Dependencies

Using Ubuntu 18.04, install the dependencies:

sudo apt-get install g++, libsctp-dev, lksctp-tools

For the server/client architecture we are going to utilize Apache server. In order to do so, we must first install it, if it is not present in our system using the command: sudo apt-get install build-essential apache2. After that, we must enable the cgi module with the command: sudo a2enmod cgi. Finally, Apache must be restarted, by typing: service apache2 restart.

Compilation

The compilation of the source code for the SCTP is done with the commands:
g++ client-sctp.cc SenderContent.cc -lsctp -o sctp_client
g++ server-sctp.cc SenderContent.cc -lsctp -o sctp_server

The compilation of the source code for the DCCP is done with the commands:
g++ client-dccp.cc SenderContent.cc -o dccp_client
g++ server-dccp.cc SenderContent.cc -o dccp_server

Command line arguments

The command line arguments for the SCTP client are:
-a : for the IP address of the server
-d : the number of bytes to transfer
-t : TTL for the data transmission
-s : the number of streams
-u : ordered or unordered transmission
-n : number of transmission cycles (e.g Two cycles of 8 bytes will transmit in total 16 bytes)
-b : time to wait between cycles in microseconds

The command line arguments for the DCCP client are:
-a : for the IP address of the server
-d : the number of bytes to transfer
-r : data transmission rate kbytes/sec
-s : the number of sockets
-n : number of transmission cycles (e.g Two cycled of 8 bytes will transmit in total 16 bytes)
-b : time to wait between cycles in microseconds

Setup – Preparation

The path where our executables will reside in order to be recognized by the Apache is /usr/lib/cgi-bin/. The two server-side files needed for the DCCP and SCTP exchange will be dccp_server and sctp_server and to begin the transfer we will need first to open a browser and make a request of type: http://<IP>/cgi-bin/<file>?bytes=<number> where IP is the server's IP address, file can be either dccp_server or sctp_server and number is the total number of bytes the server will be waiting to receive. After this, we can execute our client with the correct arguments, ./sctp_client -d 8 -a 127.0.0.1 -b 100 -s 1 -n 2.

Examples

Except from the provided pcaps where the whole traffic is shown, an example of execution can also be seen if we take a look at the images provided below (see “Image 1” and “Image 2”) where the output after a successful execution and transfer of data using SCTP is shown.

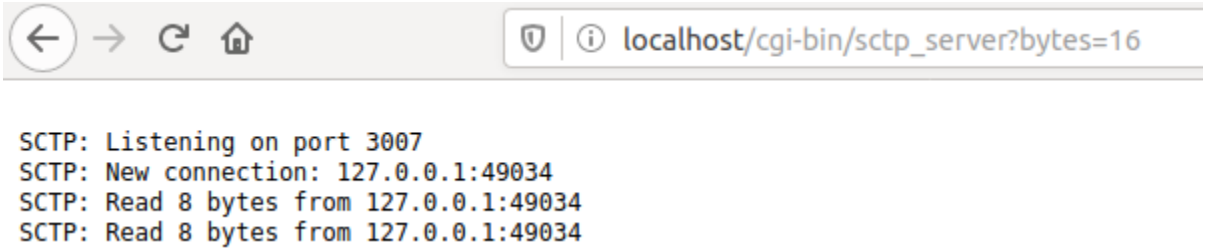

Image 1: Response from server after successful execution.

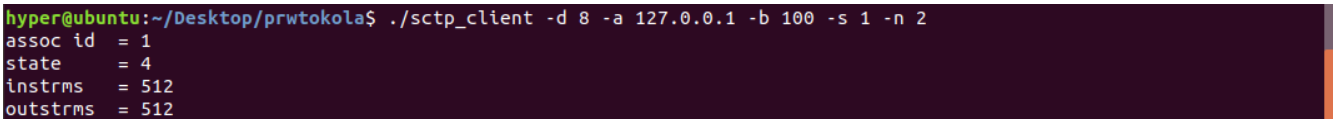

Image 2: Result of client after successful execution.