Lab 1 – Introduction to Socket Programming –

TCP Sockets

Objective

* Review how to build a simple network.
* To send and receive messages using a TCP client/server application.
* Get familiar with source and destination port numbers at the transport layer.
* Inspect the TCP connection handshake.

PLEASE GET ALL YOUR WORK VERIFIED BY LAB INSTRUCTOR BEFORE LEAVING

Task 0: Download files from Canvas first

In the next tasks we will be disconnected from the Internet. So please download the following files now at the beginning of the lab:

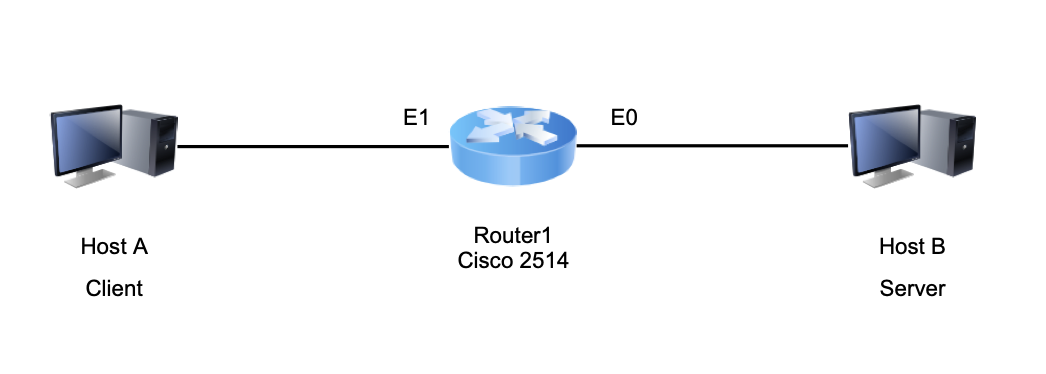
1. From our Week 1 Canvas module, download tcpclient\_lab1.py and tcpserver\_lab1.py. There is also a zip file inside the lab assignment with these two Python files. Do not click on them or try to run them, just save it in your local directory. Note: make sure to create a new folder for our labs as these files will be used in next labs.
2. If your browser does not allow you to download Python files, try to open IDLE, create a new file and just copy and paste the code in Canvas. (You can download Python from <https://www.python.org/downloads/>
3. Make sure to have Wireshark installed in your laptop. You can download Wireshark from this link: <https://www.wireshark.org>

Task 1: Setting up for communication between hosts (2-Person Team, or 3-Person team depending on the lab bench)

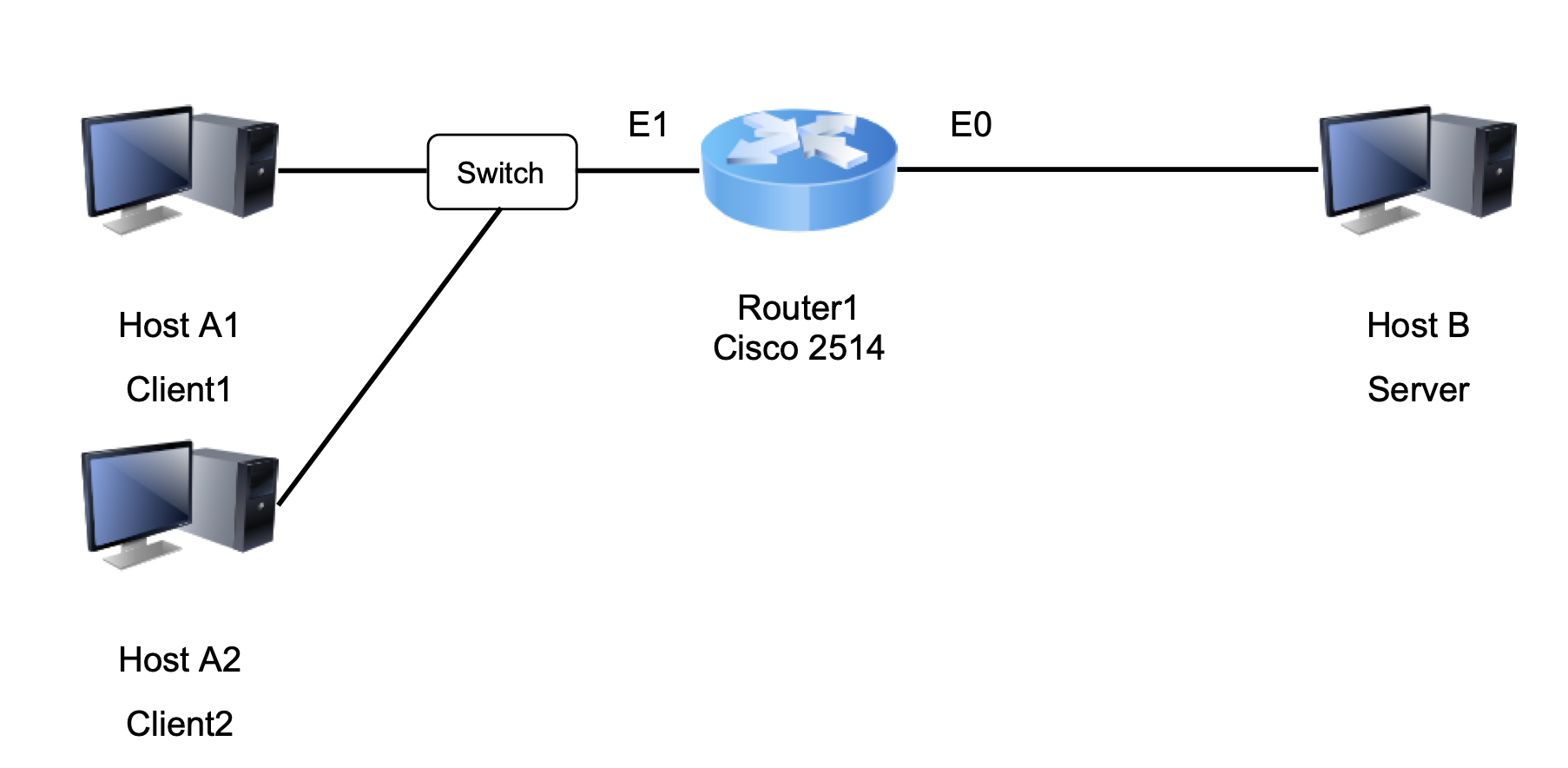
1. In Figure 1 of Figure 2, circle all the subnetworks or subnets (please remember that for a given router, each router interface is in a different network, because routers interconnect different networks; i.e., they interconnect different subnets)
2. Assign IP address to all the subnets, hosts, and router interfaces in Figure 1 or 2. Assume IP address space:

30.30.30.0/24

Please choose your own subnet masks (please see table 1 for reference) and make sure the IP addresses from one subnet do not overlap with another subnet.



*Figure 1 – Simple network with two subnets (for lab benches with two seats).*



*Figure 2 – Simple network with two subnets (for lab benches with 3 seats).*

*Table 1: Review of Subnet Masks*

|  |  |  |
| --- | --- | --- |
| *Subnet Mask (number of 1’s)* | *Subnet Mask in Decimal Form* | *Max Number of Possible IP Addresses* |
| */24* | *255.255.255.0* | *28 - 2 = 254* |
| */25* | *255.255.255.128* | *27 - 2 = 126* |
| */26* | *255.255.255.192* | *26 - 2 = 62* |
| */27* | *255.255.255.224* | *25 - 2 = 30* |
| */28* | *255.255.255.240* | *24 - 2 = 14* |
| */29* | *255.255.255.248* | *23 - 2 = 6* |
| */30* | *255.255.255.252* | *22 - 2 = 2* |

Show TA your labeled network with all IPs and subnets before you start configuring the devices.

**Connecting to Router’s Console Port**

First, use the documentation on Canvas to connect to the terminal (or Console) interface on the router. Use your personal computer with Windows or MacOS. (Check out the necessary cables and adapters with the TA).

The Week 1 Module in Canvas has instructions on how to use your device. We have instruction to connect the Console Port (serial cable) and also the Ethernet port. Here is the link: <https://canvas.tamu.edu/courses/210965/modules#module_1001719>

Make sure to open these files and check if you already have Putty (if you have Windows) or know how to configure the devices using Terminal if you have a MacBook.

* for your MacOS laptop.
* for your Windows device

**Configure the Router**

Turn on Cisco 2514 or Cisco 4221 Router. Provide some time for the router to boot. (**If a prompt is shown asking to enter the initial configuration, type “no”**). After the booting of the Internet Operating System (IOS) is complete, the prompt of the router will appear.

On the router’s prompt, you can see all the router’s interfaces and their status by typing:

Router> **show interfaces** (or in a short form, **sh int**)

Enter Privileged EXEC mode by typing “**enable**” in the command prompt. Privileged EXEC mode allows you to change the settings of the router. Users will not be able to change the settings of a router without getting into this mode. (Note: to enter this mode, in a typical router you would need to enter a password. For simplicity, we have not configured passwords in the routers.) See how the prompt changed from “Router>” to “Router#”

Enter configuration mode by typing “**config terminal**” (or “**config t**”) in the command prompt.

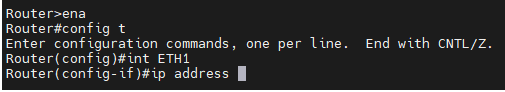
Specify interface Ethernet0 for the Interface level (submenu of configuration mode). Type “**int e0**” (for Cisco 2514) or **“int g0/0/0”** (for Cisco 4221) in the command prompt (**int** is the short for interface).

Assign interface Ethernet0 with IP address **(the IP addresses you chose for the computers in Figure 1 or Figure 2)**. You can assign the IP address and the subnet mask to the Ethernet interface – Ethernet 0, by typing for instance “**ip address 10.10.10.1 255.255.255.0**” in the command prompt.

Type “**no shutdown**”to enable the interface to send and receive packets.

Type “**end**” to get out of Privileged EXEC mode.

Type “**show running-config**” **(or “sh run”)** (to see your changes). Check if your IP address that you assigned on Step 7 was accepted to the router’s configuration.



*Figure 5: Configure Interfaces for Routers. Here it can be ETH1 or E1. Also, make sure to do a “no shutdown” command.*

**Hint:** You can use “Tab” to complete your commands. And whenever you forget a command, just type “?”. For instance: “show ?” or “ip address ?”

Make sure to configure the correct IP address in the hosts and connect them to the router. Then test using ping command between Host A and Host B.

**Question 1**: Explain your network and the IP addresses you chose (make sure to show a network diagram with the IP addresses and subnet masks clearly shown). Then, show that you achieved connectivity by showing the screenshots of your ping results.

Show TA your configuration and your pings before moving forward.

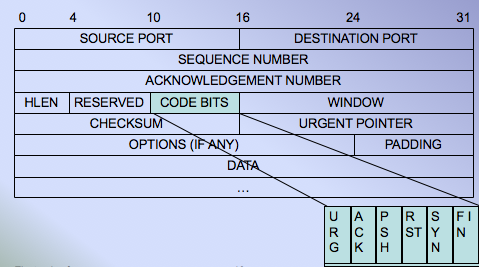
Task 2: Socket Programming (writing a client/server application)

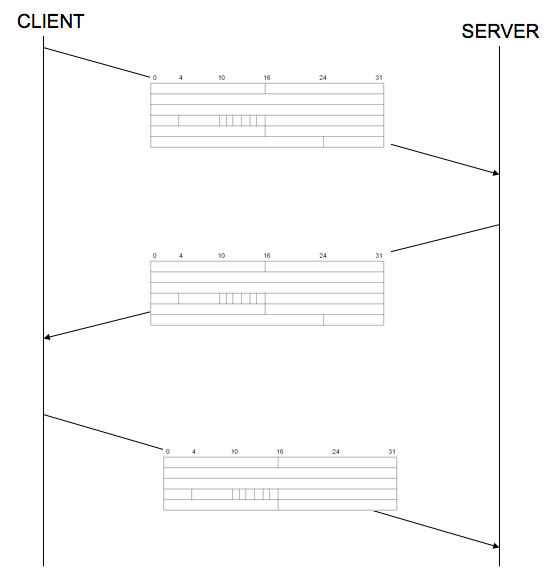
1. One person will act as the server and the other as the client (or two clients).
2. Now search for the Python graphical user interface (GUI) called IDLE under the start button and open the Python shell.
3. One team member opens the tcpserver\_lab1.py file. Make sure to change the IP address in this program to have your laptop’s IP address (based on our simple network in Figure 1 or Figure 2). You should change the variable: *host = “your ip address”*
4. First one team member starts the **server**:
   1. click on “run module”. You should see a message “waiting for connection on port 10000”. This means that the server is running all the time, and waiting for requests from other hosts. *Note: in case you need to stop the sever at a later time, use Ctrl + C.*
5. Now, the client person should go to “File” and open the tcpclient\_lab1.py file.
   1. You should change the Host IP and Port # to match the server’s which is your team member’s *Host IP and Port #.*
6. Now the client person should run his/her program.

Show TA that you are able to connect with each other.

Note that this is simple two-message application. The client sends a message that you type in the keyboard, and the server receives it and sends a response to the client.

**Question 2:** Start a Wireshark capture at either the client or server computer. You can filter your capture by typing: tcp.port == 10000. Now run the client program and observe the connection setup messages. Show a screenshot of your connection setup (3 packets) and draw the diagram of the connection handshake here showing the details of the TCP packet header from your packet capture (complete the diagram using the same values that you captured in your Wireshark packets) .





*Figure 6 - TCP connection setup*

**Question 3:** Now run the client program several times (at least 5 times), and at each time observe the request and reply messages, and the source and destination port numbers being used. Fill in the table below and add it to your report, and explain what you understood from the table. Which port numbers remain the same? Which port numbers change? Why?

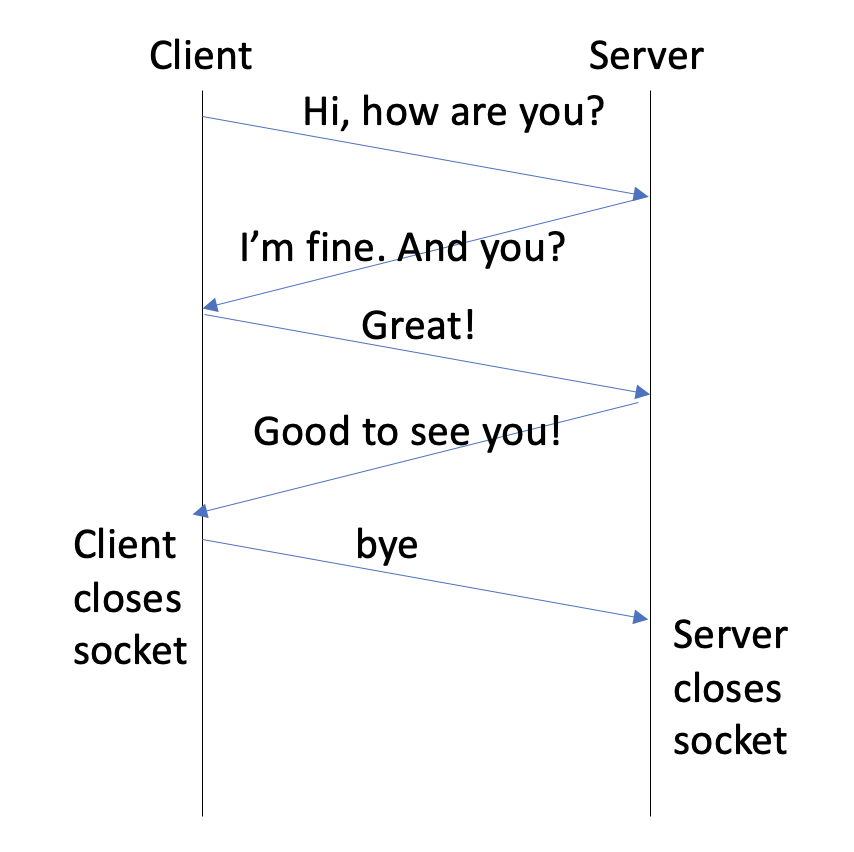
*Table 1: Observed port numbers*

|  |  |  |  |
| --- | --- | --- | --- |
| Packets sent by the Client | | Packets sent by the Server | |
| Source Port # | Destination Port # | Source Port # | Destination Port # |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Task 3: Modifying the client and server program

Work with your team to modify the server code to create a CHAT application. Here are some steps:

* Create a while loop on the server to make sure it keeps receiving and sending messages
* Create a while loop on the client so that the client can reply after it receives a message from the server.
* Only when the client sends a “bye” the client and server will close the socket, as in Figure 7.



*Figure 7 - Sample chat exchange between client and server*

**Question 4:** Start a Wireshark capture at either the client or server computer. Start a new chat and filter your TCP packets (tcp.port==10000). Inspect one of the packets to show that you can read the message typed in the chat. Please add a screenshot to show this unencrypted conversation.

Make sure to submit your new client and server Python code, with the latest modification. Make sure to add comments to explain what you did.

Finally, create your Groups on Canvas (go to People, Lab1 groups).

### Lab Reports Format

*Here are some guidelines for your lab report:*

1. ***Section 1 - Introduction*** *- What is this lab about? Explain in your words the main topic. What tasks you did in the lab? Please write a summary in your own words of what you had to do in the lab; a description of the main steps*
2. ***Section 2 – Results and answers to questions*** *– describe results from the lab tasks (e.g., any screenshots to show your work, tables, graphs) with a good explanation. Make sure to answer all post-lab questions. Remember all figure need to have a caption, and they need to be referenced in the text.*
3. ***Section 3 – Discussions (Conclusions)*** *– What are the key concepts that you learned in this lab? After completing the lab, please reflect on the lessons learned, if it helped you understand the topic, or any issues you had during the lab*
4. ***Appendix (cumulative)*** *- Reference list of formulas/commands used in the lab (to be appended after every lab)*