Programming Assignment 4

CST 311, Introduction to Computer Networks, Spring 2020

READ INSTRUCTIONS CAREFULLY BEFORE YOU START THE ASSIGNMENT.

This programming assignment is due on Saturday, February 29, 2020.

Assignment must be submitted electronically to iLearn on https://ilearn.csumb.edu by 11:59 p.m. on the due date. Late assignments will not be accepted.

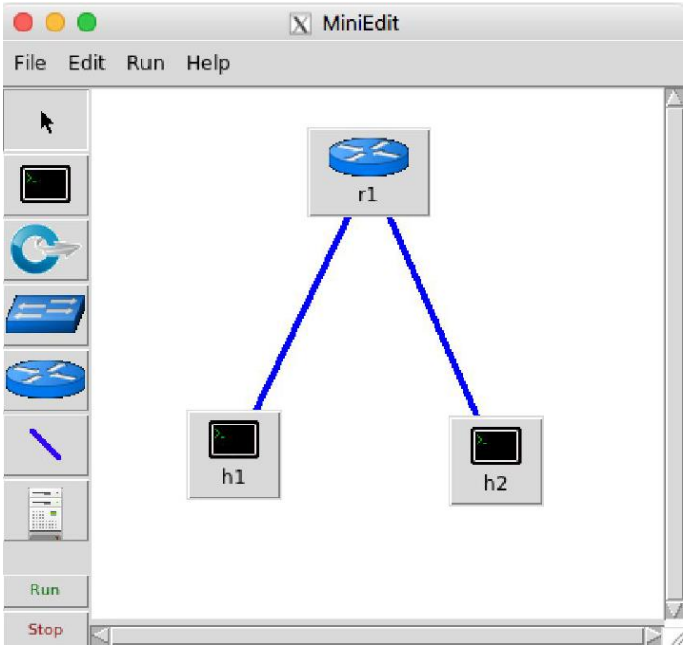
The assignment requires you to submit modified legacy_router.py program per **"Grading objectives"** and to submit a document as described in "**What to turn in"** below.

This assignment is to be done with your Team per the Programming Process document with the steps below to modify the legacy_router.py used by miniedit to configure mininet.

Put your names in the document with the answers that you turn in. You must show the network design used to make h1 to be able to ping h2 AND h2 to be able to ping h1. You must also clearly document what changes you made to modify legacy_router.py and why those changes were made. This assignment is worth 225 points.

Subnet addressing in Mininet

In this assignment you will start with Python code that builds a 2 host network connected by a legacy router. You will modify it such that the two hosts can send packets to each other. It requires that you understand subnet addressing and the function of a router.
The network is built using Miniedit on a Mininet virtual machine:
python mininet/examples/miniedit.py



In order to run miniedit.py, you must install X server (XQuartz for MAC and XMing for Windows), if it is not already installed and setup X11 forwarding on your machine.

https://uisapp2.iu.edu/confluence-prd/pages/viewpage.action?pageId=280461906

The code generated by exporting it as a Level 2 Script: (unused imports and code added by Miniedit have been removed):

```
#!/usr/bin/python
# File: legacy_router.py

from mininet.net import Mininet
from mininet.node import Host, Node
from mininet.cli import CLI
from mininet.log import setLogLevel, info

def myNetwork():

    net = Mininet( topo=None,
                   build=False,
                   ipBase='10.0.0.0/8')

    info( '*** Adding controller\n' )
    info( '*** Add switches\n')
    r1 = net.addHost('r1', cls=Node, ip='0.0.0.0')
    r1.cmd('sysctl -w net.ipv4.ip_forward=1')
```

```
    info( '*** Add hosts\n')
    h2 = net.addHost('h2', cls=Host, ip='10.0.0.2', defaultRoute=None)
    h1 = net.addHost('h1', cls=Host, ip='10.0.0.1', defaultRoute=None)

    info( '*** Add links\n')
    net.addLink(h1, r1)
    net.addLink(h2, r1)

    info( '*** Starting network\n')
    net.build()

    CLI(net)
    net.stop()

if __name__ == '__main__':
    setLogLevel( 'info' )
    myNetwork()
```

Executing this code and trying a ping results in "Destination Host Unreachable":

```
mininet@mininet-vm:~$ sudo python legacy_router.py
*** Adding controller
*** Add switches
*** Add hosts
*** Add links
*** Starting network
*** Configuring hosts
r1 h2 h1
*** Starting CLI:
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
From 10.0.0.1 icmp_seq=1 Destination Host Unreachable
From 10.0.0.1 icmp_seq=2 Destination Host Unreachable
From 10.0.0.1 icmp_seq=3 Destination Host Unreachable
```

Your task is to modify this program (perhaps using different IP addresses) such that the legacy router is able to forward packets between the two hosts. You will need to understand Internet addressing, subnets, and the function of a router as described in the "IPv4 Addressing" Section of Kurose and Ross (4.3.3 in 7th Edition, 4.4.2 in 6th Edition). You may also find the example Python programs in mininet/examples helpful; in particular linuxrouter.py; I suggest executing that program and studying it to understand how you will need to modify legacy_router.py.

Grading Rubric and Objectives

(15 %) 1. Correct Network Design which allows h1 to ping h2 and for h2 to be able to ping h1

(10%) 2. Show what lines were changed and why

(10%) 3. Screen capture of program that runs with no Python errors

(20 %) 4. Screen capture of successful ‘ h1 ping h2 ’ command at the minnet> prompt.

(20%) 5. Screen capture of successful ‘ h2 ping h1 ’ command at the minnet> prompt.

(25%) 6. Answers to these questions:

a. (5 %) What were any interesting findings and lessons learned ?
b. (5%) Why didn’t the original program forward packets between the hosts?
c. (5%) Is the line ‘ r1.cmd('sysctl -w net.ipv4.ip_forward=1') ’ required?
d. (10%) Intentionally break your working program, e.g.: change a subnet length, IP address, or default route for a host. Explain why your change caused the network to break.

The Process

1. View Professor McCann’s video from his class (which is Assignment #3 in his class. It is Programming Assignment #4 in our class). You can ignore the part about using github – we did not require that in our assignment. But the rest of Dr. McCann’s video can be very helpful for this assignment
2. Each student reads the assignment.
3. Team meets to select the Team Lead who works with the team to assign roles, discusses the assignment together for a common understanding of the problem and what is needed and then sets up a schedule. Roles for this assignment should include Team Lead; OS, mininet and network setup person; person who will do the software program changes and documentation on changes made; testing and debug; team lead backup.
4. For this assignment, everyone needs to come up with a network addressing design that will allow h1 to ping h2 AND h2 to ping h1 – this is to be turned in.
5. For this assignment, in lieu of writing pseudo code, the group will work together on how the program legacy_router.py works, what and where to make the changes needed and why. Note – as per recommendation above and in the video, you may need to study the example Python programs in mininet/examples and find them helpful; in particular linuxrouter.py
6. Team leader works with team on the plan for testing. No test data is needed on this assignment.

7. Make the changes needed and document each line that is changed
8. Team works on test and debugging until h1 can ping h2 successfully AND h2 can ping h1 successfully.
9. TA may check in with Team Lead on assignment progress.
10. Team lead collects items to turn in and submits a document with all the parts needed. Important – only 1 document needs to be submitted - by the Team lead for each team.
11. Complete assignment before deadline.

What to turn in:

Team leader will work with the team to prepare the following in 1 document to turn in:

1. Network Design from each Team member. Missing network design results in % deduction per Rubric #1
2. A list of lines that were changed and why
3. Screen capture of program running with no Python errors (needed only if Step 4 and Step 5 below do not work
4. Screen capture of successful ‘ h1 ping h2 ’ command at the minnet> prompt.
5. Screen capture of successful ‘ h2 ping h1 ’ command at the minnet> prompt.
6. Answers to these questions:
   a) What were any interesting things found and lessons learned?
   b) Why didn’t the original program forward packets between the hosts?
   b) Is the line ‘ r1.cmd('sysctl -w net.ipv4.ip_forward=1') ’ required?
   c) Intentionally break your working program, e.g.: change a subnet length, IP address, or default route for a host. Explain why your change caused the network to break.