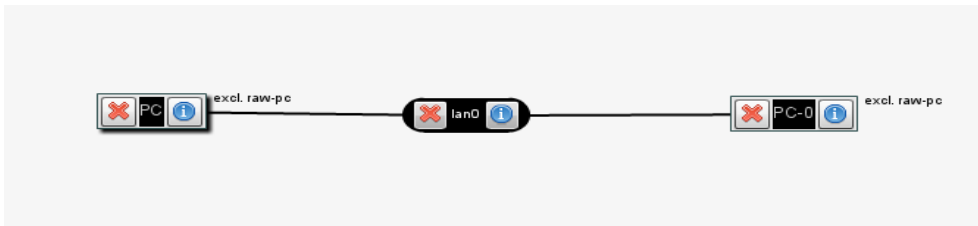


The purpose of this tutorial is to give an introduction on how to set up and run a simple experiment using the tmix TCP traffic generator on a GENI slice. This tutorial presupposes that the user has GENI credentials and has their ssh client configured with their GENI identity's keyfile.

- Other prerequisites:
- A tmix tarball complete with configuration and cvec files
 - A means of getting this tarball onto the GENI machines
 - Some familiarity with Flac

Step One-Slice Creation:

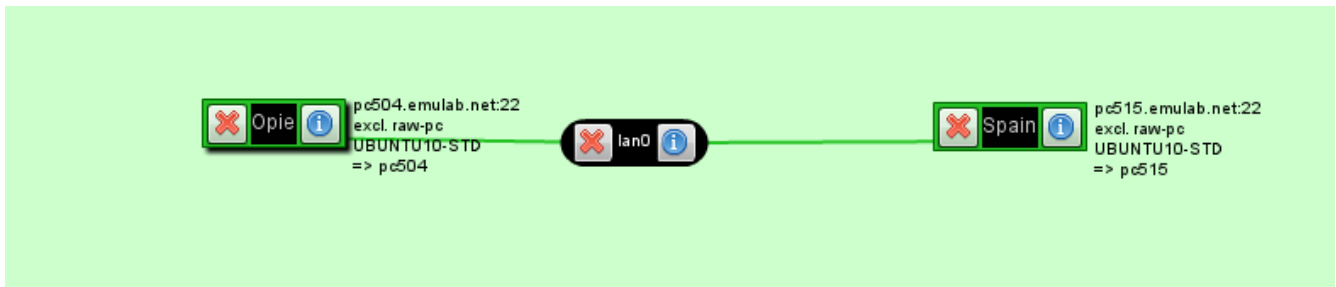
Create a slice in flack with a unique name. Something like [username]Tmix will work fine. In this slice add two raw hosts (not VMs) from utahemulab (other aggregates should work but we have the best luck with this one). Create a LAN connection between these two hosts as shown below:



Configure both nodes, by clicking on the blue 'i' icon, with a unique hostname (we will use 'opie' and 'spain' for this tutorial since these are the hostnames preconfigured in the tarball) and select the Ubuntu 32bit image from the dropdown box. After you have the nodes configured, configure the LAN by selecting its blue icon. Here you need only fill in the IP and Mask fields as shown below:

Interfaces				Properties	
Node	Interface	Physical	MAC	IP	Mask
Opie	PC:if0			10.1.1.1	255.255.255.0
Spain	PC-0:if0			10.1.1.2	255.255.255.0

With the LAN and both machines configured you can now submit the slice. Wait until Flac's background turns green and shows the routable hostnames for your machines on the canvas like this:



Step Two-Setting up tmix:

Now that your machines are up you will need to move the tarball onto them and set up tmix and usernet for delays. For brevity's sake this tutorial will only go over the steps for one machine. All of the following steps must be completed on both machines symmetrically.

First move the tmix tarball into your home folder on the machine with scp or equivalent method. It's worth noting here that the Ubuntu images from utahemulab deviate slightly in the standard directory structure such that your home folder is in /users. For instance my home folder is /users/doneill on both machines. Once the file transfer is complete extract the tarball with 'tar -xf' or similar. The scp command will look something like: `scp tmixv2.tgz:/users/<username>/` and after logging into the host run `tar -xf tmixv2.tgz`

The next step is to modify the tmix configuration file for the machine. The prepared tarball contains two configuration files in ~/take2/tmix/ one called opie.conf and the other spain.conf. You only need to modify one configuration file per machine. In the config file change Peer_Host_Name to reflect the hostname of the *other* machine (entries in /etc/hosts were added automatically) eg opie.conf would look like this:

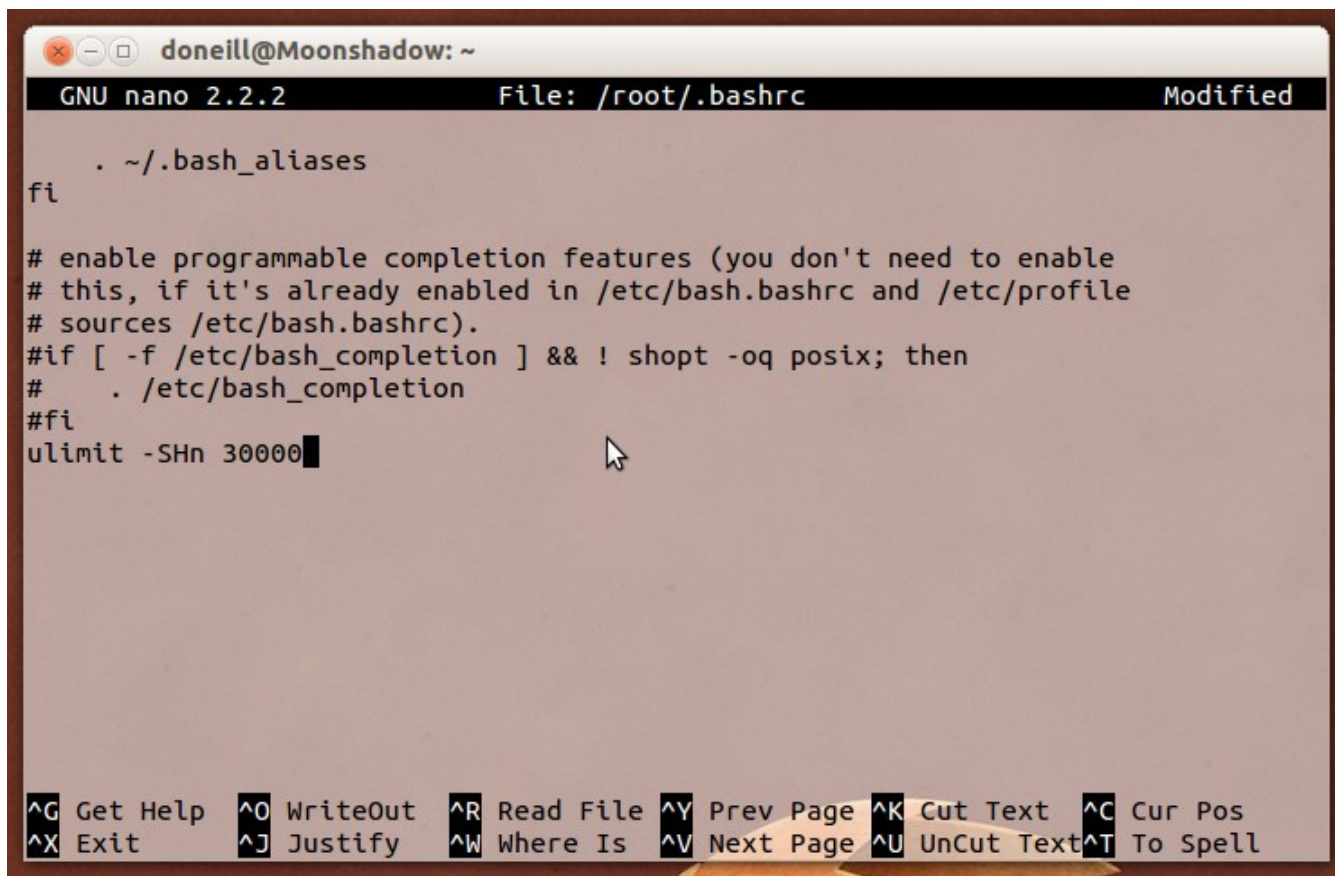
```
doneill@Moonshadow: ~  
GNU nano 2.2.2 File: take2/tmix/opie.conf  
  
# Host Name of Traffic Generation Peer  
Peer_Host_Name = spain  
  
# Number of Active Port in Traffic Generation Peer  
# (Number of Passive Port is Peer_Active_Port + 1)  
Peer_Active_Port = 1700  
  
# Number of Local Active Port  
# (Number of Passive Port is Local_Active_Port + 1)  
Local_Active_Port = 1700  
  
# Limit Duration for Traffic Generation (in seconds)  
# If undefined, the generator stops after all cvecs are reproduced  
Duration = 4200
```

While the configuration file is opened you also need to change Usernet_Delay to 1 if you intend on using that feature. Usernet is the means that tmix uses to emulate round trip times. The configuration file is also where you would change the init and recv cvec files. These files determine the traffic that tmix generates. We have provided example files preconfigured for this tutorial.

Step Three-Configuring the OS:

The next change that needs to be made is to set the operating system to allow us to have open more than the default maximum number of file descriptors. The following process may seem unnecessarily obtuse but it is the result of the Ubuntu image not allowing non-root users to use ulimit.

First, become root with `sudo -i` and open `/root/.bashrc` for editing. Add `'ulimit -SHn 30000'` to the end of the file which will set a new hard limit of 30,000 open files whenever root logs in. When you're finished it should look like this:



```
doneill@Moonshadow: ~
GNU nano 2.2.2      File: /root/.bashrc      Modified

. ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
# sources /etc/bash.bashrc).
#if [ -f /etc/bash_completion ] && ! shopt -oq posix; then
#   . /etc/bash_completion
#fi
ulimit -SHn 30000

^G Get Help  ^O WriteOut  ^R Read File ^Y Prev Page ^K Cut Text  ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^V Next Page ^U UnCut Text ^T To Spell
```

Logout of root with `exit` and then become root again with `sudo -i`. Run `ulimit -n` and confirm that the limit has been set to 30,000.

Step Four-Set up Usernet:

If you intend to use usernet delays you will need to compile the kernel module, insert it into the kernel, and attach it to the desired interface.

First, you will need to install kernel headers for the version of the kernel running on your machine. As of this writing the proper package for utahemulab's standard Ubuntu image is 'linux-headers-2.6.32-24-generic-pae'. Simply install with `apt-get install <header package>` after running `apt-get update`.

With that set up navigate to `~/take2/usernet/sch_tmix/` and run `make`. When make completes insert the newly-compiled module into the kernel with `sudo insmod sch_tmix.ko`. The next step is to attach it to the proper interface. Use `ifconfig` to see which interface has the IP address you assigned in step one.

Run the following command to attach usernet to the interface: `sudo tc qdisc add dev <interface> root usernet`.

Step Five-Run tmix:

The final step is to actually run tmix. It works best if you have two terminals open (one per host) and execute these steps concurrently. First, become root with `sudo -i` or equivalent. Then navigate to the tmix folder eg, `/users/doneill/take2/tmix/`. It's a good idea to run `date` here on both machines and ensure that both machines have synced clocks. After running `date` also note what the node's time is as it may be different from your local clock. Tmix will run with timing based on the node's timezone. To actually run tmix issue the following command on both nodes: `./tmix -s HH:MM:SS [hostname].conf` where `[hostname]` is replaced with the name of the machine it's being run on and `HH:MM:SS` is the 24hr time for the node that you want tmix to start. At this point tmix will run and output preliminary information about the cvecs it will generate. It will not actually start generating traffic until the start time specified in the command line. It's best to give a few minutes lead time when setting the start time parameter to allow for tmix's setup process and the time it takes you to run the command in both terminals.

Notes:

This tutorial does not set up any data collection mechanisms outside of tmix's own logging functionality. By default with the current tarball the logs will be saved to the tmix folder with a prefix of unc followed by the date the original data was collected.

The default experiment will run for ~1 hour. Though ssh tunnels are relatively stable it is highly recommended to use something like the 'screen' program to maintain execution in the event the ssh tunnel collapses or the connection is lost.