Programming Assignment 4

CST 311, Introduction to Computer Networks, Spring 2021

**READ INSTRUCTIONS CAREFULLY BEFORE YOU START THE ASSIGNMENT.**

This programming assignment is due on Saturday, Apr 25, 2021.

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 a 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. The naming convention of the file should be PA4\_legacyrouter\_Team<your team #>.py

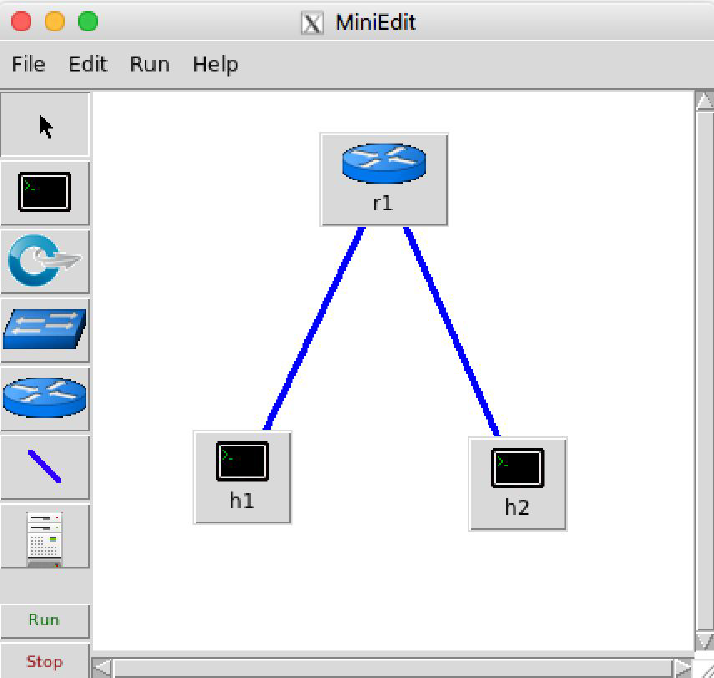
**Put your names in the document with the answers that you turn in.** You must show the network design used to make h1 to be able to ping h2 AND h2 to be able to ping h1. You must also clearly document what changes you made to modify legacy\_router.py and why those changes were made. This assignment is worth 100 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 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

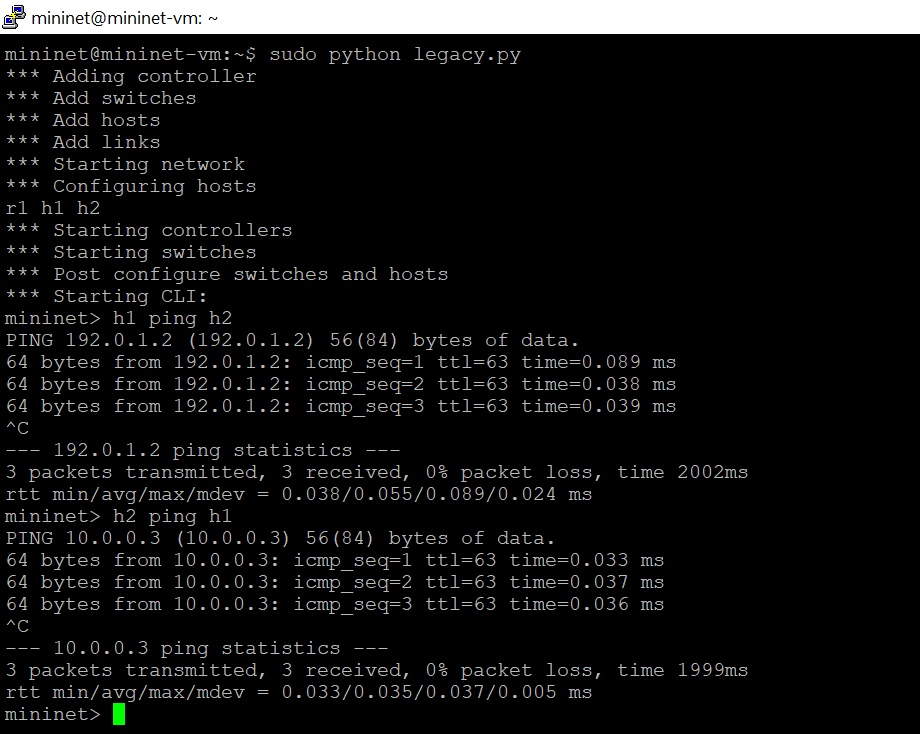
From 10.0.0.1 icmp\_seq=3 Destination Host Unreachable

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. 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.

**Expected output**



## What to Hand in

1. You will hand in the working legacy\_router.py code to iLearn.
2. Minutes of the 3 meetings.
3. Make a pdf file with screenshots of working legacy\_router.py. Please do not upload the screenshots as image files.
4. In the same pdf file with screenshots, draw the network sketches as asked in grading objective 1 and 2 and also answer the questions from the grading objective 7.

**Grading Objectives**

1. **(5 points)** Network design of the script given in this document and show in the diagram - what is missing/incorrect?
2. **(5 points)** Correct Network Design which allows h1 to ping h2 and for h2 to be able to ping h1. Draw and submit the network design in pdf format with IPs of all interfaces labelled. Also label the hosts as h1 and h2 and the switch as S/R.
3. **(5 points)** Screen capture of the program that runs with no Python errors.
4. **(5 points)** Screen capture of successful ‘ h1 ping h2 ’ at the minnet> prompt.
5. **(5 points)** Screen capture of successful ‘ h2 ping h1 ’ at the minnet> prompt.
6. **(5 points)** Your modified legacy\_router.py program as a separate python file.
7. **(35 points)** Answers to these questions:
8. What were any interesting findings and lessons learned ?
9. Why didn’t the original program forward packets between the hosts?
10. Is the line ‘ r1.cmd('sysctl -w net.ipv4.ip\_forward=1') ’ required?
11. 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.
12. **(5 points)** Submission files are in order - 1 pdf document and 1 python script. Minutes of the 3 meetings. Program must be well documented.
13. Teamwork grade: **(30 points)** Each team member will grade each other teammate out of 10 points during peer evaluation. I will average all team members’ grades and scale it to get your teamwork grade out of **30 points**. Note that 30% of your grade will come from your teamwork and team member evaluations.