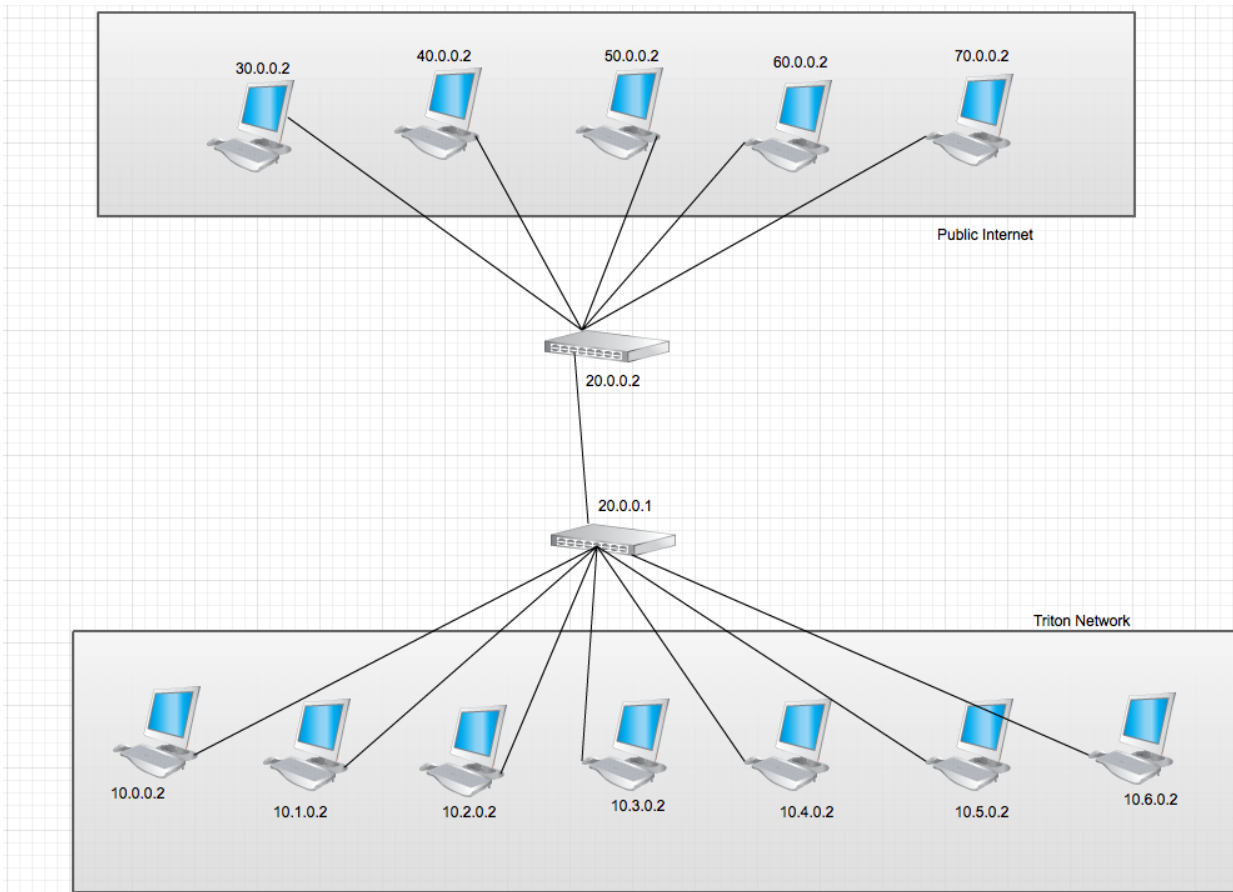


Topologies for the “Simple Router” project

Sample topology :



You can use the [source files](#) of the above topology as a bootstrap to create your own topologies for testing your router.

To test your router using the topology shown above, follow the steps below :

1. Copy the cse123 folder to /home/mininet/cse123-p2/pox_modules
2. In /home/mininet/cse123-p2/pox_modules/setup.py, change the first line under “setup” to name='cse123'.
3. Execute: mininet@mininet-vm:~/cse123-p2/pox_module\$ sudo python setup.py install
4. Copy topoa.py and IP_CONFIG_TOPOA files to /home/mininet/cse123-p2
5. Copy rtable_topoa1 and rtable_topoa2 to /home/mininet/cse123-p2/router.
6. Change run_pox.sh contents to ./pox/pox.py cse123.ofhandler cse123.srhandler
7. Change run_mininet.sh contents to sudo python topoa.py
8. Start POX and Mininet
9. Compile your router code and start two instances of sr each connecting to one of the routers in the topology shown above.
 - a. ./sr -v sw1 -r rtable_topoa1
 - b. ./sr -v sw2 -r rtable_topoa2

UPDATE : Additionally for wget to work, you need to create extra directories as shown below. Since there are five servers in this case we need three more http_server# directories.

```
mininet@mininet-vm:~/cse123-p2$ cp -r http_server1/ http_server3
mininet@mininet-vm:~/cse123-p2$ cp -r http_server1/ http_server4
mininet@mininet-vm:~/cse123-p2$ cp -r http_server1/ http_server5
```

NOTE : There is no need to create new topologies and you can do what you need for the project using the initial topology (client, server1, server2, sw0).

However, to test your routing table lookup and ICMP behaviour, you can change the entries in rtable file to have different subnet masks. For instance, to check your router's longest prefix match logic, you can have multiple entries in your rtable file that match a single destination IP. The following routing table has two matching entries for the packet destined to 192.168.2.2.

10.0.1.100	10.0.1.100	255.255.255.255	eth3
192.168.2.0	192.168.2.2	255.255.255.0	eth2
192.168.2.2	192.168.2.2	255.255.255.255	eth1
172.64.3.10	172.64.3.10	255.255.255.255	eth2

In the above routing table, taking the wrong routing table entry would lead to picking up the wrong outgoing interface in the router.

With some changes to the routing table, you can trigger a Destination Host Unreachable ICMP by making your router send ARP request to a host which is not present. To simulate that scenario, set the subnet mask of a routing table entry in such a way that it matches your destination IP, but the next-hop gateway should be a non-existing IP address.

To test your router using new topologies, follow the steps given below.

Steps to create new topologies :

1. Changes in Mininet topology code :

- Create a new IP_CONFIG file under cse123-p2 folder corresponding to the new topology you want to create. Ref: IP_CONFIG_TOPOA file.
- Create a new topology script in Python under cse123-p2 folder. You can start with topoa.py file. Change the lines marked as #TOPOCHANGE.

2. Changes in POX Controller components:

- Refer the ofhandler.py script provided. Change the lines marked as #TOPOCHANGE. (You need to provide the new routing table entries to the controller)

Each time a change is made to this component file, you need to recompile it using the following command.

```
mininet@mininet-vm:~/cse123-p2/pox_module$ sudo python setup.py install
```

3. Changes in "router" folder :

As you can see from the sample topology, we are going to have multiple routers. So we should have equal number of routing tables under the “router” folder.

In the sample topology, we have two routers “sw0” and “sw1”. The corresponding routing tables can be found in rtable_topoa1 and rtable_topoa2.

Execution :

You have to run separate “sr” instances for each router in your topology. While executing, you need to pass two parameters :

-v <router> where <router> is the router instance you want the sr instance to connect to (Eg: “sw0”).

-r <routing_table> where <routing_table> is the location of the rtable file you want to pass to <router>

```
./sr -v sw1 -r rtable_topoa1
```

```
./sr -v sw2 -r rtable_topoa2
```

If everything is working as expected, you should be able ping and traceroute among all the hosts and download the web content on the servers from all the clients.