

GENI Inter-Aggregate Stitching

<http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/StitchingTutorial>

version 20140610



GENI Inter-Aggregate Stitching

Overview:

This tutorial will guide you through a stitching experiment where you will:

- Use Flack to modify an existing experiment RSpec to generate a stitching RSpec.
- Create a sliver with 3 hosts with one stitched link
- Run an experiment to compare performance between a stitched link connecting two sites and a local link connecting two hosts in the same rack.

Prerequisites:

- A GENI account, if you don't already have one [sign up!](#)
- Familiarity with reserving GENI resources with any of the GENI Tools (GENI Experimenter Portal, Omni, Flack). See Flack [tutorial](#), or Omni [tutorial](#)
- Familiarity with [logging in to GENI compute resources](#).
- Familiarity with the Unix Command line and [Omni tools](#).

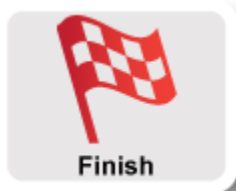
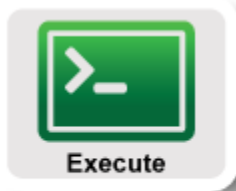
Tools:

- Omni Tools that is part of the gcf software, download it [here](#)

Where to get help:

- If you need help with GENI, or Omni Tools email help@geni.net

Tutorial Instructions

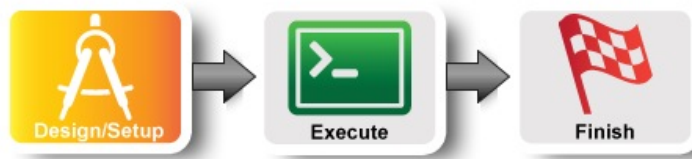


- [Part I: Design/Setup](#)
 - Step 1: Download and Import Initial Resource Request RSpec
 - Step 2: Modify RSpec to include 2 Aggregates with 1 stitched link
 - Step 3: Run sticher tool to get resources

- [Part II: Execute](#)
 - Step 4: Execute Experiment

- [Part III: Finish](#)
 - Step 5: Teardown Experiment

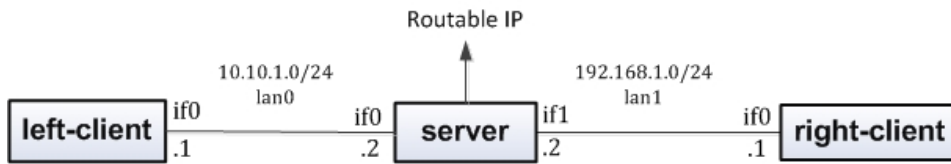
GENI Inter-Aggregate Stitching Tutorial



Step 1: Download and Import Initial Resource Request RSpec

For this tutorial we are going to use the Flack tool as an rspec editor, but we are going to do the actual reservation using Omni, a command line tool.

This tutorial starts with an existing RSpec which includes 3 nodes in a linear topology. The resources are un-bound, which means the resources are not affiliated with any rack aggregate.

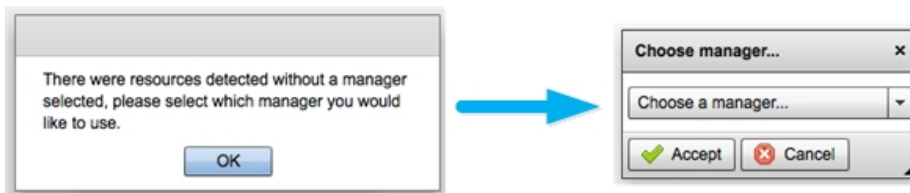


To get started download the initial RSpec file and import it into Flack to make modifications as follows:

1. Login to the GENI Portal at <https://portal.geni.net>
2. Go to the [Projects](#) panel and click on the "Create Slice" button for the tutorial project. (If you are running this tutorial on your own, you may use any existing project you belong to.) On the "Create New Slice" page enter a slice name and click on "Create slice" button. For the purpose of this tutorial we will assume the slice is named *MySlice*, but you should use a unique name because each slice name must be unique within a project.
3. On the slice page select the "Launch Flack" button. (You must wait for all aggregates to show up in the MySlice panel before you can proceed.)
4. On the Flack slice canvas, select the pull-down named "Import" and choose the "Import from the web" option. Paste this URL <http://www.gpolab.bbn.com/exp/StitchingTutorial/3nodes-1rack.rspec> and click "OK".

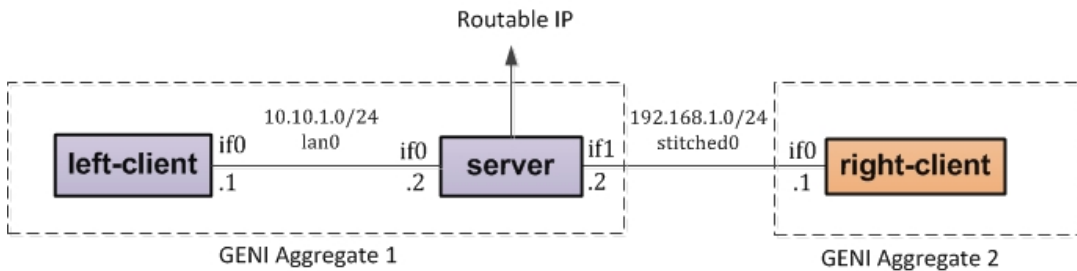


5. You will see a pop-up that states that no manager is associated with the resource; click "OK" and another pop-up will have you select an aggregate manager, select one of the two aggregate managers assigned to you and click the "Accept" button. (If you are running this tutorial on your own, you may use any aggregate found on the [GENI Network Stitching Sites](#) page.




Step 2: Modify RSpec to include 2 Aggregates with 1 stitched link


Now that you have imported the RSpec, we will modify it to include a stitched link. Modifications in this section will result in the following topology:



To get to the above topology, we must make changes to the resources imported into Flack. We will now modify "right-client" to become part of a different rack aggregate and then modify its link to be a stitched link.

6. Click on the "right-client" VM information button .

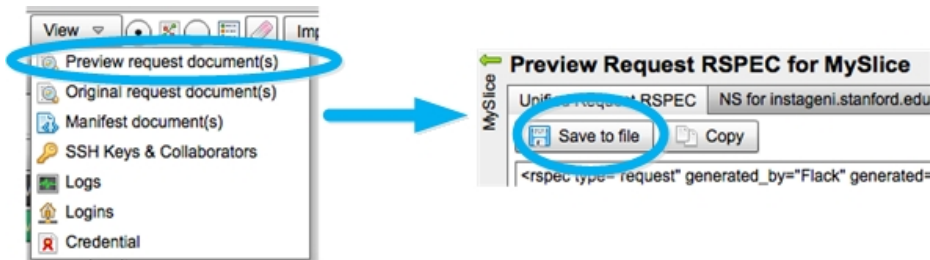
7. In the "right-client" details panel look for the "Manager" pull-down. Modify the aggregate to be the second aggregate manager that was assigned to you at the beginning of the tutorial; do not use the aggregate used earlier. Click on the "Apply" button. (If you are running this tutorial on your own, you may use any other site aggregate found on the [GENI Network Stitching Sites](#) page.)

8. You will now see that the link from the "server" VM to "right-client" VM is now a gre-tunnel link. Select the gre-tunnel link information button  to modify the link type to "stitched" and click "Apply" button.



9. You should now see that the topology has changed and that between the "server" VM and the "right-client" VM the link is now named "stitched0".

10. We will now export this topology to a file by selecting the "View" pull-down and choosing "Preview requested document(s)". You will now choose "Save to file" to export the RSpec we just modified. For the purpose of this experiment we will assume the file is saved to "MyTopology.rspec", but you can choose any name or location, just make a note of it.



We are now ready to use stitcher to create the stitched topology for our experiment!

Step 3: Run stitcher tool to get resources

The stitcher tools delivers the same features as the Omni tool, but with one major difference: stitcher will submit sliver request at all aggregate in your request RSpec. Let's now set up slivers for the request RSpec generated in the previous section:

11. Use stitcher to create a sliver, using the slice that we created in the Portal. The request will use the RSpec that was saved to a file in the setup steps. The output will look something like this:

```
$ stitcher createsliver MySlice MyTopology.rspec
11:48:18 INFO      stitcher: Loading agg_nick_cache file '/Users/lnevers/.gcf/agg_nick_cache'
11:48:18 INFO      stitcher: Loading config file /Users/lnevers/.gcf/omni_config
11:48:18 INFO      stitcher: Setting option useSliceMembers based on omni_config setting
11:48:18 INFO      stitcher: Using control framework portal
11:48:18 INFO      stitcher: Member Authority is https://ch.geni.net/MA (from config)
11:48:18 INFO      stitcher: Slice Authority is https://ch.geni.net/SA (from config)
11:48:19 INFO      stitcher: Checking that slice MySlice is valid...
11:48:19 INFO      stitcher: Slice urn:publicid:IDN+ch.geni.net:ln-prj+slice+MySlice expires on 2014-0
11:48:20 INFO      stitcher: Stitched reservation will include resources from these aggregates:
```

```
11:48:20 INFO    stitcher:    <Aggregate wisconsin-ig>
11:48:20 INFO    stitcher:    <Aggregate stanford-ig>
11:48:20 INFO    stitcher:    <Aggregate ion>
11:48:20 INFO    stitch.Aggregate:
    Stitcher doing createsliver at https://www.instageni.wisc.edu:12369/protogeni/xmlrpc/am
11:48:38 INFO    stitch.Aggregate: Allocation at <Aggregate wisconsin-ig> complete.
11:48:38 INFO    stitch.Aggregate:
    Stitcher doing createsliver at https://www.instageni.stanford.edu:12369/protogeni/xmlrpc/am
11:48:58 INFO    stitch.Aggregate: Allocation at <Aggregate stanford-ig> complete.
11:48:58 INFO    stitch.Aggregate:
    Stitcher doing createsliver at http://geni-am.net.internet2.edu:12346
11:49:43 INFO    stitch.Aggregate: DCN AM <Aggregate ion>: must wait for status ready....
11:49:43 INFO    stitch.Aggregate: Pause 30 seconds to let circuit become ready...
11:50:20 INFO    stitch.Aggregate: DCN circuit 74871 is ready
11:50:27 INFO    stitch.Aggregate: Allocation at <Aggregate ion> complete.
11:50:27 INFO    stitch.launcher: All aggregates are complete.
11:50:27 INFO    stitcher: Saved combined reservation RSpec at 3 AMs to file MySlice-manifest-rspec-
Stitching success: Reserved resources in slice MySlice at 3 Aggregates (including 1 intermediate aggr
```

We are now ready to run the experiment!

Introduction

Next: Execute

GENI Inter-Aggregate Stitching Tutorial



Step 4: Execute Experiment

To run the experiment we will use two terminals to login to the *"right-client"* and the *"left-client"* VMs and one browser to connect to the *"server"* VM to review results. To get the login information you can use the `readyToLogin` script or the portal. If you don't know how to login to your hosts look at [Appendix A](#).

1. Open two windows, one that you are logged in to the `right-client` and one that you are logged in to the `left-client`.
2. Start a browser and navigate to the *"server's"* hostname; if you don't know how to find the hostname look at [Appendix A](#). For this example experiment the hostname allocated is `pcvm1-15.instageni.stanford.edu`:



3. Now let's generate some traffic to determine how much capacity we can use on each link type:

In the *"right-client"* window issue:

```
right-client$ /usr/bin/iperf -c 192.168.1.2 -t 30 -P 5
```

In the *"left-client"* window issue:

```
left-client$ /usr/bin/iperf -c 10.10.1.2 -t 30 -P 5
```

4. Back in the browser. Refresh the web page and you should see the results for both iperf tests:

Note: The interface capacity for all links was set to **20 Mbits/sec** in the RSpec used for this tutorial.

Iperf Server Log content:

```
-----  
Server listening on TCP port 5001  
TCP window size: 85.3 KByte (default)  
-----  
[ 4] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 57503  
[ 5] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 57501  
[ 6] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 57500  
[ 7] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 57502  
[ 8] local 10.10.1.2 port 5001 connected with 10.10.1.1 port 57504  
[ ID] Interval      Transfer      Bandwidth  
[ 5]  0.0-31.4 sec  17.2 MBytes  4.60 Mbits/sec  
[ 8]  0.0-31.5 sec  15.4 MBytes  4.10 Mbits/sec  
[ 6]  0.0-31.6 sec  12.8 MBytes  3.39 Mbits/sec  
[ 7]  0.0-31.8 sec  12.5 MBytes  3.30 Mbits/sec  
[ 4]  0.0-31.9 sec  11.5 MBytes  3.02 Mbits/sec  
[SUM] 0.0-31.9 sec  72.4 MBytes  19.1 Mbits/sec  
[ 9] local 192.168.1.2 port 5001 connected with 192.168.1.1 port 38079  
[ 4] local 192.168.1.2 port 5001 connected with 192.168.1.1 port 38080  
[ 5] local 192.168.1.2 port 5001 connected with 192.168.1.1 port 38081  
[ 6] local 192.168.1.2 port 5001 connected with 192.168.1.1 port 38078  
[ 7] local 192.168.1.2 port 5001 connected with 192.168.1.1 port 38077  
[ 9]  0.0-31.0 sec  15.2 MBytes  4.13 Mbits/sec  
[ 4]  0.0-31.6 sec  10.0 MBytes  2.65 Mbits/sec  
[ 5]  0.0-33.1 sec  13.2 MBytes  3.36 Mbits/sec  
[ 7]  0.0-33.5 sec  20.5 MBytes  5.13 Mbits/sec  
[ 6]  0.0-33.6 sec  10.6 MBytes  2.65 Mbits/sec  
[SUM] 0.0-33.6 sec  69.6 MBytes  17.4 Mbits/sec
```

Local Link

Stitched Link

5. Time permitting! Let's exchange some layer 2 traffic by using PingPlus.

Add a third terminal window which is logged into the "server" host. Determine the 2 Data Plane interfaces and remove the IP of the two interfaces. Take note of the MAC address for the 192.168.1.2 and 10.10.1.2 interfaces and start the ping listener.

```
server$ sudo /sbin/ifconfig ethXXX 0  
server$ sudo /sbin/ifconfig ethYYY 0  
server$ sudo /local/stitch-tutorial/pingPlus_v3/pingPlusListener 10000
```

Note: A ping listener process was started for you by the installation script.

In the "right-client" window remove the IP from the dataplane interface (ethXXX) and run layer2 ping to "server's" MAC address (XX:XX:XX:XX:XX:XX) for the 192.168.1.0 network via the ethXXX interface:

```
right-client$ sudo /sbin/ifconfig ethXXX 0  
right-client$ sudo /local/stitch-tutorial/scripts/runPingPlus.sh \  
XX:XX:XX:XX:XX:XX ethXXX
```

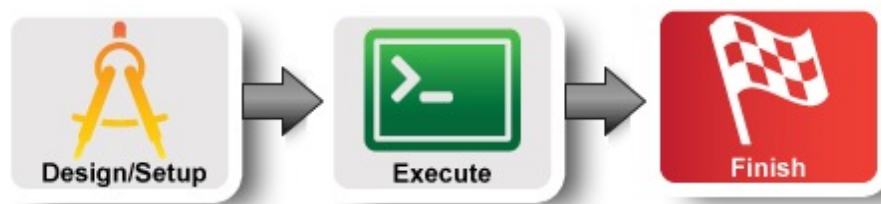
In the "left-client" window remove the IP from the dataplane interface (ethYYY) and run layer2 ping to "server's" MAC address (YY:YY:YY:YY:YY:YY) for the 10.10.1.0 network via the ethYYY interface:

```
left-client$ sudo /sbin/ifconfig ethYYY 0  
left-client$ sudo /local/stitch-tutorial/scripts/runPingPlus.sh \  
YY:YY:YY:YY:YY:YY ethYYY
```

The experiment is now completed; time to release resources!

Introduction

GENI Inter-Aggregate Stitching Tutorial



Step 5. Tear down Experiment

After you are done with this experiment, release your resources using the stitcher tool.

```
$ stitcher deletesliver MySlice
14:22:19 INFO      stitcher: Loading agg_nick_cache file '/Users/lnevers/.gcf/
14:22:19 INFO      stitcher: Loading config file /Users/lnevers/.gcf/omni_conf
14:22:19 INFO      stitcher: Setting option useSliceMembers based on omni_conf
14:22:19 INFO      stitcher: Using control framework portal
14:22:19 INFO      stitcher: Member Authority is https://ch.geni.net/MA (from
14:22:19 INFO      stitcher: Slice Authority is https://ch.geni.net/SA (from c
14:22:19 INFO      stitcher: Reading stitching slice MySlice aggregates from f
14:22:19 INFO      stitcher: Adding aggregate option https://www.instageni.wis
14:22:19 INFO      stitcher: Adding aggregate option https://www.instageni.sta
14:22:19 INFO      stitcher: Adding aggregate option http://geni-am.net.intern
14:22:19 INFO      omni: Loading agg_nick_cache file '/Users/lnevers/.gcf/agg_
14:22:19 INFO      omni: Loading config file /Users/lnevers/.gcf/omni_config
14:22:19 INFO      omni: Using control framework portal
14:22:19 INFO      omni: Member Authority is https://ch.geni.net/MA (from confi
14:22:19 INFO      omni: Slice Authority is https://ch.geni.net/SA (from confi
14:22:20 INFO      omni: Slice urn:publicid:IDN+ch.geni.net:ln-prj+slice+MySli
14:22:25 INFO      omni: Deleted sliver urn:publicid:IDN+ch.geni.net:ln-prj+sl
14:23:21 INFO      omni: (PG log url - look here for details on any failures:
14:23:22 INFO      omni: Deleted sliver urn:publicid:IDN+ch.geni.net:ln-prj+sl
14:24:51 INFO      omni: (PG log url - look here for details on any failures:
14:24:52 INFO      omni: Deleted sliver urn:publicid:IDN+ch.geni.net:ln-prj+sl
Deleted slivers on 3 out of a possible 3 aggregates
```

Tip: The stitcher creates a text file to track which aggregates are part of a stitched slice in the `~/gcf` directory when `createsliver` is run. The file is usually named "ch-geni-net-ProjXXX-SliceYYY-amlist.txt", (ProjXXX=your project, SliceYYY=your slicename). If you invoke the `deletesliver` command in the same directory where `createsliver` was invoked, then stitcher will delete the slivers only at aggregates in the AM list file. If you invoke `deletesliver` from a different location then stitcher will try to delete the slivers at all known aggregates, which takes a long time.

Now you can start designing and running your own experiments!!!

Appendix

Introduction

GENI Inter-Aggregate Stitching Tutorial

Appendix A

If you have not downloaded your keys from the portal then you should download them add them to the SSH agent. You can download the SSH keys from the profile tab in the GENI portal.

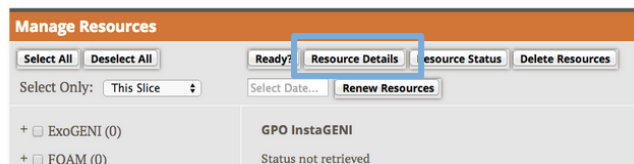
[<http://groups.geni.net/geni/wiki/GENIExperimenter/Tutorials/StitchingTutorial/AppendixA> ...
Appendix A
SSH to hosts and hostname info
GENI Portal
readyToLogin script

SSH to hosts and hostname info

There are multiple ways that you can get ssh and hostname information for your slice. In this appendix we are going to describe two ways: through the GENI portal and the readyToLogin script. ✓ If you are on a Windows computer use the GENI Portal method. For Mac OS and linux systems both methods are good.

GENI Portal

After you login to the GENI Portal find your slice and load it. After you do that press on the



Resource Details button:

This will load a new page with the information for all the hosts in the topology (scroll below the diagram for the second aggregate). The page will look like that:

Node #2				
Client ID	Component ID	Exclusive	Type	Hostname
server	pc1	not exclusive	emulab-openvz	server.xuan-tutorial.ch-geni-net.:
Login	SSH info/link	ssh xuanliu@pcvm1-20.instageni.gpolab.bbn.com		hostname
Interfaces		MAC	Layer 3	
server:if0	pc1:lo0	02e2bf9c6d3e	ipv4:	10.10.1.2
server:if1	pc1:eth1	02223b639b34	ipv4:	192.168.1.2

You can click on the SSH link to load a terminal. This will work on Mac OS and on Firefox browser with the FireSSH installed (independent of the operating system). In Linux you can copy and paste the SSH command to a terminal.

readyToLogin script

readyToLogin script is part of the Omni/gcf package. Open a terminal and run:

```
readyToLogin MySlice --useSliceAggregates
```

The end of the output should look like:

```
=====  
LOGIN INFO for AM: https://boss.instageni.gpolab.bbn.com:12369/protogeni/xmlr  
=====
```

```
For more login info, see the section entitled:
  'Providing a private key to ssh' in 'readyToLogin.py -h'

left-client's geni_status is: ready (am_status:ready)
User xuanliu logs in to left-client using:
  ssh -p 34106 -i /Users/xliu/.ssh/id_rsa xuanliu@pcl.instageni.gpolab.bbn.com
  ssh -p 34106 -i /Users/xliu/.ssh/geni_cert_portal_key xuanliu@pcl.instageni.gpolab.bbn.com

server's geni_status is: ready (am_status:ready)
User xuanliu logs in to server using:
  ssh -i /Users/xliu/.ssh/id_rsa xuanliu@pcvml-20.instageni.gpolab.bbn.com
  ssh -i /Users/xliu/.ssh/geni_cert_portal_key xuanliu@pcvml-20.instageni.gpolab.bbn.com

NOTE: If your user is not listed, try using the --no-keys option.
```

You can copy and paste the ssh command to a terminal (for Linux or Max OS) or you can use the information from the command on any ssh client.

Attachments

- [details.png](#) (56.1 KB) - added by [nriga@bbn.com](#) 4 days ago.
- [details-info.png](#) (80.2 KB) - added by [nriga@bbn.com](#) 4 days ago.