is recommended that a web browser be used to view the documentation if possible. There is detailed

Building the project

A makefile is included. The user can run 'make clean' and then 'make' to build both the client and server applications (myClient and myServer respectively).

Running the server

The server can be run after building with './myServer'. It takes the following command line arguments:

- port - This argument specifies the port to connect to. The default is 54321.

The server will run multiple connections at a time. This is implemented by accepting new connections and then forking off a process to handle messages. This allows multiple messages to be processed at the same time and any error in a child process should not bring down the server. There is no mechanism to limit the number of children however. This is obviously

undesirable for the real world. Additionally, there is no mechanism to time out dormant connections.

The server will save files to the local filesystem if the correct flag is set in the header. The files are stored in the `./files/` subdirectory.

There is no built in method to gracefully exit from the server. The user is directed to use `[control] + C` to exit the program.

The server only handles one client at a time. If a client sends a message while another client is transmitting, the first client's message will be dropped. Ideally, I would create a hashtable to handle concurrent messages, with the keys computed from the sender's address, port, UIN, and transaction number. I chose not to because messages that are not fully sent would need to be cleaned up sporadically and dead messages would sit in the table until program exit. I feel this is out of the scope of this assignment.

Running the client

The client can be run after building with `./myClient`. It takes the following command line arguments:

- **server** - This is the IP address of the final destination that the packet is headed to. The default is 127.0.0.1.
- **server port** - This is the port of the final destination that the packet is headed to. The default is 54321.
- **relay** - This is the IP address of an intermediate relay. The default is 127.0.0.1.
- **relay port** - This is the port of an intermediate relay. The default is 54320.

Please note that the parsing of these inputs is brittle at best. Any incorrect value will cause the program to exit.

The client is generally interactive, looping through and asking the user to enter information interactively using basic menus. Menu items can be selected by entering the number next to the desired option. The client records basic statistics regarding time and number of bytes sent, on a per message and on a per session basis.

Interactive Options

The client handles dropped packets. In order to support this, the client has two interactive options:

- **sleep time** - This is the time in microseconds that the client sleeps between attempts to read an ack
- **timeout** - This is the number of times to try to read an ack

The total timeout can be approximated as (sleep time * timeout). The user can either use hardcoded defaults, or enter specific values for either of these. The default values were tested against the server in Japan to minimize resends. For servers on the same machine or nearby, it may be appropriate to decrease the values. For links that show a lot of packet loss, it would make sense to increase the timeouts. However this could potentially decrease the network utilization. Ideally, a client would want to fine tune these parameters on a running basis. For example, the client could calculate a running average round trip time (RTT). This RTT could be used as the basis for the sleep time, with a reasonable multiple of this for the number of tries. This would lessen the processing on the client, but is much more code to implement. I will leave this to later assignments.

Analysis

Overall, the client transmits data to local servers and to remote servers correctly. The client will also transmit data through relays. I chose to do four sets of comparisons. The first comparison was to set different timeout values and send differing sized files to the server in Tokyo. The test was repeated with the server in Virginia. The second test was to send files of various sizes to all of the available servers. The third test was to serve files to all of the available servers through a relay with the garble chance set to 20%. Finally, the test was to send a large file to both the class server in Virginia and my implementation located on a server in Virginia.

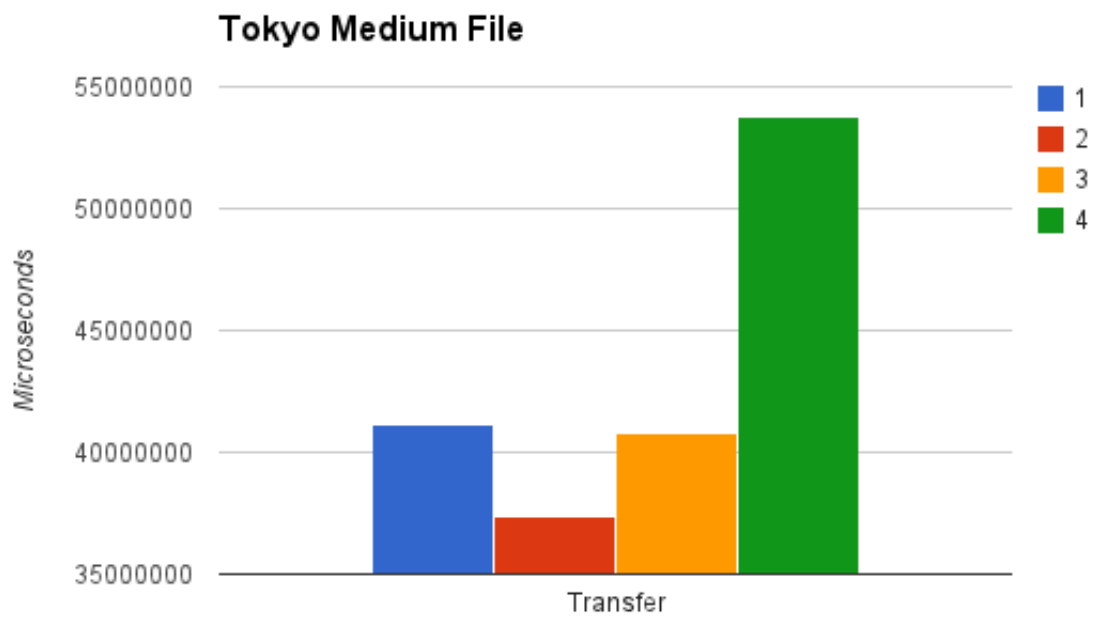
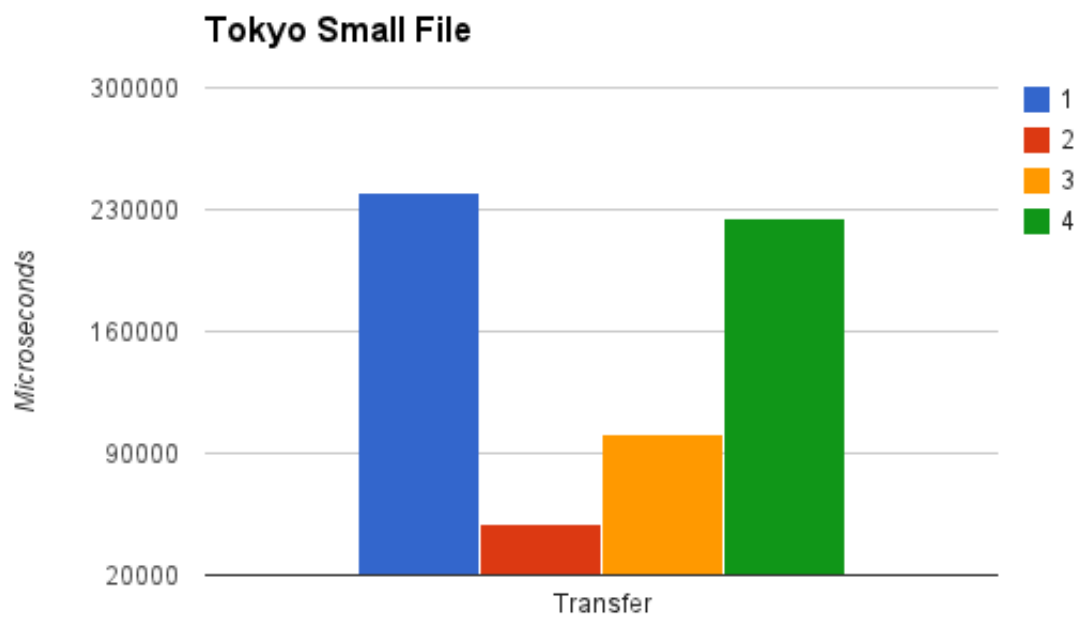
For reference, the small file was 30 bytes (1 packet), the medium file was 628361 bytes (176 packets), and the large file was 9437184 bytes (2634 packets).

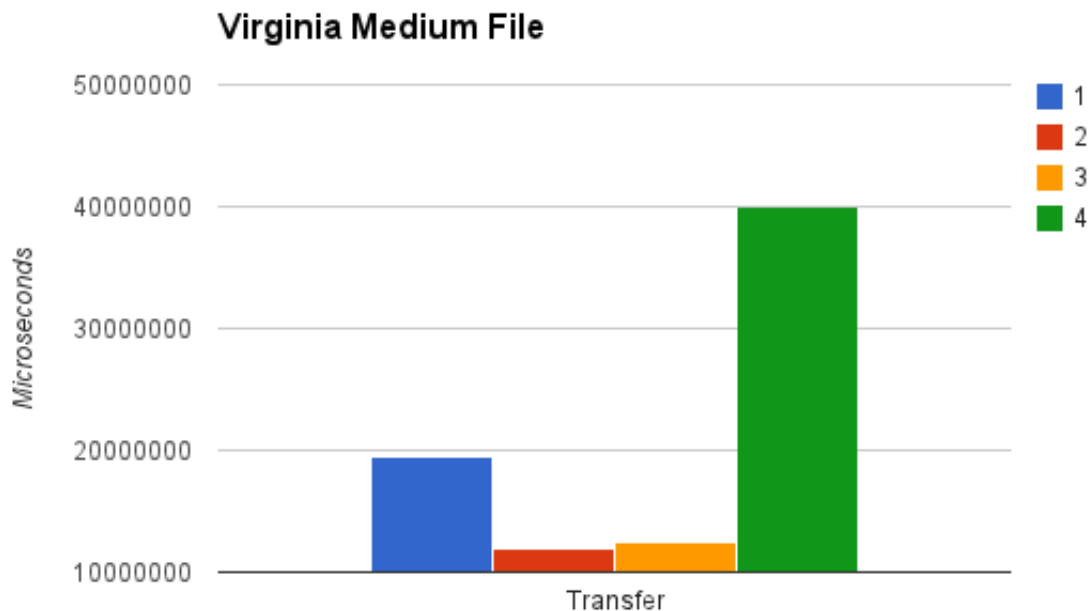
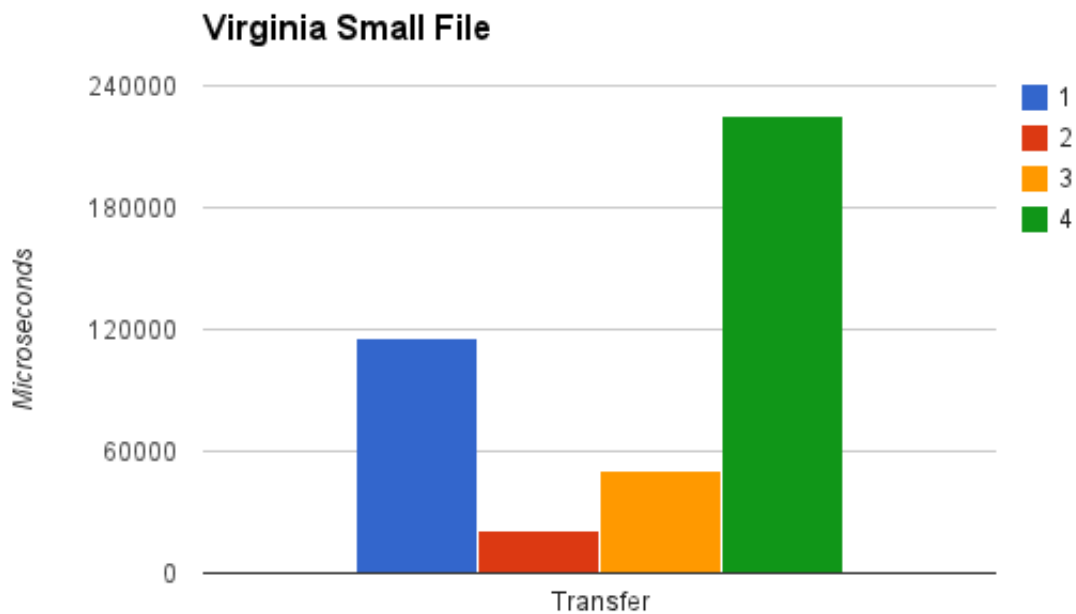
Timeout

For this test, I sent the small and medium files to the servers in Tokyo and Virginia. The goal was to find a timeout that balanced speed without sending too many duplicate packets. I used the following values (chosen arbitrarily) for the timeout:

Run	Time Between Reads	Number of Reads	Total Time (uSec)
1	300	200	6000
2	3000	200	60000
3	10000	100	1000000
4	225000	3	675000

The results are shown in the following graphs:





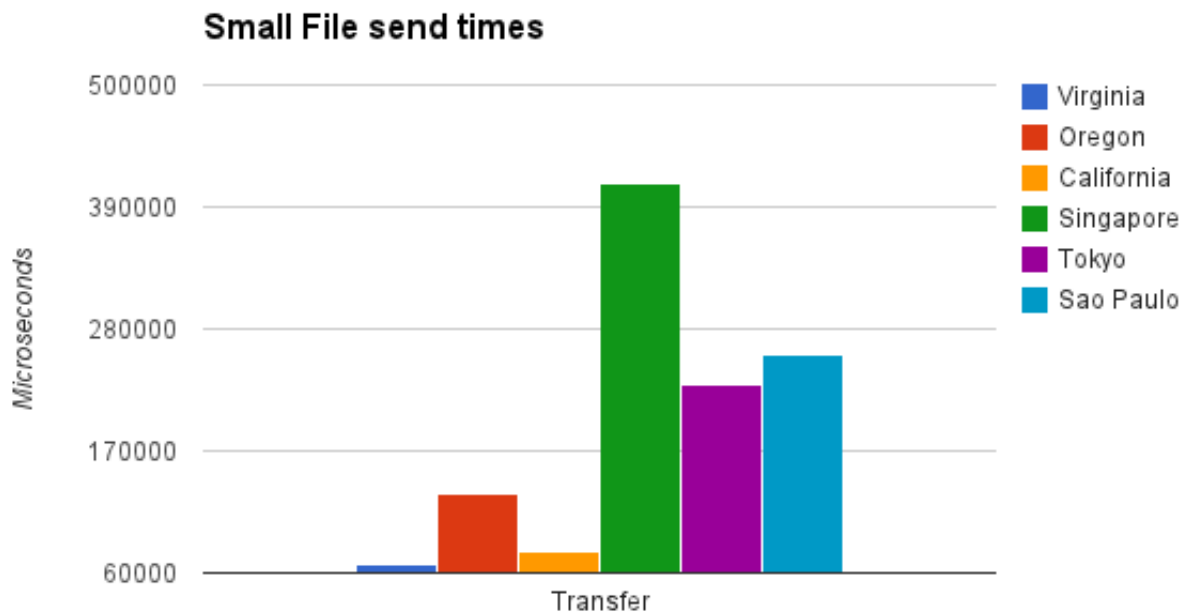
It is apparent from these that the sweet spot is #2, which correlates to read every 3,000 microseconds on the socket file descriptor with 200 repetitions. This trend is very apparent in the packets sent to Tokyo, but less so in the packets sent to Virginia.

It is interesting to note that a timeout that is too short is undesirable, as the packets do not have time to make the round trip. This causes lots of duplicates to be sent. On the other end of the spectrum, if the read time is too long, then the packets spend a long time in the buffer prior to

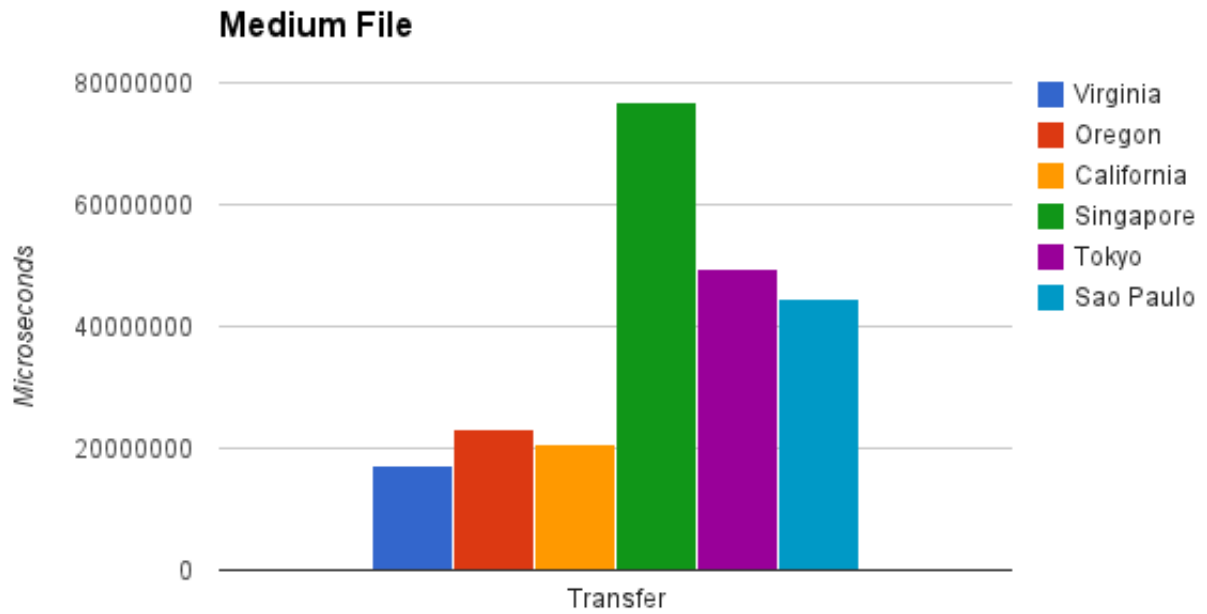
being read. While this approach is better on system resources, it limits network utilization.

Location

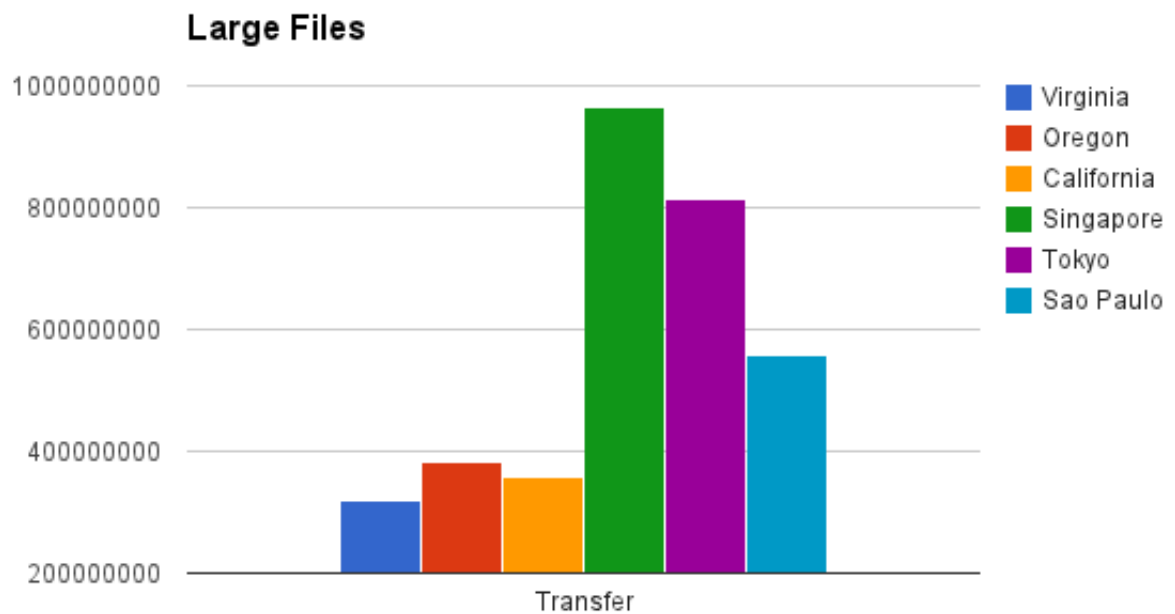
As was to be expected, send times for small files were highly correlated with location. Those in the continental United States, saw drastically lower send times. Since the small file fit into a single packet, the times listed are essentially the round trip times. If one was going to tune the sending protocol, it would make sense to use these values to tune the timeouts.



The results for the medium files was similar, although the packets travelling within the continental United States were a bit closer to each other than in the small files.

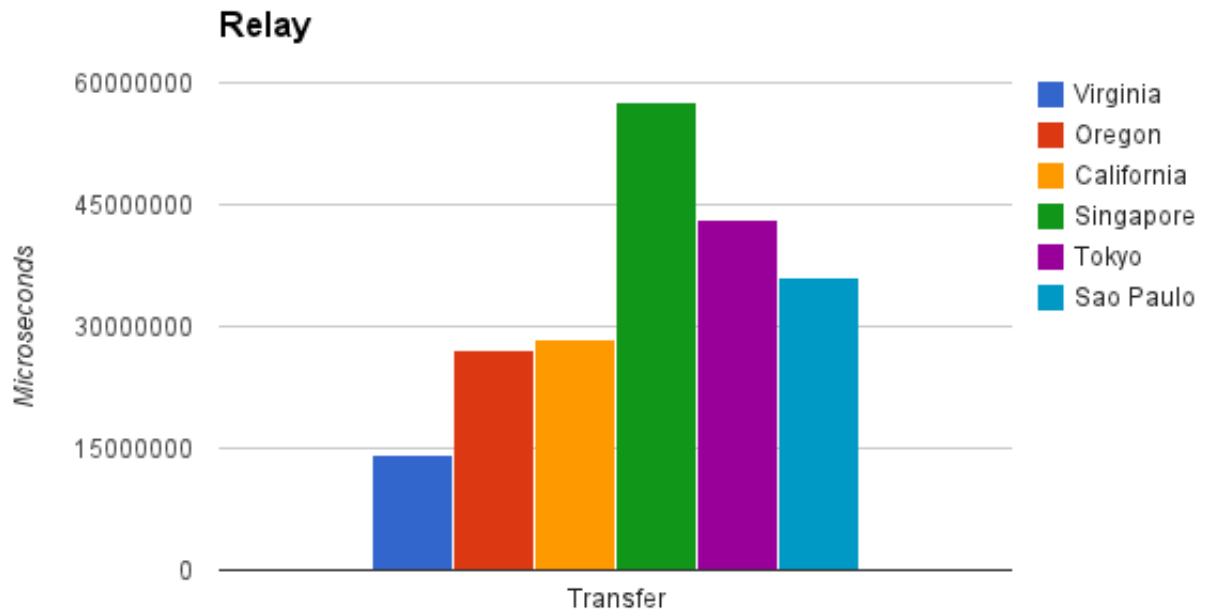


For the large files, the results were similar.



Relays

To test the relays, I sent packets to all of the servers through the Virginia relay. The results are below and similar to the previous results.



Different Implementations

For this test, I sent a large file to both the class server in Virginia and to my server in Virginia. My expectation is that the class server will much faster than my own. The results are that my server took almost 210 seconds (208,886,637 microseconds) and I had to resend 2 packets. With the class server, it took 187 seconds (187,852,352) to send the file. This was much closer than I thought it would be, as my server is caching the packets and saving them to disk once they are all successfully sent.

Going forward, I think I could prioritize the sending of acks. This would decrease the send time to my server.

TestScript

The test script is listed here for convenience

```
#!/bin/sh

make all
#####
#####
HR='++++++'
  '
echo $HR
echo "Test Run at: `date`"
echo $HR

#####
#####
FN="temp.txt"
BIG="diver.raw"
```



```
MED="new.jpg"
SMALL="test.txt"

SERVERPORT='54323'
RELAYPORT='54322'

VIRGINIA='54.84.21.227'
OREGON='54.213.83.180'
CALIFORNIA='54.193.35.191'
IRELAND='54.194.234.13'
SINGAPORE='54.251.177.165'
TOKYO='54.199.136.22'
SAOPAULO='54.207.17.177'

#####
#####
echo $HR
echo SENDING TO TOKYO WITH VARIOUS TIMEOUTS SMALL FILE
#####
#####
echo $HR

echo 300 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR

echo 3000 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR

echo 10000 > $FN # Time between reads
echo 100 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR

echo 225000 > $FN # Time between reads
echo 3 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR
```

```
echo SENDING TO TOKYO WITH VARIOUS TIMEOUTS MEDIUM FILE
#####
#####
echo $HR

echo 300 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR

echo 3000 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR

echo 10000 > $FN # Time between reads
echo 100 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR

echo 225000 > $FN # Time between reads
echo 3 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $TOKYO < $FN

#####
#####
echo $HR
echo SENDING TO VIRGINIA WITH VARIOUS TIMEOUTS SMALL FILE
#####
#####
echo $HR

echo 300 > $FN # Time between reads
echo 10 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR
```

```
echo 300 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 3000 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 10000 > $FN # Time between reads
echo 100 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 225000 > $FN # Time between reads
echo 3 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR
echo SENDING TO VIRGINIA WITH VARIOUS TIMEOUTS MEDIUM FILE
#####
#####
echo $HR

echo 300 > $FN # Time between reads
echo 10 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 300 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN
```

```
./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 3000 > $FN # Time between reads
echo 200 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 10000 > $FN # Time between reads
echo 100 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR

echo 225000 > $FN # Time between reads
echo 3 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MED >> $FN
echo 2 >> $FN

./myClient $VIRGINIA < $FN

#####
#####
echo $HR
echo SENDING SMALL MESSAGES TO HOSTS
#####
#####
echo 3000 > $FN # Time between reads
echo 300 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $SMALL >> $FN
echo 2 >> $FN

echo $HR
echo SENDING TO VIRGINIA
echo $HR
./myClient $VIRGINIA < $FN
echo $HR
echo SENDING TO OREGON
echo $HR
./myClient $OREGON < $FN
echo $HR
echo SENDING TO CALIFORNIA
echo $HR
./myClient $CALIFORNIA < $FN
echo $HR
echo SENDING TO IRELAND
echo $HR
./myClient $IRELAND < $FN
echo $HR
echo SENDING TO SINGAPORE
```

```
echo $HR
./myClient $SINGAPORE < $FN
echo $HR
echo SENDING TO TOKYO
echo $HR
./myClient $TOKYO < $FN
echo $HR
echo SENDING TO SAOPAULO
echo $HR
./myClient $SAOPAULO < $FN

#####
#####
echo $HR
echo SENDING MEDIUM MESSAGES TO HOSTS
#####
#####
echo 30000 > $FN # Time between reads
echo 300 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $MEDIUM >> $FN
echo 2 >> $FN

echo $HR
echo SENDING TO VIRGINIA
echo $HR
./myClient $VIRGINIA < $FN
echo $HR
echo SENDING TO OREGON
echo $HR
./myClient $OREGON < $FN
echo $HR
echo SENDING TO CALIFORNIA
echo $HR
./myClient $CALIFORNIA < $FN
echo $HR
echo SENDING TO IRELAND
echo $HR
./myClient $IRELAND < $FN
echo $HR
echo SENDING TO SINGAPORE
echo $HR
./myClient $SINGAPORE < $FN
echo $HR
echo SENDING TO TOKYO
echo $HR
./myClient $TOKYO < $FN
echo $HR
echo SENDING TO SAOPAULO
echo $HR
./myClient $SAOPAULO < $FN

#####
#####
echo $HR
echo SENDING BIG MESSAGES TO HOSTS
#####
#####
echo 30000 > $FN # Time between reads
echo 300 >> $FN # Number of times
echo 1 >> $FN # Send a file
echo $BIG >> $FN
echo 2 >> $FN

echo $HR
echo SENDING TO VIRGINIA
echo $HR
./myClient $VIRGINIA < $FN
echo $HR
```

```

echo SENDING TO OREGON
echo $HR
./myClient $OREGON < $FN
echo $HR
echo SENDING TO CALIFORNIA
echo $HR
./myClient $CALIFORNIA < $FN
echo $HR
echo SENDING TO IRELAND
echo $HR
./myClient $IRELAND < $FN
echo $HR
echo SENDING TO SINGAPORE
echo $HR
./myClient $SINGAPORE < $FN
echo $HR
echo SENDING TO TOKYO
echo $HR
./myClient $TOKYO < $FN
echo $HR
echo SENDING TO SAOPAULO
echo $HR
./myClient $SAOPAULO < $FN

#####
#####
echo $HR
echo SENDING MESSAGES TO HOSTS THROUGH RELAY
#####
#####
echo 10000 > $FN # Time between reads
echo 100 >> $FN # Number of times 9437184
echo 1 >> $FN # Send a file
echo $MEDIUM >> $FN
echo 2 >> $FN

echo $HR
echo SENDING TO VIRGINIA THROUGH VIRGINIA
echo $HR
./myClient $VIRGINIA $SERVERPORT $VIRGINIA $RELAYPORT < $FN
echo $HR
echo SENDING TO OREGON THROUGH VIRGINIA
echo $HR
./myClient $OREGON $SERVERPORT $VIRGINIA $RELAYPORT < $FN
echo $HR
echo SENDING TO CALIFORNIA THROUGH VIRGINIA
echo $HR
./myClient $CALIFORNIA $SERVERPORT $VIRGINIA $RELAYPORT < $FN
echo $HR
echo SENDING TO IRELAND THROUGH VIRGINIA
echo $HR
./myClient $IRELAND $SERVERPORT $VIRGINIA $RELAYPORT < $FN
echo $HR
echo SENDING TO SINGAPORE THROUGH VIRGINIA
echo $HR
./myClient $SINGAPORE $SERVERPORT $VIRGINIA $RELAYPORT < $FN
echo $HR
echo SENDING TO TOKYO THROUGH VIRGINIA
echo $HR
./myClient $TOKYO $SERVERPORT $VIRGINIA $RELAYPORT < $FN
echo $HR
echo SENDING TO SAOPAULO THROUGH VIRGINIA
echo $HR
./myClient $SAOPAULO $SERVERPORT $VIRGINIA $RELAYPORT < $FN

```

Raw Output

The raw output of the test script is listed below.

```

make: Nothing to be done for `all'.
+++++
+++++
Test Run at: Sun Mar 9 21:29:14 CDT 2014
+++++
+++++
+++++
+++++
SENDING TO TOKYO WITH VARIOUS TIMEOUTS SMALL FILE
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 300 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 240053 microseconds.
Sent: 1
Resent: 3

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 49099 microseconds.
Sent: 1
Resent: 0

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 100779 microseconds.
Sent: 1
Resent: 0

1. Send a file
2. Exit
+++++
+++++

```

```
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 225000 microseconds.
Enter the number of times to try [200]:
Using 3 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 225078 microseconds.
Sent: 1
Resent: 0

1. Send a file
2. Exit
+++++
+++++
SENDING TO TOKYO WITH VARIOUS TIMEOUTS MEDIUM FILE
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 300 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 41118839 microseconds.
Sent: 176
Resent: 451

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 37358254 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit
```



```
Enter the filename:
This transfer took 40789942 microseconds.
Sent: 176
Resent: 2

1. Send a file
2. Exit
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 225000 microseconds.
Enter the number of times to try [200]:
Using 3 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 53800665 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
      +++++
SENDING TO VIRGINIA WITH VARIOUS TIMEOUTS SMALL FILE
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 300 microseconds.
Enter the number of times to try [200]:
Using 10 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 56983 microseconds.
Sent: 1
Resent: 15

1. Send a file
2. Exit
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 300 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 115563 microseconds.
Sent: 1
Resent: 1

1. Send a file
2. Exit
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 200 times.
```

1. Send a file
2. Exit

```
Enter the filename:
This transfer took 21513 microseconds.
Sent: 1
Resent: 0
```

1. Send a file
2. Exit

```
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.
```

1. Send a file
2. Exit

```
Enter the filename:
This transfer took 50399 microseconds.
Sent: 1
Resent: 0
```

1. Send a file
2. Exit

```
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 225000 microseconds.
Enter the number of times to try [200]:
Using 3 times.
```

1. Send a file
2. Exit

```
Enter the filename:
This transfer took 225104 microseconds.
Sent: 1
Resent: 0
```

1. Send a file
2. Exit

```
+++++
+++++
SENDING TO VIRGINIA WITH VARIOUS TIMEOUTS MEDIUM FILE
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 300 microseconds.
Enter the number of times to try [200]:
Using 10 times.
```

1. Send a file
2. Exit

```
Enter the filename:
```

```
This transfer took 141553656 microseconds.
Sent: 176
Resent: 38838

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 300 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 19546616 microseconds.
Sent: 176
Resent: 168

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 200 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 11878706 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 12447982 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 225000 microseconds.
Enter the number of times to try [200]:
```

Using 3 times.

1. Send a file
2. Exit

Enter the filename:

This transfer took 40071541 microseconds.

Sent: 176

Resent: 0

1. Send a file
2. Exit

+++++

+++++

SENDING SMALL MESSAGES TO HOSTS

+++++

+++++

SENDING TO VIRGINIA

+++++

+++++

myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:

Using 3000 microseconds.

Enter the number of times to try [200]:

Using 300 times.

1. Send a file
2. Exit

Enter the filename:

This transfer took 67497 microseconds.

Sent: 1

Resent: 0

1. Send a file
2. Exit

+++++

+++++

SENDING TO OREGON

+++++

+++++

myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:

Using 3000 microseconds.

Enter the number of times to try [200]:

Using 300 times.

1. Send a file
2. Exit

Enter the filename:

This transfer took 131993 microseconds.

Sent: 1

Resent: 0

1. Send a file
2. Exit

+++++

+++++

SENDING TO CALIFORNIA

+++++

+++++

myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:

Using 3000 microseconds.

Enter the number of times to try [200]:

Using 300 times.

```
1. Send a file
2. Exit
```

```
Enter the filename:
This transfer took 79823 microseconds.
Sent: 1
Resent: 0
```

```
1. Send a file
2. Exit
```

```
+++++
      +++++
SENDING TO SINGAPORE
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 300 times.
```

```
1. Send a file
2. Exit
```

```
Enter the filename:
This transfer took 411216 microseconds.
Sent: 1
Resent: 0
```

```
1. Send a file
2. Exit
```

```
+++++
      +++++
SENDING TO TOKYO
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 300 times.
```

```
1. Send a file
2. Exit
```

```
Enter the filename:
This transfer took 230180 microseconds.
Sent: 1
Resent: 0
```

```
1. Send a file
2. Exit
```

```
+++++
      +++++
SENDING TO SAOPAULO
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)
```

```
Enter time to sleep between reads in microseconds [3000]:
Using 3000 microseconds.
Enter the number of times to try [200]:
Using 300 times.
```

```
1. Send a file
2. Exit
```

```
Enter the filename:
This transfer took 257751 microseconds.
Sent: 1
Resent: 0

1. Send a file
2. Exit
make: Nothing to be done for `all'.
+++++
+++++
Test Run at: Sun Mar 9 22:54:35 CDT 2014
+++++
+++++
+++++
SENDING MEDIUM MESSAGES TO HOSTS
+++++
+++++
SENDING TO VIRGINIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 17209432 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
SENDING TO OREGON
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 23074420 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
SENDING TO CALIFORNIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
```

Using 300 times.

1. Send a file
2. Exit

Enter the filename:

This transfer took 20851416 microseconds.

Sent: 176

Resent: 0

1. Send a file
2. Exit

++++++
++++++

SENDING TO SINGAPORE

++++++
++++++

myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:

Using 30000 microseconds.

Enter the number of times to try [200]:

Using 300 times.

1. Send a file
2. Exit

Enter the filename:

This transfer took 76793016 microseconds.

Sent: 176

Resent: 3

1. Send a file
2. Exit

++++++
++++++

SENDING TO TOKYO

++++++
++++++

myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:

Using 30000 microseconds.

Enter the number of times to try [200]:

Using 300 times.

1. Send a file
2. Exit

Enter the filename:

This transfer took 49634648 microseconds.

Sent: 176

Resent: 1

1. Send a file
2. Exit

++++++
++++++

SENDING TO SAOPAULO

++++++
++++++

myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:

Using 30000 microseconds.

Enter the number of times to try [200]:

Using 300 times.

1. Send a file
2. Exit

```
Enter the filename:
This transfer took 44582901 microseconds.
Sent: 176
Resent: 1

1. Send a file
2. Exit
+++++
+++++
SENDING BIG MESSAGES TO HOSTS
+++++
+++++
SENDING TO VIRGINIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 319007508 microseconds.
Sent: 2634
Resent: 2

1. Send a file
2. Exit
+++++
+++++
SENDING TO OREGON
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 384069940 microseconds.
Sent: 2634
Resent: 2

1. Send a file
2. Exit
+++++
+++++
SENDING TO CALIFORNIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit
```



```
Enter the filename:
This transfer took 358203024 microseconds.
Sent: 2634
Resent: 6

1. Send a file
2. Exit
+++++
      +++++
SENDING TO SINGAPORE
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 964195345 microseconds.
Sent: 2634
Resent: 21

1. Send a file
2. Exit
+++++
      +++++
SENDING TO TOKYO
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 813463648 microseconds.
Sent: 2634
Resent: 19

1. Send a file
2. Exit
+++++
      +++++
SENDING TO SAOPAULO
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 30000 microseconds.
Enter the number of times to try [200]:
Using 300 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 559483381 microseconds.
Sent: 2634
Resent: 4
```

```
1. Send a file
2. Exit
+++++
      +++++
SENDING MESSAGES TO HOSTS THROUGH RELAY
+++++
      +++++
SENDING TO VIRGINIA THROUGH VIRGINIA
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 14225801 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
      +++++
SENDING TO OREGON THROUGH VIRGINIA
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 27043088 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
      +++++
SENDING TO CALIFORNIA THROUGH VIRGINIA
+++++
      +++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 28343124 microseconds.
Sent: 176
Resent: 0
```

```
1. Send a file
2. Exit
+++++
+++++
SENDING TO SINGAPORE THROUGH VIRGINIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 57708929 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
SENDING TO TOKYO THROUGH VIRGINIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 43108432 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
+++++
+++++
SENDING TO SAOPAULO THROUGH VIRGINIA
+++++
+++++
myClient by William Montgomery (wmontg2@uic.edu)

Enter time to sleep between reads in microseconds [3000]:
Using 10000 microseconds.
Enter the number of times to try [200]:
Using 100 times.

1. Send a file
2. Exit

Enter the filename:
This transfer took 35953371 microseconds.
Sent: 176
Resent: 0

1. Send a file
2. Exit
```

