ECE 4380/6380: Computer Communications **Spring 2018**
Machine Problem 2 **Due: 11:59 pm Monday, April 2**

Extended LANs and GENI

The goal of this machine problem is to experiment with extended LANs and the spanning tree algorithm and to familiarize you with GENI. The first step is to establish an account on GENI by joining the project for this class. Here is a quick start guide, but see the tutorial below for full details.

- Go to https://portal.geni.net/ and click on the "Use GENI" link
- Select the Clemson University icon if visible, or use the "show a list of organizations" link
- Go to https://portal.geni.net/secure/join-project.php and search for ece-4380-6380
- Click on the "join" button for this project
- After reading the rules for using GENI, request access. Approval may take some time as a system administrator must review and approved your request. So, plan ahead and do this step a few days before you are ready to start the project.
- Once you receive a confirmation email you can begin the project.

General Notes

If you have system-specific problems related to GENI first send email to our local GENI administrators at clemson-openflow-group@googlegroups.com. They can answer most questions related to using GENI for our project.

The tutorial required to get started is

https://clemsonopenflowgroup.atlassian.net/wiki/spaces/ece44006400sp18/pages/93388801/Introduction+to+GENI

Procedure

Login into GENI

Follow the directions on the tutorial page to navigate to the portal. On the Projects tab click on "Join a Project" button. Search for our "ece-4380-6380" project. This step requires an administrator to approve your request.

Create your SSH public/private key pair. Make sure you download your private key. This step can be completed before you are approved to join the project. See the Tips at the end of this document for additional notes about using the keys.

Create a new slice

Follow the directions on the tutorial web page to create a new slice. Create an extended LAN by placing at least 6 "OF OVS" switches. Connect the switches with links so that there are at least 3 loops. We will

configure these switches as bridges and each bridge will run the spanning tree algorithm. Connect one host to each bridge using the "Xen VM". The "OF OVS" switches will use an image that has all the necessary software already installed. However, for the "Xen VM" hosts, select Ubuntu 14.04 as the disk image. Rename the nodes using a pattern such as bridge-1 and host-1. This sets the VM host names so when you log into the VM you will know the type of the node.

Save your rspec design before you reserve any resources.

For example, below is the extended LAN we used to study the spanning tree algorithm in the Set 6 lecture notes. You MUST create an extended LAN that is different than this example. Homework Set 5 provides a few additional examples, but you should create your own variation.

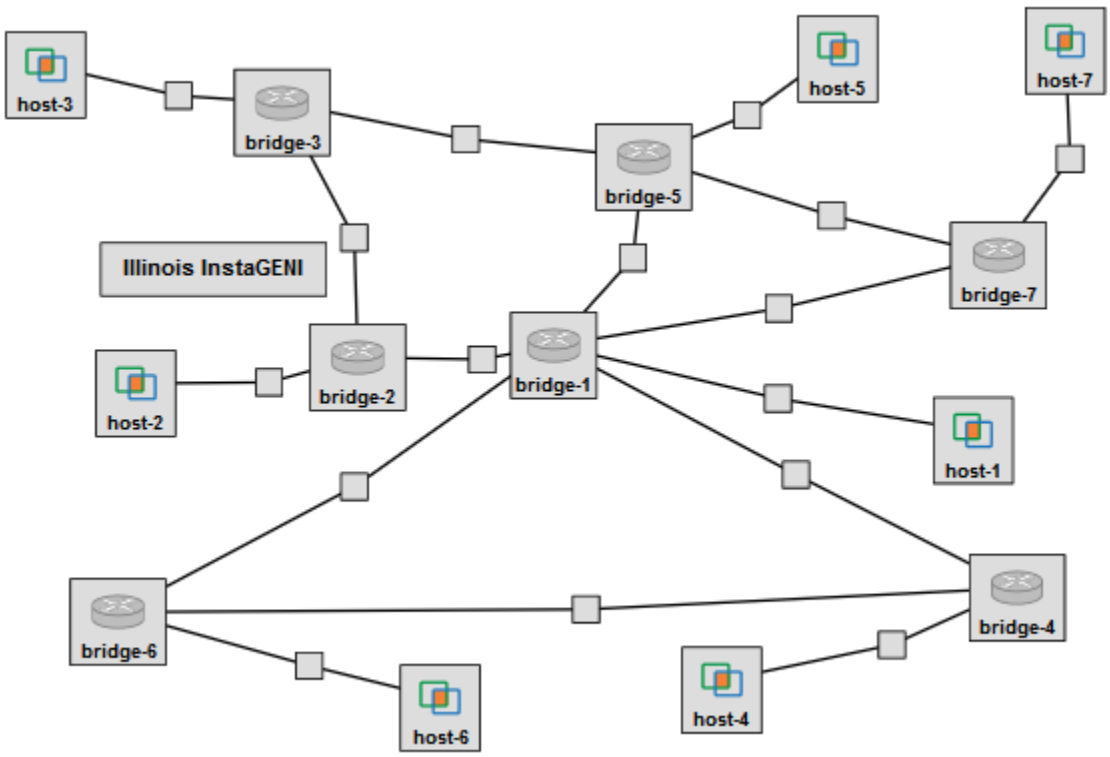

Figure 1. Example ELAN

Reserve Resources

Make sure the site you have selected is one of the "InstaGENI" options. As described in the tutorial, check this web page to see which InstaGENI sites are not too heavily loaded:
http://groups.geni.net/geni/wiki/ExpGraphs

This web page shows you the status (since a site with a low load might also be down):
https://portal.geni.net/amstatus.php

For my example at the Illinois InstaGENI site, the first VM came up after 8 minutes. It took about 30 minutes for all of them to boot. I should have picked a site that was less busy. Tip: once all the VS's are running, they will remain up until the slice expires or you select an icon and click the delete button.

Illustrate Figure

Because there are a large number of IP addresses and Ethernet interfaces to track, make an illustration with the details for your network. SSH into each host and bridge (review the tutorial for details on how to use your private key with **`ssh`**). On each VM use **`ifconfig`** to determine the IP address that is assigned to each Ethernet interface. You can use the rspec file to find the machine name and port number for each VM. Recall that the interface **`eth0`** should not be used in any of our experiments. In my figure below I have indicated the machine name and port number (for use with **`ssh`**) along with the IP address for each Ethernet interface for all the hosts and bridges.

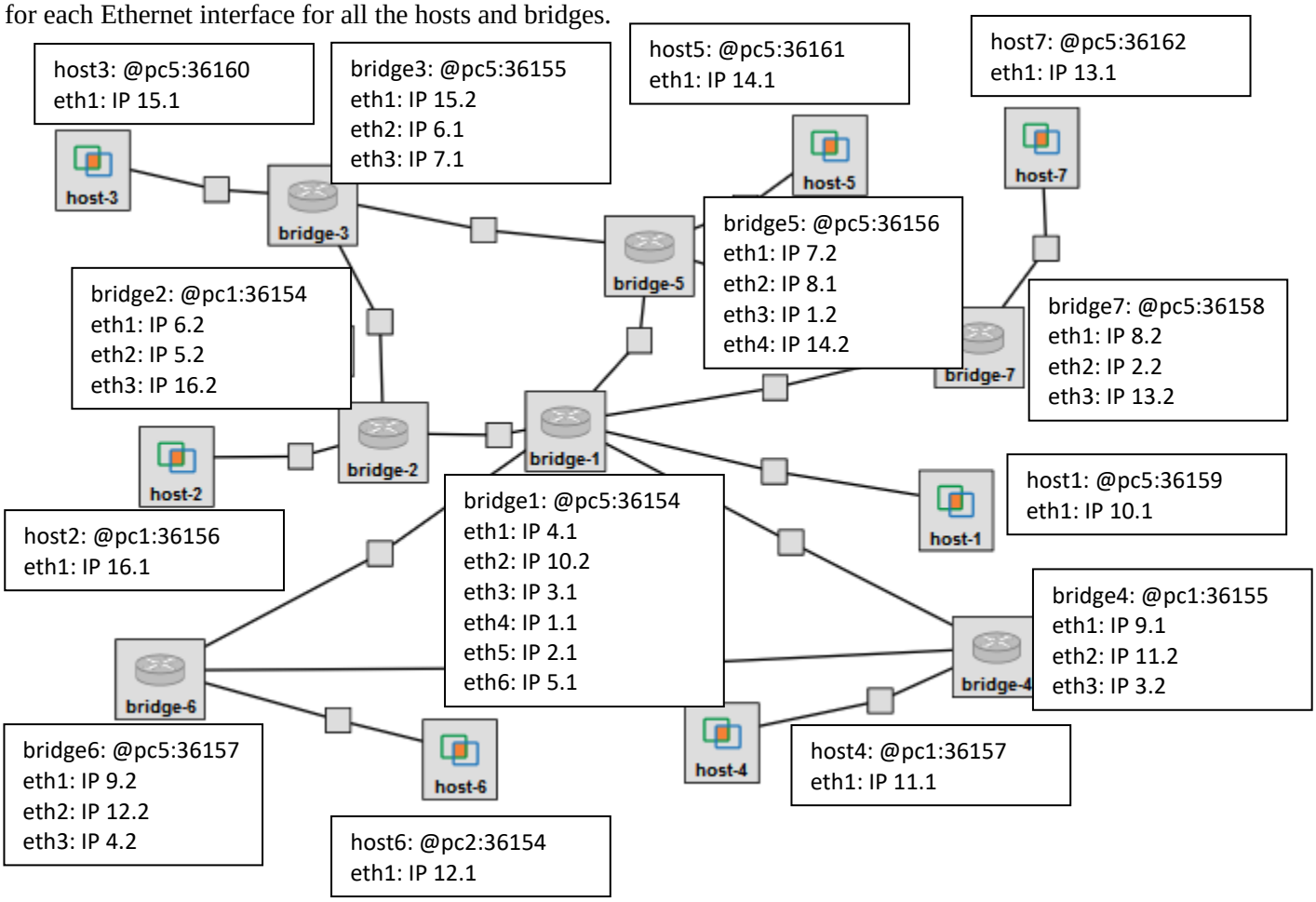

Figure 2. Subnet 10.10.0.0/16

Once you have the mapping from IP addresses to Ethernet interfaces you can now determine which interfaces are used to connect the bridges. Redraw the figure to show the mapping.

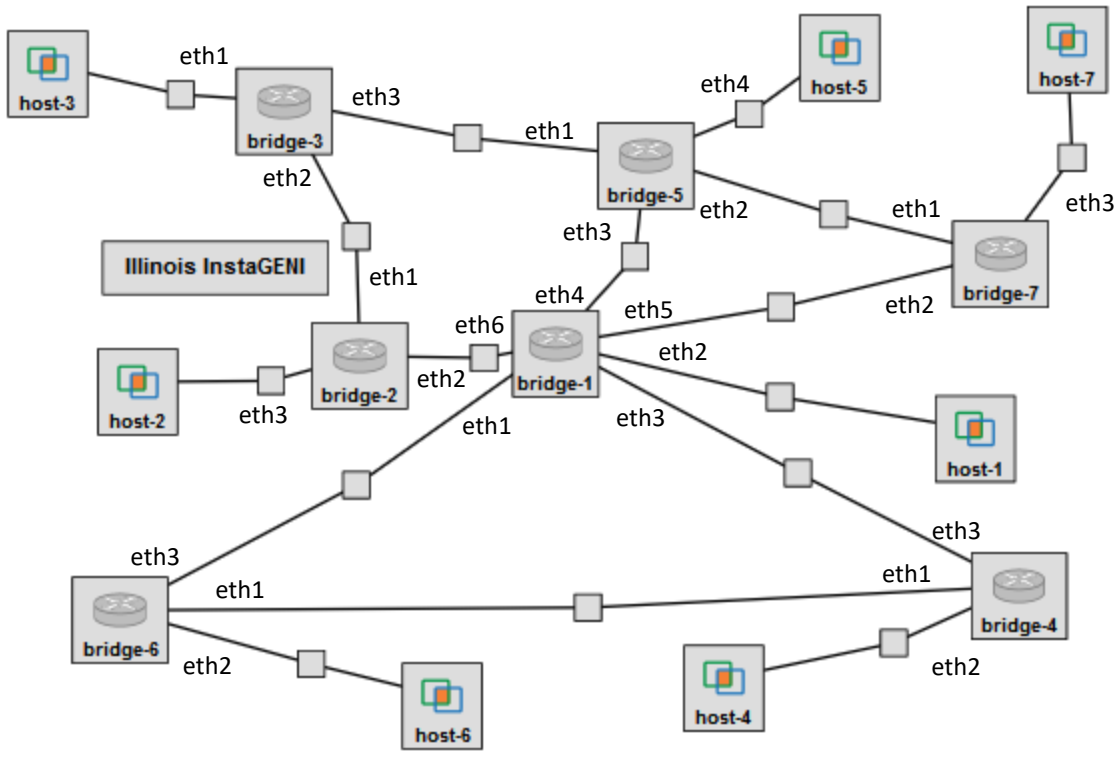

Figure 3. Interface mappings for ELAN

Configure Bridges

Log into each bridge and enable the software to operate the VM as a bridge instead of a router. Each bridge must run the spanning tree algorithm. Because the bridges are layer-2 devices, they do not need IP addresses.

Here are the commands used to configure **`bridge-1`** in the example network.

Create a new bridge, enable the spanning tree protocol, and add all the interfaces except **`eth0`**

```
sudo ovs-vsctl add-br br0
sudo ovs-vsctl set bridge br0 stp_enable=true
sudo ovs-vsctl add-port br0 eth1
sudo ovs-vsctl add-port br0 eth2
sudo ovs-vsctl add-port br0 eth3
sudo ovs-vsctl add-port br0 eth4
sudo ovs-vsctl add-port br0 eth5
sudo ovs-vsctl add-port br0 eth6
sudo ovs-vsctl show
```

Remove the IP addresses for all the interfaces

```
sudo ifconfig eth1 0
sudo ifconfig eth2 0
```

```
sudo ifconfig eth3 0
sudo ifconfig eth4 0
sudo ifconfig eth5 0
sudo ifconfig eth6 0
```

Bring the bridge up

```
sudo ifconfig br0 up
```

Repeat the above steps for each bridge.

Tip: Because you will need to log into the various nodes multiple times, create aliases to make keyboard shortcuts. Because the configuration for each bridge is similar, create a file for each bridge with all of the commands for that bridge. The notes section at the end of this file describes how to create **bash** aliases and to use **sftp** to copy files.

After the bridges are configured, all the hosts are connected to the same extended LAN. The IP address and mask must be reconfigured for each host so that all the hosts are on the same subnet. There are two simple options. One approach is to reassign the IP address at each host and leave the subnet mask as **255.255.255.0**. The second approach is to only change the subnet mask so that all the existing IP addresses are part of the same subnet. One simple subnet mask is **255.255.0.0** since all the hosts have IP addresses with the common prefix **10.10**. Recall that the class A address **10.0.0.0/8** is for private networks, so you can use any subnet mask you like.

As an example, consider **host-1**. To change the IP address so all hosts use subnet number 123.21 (you can pick any valid subnet number):

```
sudo ifconfig eth1 10.123.21.1/24
```

Notice you need to include the **/24** to set the mask to **255.255.255.0**. If you don't give the CIDR mask, the interface will default to **255.0.0.0** since this is a class A address. Alternatively, to change just the mask at **host-1**in the example network (and not its IP address):

```
sudo ifconfig eth1 netmask 255.255.0.0
```

Monitor the spanning tree configuration messages

Open an ssh window for each host and verify that the spanning tree configuration messages are visible. For example, on **host-1** do

```
sudo tcpdump -i eth1
```

This will show all packets that appear on interface **eth1** at **host-1**. You should see a "**STP 802.1d**" packet about once every 2 seconds. You will also see other packets occasionally, such as **IP6 ICMP6** multicast query messages. Filters are used to limit which packets are displayed. To see only **stp** messages (and additional details about these packets) use:

```
sudo tcpdump -i eth1 stp -vv
```

Notice in the detailed version of the **stp** packets that the **bridge-id** is the source of the packet, the **root-id** is the bridge number for the root bridge, and the **root-pathcost** for the cost of the path to the root bridge.

ARP messages

Stop the **tcpdump** on one of the hosts (e.g., **host-7**). Change the **tcpdump** filters on all other hosts to no longer print **stp** packets but instead show **icmp** and **arp** packets:

```
sudo tcpdump -i eth1 icmp or arp -vv
```

View the **arp** table on **host-7** using:

```
arp
```

Find one host that is not listed in the table and ping that host to generate **arp** messages. If the **arp** table lists all of the hosts, delete one of the entries using **-d**. For example, if **host-1** has IP address **10.10.10.1** in the example network, then on **host-7** do:

```
sudo arp -d 10.10.10.1
ping 10.10.10.1
```

Notice that all of the hosts will receive an **ARP Request**, and only **host-1**, in this example, will generate an **ARP Reply**. Also, only **host-1** will receive **ICMP echo request** packets and generate **echo reply packets**.

Discover the spanning tree

Use **ping** packets at the hosts and **tcpdump** captures at the bridges to determine which links are included in your spanning tree. For example, in the example network pings from host 7 to host 6 are created using:

```
host-7:~$ ping 10.10.12.1
```

Then trace the path that the ping packets follow at each bridge. For the network shown in Figure 3, at **bridge-7** the ping packets are seen on the **eth1** interface but not the **eth2** interface.

```
bridge-7:~$ sudo tcpdump -i eth1 icmp
bridge-7:~$ sudo tcpdump -i eth2 icmp
```

Because **bridge-7** does not use the **eth2** interface, this shows that the link between **bridge-1** and **bridge-7** is not included in the spanning tree. Likewise, examining **bridge-5**, these ping packets are seen on **eth3** but not **eth1**. At **bridge-1**, the ping packets are seen on **eth4** and **eth1**. Finally at **bridge-6**, the packets are seen on **eth3** and **eth2**. Changing **host-7** to ping **host-2**:

```
host-7:~$ ping 10.10.16.1
```

and using **tcpdump** at the bridges shows that the packets follow the bridges on the path 7 -- 5 -- 3 -- 2. This shows that the link between **bridge-1** and **bridge-2** is not in the spanning tree. Finally, setting

up a ping at **host-6** to **host-4** shows these ping packets follow the bridges on the path 6 -- 1 -- 4, showing that the link between **bridge-4** and **bridge-6** is not in the tree.

The root bridge can be discovered by using the **tcpdump** filter **stp -vv** to show the **root-id**. Alternatively, at a bridge use

```
sudo ovs-vsctl list bridge
```

and find the "**status**" field. The **stp_bridge_id** shows the bridge number for the spanning tree protocol, and the **stp_designated_root** shows the identification number of the root bridge. Check the bridges to find which one has the same identification number as the **root-id**. In the example network, it is **bridge-3** that is the root.

The following figure shows the root bridge and the links that do not participate in the spanning tree.

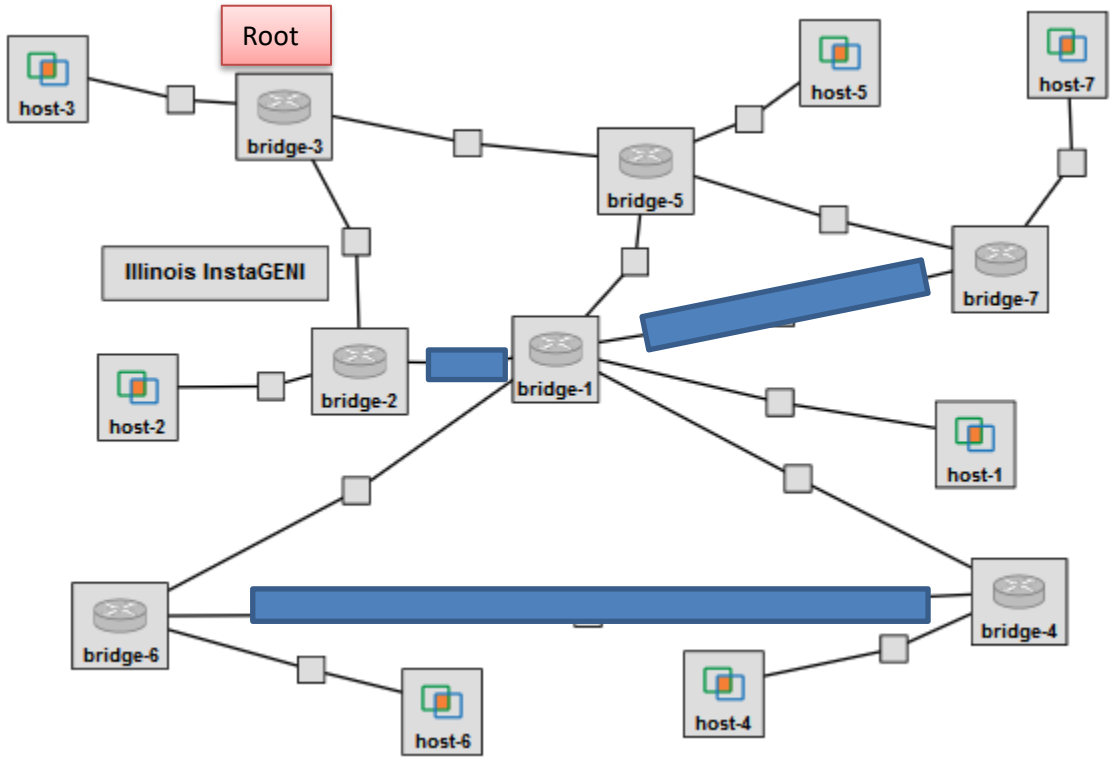

Figure 4. Spanning tree showing the links not included for forwarding packets

Further Reading

The spanning tree protocol is described in the IEEE standard 802.1d. There have been numerous extensions and improvements, including the rapid spanning tree protocol, better support for virtual LANs, and shortest path bridging (IEEE 802.1aq).

Questions

1. Illustrate your spanning tree

Create figures similar to Figures 1 through 4 in this document to show the spanning tree for your example network.

2. Link failure and recovery

Identify the bridges that are leaves in your spanning tree (e.g., in the example network, bridges 2, 4, 6, and 7 are leaves). Start a ping trace from one host attached to a leaf bridge to another host attached to a leaf bridge. Identify a link in the path between these two hosts that is part of a loop. Bring the link down and observe that the ping packets are interrupted until the spanning tree is repaired. How many ping packets are lost until the tree is repaired?

For example, in the example network, host 7 pings 2, and host 6 pings 7. The link at bridge-5 to bridge-3 is brought down.

```
host-7:~$ ping 10.10.16.1
host-6:~$ ping 10.10.13.1
bridge-5:~$ sudo ifconfig eth1 down
```

After a few minutes, bring the link back up and again determine how many ping packets are lost.

3. Change the root bridge

Identify the longest path between two hosts that is also not the minimum path. In the example network the path from host-2 to host-4 involves five bridges but the minimum path would utilize three bridges only. Change the bridge **stp-priority** for one or more bridges so that your selected path now follows the shorter route. Describe how you changed the bridge priority values and draw a figure for the new spanning tree.

From the ovs-vsctl manual that is on-line:

other_config : stp-priority: optional string, containing an integer, in range 0 to 65,535

> The bridge's relative priority value for determining the root bridge (the upper 16 bits of the bridge-id). A bridge with the lowest bridge-id is elected the root. By default, the priority is 0x8000.

Use this command to change the bridge-id (substitute **<some-hex-value>** with your integer):

```
sudo ovs-vsctl set bridge br0 other_config:stp-priority=0x<some-hex-value>
```

Lab Report

Your laboratory report must include the following sections:

- A cover page with your name, course information, lab title, and date of submission
- Summary: a summary of the objectives of the lab

- Implementation: a brief description of your implementation
- Results: figures and answers to the given questions
- Conclusion: a brief summary that includes what you learned, difficulties you encountered, and any suggested extensions or improvements.

The report has to be electronic (in PDF format) and must be submitted to Canvas by completing the assignment (submit on the same page that has the PDF for the assignment).

While you are encouraged to discuss the general approach of the problem with others, you are expected to do your own work. **In particular, you must write your own lab report**. **Do not share text, figures, or your rspec file with other students.**

The machine problem is worth 100 points. For each day your submission is late your score will be reduced by an additional 2^i points where $i = 1$ for a submission after the due date any anytime during the next 24 hours, etc. (Note that this rule applies to machine problems only. Late homework sets are not accepted.)

Turning in a PDF is required. Microsoft Word can save as PDF. There are also many online sites that will convert common file formats to PDF. Examples include Adobe, PDFonline or FreePDFConvert. There is also free or cheap software that installs as a printer driver and performs that conversion, including PrimoPDF.

Tips

1. Working in the unix home directory (such as on your laptop, one of the CES apollo machines, or one of the VMs), you can create (or add to) a file called "**`.bash_aliases`**". Note the file starts with a dot. Files and directories that start with a dot are not shown unless you use **`ls -a`** to see all. After you create the file, either log out and back into your account or use **`"source .bashrc"`**. After the aliases are loaded you can type a shortcut in a terminal.

Here is a partial example of the **`.bash_aliases`** file I created:

```
alias b1="ssh harlanr@pc5.instageni.illinois.edu -p 36154"
alias b2="ssh harlanr@pc1.instageni.illinois.edu -p 36154"
alias h1="ssh harlanr@pc5.instageni.illinois.edu -p 36159"
alias h2="ssh harlanr@pc1.instageni.illinois.edu -p 36156"
```

Update the entries with the ssh information found in your rspec file. If you are asked for a password, then add the **`-i`** option or run the **`ssh-add`** command on your local machine as described in the tutorial.

Tip: On the Apollo machines it appears that the **`ssh-agent`** is not running. This means you have to re-enter you ssh passphrase every time! If you attempt to run "**`ssh-add`**" and get the message "**`Could not open a connection to your authentication agent`**", then do this:

```
apollo08:~$ exec ssh-agent bash
apollo08:~$ ssh-add .ssh/id_geni_ssh_rsa
```

2. There is a considerable amount of repetitive typing required to configure a bridge. You can put all of the commands into a file and then execute the file. For example, create a file for the first bridge and then make and edit copies for the other bridges. For example

```
sudo ovs-vsctl add-br br0
sudo ovs-vsctl add-port br0 eth1
sudo ovs-vsctl add-port br0 eth2
sudo ovs-vsctl add-port br0 eth3
sudo ovs-vsctl set bridge br0 stp_enable=true
sudo ifconfig eth1 0
sudo ifconfig eth2 0
sudo ifconfig eth3 0
sudo ifconfig br0 up
sudo ovs-vsctl show
ifconfig -a
```

After the files are prepared copy them to each VM using **sftp** or **scp**. For example, to copy a file to **bridge-1** use (or make additional aliases):

sftp -oPort=port [user@]host

For example, my local user name is the same as my remote user name so I don't have to include it:

sftp -oPort=36154 pc5.instageni.illinois.edu

then at the **sftp** prompt use **put** to move the files to the remote machine

sftp> **put filename**

Then **ssh** into the remote machine and do

sh filename

Be sure to check the output for errors.