

GENI Tutorial with tmix

Derek O'Neill

UNC Chapel Hill

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 flack

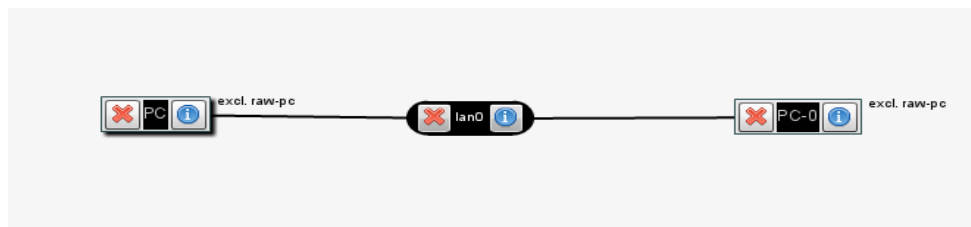
[Help With Prerequisites:](#)

Flack: <http://www.protogeni.net/trac/protogeni/wiki/FlackTutorial>

SSH: <http://groups.geni.net/geni/wiki/GEC11NetworkingExperiments/Tutorial#VIII.GetYourOwnAccount>

Step One-Slice Creation:

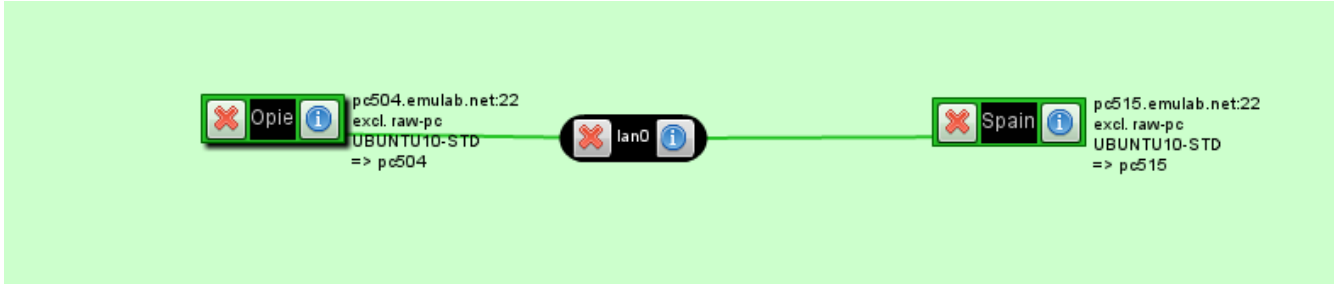
Create a slice in flack with a unique name. Something like [username]Tmix will work fine. In this slice add two raw hosts (not VMs) from utahemulab (other aggregates should work but we have the best luck with this one). Create a LAN connection between these two hosts as shown below:



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 flack'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.

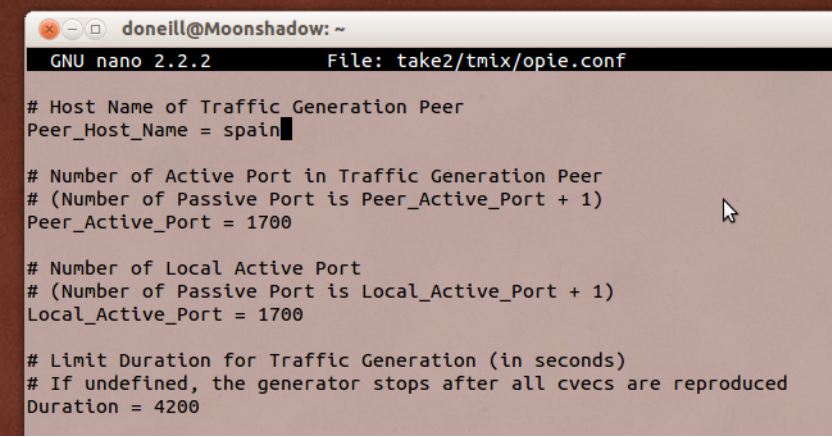
First move the tmix tarball into your home folder on the machine with scp or equivalent method. It's worth noting here that the Ubuntu images from utahemulab deviate slightly in the standard directory structure such that your home folder is in /users. For instance my home folder is /users/doneill on both machines. The scp command will look something like: `scp tmixv2.tgz`

`<username>@<hostname>:/users/<username>/` where <hostname> is the net-accessible domain name of the machine. After logging into the host run `tar -xf tmixv2.tgz` to extract the tarball. This entire process will look like this in the terminal:

```
doneill@Moonshadow: ~
doneill@Moonshadow:~$ scp tmixv2.tgz pc203.emulab.net:/users/doneill
The authenticity of host 'pc203.emulab.net (155.98.39.3)' can't be established.
RSA key fingerprint is 46:63:92:67:c8:75:20:4e:52:9f:2d:f6:cb:58:16:77.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'pc203.emulab.net,155.98.39.3' (RSA) to the list of k
nown hosts.
tmixv2.tgz          100% 27MB 127.4KB/s   03:34
doneill@Moonshadow:~$ ssh pc203.emulab.net
Linux opie.doneiltutorial.pgeni-gpolab-bbn-com.emulab.net 2.6.32-24-generic-pae
#38+emulab1 SMP Wed Aug 4 10:25:17 MDT 2010 i686 GNU/Linux
Ubuntu 10.04.1 LTS

Welcome to Ubuntu!
 * Documentation: https://help.ubuntu.com/
opie:~> tar -xf tmixv2.tgz
opie:~> █
```

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

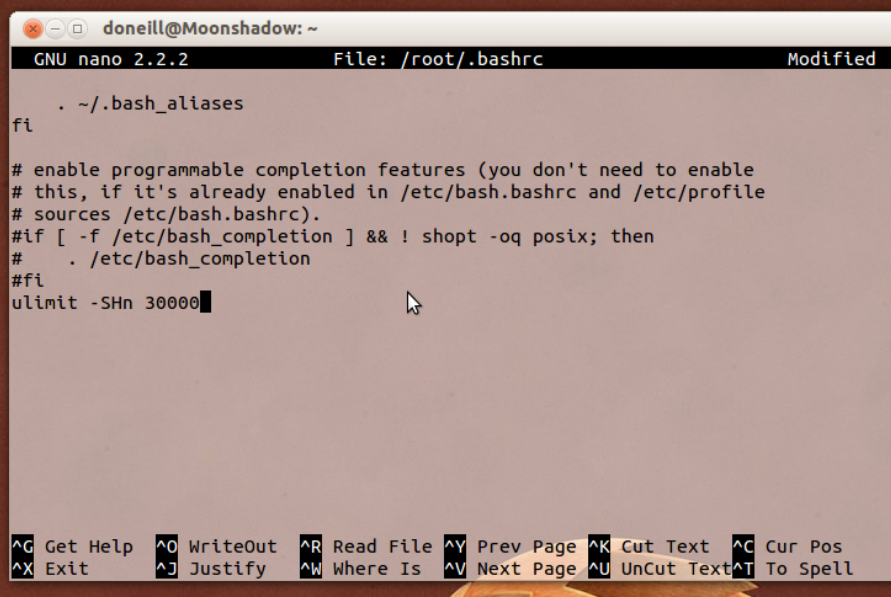
# Limit Duration for Traffic Generation (in seconds)
# If undefined, the generator stops after all cvecs are reproduced
Duration = 4200
```

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 rcv 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.

Fist, 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:



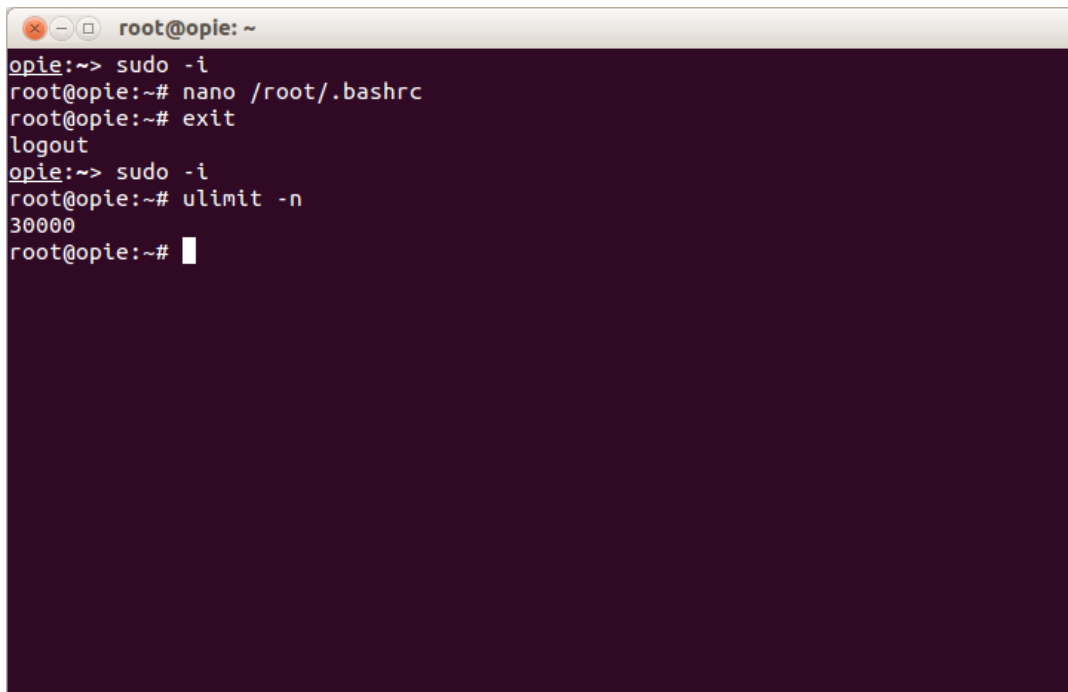
```
doneill@Moonshadow: ~
GNU nano 2.2.2 File: /root/.bashrc Modified

. ~/.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  ^V Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^N 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. After all of this your terminal should look like this:



```
root@opie: ~
opie:~> sudo -i
root@opie:~# nano /root/.bashrc
root@opie:~# exit
logout
opie:~> sudo -i
root@opie:~# ulimit -n
30000
root@opie:~#
```

Step Four-Set up Usenet:

If you intend to use usenet 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'. Assuming you are still root, simply install with `apt-get update` followed by `apt-get install <header package>`.

With that set up, navigate to `/users/<username>/take2/usenet/sch_tmix/` and run `make`. When `make` completes insert the newly-compiled module into the kernel with `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: `tc qdisc add dev <interface> root usernet`. The ifconfig and tc steps will look like this:

```
root@opie: /users/doneill/take2/usernet/sch_tmix
collisions:0 txqueuelen:100
RX bytes:43173141 (43.1 MB) TX bytes:1460509 (1.4 MB)

eth2    Link encap:Ethernet HWaddr 00:04:23:b7:23:a0
        inet addr:10.1.1.1 Bcast:10.1.1.255 Mask:255.255.255.0
        inet6 addr: fe80::204:23ff:feb7:23a0/64 Scope:Link
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1
        RX packets:0 errors:0 dropped:0 overruns:0 frame:0
        TX packets:12 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:1000
        RX bytes:0 (0.0 B) TX bytes:1480 (1.4 KB)

lo      Link encap:Local Loopback
        inet addr:127.0.0.1 Mask:255.0.0.0
        inet6 addr: ::1/128 Scope:Host
        UP LOOPBACK RUNNING MTU:16436 Metric:1
        RX packets:83 errors:0 dropped:0 overruns:0 frame:0
        TX packets:83 errors:0 dropped:0 overruns:0 carrier:0
        collisions:0 txqueuelen:0
        RX bytes:7707 (7.7 KB) TX bytes:7707 (7.7 KB)

root@opie:/users/doneill/take2/usernet/sch_tmix# tc qdisc add dev eth2 root user
net
root@opie:/users/doneill/take2/usernet/sch_tmix#
```

Notice that eth2 has the 10.1.1.1 address.

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 if you haven't already. 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. This part of the process should look like this in the terminal:

```
root@opie: /users/doneill/take2/tmix
opie:~> sudo -i
root@opie:~# cd /users/doneill/take2/tmix/
root@opie:/users/doneill/take2/tmix# date
Sat Nov  3 17:13:48 MDT 2012
root@opie:/users/doneill/take2/tmix# ./tmix 17:16:00 opie.conf
Compiled on Jun 13 2012 (12:00:18) with parameters:
  FD_SETSIZE=16384 BUSY_CONNS
Parsing command-line arguments...
Initializing tmix (config file opie.conf)...
Parsing configuration file...
Peer Host Name:      spain
Peer Active Port:    1700
Peer Passive Port:   1701
Local Active Port:   1700
Local Passive Port:  1701
Duration:            4200 seconds
Start Time Scaling:  1.00
(assuming sync start time for tomorrow (10:24:00): 1351959840 < 1351984453)
Synchronized Start Time: Sat Nov  3 10:24:00 2012
tmix: Warning: start time delay from config file not used
Connection Timeout:  75 seconds
Output_Prefix: unco8.linux1
Cvec Minimum ta:     500
Cvec Minimum tb:     500
```

The information at the bottom of the terminal is the output from tmix which continues off of the screen.

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.