

COSC402 Lab 4: Socket Options, DNS & IPv6

Part I: Socket Options & Performance

In this lab, you will practice to:

- get and set socket options using `getsockopt()` and `setsockopt()`;
- optimise network throughput.

Lab setup

Change into `/402lab/socketopt`, read the `checkouts.c`. Compile `checkouts` and run it to check the default settings for socket options.

Measure and optimise network throughput

Programming

Copy `Lab4` from the `pickup` directory to `402lab`. Implement the following functions:

- (1) `set_buf_size()`: set a new value to a socket option.
- (2) `print_buf_size()`: print the sizes of the transmit and receive buffers.

After implementing these two functions/procedures, copy the two source files to the `/lib` folder. Open the compiling configuration file `Make.defines` in the `402lab` directory, and add `set_buf_size.o` and `print_buf_sizes.o` to `LIB_OBJS`. Change into the `lib` folder, and type `make` to recompile the library functions.

In the `tcp_sender.c` and `tcp_receiver.c` source files, add corresponding codes to set the value for `SO_SNDBUF` for the sender and `SO_RCVBUF` for the receiver. Note that:

- the sizes of the send and receive buffer are inputted from the command line and stored in the global variables `size_sndbuf` and `size_rcvbuf`, respectively (see the main function).
- when setting the size of the receive buffer, the ordering of the function call is important. Be careful with the position where the codes to be put in.

Throughput measurement

Perform measurements on a pair of sender and receiver, with buffer sizes 2k, 4k, 8k, 16k, 32k, 62k, 128k, respectively. Note that send buffer and receive buffer should be set to the same size in each run.

```
%receive:
./tcp_receiver 127.0.0.1 65530    RCV_BUF_SIZE
```

In each run, replace `RCV_BUF_SIZE` with the actual buffer size. Note that the unit for this parameter is Kbyte, if you want to set the receive buffer to be 2K bytes, `RCV_BUF_SIZE` should be set with a value of 2.

```
% send a gig
time ./tcp_sender 127.0.0.1 65530  SND_BUF_SIZE  AMOUNT_DATA
```

Replace `SND_BUF_SIZE` with the exact buffer size and `AMOUNT_DATA` with the amount of data you want to transmit (unit is MB). Plot a graph to show the throughput achieved with different buffer size setting.

If you want to test it in a network environment, you'd better use two machines in the lab in case that it produces unuseful Internet traffic.

Part II: DNS & IPv6

In this lab, you will practice:

- using `getaddrinfo()` to do name-to-address and server-to-port translation.
- setting up TCP connections using IPv6.

Programming

Change into *Lab4*, and complete the following functions based on the provided skeleton code.

`tcp_easy_connect()` in `tcp_easy_ipv46_client.c`

- use `getaddrinfo()` to do name-to-address translation.
- try to connect to the server using the returned socket addresses until success or all socket addresses have been tried.

`tcp_easy_listen()` in `tcp_easy_ipv46_server.c`

- use `getaddrinfo()` to do name-to-address translation.
- try to bind one returned socket address until success or all socket addresses have been tried.

Testing

Change the setting for *ai_family* between *AF_INET*, *AF_INET6*, and *AF_UNSPEC*, and see if you can set up the connection using IPv4 or IPv6.