

**Technical Report for the project of  
“Integrating a CRON (Cyberinfrastructure of Reconfigurable Optical Network)  
Testbed into ProtoGENI” (GENI Proposal 1794)**

Period: January 2012 – March 2012

PI: Seung-Jong Park  
Computer Science and Center for Computation & Technology  
Louisiana State University  
289 Coates Hall  
Baton Rouge, LA 70803  
[sjpark@csc.lsu.edu](mailto:sjpark@csc.lsu.edu)

Co-PI: Rajgopal Kannan  
Computer Science  
Louisiana State University  
289 Coates Hall  
Baton Rouge, LA 70803  
[rkannan@csc.lsu.edu](mailto:rkannan@csc.lsu.edu)

## I. Major Accomplishments

### Scope

This effort provides a reconfigurable optical network emulator aggregate connected to the GENI backbone over Louisiana Optical Network Initiative (LONI). The role of optical network emulation in GENI is to provide a predictable environment for repeatable experiments, and to perform early tests of network research experimentation prior to acquiring real network resources. The tools and services developed by this project will integrate with the ProtoGENI suite of tools. The aggregate manager and network connections between LONI and GENI for this project will also allow other LONI sites to participate in GENI.

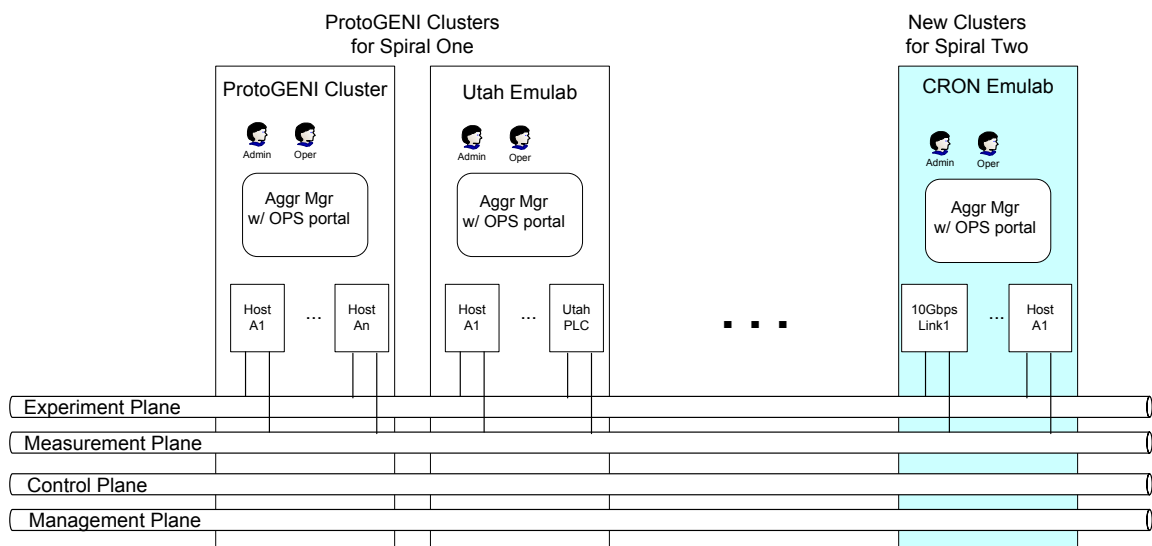


Figure 1. Implementation of Federation of CRON into GENI at the Spiral 4

### A. Milestones Achieved Between January 2012- March 2012

Note: (6mo) means activity that completes by the 6<sup>th</sup> month after start of project. Also, (7-12mo) means activity that starts in the 7<sup>th</sup> month and continues through the 12<sup>th</sup> month.

#### **Milestone 1: CRON Tutorial and Stitching Due 17/03/2012**

#### **Status: Accomplished**

At GEC 13, PIs have demonstrated tree-mode VLAN stitching using the GENI API between MAX and CRON using the Internet2 ION service.

## B. Deliverable Made

Tutorials and Demonstration videos have been uploaded to CRON website.  
(<http://www.cron.loni.org>)

# II. Description of work performed during last quarter

## A. Activities and Findings

### 1) Integrating Hardware Emulator into CRON to emulate impairment parameters

The 10 Gbps hardware network emulator or XGEM is a precision test instrument for 10Gbps Ethernet emulation testing. It can be configured for both uni-directional and bi-directional communication tests, accurately and precisely emulates signal delay and impairments that occur during the transmission of 10 Gigabit Ethernet signals.

User requests for the emulator by specifying delay value starting with 'H' in the input ns file to the CRON test bed. The parser understands this request and updates the database hardware emulator field in the virt\_lans table. When the assign function allocates the resources and if the this field is set then hardware emulator is used for the link emulation. Once the emulator is allocated to the experiment, users can login to the XGEM box via web interface to check about the parameter settings and modify the values if need to change the values.

Virtual topology indicating boss communicating with Emulator to set the emulation parameters. Boss uses TCL interface to set the parameters during experiment initialization

Below is the sample script to request multiple hardware emulators.

```
set ns [new Simulator]
source tb_compat.tcl
set sender1 [$ns node]
set sender2 [$ns node]
set receiver [$ns node]
set router [$ns node]
set link1 [$ns duplex-link $router $sender1 10000Mb H50ms DropTail]
set link2 [$ns duplex-link $router $sender2 10000Mb H45ms DropTail]
set link3 [$ns duplex-link $router $receiver 10000Mb 30ms DropTail]
#tb-set-node-os $sender1 U10-64-SYNC+
#tb-set-node-os $sender2 U10-64-SYNC+
#tb-set-node-os $receiver U10-64-SYNC+
```

```

#tb-set-node-os $router U10-64-100G-ROUTER
#tb-fix-node $sender1 pc5
#tb-fix-node $sender2 pc6
#tb-fix-node $receiver pc1
#tb-fix-node $router pc9
$ns rtproto Static $ns run

```

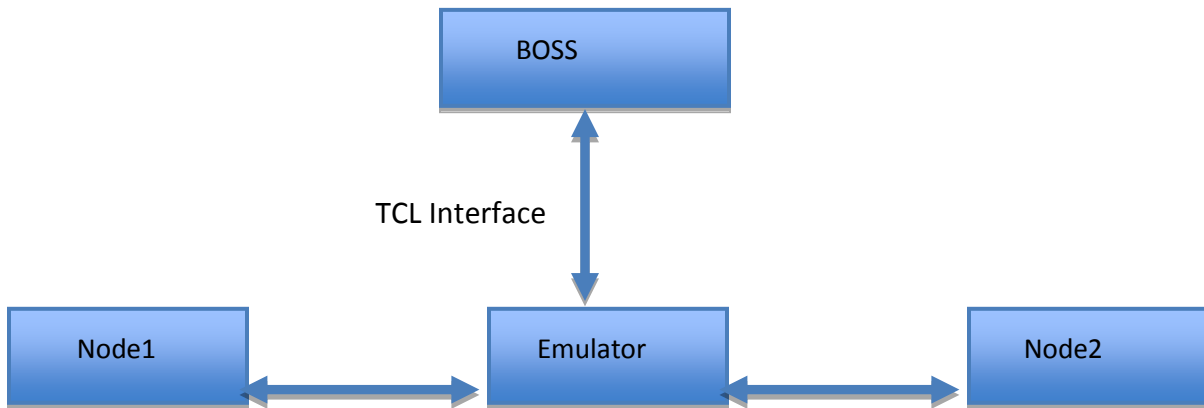


Figure 2. Configuration among emulators and other nodes at CRON

## 2) Source Code Changes

1. **Parse.tcl** – Parse the NS script to read for the delay parameter. If the delay parameter value starts with value **H** **Then** it is interpreted as hardware emulator is requested by the user.

```

set strH [ string first H $dspec]
if { $strH == -1 } {
if {[scan $dspec "%f%s" delay unit] == 1} {
    set unit s
}
}

```

2. **sim.tcl** - sets the link object with hardware emulator

#usage - we created an instance of Link in the above statement just use that.

```
$curlink hardware_emulator $hdelay
```

3. **lanlink.tcl** – Updates the database to set hardware\_emulator parameter for virt\_lans table.

Parse.tcl, sim.tcl and lanlink.tcl together parse the input NS script by the user. Identify if the emulator is requested by the user and accordingly sets the virt\_lans table of the database.

4. **tbswap.tcl** – From this script boss calls emulator to set the various parameters.

```

my $query_result =
DBQueryFatal("select d.delay0,e.IP,e.id from delays as d left join Emulators as e on
  d.node_id=e.id right join Emulators as ee on d.node_id=ee.id".
  " where d.pid='$pid' and d.eid='$eid'");
if ($query_result->numrows) {
while( my ($hdelay,$IP,$nodeID) = $query_result->fetchrow_array())
{
  print "$nodeID and $hdelay and $IP and rows $query_result->numrows" ;
system("/usr/local/bin/tclsh8.4 $tb_set $IP $hdelay")==0 or fatal("Error while executing
  emulator script" );
}

```

5. **anue\_comms\_library.tcl** - Anue XGEM libraries needed to connect to emulator

6. **anue\_tcl\_library.tcl** - Anue XGEM functions to set the parameters

7. **tb\_set\_params.tcl** - This is the script containing the function calls to the emulator. This is being called by boss during the experiment initialization.

## 2) Tree Mode stitching Using OMNI tool to federate CRON with MAX Planetlab Central

This report describes how to connect those two federated aggregates (CRON and MAX) through Internet2 ION service. The following figure shows the setup for networking and computing environment between two federated aggregates. First, an experimenter will get slivers including several machines from CRON at cron.ioni.org (ProtoGeni Framework) and several machines from MAX (PlanetLab Framework) based on the following steps.

A user uses the Omni tool to send *RSpec* to three GENI AMs, including CRON AM, MAX AM, and ION AM as shown in Figure 3.

At first, a GENI *clearinghouse* authenticates experimenters and issues them credentials needed to obtain GENI resources for experimentation. Then GENI *aggregates* provide resources to experimenters with GENI credentials. A GENI *slice* holds a collection of computing and communications resources capable of running an experiment or a wide area service. And *RSpec* is the mechanism for advertising, requesting, and describing the resources used by slice. The Internet2 ION Aggregate Manager does VLAN stitching to connect CRON and MAX as a coherent network. GENI network stitching operation is to construct a topology of substrates as represented by their Aggregate Managers. Each Aggregate Manager has a unique Rspec which defines its Substrate resources.

A Rspec example:

```
<stitchingResourceid="urn:aggregate=geni.maxgigapop.net:rspec=maxpl_1sutestslice1:stitching=3022" type="p2pvlan">
```

We got slivers including several machines from CRON at cron.ioni.org (ProtoGeni) and several machines from MAX (PlanetLab). The steps to use Omni to get slivers at each site are as following:

1. Use Omni to request sliver on MAX AM
  - a. get the resource from MAX
  - b. get the VLAN number 1 on MAX side
2. Use Omni to request sliver on CRON AM
  - a. get the VLAN number 2 on CRON side
3. Use Omni to request sliver on ION AM to use VLAN number 1 and VLAN number 2
4. ION will do VLAN translation
5. CRON to MAX Internet 2 connection done

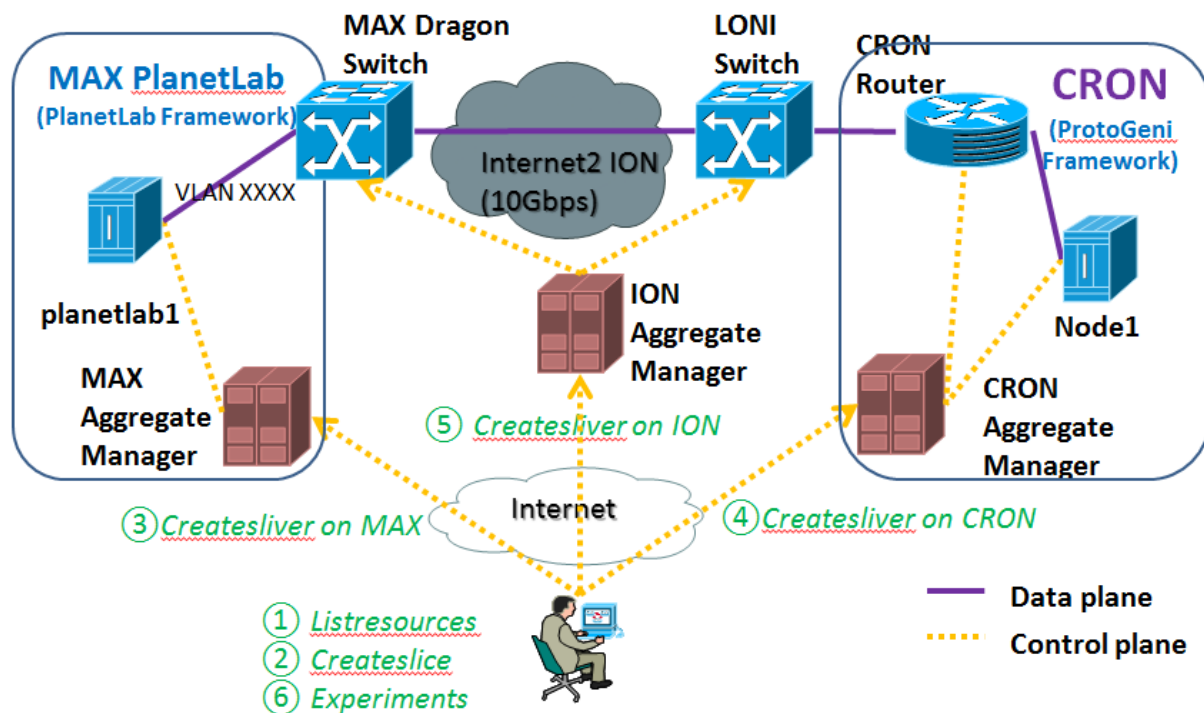


Figure 3. Configuration between CRON and MAX and Flowchart to perform networking stitching using GENI API

OMNI Configuration using CRON credentialed user:

```
[omni]
default_cf = cron
users = linxue
# =====
# Define AM here.
[emulab]
type = pg
ch = https://www.emulab.net:443/protogeni/xmlrpc/ch
sa = https://www.emulab.net:443/protogeni/xmlrpc/sa
cert = ~/.ssl/encrypted_emulab.pem
key = ~/.ssl/encrypted_emulab.pem
[cron]
type = pg
ch = https://www.emulab.net:443/protogeni/xmlrpc/ch
sa = https://www.cron.loni.org:443/protogeni/xmlrpc/sa
cert = ~/.ssl/encrypted_cron.pem
key = ~/.ssl/encrypted_cron.pem
#cert = ~/.ssl/encrypted_emulab.pem
#key = ~/.ssl/encrypted_emulab.pem
[plc]
```

```

type=sfa
authority=maxpl.maxpl
user=maxpl.maxpl.lxue2
cert=~/.gcf/maxpl.maxpl.lxue2.gid
key=~/.work/keys/id_rsa
#key=~/.work/keys/server.key
#key=~/.gcf/maxpl.maxpl.lxue2.pkey
registry=http://max-myplc.dragon.maxgigapop.net:12345
slicemgr=http://max-myplc.dragon.maxgigapop.net:12347
# =====
# Define users here.
[linxue]
urn = urn:publicid:IDN+cron.loni.org+user+linxue
keys = ~/.ssh/id_rsa.pub

```

### Rspec:

Because now GENI uses tree mode experiment stitching, so we need to send three separate Rspec to MAX, CRON and ION. Following are three Rspec examples for these three AMs

#### 1. Rspec for CRON

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Resources at AM: URN: urn:publicid:IDN+cron.loni.org+authority+sa URL:
http://cron.loni.org:443/protogeni/xmlrpc/am -->
<rspec xmlns="http://www.protogeni.net/resources/rspec/2"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.protogeni.net/resources/rspec/2
http://www.protogeni.net/resources/rspec/2/request.xsd" type="request">
<node client_id="geni13" component_id="urn:publicid:IDN+cron.loni.org+node+pc22"
component_manager_id="urn:publicid:IDN+cron.loni.org+authority+cm" exclusive="true">
</node>
</rspec>

```

#### 2. Rspec for MAX

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Resources at AM: URN: urn:publicid:IDN+maxpl+authority+sa URL: http://max-
myplc.dragon.maxgigapop.net:12346 -->
<RSpec type="SFA">
<rspec id="ionstitching1" xmlns="http://geni.maxgigapop.net/aggregate/rspec/20100412/"
xmlns:CtrlPlane="http://ogf.org/schema/network/topology/ctrlPlane/20080828/">
<aggregate>geni.maxgigapop.net</aggregate>
<description>Stitching Demo Request RSPEC</description>
<lifetime id="time-1330876800-1332219600">
<CtrlPlane:start type="CtrlPlane:TimeContent">1330876800</CtrlPlane:start>

```



```

<CtrlPlane:end type="CtrlPlane:TimeContent">1332219600</CtrlPlane:end>
</lifetime>
<user email="lxue2@tigers.lsu.edu"/>
<computeResource id="urn:aggregate=geni.maxgigapop.net:rspec=ionstitching1">
<planetlabNodeSliver
id="urn:aggregate=geni.maxgigapop.net:rspec=ionstitching1:domain=dragon.maxgigapop.net:node=planetlab5">
<address>206.196.176.133</address>
<computeCapacity>
<cpuType>generic</cpuType>
<cpuSpeed>2.0GHz</cpuSpeed>
<numCpuCores>1</numCpuCores>
<memorySize>256MB</memorySize>
<diskSize>16GB</diskSize>
</computeCapacity>
<networkInterface
id="urn:aggregate=geni.maxgigapop.net:rspec=ionstitching1:domain=dragon.maxgigapop.net:node=planetlab5:interface=eth1.any_stitching">
<deviceType>Ethernet</deviceType>
<deviceName>eth1</deviceName>
<capacity>100Mbps</capacity>
<ipAddress>192.168.44.2/24</ipAddress>
<vlanRange>3211</vlanRange>
<attachedLinkUrn>urn:ogf:network:domain=dragon.maxgigapop.net:node=MAX:port=1-0-8:link=*</attachedLinkUrn>
</networkInterface>
</planetlabNodeSliver>
</computeResource>
<stitchingResource id="urn:aggregate=geni.maxgigapop.net:rspec=ionstitching1:stitching=*"
type="p2pvlan">
<networkInterfaceUrn>urn:aggregate=geni.maxgigapop.net:rspec=ionstitching1:domain=dragon.maxgigapop.net:node=planetlab5:interface=eth1.any_stitching</networkInterfaceUrn>
<networkInterfaceUrn>urn:ogf:network:domain=dragon.maxgigapop.net:node=CLPK:port=1-2-3:link=*</networkInterfaceUrn>
<externalResourceId>urn:aggregate=ion.internet2.edu:rspec=legacy_item</externalResourceId>
</stitchingResource>
</rspec>
</RSpec>

```

For MAX Rspec, you need to change corresponding

```
<ipAddress>192.168.44.2/24</ipAddress>
```

```
<vlanRange>3211</vlanRange>
```

You can use vlan ranging from 3211 to 3219.

Since the interfaces are in different VLANs, you need to change the subnet of the interface for different links.

### 3. Rspec for ION

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Resources at AM: URN: urn:publicid:IDN+ionpl+authority+sa URL:
http://alpha.east.isi.edu:12346 -->
<RSpec type="SFA">
<rspec id="ionstitching1" xmlns="http://geni.maxgigapop.net/aggregate/rspec/20100412/"
xmlns:CtrlPlane="http://ogf.org/schema/network/topology/ctrlPlane/20080828/">
<aggregate>ion.internet2.edu</aggregate>
<description>ION stitching demo rspec</description>
<lifetime id="time-1330876800-1332219600">
<CtrlPlane:start type="CtrlPlane:TimeContent">1330876800</CtrlPlane:start>
<CtrlPlane:end type="CtrlPlane:TimeContent">1332219600</CtrlPlane:end>
</lifetime>
<user email="lxue2@tigers.lsu.edu"/>
<stitchingResource id="urn:aggregate=ion.internet2.edu:rspec=ionstitchingslice1:stitching="*
type="p2pvlan">
<networkInterface id="urn:ogf:network:domain=ion.internet2.edu:node=rtr.newy.port=x-
0/0/3:link="*>
<deviceType>Ethernet</deviceType>
<capacity>100Mbps</capacity>
<vlanRange>3211</vlanRange>
<peerNetworkInterface>urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-
3:link="*</peerNetworkInterface>
</networkInterface>
<networkInterface id="urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-3:link="*>
<deviceType>Ethernet</deviceType>
<capacity>100Mbps</capacity>
<vlanRange>3211</vlanRange>
<peerNetworkInterface>urn:ogf:network:domain=ion.internet2.edu:node=rtr.newy.port=x-
0/0/3:link="*</peerNetworkInterface>
</networkInterface>
</stitchingResource>
</rspec>
</RSpec>

```

For ION Rspec, you need to set the same <vlanRange> according to the VLAN you set at MAX site. You can see that this ION circuit is between:

```

urn:ogf:network:domain=ion.internet2.edu:node=rtr.newy.port=x-0/0/3:link=*
urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-3:link=*

```

Which creates an ION I2 circuit all the way from LONI switch to the point where MAX site connects to.

Behind the scene, OSCAR log example shows:

Circuit set up succeeded. GRI: dragon.maxgigapop.net-2048 description: maxpl\_lsu\_demo\_test stitching-resource: p2pvlan-urn:ogf:network:domain=dragon.m login: internet2 status: ACTIVE start time: 10/27/11 9:01 PM UTC end time: 11/9/11 3:55 PM UTC bandwidth: 100000000 path setup mode: timer-automatic layer: 2 source:

urn:ogf:network:domain=dragon.maxgigapop.net:node=MAX:port=1-0-9:link=\* destination: urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-3:link=\* source VLAN tag: 3022 source tagged: true destination VLAN tag: 3022 destination tagged: true intradomain hops: urn:ogf:network:domain=loni.org:node=loni-dcn-e1:port=0-1-4:link=\*, VLAN=3022 urn:ogf:network:domain=loni.org:node=loni-dcn-e1:port=0-2-4:link=10.100.5.2, VLAN=3022 urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-4:link=10.100.5.1, VLAN=3022 urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-3:link=\*, VLAN=3022 interdomain hops: urn:ogf:network:domain=dragon.maxgigapop.net:node=MAX:port=1-0-9:link=\*, VLAN=3022 urn:ogf:network:domain=dragon.maxgigapop.net:node=MAX:port=1-2-1:link=140.173.97.93 urn:ogf:network:domain=dragon.maxgigapop.net:node=CLPK:port=1-2-2:link=140.173.97.94 urn:ogf:network:domain=dragon.maxgigapop.net:node=CLPK:port=1-2-3:link=\*, VLAN=3022 urn:ogf:network:domain=ion.internet2.edu:node=rtr.newy:port=x-0/0/3:link=\*, VLAN=3022 urn:ogf:network:domain=ion.internet2.edu:node=rtr.hous:port=x-0/3/0:link=\*, VLAN=3022 urn:ogf:network:domain=loni.org:node=loni-dcn-e1:port=0-1-4:link=\*, VLAN=3022 urn:ogf:network:domain=loni.org:node=lsu-dcn:port=0-2-3:link=\*, VLAN=3022

### Set VLAN on CRON datasw port

CRON data switch only has one link to LONI switch which connect to Internet2, and that link is already set to trunk link and allow any VLAN. So all you need to set is the datasw port which connects to your experiment machines. You need to manually set that port into the vlan which you set both in MAX and ION Rspec. You can do that before step three so that whenever you have all the Rspec sent you can have that active link.

### Send OMNI command

Here are the OMNI command examples

1. get version

```
./omni.py -c ../omni_config -a https://www.cron.loni.org/protogeni/xmlrpc/am getversion
```

2. list resources

```
./omni.py -c ../omni_config -a https://www.cron.loni.org/protogeni/xmlrpc/am -o -t ProtoGENI 2 listresources
```

3. create slice

```
./omni.py -c ../omni_config -a https://www.cron.loni.org/protogeni/xmlrpc/am createslice crontest
```

4. renew slice

```
./omni.py -c ../omni_config -a https://www.cron.loni.org/protogeni/xmlrpc/am renewslice  
crontest 2012-03-20
```

5. create sliver on MAX

```
./omni.py -c ../omni_config -a http://max-myplc.dragon.maxgigapop.net:12346 -o createsliver --  
api-version 2 crontest ../max.rspec
```

6. sliver status on MAX

```
./omni.py -a c -c ../omni_config -a http://max-myplc.dragon.maxgigapop.net:12346 -o  
sliverstatus -t geni 3 --api-version 2 crontest
```

7. log on to MAX machines

```
sudo ssh -i ~/.ssh/id_rsa cronloniorg_crontest@planetlab5.dragon.maxgigapop.net
```

8. delete sliver on MAX

```
./omni.py -a http://max-myplc.dragon.maxgigapop.net:12346 -c ../omni_config deletesliver -t  
geni 3 --api-version 2 crontest
```

9. create sliver on ION

```
./omni.py -c ../omni_config -a http://alpha.east.isi.edu:12346 -o createsliver --api-version 2  
crontest ../ion.rspec
```

10. sliver status on ION

```
./omni.py -a c -c ../omni_config -a http://alpha.east.isi.edu:12346 -o sliverstatus -t geni 3 --api-  
version 2 crontest
```

11. delete sliver on ION

```
./omni.py -a http://alpha.east.isi.edu:12346 -c ../omni_config deletesliver -t geni 3 --api-version 2  
crontest
```

12. create sliver on CRON

```
./omni.py -c ../omni_config -a https://www.cron.loni.org/protogeni/xmlrpc/am createsliver  
crontest ../cron2.rspec
```

13. sliver status on CRON

```
./omni.py -a https://www.cron.loni.org/protogeni/xmlrpc/am -c ../omni_config -o sliverstatus -t  
geni 3 --api-version 2 crontest
```

14. delete sliver on CRON

```
./omni.py -a https://www.cron.loni.org/protogeni/xmlrpc/am -c ../omni_config deletesliver  
crontest
```

15. stitching

```
python ./sample.py --real -c ../omni_config ionstitching1 cron2.rspec ion.rspec max.rspec
```

16. set interface up in CRON

```
ifconfig eth4 192.168.45.3 netmask 255.255.255.0 up
```

Start to do experiments after all these setups!

## **B. Project Participants**

PI: Seung-Jong Park

co-PI: Rajgopal Kannan

Graduate Students: Cheng Cui, Lin Xue, Chui-Hui

## **C. Publications**

N/A

## **D. Outreach Activity**

N/A

## **E. Collaborations**

ProtoGENI Project Group at University of Utah

OnTimeMeasure Project Group at Ohio State University

GMOC Project Group at Indiana University

MAX project at PlanetLab Control Framework

Openflow Switch Project Group at Stanford University

## **F. Other Contributions**