

ARIZONA STATE UNIVERSITY  
CSE 434, SLN 70569 — Computer Networks — Fall 2021

**Lab #1**

Lab report due electronically on Canvas before 11:59pm on Sunday, 09/12/2021

This lab has two major parts:

1. An introduction to the racks in BYENG 217 in §1 through exercises and experiments with a single-segment IP network in §2. Be sure to take a USB flash drive to the lab with you to save data from your experimentation.
2. An introduction to GENI in §3 through an experiment to set up your `ssh` keys, and solving a problem in designing subnets on GENI with exercises and experiments in §4.

## Preliminaries

### Join a Group on Canvas, Reserve a Timeslot on a Rack in BYENG 217

This lab may be completed individually, or in a group of size two. Either way, everyone must join a group because the lab is a group assignment. On Canvas, go to People/Groups and choose the **Lab Groups** tab. Self sign-up is enabled.

Once you've created a group, sign up for a three-hour slot on one of the six racks in BYENG 217 [using this sign-up sheet](#).

### Lab Report Format

For each exercise:

1. Give the exercise number.
2. Include the output and/or screen shots as requested in the exercise.
3. Provide explanations and/or answer any questions, as requested in the exercise.

As you prepare your report, extract the portion of your output (saved to a file, or screen capture) that relates directly to the exercise. Do not just copy and paste in pages of output hoping the grader can find what's relevant; that is your determination. All screen shots must include the ASUrite id of a group member.

Include your group name, and the names of the member(s) in the group on the first page of your report.

## 1 Introduction to the Racks in BYENG 217

The exercises in this section of the lab introduce you to commands used later in this lab and in future labs.

1. **man pages:** The machines run the Linux operating system. This lab asks you to review some Linux commands. Manual pages (`man` pages) exist on every lab machine. See also [Linux Man Pages Online](#). If you are not familiar with Linux, it is suggested that for each of the following commands, type the name of the command as a search term. The search will return the appropriate man page to read.

```
man mv rmdir pwd cp chmod tcpdump
ls rm kill more mkdir ping
```

2. **Wireshark:** The man page for `Wireshark`, a network analyzer tool, can be found on every lab machine. You can also read more about the tool at the [Wireshark network analyzer site](#). The manual pages of Wireshark can be found under "Command-line Manual Pages."

## Questions

It is expected that you know the answers to the questions below. If you do not, you should read the appropriate man pages.

1. How can you use the command `ls` to find the size of file `/etc/grub.conf`?
2. What happens if you have two files with names `file1` and `file2` and you type `mv file1 file2`? Which option of `mv` issues a warning in this situation?
3. What is the command that you issue if you are in directory `/` and want to copy the file `/mydata` to directory `/labdata`?
4. What is the command that you issue if you are in directory `/` and want to copy all files and directories under directory `/mydirectory` to directory `/newdirectory`?
5. What happens if you type the command `rm *` in a directory?
6. What is the command that you issue if you want to delete all files and directories under the directory `/mydirectory`?

### 1.1 Acquainting yourself with the Racks

In this part of Lab #1, you will acquaint yourself with the racks of equipment in BYENG 217, the Linux operating system, and some traffic measurement tools.

#### Becoming Familiar with the Equipment

The equipment that you are working with in the lab has a setup similar to that in Figure 1.



Figure 1: Lab equipment in BYENG 217.

Take a few minutes to compare the following description with the actual equipment:

- A 19 inch rack that houses most of the equipment.
- Four Linux PCs, which are labelled as *Computer A*, *Computer B*, *Computer C*, and *Computer D*. The PCs have the Redhat operating system installed. Each Linux PC has two Ethernet network interface cards (NICs) installed, which are labelled *p2p1* and *p2p2*.
- Four Cisco routers (silver, near top of rack beneath the SDN switch) are labelled as *Router A*, *Router B*, *Router C*, and *Router D*.
- Four Ethernet switches (blue, near middle of rack), each with eight ports. The data rates of the ports are 10 Mbps, 100 Mbps, or a dual speed at 10/100 Mbps.
- Two Ethernet hubs (white, near bottom of rack), each with 5 ports.
- A monitor, a keyboard, a mouse, and a KVM (keyboard-video-mouse) switch. The KVM switch connects the keyboard, monitor, and mouse to a PC. The KVM switch gives you control over all four Linux PCs from one keyboard, monitor, and mouse, but you can access only one computer at a time.
- Ethernet cables (hanging over top rack). Note that there are two kinds: straight-through Ethernet cables and crossover Ethernet cables. The crossover cables should be colour-coded or labelled.

### Using the KVM switch, logging in to a Linux PC, and exploring the desktop.

1. Set the KVM switch to computer A (the first light labelled A should light up). Log in as `root`. To do so, select “other” then enter `root` as the account you want to login to; no password is needed.
2. Use the KVM switch to switch to computer C and log in as root.
3. Explore the desktop environment of computer C.
4. Create a terminal window by selecting `Terminal` under the `Applications/Systems Tools` menu. Recall that all Linux commands can be typed from a terminal window.
5. Set the KVM switch to computer A and reboot it by typing `reboot` on the command line at the prompt in the terminal window.  
`[root@hostA ~]# reboot`

### Setup of the network.

In this part of the lab, all four Linux PCs are attached to the same Ethernet switch.

1. Attach each Linux PC to the same Ethernet switch with (straight-through) Ethernet cables. That is, for each computer, connect one end of an Ethernet cable to the interface with label *p2p1* and the other end of the cable into an Ethernet switch.
2. When you reboot the Linux PCs, the IP addresses of the computers should be configured as shown in Table 1. The IP addresses listed in the table are associated with the Ethernet card labelled *p2p1*. In this lab, the second Ethernet card of the Linux PCs, labelled *p2p2*, is not used.

Table 1: Default IP addresses

Linux PC	IP Address and Mask of Interface <i>p2p1</i>
Computer A	10.0.1.11/24
Computer B	10.0.1.12/24
Computer C	10.0.1.13/24
Computer D	10.0.1.14/24

### Testing connectivity between computers.

After connecting the four Linux PCs to the Ethernet switch, all four computers should be able to communicate with one another. The following steps verify that the Linux PCs are properly connected. The test consists of running a `ping` test between two Linux PCs.

1. On each PC, first verify that the IP address and mask are set according to Table 1 by issuing the following command on each PC. Among other things, the output shows you the IP address and mask of the p2p1 interface on the computer on which it is issued:

```
ifconfig p2p1
```

If the machines are not configured as expected, see §2.2 to reconfigure the IP addresses and masks.

2. If the machines are configured according to Table 1, then you should be able to ping each host from any other one as they are all on the same /24 network, e.g., from computer A, B, or C, the following command pings computer D:

```
ping -c 5 10.0.1.14
```

## 1.2 Using the Linux Operating System

Here you explore the Linux system by trying out commands that can be typed in a terminal window. Some basic Linux commands are reviewed next. See the `man` pages for a more detailed description.

### Using Linux Commands.

If you are not familiar with Linux or other Unix-like systems, try out some Linux commands by performing the following tasks on computer A:

1. Create a terminal window.
2. Change to the home directory of the root account (`/root`).
3. Create a directory `test` in that directory (unless it already exists).
4. Copy the file `/etc/hosts` to the directory `test`.
5. Change the current directory to directory `test`.
6. Change the name of file `hosts` to `hostfile`.
7. List the content of directory `test`.
8. Edit file `hostfile` with `gedit` (or some other editor). Run `gedit` in the background.
9. Switch `gedit` to run in the foreground.
10. Change the content of the `hostfile` in the editor and save the results. Quit the editor.
11. List the content of `hostfile`.
12. Remove all files in directory `test`.
13. Remove directory `test`.

## 1.3 Saving Your Data

Most lab exercises ask you to save data that is displayed on your monitor to a file. Familiarize yourself with some methods to save data to a file.

**Note:** Whenever you create a file, place the file in the directory `/root`. Since other students may purge the files in this directory, remember to save your files to a USB drive at the end of your lab session.

Here are two methods to save data to a file on a Linux system.

1. **Save data to a file with the redirection operators:** Linux provides an easy way for redirecting the output of a command to a file via the redirection `>` and append operators `>>`.
2. **Save data with a text editor (with copy and paste):** If you have experience with a Unix-like operating system, you may have your favourite text editor (e.g., `vi`, `emacs`, `nano`, etc.). If you have never edited a file on a Unix-like system, we recommend the `gedit` editor. To edit a file with name `fname` using `gedit`, simply type:

```
gedit fname
```

If you use the text editor `gedit`, you can copy text by highlighting the text and pressing Ctrl-C. Then paste the text by pressing Ctrl-V. If you are copying from a terminal window, you need to highlight the text with the mouse and press Ctrl-Shift-C instead.

**On computer A try each of the preceding methods to save data to a file.**

Save the output of the command `ls -l /etc` to a file named `/root/etcfile_x`, where `x` refers to the method used for saving.

## 1.4 Copying Files to a Flash Drive

In all labs you need the data saved in the lab sessions to complete the lab report. Since the equipment in BYENG 217 is not connected to the Internet, the most convenient way to transfer your saved data is with a USB flash drive. This part of the lab acquaints you with the basic commands for accessing a flash drive on a Linux system.

### A Review of Using Flash Drives in Linux

1. **Mounting:** USB flash drives are automatically mounted by the version of Linux installed on the lab machines. The mount point will be `/media/CDROM/LABEL`, where `LABEL` is either the disk label (if you previously assigned one on another computer), or a hexadecimal number of the form `####-####`. You can run the command `ls /media/CDROM/LABEL` after inserting your drive to find out its mount point. You can also issue the command `mount` to list mount points; the last is the most recently mounted, and should begin like:

```
/dev/sdb on /media/CDROM/0000-0000 type vfat
```

The part between `on` and `type` is the mount point. The `type` will be `vfat` unless you have reformatted your flash drive to use a different file system.

2. **Using the file system:** After mounting you can perform any read and write operation on the flash drive. Everything that you read from or write to the mount point will be read from or written to the flash drive. You can copy files to and from this directory, add or delete subdirectories or files, or make this directory the current directory.
3. **Unmounting:** Before you remove the flash drive, you must first “unmount” the file system on it. If you skip this step, you may lose recently-written data and may lose everything on the drive. When you unmount a drive, the current working directory should not be its mount point or any of its subdirectories. If necessary, change the current working directory with the `cd` command. The command for mounting is:

```
umount /media/CDROM/LABEL
```

where `/media/CDROM/LABEL` is the mount point you identified before. Note the spelling of the command. (It is `umount` and not `unmount`.) You can safely remove the drive after you have unmounted the file system. In the event that the system has trouble unmounting the flash drive, try using these optional arguments with the `umount` command:

```
umount -f /media/CDROM/LABEL
```

```
umount -l /media/CDROM/LABEL
```

### Saving data to a flash drive.

1. Use the previous commands to save a file on computer A to a flash drive.
2. On computer A, run the command `df` to obtain a list of all file systems currently mounted on your system. Save the output of the command to a file and save the file to the flash drive.

## 1.5 Locating Configuration Files in Linux

Linux has numerous configuration files that set the environment variables of the operating system. Studying configuration files also provides a way of learning what network configuration options are available to you.

In all labs, you will use Redhat. A list of the most important network configuration files follows:

**Important:** Do not modify configuration files unless asked to do so. Certain changes to the configuration files may require a reinstallation of the operating system.

**Note:** Configuration files are fundamentally different across different versions of Unix-like operating systems (e.g., AIX, Solaris, Linux, FreeBSD). Sometimes the structure of configuration files changes between releases of the same Unix version. Furthermore, the configuration files between different versions of the same Linux distribution can have significant differences.

- **/etc/sysconfig/network**

This file defines global parameters of the network configuration, such as the host name, domain name, and IP address of the default gateway. It also includes a line to determine whether the Linux PC acts as a router or not.

- **/etc/sysconfig/network-scripts/ifcfg-lo**  
**/etc/sysconfig/network-scripts/ifcfg-p2p1**  
**/etc/sysconfig/network-scripts/ifcfg-p2p2**

These files define the configuration of the network interfaces. There is one configuration file for each network interface. The files **ifcfg-p2p1** and **ifcfg-p2p2** are for the two installed Ethernet interface cards. The file **ifcfg-lo** is for the loopback interface.

- **/etc/sysctl.conf**

This file specifies many kernel options related to the network configuration.

- **/etc/hosts**

This file specifies the mapping between host names and IP addresses for network devices. This file also determines the name of the local Linux system.

## 1.6 Using Ping

One of the most basic, but also most effective, tools to debug IP networks is the **ping** command. The **ping** command tests whether another host or router on the network is reachable. The **ping** command sends an **ICMP Echo Request** datagram to an interface and expects an **ICMP Echo Reply** datagram in return.

- On Linux systems, **ping** continues to send packets until you interrupt the command with Ctrl-C.
- When using **ping** on the Linux PCs, we recommend to always send at least two **ICMP Echo Request** packets. We have observed that the first **ICMP Echo Request** may often be dropped at the receiver (Time Exceeded Type 11, Code 0 or 1). This occurs when the **ICMP Echo Request** packet does not reach its destination within a certain amount of time or number of hops, e.g., when waiting for an **ARP Reply** or **ICMP Redirect**. This is explained later in this lab.

### Issuing ping commands.

1. From computer A, send five ping messages (using the **-c** option) to computer B. Save the output.  
**ping -c 5 10.0.1.12**
2. On computer B, issue a ping to the IP address of computer A. Limit the number of pings to five. Save the output.

## 1.7 Basics of tcpdump

**tcpdump** allows you to capture traffic on a network and display the packet headers of the captured traffic. **tcpdump** can be used to identify network problems or to monitor network activities.

### Simple **tcpdump** exercise.

Use **tcpdump** to observe the network traffic that is generated by issuing **ping** commands.

1. Switch to computer A. Start **tcpdump** so that it monitors all packets that contain the IP address of computer B, by typing:

```
tcpdump -i p2p1 host 10.0.1.12
```

2. Open a new window and execute:  
`ping -c 1 10.0.1.12`
3. Observe the output of `tcpdump`. Save the output to a file.

**Another `tcpdump` traffic exercise.**

1. On computer A, start capturing packets using the `tcpdump` command.
2. Issue a `ping` to the nonexistent IP address 111.111.111.111:  
`ping -c 1 111.111.111.111`
3. Issue a `ping` to the broadcast address 10.0.1.255 using the command:  
`ping -c 2 -b 10.0.1.255`
4. Save the output of `ping` and `tcpdump` to a file.

## 1.8 Basics of Wireshark

Wireshark is a network protocol analyzer with a graphical user interface. Using Wireshark, you can interactively capture and examine network traffic, view summaries, and get detailed packet information.

### Running Wireshark.

This exercise walks you through the steps of capturing and saving network traffic with Wireshark. The exercise is conducted on computer A.

1. **Starting Wireshark:** On computer A, start Wireshark by typing:  
`wireshark`  
or by double clicking on the `Wireshark Network Analyzer` icon on the Desktop (it looks like a shark fin). This displays the Wireshark main window on your desktop similar to that in Figure 2.
2. **Selecting the capture options:** Set the options of Wireshark in preparation for capturing traffic. Use the same options in other labs, whenever Wireshark is started.
  - (a) From the main window, click on `p2p1` in the *Capture* column.
  - (b) Click *Capture Options* beneath the interface list.
  - (c) This displays the *Capture Options* window.
  - (d) Select *Use promiscuous mode on all interfaces*.
  - (e) Select *Update list of packets in real time*.
  - (f) Select *Automatically scroll during live capture*.
  - (g) Unselect *Resolve MAC addresses*.
  - (h) Unselect *Resolve network-layer names*.
  - (i) Unselect *Resolve transport-layer names*.
3. **Starting the traffic capture:** Start the packet capture by clicking *Start* in the *Capture Options* window.
4. **Generating traffic:** In a separate window on computer A, execute a `ping` command to computer C.  
`ping -c 2 10.0.1.13`  
Observe the output in the Wireshark main window. Click and highlight a captured packet in the Wireshark window and view the headers of the captured traffic.
5. **Stopping the traffic capture:** Click the stop button (red square) on the toolbar in the Wireshark main window.
6. **Saving captured traffic:** Save the results of the captured traffic as a plain text file. This is done by selecting *Print* in the *File* menu. When a *Print* window pops up, select the options and set a filename.
  - (a) Select the format *Plain text*.
  - (b) Select the *Output to file* checkbox and type the filename in the field next to it.

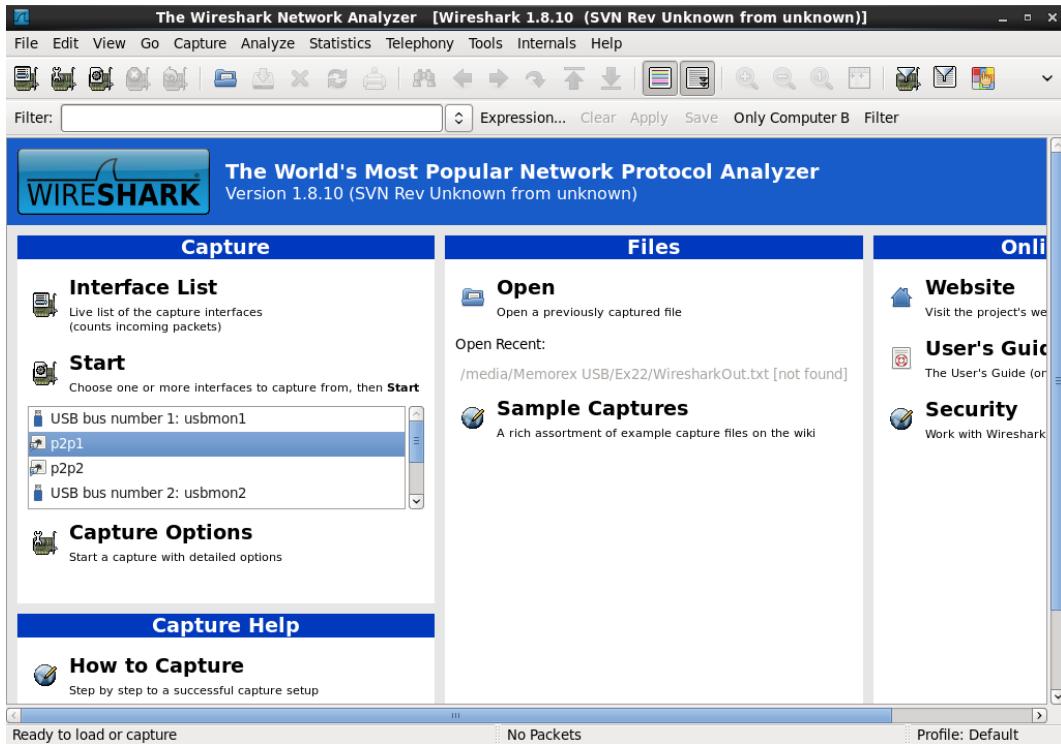


Figure 2: Wireshark main window.

- (c) Unselect *Packet details* if you want to save only some high-level information on each packet. This is usually sufficient. Select *Packet details* and *All expanded* if you want to save all details of all packets at all levels.
- (d) Click the *Print* button to complete the save operation.

Unless asked to do otherwise, always unselect the *Packet details* option when you include saved data in the lab report. If detailed information is required, you will be asked to save details of the captured traffic. In this case, select the *Packet details* option.

If you select *Save* from the *File* menu, the captured data is saved in the format of a `libpcap` file. This format can be interpreted by both `tcpdump` and Wireshark. Measurements saved in `libpcap` format can be analyzed at a later time. `libpcap` files are not plain text files and are not useful for preparing your report.

Unless you have `tcpdump` and/or Wireshark tools available on a system outside of the lab, which allows you to view and save captured traffic at a later time, always save captured traffic in plain text format.

## 2 Single-Segment IP Networks

In this section of the lab, you become acquainted with IP configuration issues on a single Ethernet segment using the equipment in BYENG 217.

- If you have not already done so, before you get started, it is a good idea to reboot the Linux PCs.
- During the lab, you need to save data to files. Save all files in the directory `/root`.
- Save your files to a USB flash drive before the end of the lab. You will need the files when you prepare your lab report.

## 2.1 Setup

- The setup for this section of the lab is identical to that in §1. All Linux PCs are connected to the same Ethernet segment by a single switch.
- The IP addresses for the Linux PCs should be configured as shown in Table 1. Whenever a Linux PC is rebooted, the IP addresses should be set to the values displayed in the table; if not, see §2.2.

## 2.2 Configuring IP Interfaces in Linux

The `ifconfig` command is used to configure parameters of network interfaces on a Linux system, such as enabling and disabling of interfaces, and setting the IP address. The `ifconfig` is usually run when a system boots up. In this case, the parameters of the commands are read from a file. Once the Linux system is running, the `ifconfig` command can be used to modify the network configuration parameters.

This list shows how `ifconfig` is used to query the status of network interfaces.

- `ifconfig`  
Displays the configuration parameters of all active interfaces.
- `ifconfig -a`  
Displays the configuration parameters of all network interfaces, including the inactive interfaces.
- `ifconfig interface`  
Displays the configuration parameters of a single interface. For example, `ifconfig p2p1` displays information on interface p2p1.

There are numerous options for configuring a network interface with `ifconfig`. The examples in this list shows how to enable and disable an interface and how to change the IP configuration.

- `ifconfig p2p1 down`  
Disables the p2p1 interface. No traffic is sent or received on a disabled interface.
- `ifconfig p2p1 up`  
Enables the p2p1 interface.
- `ifconfig p2p1 10.0.1.8 netmask 255.255.255.0 broadcast 10.0.1.255`  
Assigns interface p2p1 the IP address 10.0.1.8/24 and a broadcast address of 10.0.1.255.

**Exercise 1: Changing the IP address of an interface.** Use the `ifconfig` command to modify the IP address of the p2p1 interface of computer D.

1. On computer D, run `ifconfig -a` and save the output.
2. Change the IP address of interface p2p1 of computer D to 10.0.1.16/24.
3. Run `ifconfig -a` again and save the output.

Include the saved output in your report and explain the fields of the `ifconfig` output.

## 2.3 Changing Netmasks

In this part of the lab you test the effects of changing the IP addresses and netmasks of a network configuration. In Table 2, all hosts have been assigned different IP addresses and network prefixes.

Table 2: IP addresses for Exercise 2

Linux PC	IP Address of Ethernet Interface <i>p2p1</i>	Network Mask
Computer A	128.143.71.201/16	255.255.0.0
Computer B	128.143.71.21/24	255.255.255.0
Computer C	128.143.137.144/26	255.255.255.192
Computer D	128.143.137.32/26	255.255.255.192

**Exercise 2: Changing netmasks.**

1. Configure the interfaces of the hosts as shown in Table 2.
2. Run Wireshark on computer A and capture the packets for the following `ping` commands (running each on the “from” computer). Save the Wireshark output to a text file (clearing the *Packet details* option), and save the output of the `ping` commands, including any error messages.
  - (a) From A to C: `ping -c 3 128.143.137.144`
  - (b) From A to B: `ping -c 3 128.143.71.21`
  - (c) From A to D: `ping -c 3 128.143.137.32`
  - (d) From D to A: `ping -c 3 128.143.71.201`
  - (e) From B to D: `ping -c 3 128.143.137.32`
  - (f) From B to C: `ping -c 3 128.143.137.144`

Include selected output from each `ping` in your report and briefly explain the `ping` result.

**Before you leave the lab:**

1. Make a copy of all files you’ve created on each computer onto your USB flash drive and then delete the files!
2. Reset the interfaces to their original values given in Table 1.
3. Disconnect the Ethernet cables from the Ethernet switch and from the PC interfaces.

## 3 An Introduction to GENI

### 3.1 GENI Project

Your ASURite has been imported into the GENI project named `ASU-CSE434-Fall2021` for this course. To use GENI login to the [portal](#) using your ASU credentials. Whitelist `geni.net` if you run an ad blocker.

### 3.2 A First Experiment on GENI using the Jacks Tool

Follow the instructions for your [first GENI experiment](#). It describes how to set up your `ssh` keys; these are required to be able to add any resources into a GENI slice. See also the video accompanying this lab.

You will create a slice and use a tool called Jacks to add resources into the slice, *i.e.*, to create your virtual network topology. You will then instantiate the slice on an InstaGENI site. When the resources are available, *i.e.*, the VMs have booted, you will be able to login to the VMs and perform a simple experiment.

The goal of this experiment is to validate that your `ssh` keys are set up properly. If they are not, you will not be able to `ssh` successfully into any virtual machine. If something has gone wrong, reinstall the `ssh` keys and then [Update ssh Keys](#) on the slice.

**Exercise 3: Testing ssh keys.** Be sure that all screen shots include the ASURite id of a group member. This should be in the login prompt when you `ssh` into a VM.

1. Take a screen shot of your topology once it is ready (it should resemble Figure 3-9) and include it in your report.
2. Give the interface names and IPv4 addresses of the client and server assigned to the data plane and the control plane and include them in your report. (The data plane interfaces have IPv4 addresses of the form 10.1.x.y.)
3. Take a screen shot of `ping` output on the `client` in 5.1(c) and 5.1(d) of the instructions.
4. Delete the resources from this slice when you are finished. This results in an empty slice that can be reused, *i.e.*, other resources can be added to into it.

You are welcome to work through the rest of the experiment if you wish.

## 4 Designing Subnets

**Do not proceed with this part of the lab if you have not completed the first experiment in §3.**

Follow the instructions in [Designing Subnets](#). In this experiment, your task is to set up subnets in a few small *local area networks* (LANs) to meet given design requirements.

Read the section **Background** and work through the detailed example provided. If you understand this example, you will be able to solve the subnet design problem that is the goal of the lab.

**Exercise 4: Design Subnets.** Solve the subnet design problem posed in the section **Challenge: Design Subnets** that meets the requirements.

*For each of the three subnets* in the design problem, provide a table in your report giving:

1. The subnet mask.
2. The network address.
3. The smallest IPv4 address that may be assigned to a host in the subnet.
4. The broadcast address for the subnet.
5. The highest IPv4 address that may be assigned to a host in the subnet.

Be sure to follow the conventions on addressing given, *i.e.*, assign the lowest IP address in the subnet to the LAN-facing interface of the router, assign the highest IP address in the subnet to one node, and any other IP address in the subnet to the other node.

In the section **Implement your Design on each LAN**, in each LAN, use `ifconfig` command to configure the IP address and subnet masks of each host in the LAN. After configuration, each host should be able to reach every other host in the same LAN.

**Exercise 5: Implement your Design.** Configure the IPv4 address and subnet mask of each host in each LAN using your solution to Exercise 4. Recall the instructions on configuring IP addresses from §2.2.

1. Take a screen shot showing the output of the `ping -c 5 IP` command between the two hosts in the same LAN and include each one in your report. (You should have 3 screen shots, one for each LAN.)
2. Take a screen shot showing the output of the `ping -c 5 IP` between a host in LAN A and a host in LAN B, between a host in LAN B and a host in LAN C, and between a host in LAN C and a host in LAN A, and include them in your report. (Again, you should have 3 screen shots.)

If your `ping` output is not as expected, you may need to return to the design problem (Exercise 4) and/or check your node configurations.

In the section **Adding Routing Rules**, on each host, follow the instructions for adding a rule that describes how to reach the other two LANs. Use the `route add` command for this purpose. In addition to adding such rules to every host, you also need to add a rule on each router.

If you make a mistake you can use the `route del` command and try again. When you have set up these rules correctly, every host in every LAN should now also be able to `ping` every host in *every other LAN* in your topology.

**Exercise 6: Add Routing Rules.** Configure the route on each host in each LAN, and also on each router.

1. Take a screen shot of the output of the `route -n` command on each node and on each router and include it in your report. (You should have 9 small screen shots.)
2. Take a screen shot of the output of the `ping -c 5 IP` command from `node-11` to `node-4`, from `node-5` to `node-6`, and from `node-10` to `node-7` and include them in your report. (You should have 3 screen shots.)
3. Follow the instructions to configure the routing rules for the `10.10.100.0/24` network. Now, take a screen shot of the output of the `traceroute IP` command from `node-11` to `node-4` and include it in your report. (The `traceroute` should be able to trace the route taken.)

Once you have completed all exercises in the **Designing Subnets** portion of this lab, and you no longer wish to experiment in your slice, delete the resources in the slice to free them up for other experimenters.