

Introduction to Computer Networks

Homework 1

Due: Oct. 16, 2023 23:55

Assignment 1:

Mail Client

In this assignment, you will create a simple mail client that sends email to any recipient. Your client will need to establish a TCP connection with a mail server (you may need to install a test server), dialogue with the mail server using the SMTP protocol, ask the user to input the recipient's email address and the mail content, send the email message to the recipient via the mail server, and finally close the TCP connection with the mail server.

Note: Since we discussed the socket API in C programming language in the class, we suggest that you write the program in C or C++ on a Linux machine. There is a tutorial on how to develop C programs on Linux on the course website. However, you are allowed to use other languages. The principle of thumb is that you must create the TCP socket to the server and send/receive the SMTP messages through the socket by your own program. Other libraries, like the `smtplib` or `email` modules in python, are not allowed to be used in your program.

Hand in:

1. The source code.
2. A readme file for how to run your program.

Assignment 2:

Iperf is a common tool used to measure network bandwidth. You will write your own version of this tool using sockets. You will then use your tools to measure the performance of certain networks. Your tool, called Iperfer, will send and receive TCP packets between a pair of hosts using sockets.

When operating in client mode, Iperfer will send TCP packets to a specific host for a specified time window and track how much data was sent during that time frame; it will calculate and display the bandwidth based on how much data was sent in the elapsed time. When operating in server mode, Iperfer will receive TCP packets and track how much data was received during the lifetime of a connection; it will calculate and display the bandwidth based on how much data was received and how much time elapsed between received the first and last byte of the data.

Client Mode

To operate Iperf in client mode, it should be invoked as follows:

Iperf -c -h <server hostname> -p <server port> -t <time>

- -c indicates this is the iperf client which should generate data
- server hostname is the hostname or IP address of the iperf server which will consume data
- server port is the port on which the remote host is waiting to consume data; the port should be in the range $1024 \leq \text{server port} \leq 65535$
- time is the duration in seconds for which data should be generated

You can use the presence of the -c option to determine Iperf should operate in client mode. If any arguments are missing or additional arguments are provided, you should print the following and exit:

```
Error: missing or additional arguments
```

If the server port argument is less than 1024 or greater than 65535, you should print the following and exit:

```
Error: port number must be in the range 1024 to 65535
```

When running as a client, Iperf must establish a TCP connection with the server and send data as quickly as possible for time seconds. Data should be sent in chunks of 1000 bytes and the data should be all zeros. Keep a running total of the number of bytes sent. After time seconds have passed, Iperf must stop sending data and close the connection. Iperf must print a one line summary that includes:

- The total number of bytes sent (in kilobytes)
- The rate at which traffic could be sent (in megabits per second (Mbps))

For example:

```
sent=6543 KB rate=5.234 Mbps
```

You should assume 1 kilobyte (KB) = 1000 bytes (B) and 1 megabyte (MB) = 1000 KB. As always, 1 byte (B) = 8 bits (b).

Server Mode

To operate Iperf in server mode, it should be invoked as follows:

Iperf -s -p <listen port>

- -s indicates this is the iperf server which should consume data

- listen port is the port on which the host is waiting to consume data; the port should be in the range $1024 \leq \text{listen port} \leq 65535$

You can use the presence of the -s option to determine Iperfer should operate in server mode. If arguments are missing or additional arguments are provided, you should print the following and exit:

```
Error: missing or additional arguments
```

If the listen port argument is less than 1024 or greater than 65535, you should print the following and exit:

```
Error: port number must be in the range 1024 to 65535
```

When running as a server, Iperfer must listen for TCP connections from a client and receive data as quickly as possible until the client closes the connection. Data should be read in chunks of 1000 bytes. Keep a running total of the number of bytes received. After the client has closed the connection, Iperfer must print a one line summary that includes:

- The total number of bytes received (in kilobytes)
- The rate at which traffic could be read (in megabits per second (Mbps))

For example:

```
received=6543 KB rate=4.758 Mbps
```

The Iperfer server should shut down after it handles one connection from a client.

Testing

You can test Iperfer on any machines you have access to. **You should receive the same number of bytes on the server as you sent from the client.** However, the timing on the server may not perfectly match the timing on the client. Hence, the bandwidth reported by client and server may be slightly different. In general, they should not differ by more than 2 Mbps. Note, this behavior mirrors the behavior of the actual iperf tool.

Hand in:

3. The source code.
4. A short description for how to compile and run your program.