Final Project: Implementing a Simple Router

Due: Friday, March 17th, 11:55 PM (Must be Demo'd to TA & Submitted)

In the previous lab you implemented a simple firewall that allowed ARP and TCP packets, but blocked all other packets. For your final project, you will be expanding on this to implement routing between devices on different subnets, and implementing firewalls for certain subnets. The idea is to simulate an actual production network. You will be using ideas from Lab 1 to help construct the mininet topology, and ideas from Lab 3 to implement the rules allowing for traffic to flow through your network. Please refer back to those Labs for guidance on how to complete this assignment.

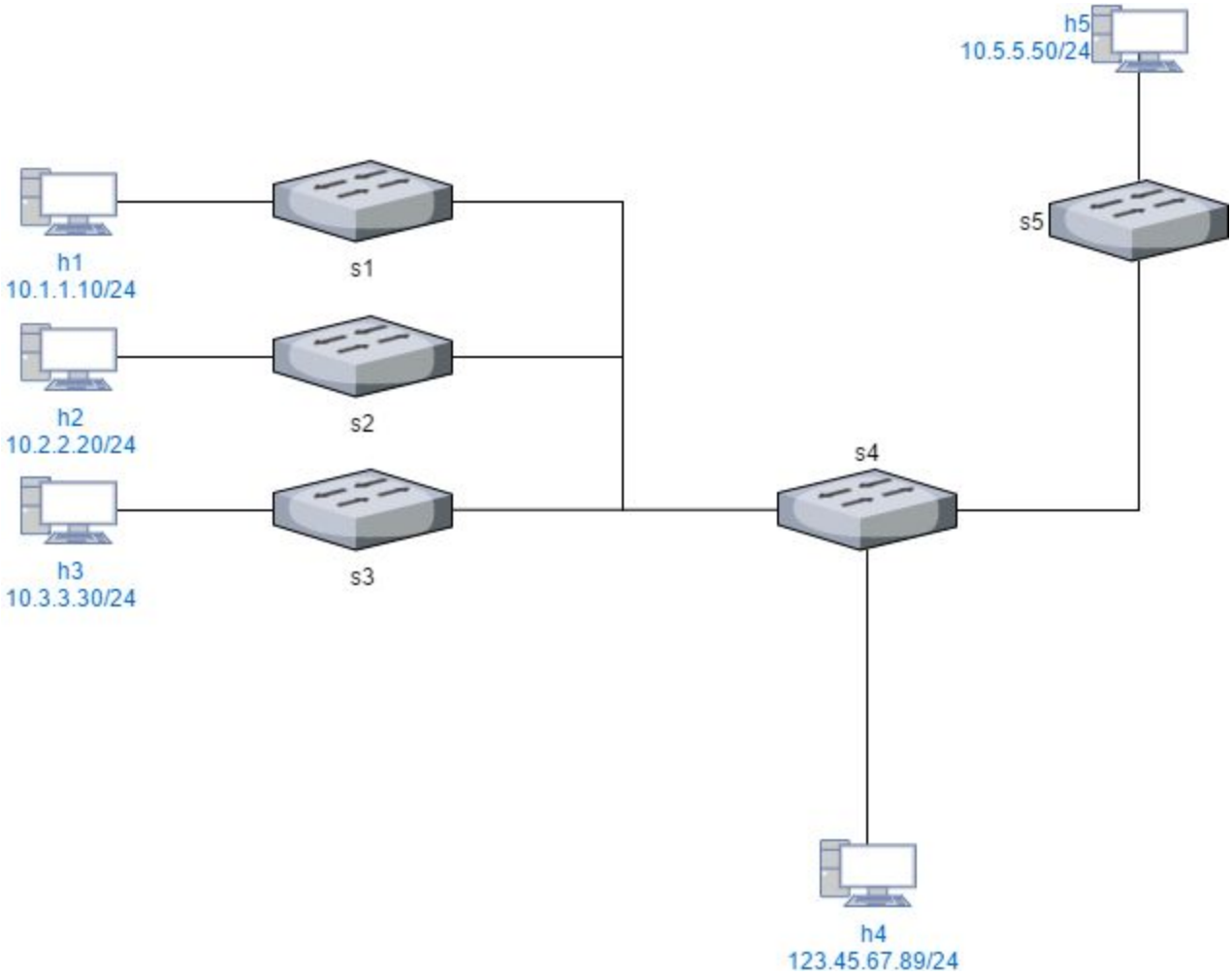
Assignment:

For this lab, we will be constructing a network for a small company. The company has a 3-floor building, with each floor having its own switch and subnet. Additionally, we have a switch and subnet for all the servers in the data center, and a core switch connecting everything together.

Your device's roles and IP addresses are as follows:

Device	Mininet Name	IP Address	Description
Floor 1 Host	h1	10.1.1.10/24	A computer on floor 1 of the company.
Floor 2 Host	h2	10.2.2.20/24	A computer on floor 2 of the company.
Floor 3 Host	h3	10.3.3.30/24	A computer on floor 3 of the company.
Untrusted Host	h4	123.45.67.89/24	A computer outside our network. We treat this computer as a potential hacker.
Server	h5	10.5.5.50/24	A server used by our internal hosts.

The topology will look as follows:



Your goal will be to allow traffic to be transmitted between all the hosts. In this assignment, you will be allowed (and encouraged) to flood all non-IP traffic in the same method that you did in Lab 3 (using a destination port of of.OFPP_FLOOD). However, you will need to specify specific ports for all IP traffic. You may do this however you choose-- however, you may find it easiest to determine the correct destination port by using the destination IP address and source IP address, as well as the source port on the switch that the packet originated from. Additional information has been given to you in the do_final() function to allow you to make these decisions. Please see the comments in the provided code for guidance.

Additionally, to protect our servers from the untrusted Internet, we will be blocking all IP traffic from the Untrusted Host to the Server. To block the Internet from discovering our internal IP addresses, we will also block all ICMP traffic from the Untrusted Host to anywhere internally.

You will be required to demonstrate your project to the TAs. You will need to explain how you implemented the various requirements and show that they work properly. If you need help figuring out how to do this, look back to previous assignments and see how you tested them.

Provided Code:

Available in a ZIP file here.

We have provided you with starter code (skeleton files) to get you started on this assignment. The controller file (final_controller_skel.py) needs to be placed in ~/pox/pox/misc, and the mininet file (final.py) should be placed in your home directory (~). This time, you will need to modify both files to meet the lab requirements.

You will be using slightly different commands to create the Hosts and Links in the Mininet file to give you more information to make decisions within the Controller file. Additionally, you will notice that you have additional information provided in the do_final function. This is documented in the comments within the files.

Summary of Goals:

- Create a Mininet Topology (See Lab 1 for help) to represent the above topology.
- Create a Pox controller (See Lab 3 for help) with the following features:
    - All hosts able to communicate, EXCEPT:
        - Untrusted Host cannot send ICMP traffic to Host 1, Host 2, Host 3, or the Server.
        - Untrusted Host cannot send any IP traffic to the Server.

Testing:

You may test with ping commands, xterm windows, and observing packets with Wireshark inside your VM. Please have a plan in place to show you have implemented all the requirements during your demonstration to the TAs.

Grading Rubric:

Total: 100 points

30 points: Mininet Topology
    10: Devices successfully created.
    10: Links successfully created.
    10: IP addresses correct.
50 points: Pox Controller
    25: All hosts can communicate.
        15 point deduction if rules not installed in flow table.
        20 point deduction if IP traffic is implemented using OFPP_FLOOD.
    15: Untrusted Host cannot send ICMP traffic to Host 1, Host 2, Host 3, Server
        10 point deduction if Untrusted Host cannot send ANY traffic to these hosts.
    10: Untrusted Host cannot send any IP traffic to Server
20 points: Demonstration

Partial credit may be awarded for incomplete assignments based upon demonstration and explanations as to why something may not be functioning properly.