Install necessary software:

1. Start VirtualBox VM: `GENI_VM_Silver.ova` (and log in using username "`Geni Tutorial`" and password "`geniuser`")
2. Copy (by drag and drop) three files from the USB drive to the VM desktop: `eclipseforsummercamp.tgz`, `primexforsummercamp.tgz`, and `summercampmodels.tgz`. (You may first need to change the VirtualBox VM setting to allow the VM to access to the USB device).
3. Install eclipse and PRIMEX (the real-time network simulator):
    - `mkdir ~/bin`
    - `cd ~/bin`
    - `tar xzvf ~/Desktop/eclipseforsummercamp.tgz`
    - `cd ~/workspace`
    - `tar xzvf ~/Desktop/primexforsummercamp.tgz`
4. Create slingshot project in eclipse:
    - Start eclipse located in folder: `~/bin/eclipse`
    - Create new java project, and name it "slingshot"
    - Import slingshot source code:
        - Right click on the project and click "import"
        - Go to General->File System; click 'Next'
        - Click 'Browse' and choose '`/home/geni/workspace/primex/netIDE`' folder
        - Check the box on left of netIDE and click 'Finish'
    - Choose 'Yes to All' when prompted
5. Set up slingshot (run from within eclipse):
    - Expand the slingshot project and right click on 'slingshot_linux_32.product' and choose: "Run as->Eclipse Application".
    - A small window opens up prompting for a workspace folder; we use the default: "/home/geni/slingshot"; check the box "Remember the workspace", and click "OK".
    - When prompted for creating a new folder, click "OK".
    - Quit from slingshot and install the models
        - `cd ~/slingshot/Models`
        - `tar xzvf ~/Desktop/summercampmodels.tgz`

A simple simulation experiment

1. Start slingshot (you'll first see the "project perspective")
2. Select the file Models/basics/BasicDumbbellModel.py in the "project explorer"; double click to open the file in the "file editor".
3. Inspect the python model:
    - Create top-level network "topnet"
    - Create subnet "leftnet"
    - Create four hosts within the subnet, each with a network interface
    - Create a router "r"
    - Create four links connecting a newly created interface at the router and the interface at each of the four hosts
    - Create an exact copy of the subnet to be "rightnet"
    - Create a link connecting the two routers at "leftnet" and "rightnet"
4. Create experiment:
    - At prompt, enter experiment name (say, test1), and browse for the source code for the model
    - Slingshot brings you to the "model perspective", which consists of a "prefuse window", a "tree view", and a "python console"
    - The model is displayed in the "prefuse model" (using right button or control-right butter to center the network; changing view depth to view expand or collapse the network)
    - Inspect the attributes in "tree view"
    - Change the middle link's bandwidth from 1e8 to 1e7 (you need to use "control-return" to confirm the change)
5. Compile the experiment and run it
6. Add dynamic traffic (at the "python console"):
    - topnet = exp.getRootNode() #(or select the topnet by using the mouse, then at the console: topnet=sel)
    - tf = exp.createDynamicTrafficFactory(topnet)
    - left_h1 = topnet.get("leftnet.h1")
    - right_h1 = topnet.net.get("rightnet.h1")
    - tf.createSimulatedTCP(10, 1000000000, left_h1, right_h1) #(at time 10, left_h1 will request download of 1 GB from right_h1)
    - You can also select nodes using mouse and use console to assign the selected node to a variable:
        i. (select one node)
        ii. x = sel
        iii. (select another node)
        iv. y = sel
        v. tf.createSimulatedTCP(40, 1000000000, x, y)

Create a protogeni environment

1. Untar the manifests.tgz file provided in ~/slingshot/.
2. Click on Experiment->Create/Edit Environments
3. Name the enviroment "utah" and select "ProtoGENI" as the type of environment. Click

on “Add”.

4. You were assigned a group number (if you do not know please ask). Then, load the file ‘primo-scX’ where X is your group number and click on “Add”.
5. Now, you must select which machine will be your machine that will host the simulator (slave). In this case, such machine is the one which has 3 nics. Select that machine and check the ‘Slave’ checkbox. Click on Finish twice.
6. **We need to set up the password-less login to the machines**

A simple emulation experiment

1. Before we create this simple emulation experiment, we need to setup another environment with no external hosts (portals) in it. Create another environment using the same manifest used above. Edit it: Experiment->Create/Edit Environments, then select the just create second environment, click ‘Edit’, select the node under ‘Control’, click ‘Edit’, then in the Portals box, remove both portals, click “Finish”.
2. We create another experiment with the dumbbell model, name it “test2”
3. At the console, we select a node to be emulated: sel.enableEmulation()
4. Compile and run experiment on the protogeni environment
5. Click on the first emulated host, you should note the “compute node” at the “tree view” indicate the machine on which the emulated host is running; also make a note of the “uid” and “ip_address” of the host
6. In a command shell, ssh to the machine
7. Become root: sudo su -
8. Check the emulated host is running: vzlist
9. Enter the emulated host: vzctl enter 30 (assuming 30 is the uid)
10. Once in the emulated host: ifconfig
11. You can ping other nodes, for example: ping 192.1.0.33
12. You can download from other nodes, for example: wget 192.1.0.33/100000.html

An emulation experiment with portals

1. Create a new experiment named ‘test3’.
2. Load the model from Models/summercamp/traffic/ CampusPortalEmulationHTTPClientTrafficForUtah.java.
3. At this point, the network topology should be showing on your screen. For better visualization, expand the model as desired using the ‘View Depth’ option.
4. If you expand ‘net01’ by double clicking into it, you will notice that some nodes are bigger than the others and they are highlighted. These are “emulated hosts”, i.e., hosts which will be mapped to virtual machines but whose traffic is intercepted by the simulator. We have two emulated hosts which are named ‘emuhost’.
5. Compile the model by clicking in “Compile Experiment” on the bottom right corner of your screen. Select the default IP space and click “Finish”.
6. Run the experiment.
7. Select the ‘utah’ enviroment just created and click “Next”.
8. Input the number of seconds you want to run this exp, e.g., 1000 seconds.
9. Match the IPs from the model “Portals” to those in the physical machines “Interfaces”. Click on “Finish”.

10. Wait until the experiment is being instantiated. Click on the “Log” tab if you want to follow the steps performed. Click “OK” when popup window displays.
11. Now the experiment is running and you should observe a value next to “Pacing”; which shows the pace of the simulator (1 if running on real time).
12. Copy id_dsa and id_dsa.pub to ~/.ssh so that we can log in to nodes in your slice without using password.
13. Take a look at the manifest you used to create the environment, the first node is an external node, log in to that node and become root (sudo su -).
14. Add a route in this external node so that packets to the IP space you chose above are routed to the simulator: ‘ip route add via 10.10.1.2 dev eth2’ (sometimes eth2 can be eth4 or eth3).
15. Now, from that node you can reach any node in the simulation, including an ‘emulated node’. Try:
    - traceroute -I 192.1.8.1
    - ping 192.1.8.26
    - wget 192.1.8.26

    They should all succeed. While you do that take a look at the route that those packets follow.
16. Log in to the ‘primogeni1’ node as indicated in the manifest used.
17. Execute ‘vzlist’. You should that two openvz containers with ids 277 and 301 are running.
18. Log in to the first container using ‘vzctl enter 277’
19. Ping the external node from there: ‘ping 10.10.1.1’.
20. This shows that connectivity between the simulated, emulated and external nodes exists.