We suggest that you perform all of the described exercises on your Pi using both IPv4 and IPv6 addresses.

1) Configuring network interfaces
In this section, you should use both the `ifconfig` and `ip` command to assign static IP address to an interface on your Pi. `ip` is a new interface configuration command that replace ifconfig. Learning the format of the `ifconfig` command is necessary since the `ip` command is not available on all systems. Using these two command, perform the following exercises:

- **Explicit Assignment:** Assign a static IP address to an interface of your Pi. Note this would run into conflicts with the DHCP client that is running on their machine since it would periodically requests an IP address from the DHCP server and assigns it to each interface of your computer. Therefore, you should disable the DHCP client on your computer. Boot up
- **Assignment:** Another way to assign static IP address to an interface is during the boot time of your Pi through the `/etc/network/interfaces` file.
- **Another piece of information that a node needs to be able to operate, is the address of a (local) DNS server. This information is usually specified in the `/etc/resolve.conf` file.
- **You can explore the above issues on your own to have more fun or refer to the description at the following URL [https://help.ubuntu.com/lts/serverguide/network-configuration.html](https://help.ubuntu.com/lts/serverguide/network-configuration.html)**
- **One way to to check whether the assigned IP address is correct, is to ping it from another node on the same subnet.**

2) Managing routing table of local computer
A computer can forward packets between its interfaces similar to a router. To do this, it requires a forwarding/routing table that shows how individual packets should be forwarded. Similar to the previous case, routing table entries may get modified by a DHCP client that is running on your node. Make sure to disable the DHCP client. The configuration for the routing entries of the computer could be done through two set of commands, namely `route` and `ip`. Using these commands practice to define static routing table entry and default route entry for the wireless interface, and manage routing entries for multiple interfaces (wired and ethernet). To test the static or default route, you can serf them to the wireless interface and check that you can browse the Internet but you should not be able to ping other Pi's interfaces on the Ethernet subnet. To test the multi-interface setup, the connections associated with each interface should be properly routed through them to her corresponding subnet (wireless or Ethernet).

3) Firewall configuration NAT/iptables
Firewalls are used to control/manage the connections that are established to a network (e.g. home network). For firewall configuration you should become familiar with the `iptables` command. Note that there is a simplified rapper around the `iptables` command named `ufw` that you need to install but it doesn't give you the full expressiveness/flexibility of `iptables`. `iptables` has five tables, namely `raw`, `filter`, `nat`, `mangle` and `security`, and each table has a set of chains. Possible chains are `INPUT`, `OUTPUT`, `PREROUTING`, `POSTROUTING` and `FORWARD` where each chain includes a set of rules. These chains are traversed in a predefined manner, and when a packet matches a rule in one chain, the packet my not be compared (depending on the action) against the rules in other tables (i.e. the final decision for handling the packet is made). Each chain as a default action. The flow chart of a packet processing that shows all the tables and chains in a firewall is presented in the following figure: [http://riworkman.net/howtos/iptables/images/tables_traverse.jpg](http://riworkman.net/howtos/iptables/images/tables_traverse.jpg)

Each rule could have a target that defines what action should be performed if a packet matches the defined rule. The complete list of actions (or targets) is fairly large and include `DROP`, `ACCEPT`, `RETURN`, `LOG`. The first two actions are clear. `RETURN` causes a packet to be passed to the previous calling chain. `LOG` causes a packet to be logged to the log file of iptables.

Check online resources for other actions.

Using `iptables` command perform the following exercises: 1) proper firewall configuration by defining default blacklist rules for input packet, 2) whitelisting benign traffic: depending on the services that you are running on you node. 3) logging firewall events: define a set of rules for packets that should be logged and generate the packets that satisfy those rules and are logged.

To experiment with these capabilities, you can block traffic based on source IP addresses (incoming flows from another Pi) or white lists incoming flows from certain sources that are destined to those services (based on their destination IP and port) on the local node.

4) Project

**Part I:** Configure your Pi into an Internet gateway (for all students)
For your project, you should configure their Pi to act as an Internet gateway. Given a Pi, you should configure its ethernet interface as a shared interface for other nodes on the ethernet subnet to connect the Internet. This Pi that acts as a gateway, should route all the incoming flows that are destined to the Internet through the wireless interface to their destination. Your gateway node should also implement a NATing and replace the source address of the packet as it traverses the gateway. For the Internet sharing scenario, you have to enable IP forwarding for the Linux kernel which could be accomplished through various means such as the `sysctl` command or the `/proc/sys/net/ipv4/ip_forward` file.

To run this experiment, you need to pair your Pis or have two nodes on the same ethernet subnet where one node acts as the client and the other one act as the gateway. You can assign static IP address to the ethernet interface (so there is not need to run a DHCP server).

**Part II:** Logging & Processing Firewall Events (for graduate students, optional for undergrads)
Given the setting for part I, you should set the firewall rules (on your gateway) to log certain events associated with incoming flows
and then write a simple parser to report on basic statistics of the observed events. For example, you could log everything that is not whitelisted and generate a report on that log. Your report shows the total number of requests their break down across different types of protocols (http, https, ...), the frequency per request from each external IP address, etc. Note that you need to produce this incoming flows from another external node towards your gateway (or it might be generated by other external entities that are always scanning the network :-)