

GENI

Plastic Slices project report-out

Josh Smift, GPO
Denver, Colorado July 26, 2011
www.geni.net

- Spiral 3 lays the foundation for GENI production operations
 - Common control software at mesoscale aggregates
 - Nationwide managed GENI data plane (ethernet VLANs), control plane (IP) and GENI resources (campuses, backbones, and regionals)
 - Operations support from campuses, GMOC, and GPO
 - Beginnings of GENI agreements and procedures
- Most things are still under construction
- Brave experimenters are using the GENI mesoscale environment now
- How would we do with not-so-brave experimenters or even plain old application users with no GENI knowledge?
- Try continuous simplistic (but representative) “plastic” experiments and see how GENI infrastructure, people, and procedures fare
- Provide input/information for future community work

- Run ten GENI slices continuously for months
- Gain experience managing and operating production-quality mesoscale GENI resources
 - Campuses managing local resources
 - GMOC performing meta-operations activities
 - Experimenters running experiments (GPO filling in for this role)
- Discover and record issues that early experimenters are likely to encounter
 - Software (both user tools and aggregates)
 - Isolation from other experiments
 - Ease of use
 - Availability
- All documented on the GENI wiki, and reproducible

- Engineered VLANs at campuses, regionals, & backbones
- Core OpenFlow resources at Internet2 and NLR
- MyPLC and OF resources at eight campuses
- Monitoring data collection and OF support at GMOC
- Ten GENI slices, at different subsets of the campuses
- Five artificial experiments (two slices each)
- Eight baselines, with representative traffic flows
- Resources allocated with Omni via GENI AM API
- Simplistic experimenter tools for managing slices
- Draft operations procedures, mailing lists, chatrooms

- Question: Is mesoscale GENI ready for operations with more experimenters?
- Answer: Yes, *but*
 - Resource operators need to communicate more
 - With each other about plans and other coordination
 - With experimenters about outages
 - Identifying relationships between pieces (resources, slivers, slices, users) is still hard
 - We had workarounds for Plastic Slices (naming conventions)
 - These won't scale well

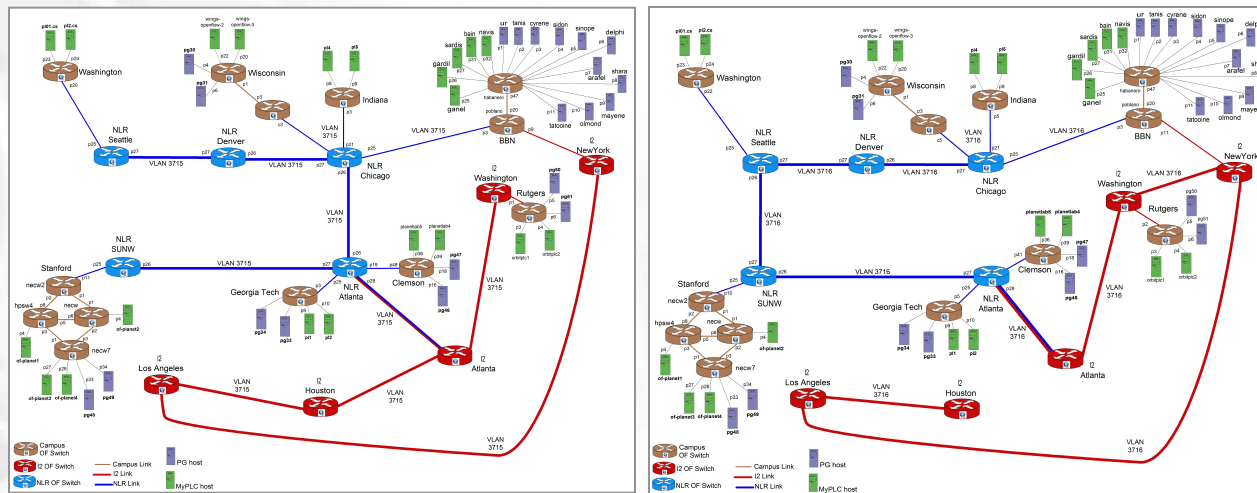
- Question: Is mesoscale GENI ready for operations with more experimenters?
- Answer: Yes, *but*
 - Uptime needs improvement
 - Aggregate managers are sometimes down
 - Software is still buggy (but developers are very responsive)
 - Software revision/release management needs improvement
- Ideas for improvement:
 - Build agreements to set and measure targets for uptime
 - Give feedback/input to software developers on features and priorities
 - Recruit more real (and brave) experimenters

- Question: Is mesoscale GENI software ready to use in a more production environment?
- Answer: Yes, *but*
 - Much of the software we rely on is still new
 - GENI may be the first large-scale test for some things
 - On the plus side, problems are generally fixed quickly
- Ideas for improvement:
 - GENI racks, making production environments more similar
 - InCNTRE (SDN initiative at Indiana), which will emphasize interoperability & commercial use of OpenFlow
 - GENI slices/resources dedicated to testing software
 - More professional software engineers

- Question: Are mesoscale GENI experiments isolated from each other?
- Answer: Only somewhat
 - MyPLC plnodes are VMs on a shared server
 - FlowVisor flowspace is shared with all users
 - Topology problems can cause outages or leak traffic
 - All bandwidth is shared – no dedicated reservations
- Ideas for improvement:
 - This is already an active area of work within GENI
 - Develop better procedures to handle communication (between ops folks and with experimenters) when there are issues
 - More information-sharing – recommendations, tips & tricks, etc
 - QoS in OpenFlow protocol and backbone hardware

- Question: Is mesoscale GENI easy for experimenters to use?
- Answer: It depends
 - Doing simple things is easy (low barriers to entry)
 - Experimenter tools are just now interoperating with GENI
 - OpenFlow opt-in requires manual intervention from multiple people
- Ideas for improvement:
 - This is another area where work is already active within GENI
 - Most of the Experimenter track at this GEC focuses on tools
 - Experimenter demand is starting to drive this
 - GENI slices/resources dedicated to testing experimenter tools
 - Stitching can help with opt-in

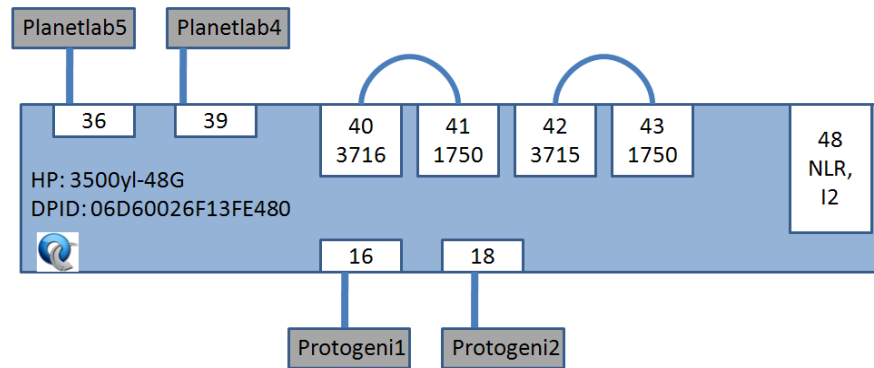
- The GENI network core, in Internet2 and NLR
 - Two VLANs on ten OpenFlow switches
 - Two Expedient OpenFlow aggregates managing them
 - A different approach to VLANs from GEC 9
 - The underlying VLANs are engineered manually
 - OpenFlow allows multiple experiments to slice and share them



(Maps of the topology of the two current OpenFlow network core VLANs, 3715 and 3716.)

<http://groups.geni.net/geni/wiki/NetworkCore>

- Compute and network resources at campuses
 - Private VLAN connected to the backbone VLANs
 - An Expedient OpenFlow aggregate managing it
 - A MyPLC aggregate with two (or more) plnodes
 - Wide-Area ProtoGENI hosts (controlled by Utah Emulab)
 - Campuses: BBN, Clemson, Georgia Tech, Indiana, Rutgers, Stanford, Washington, and Wisconsin



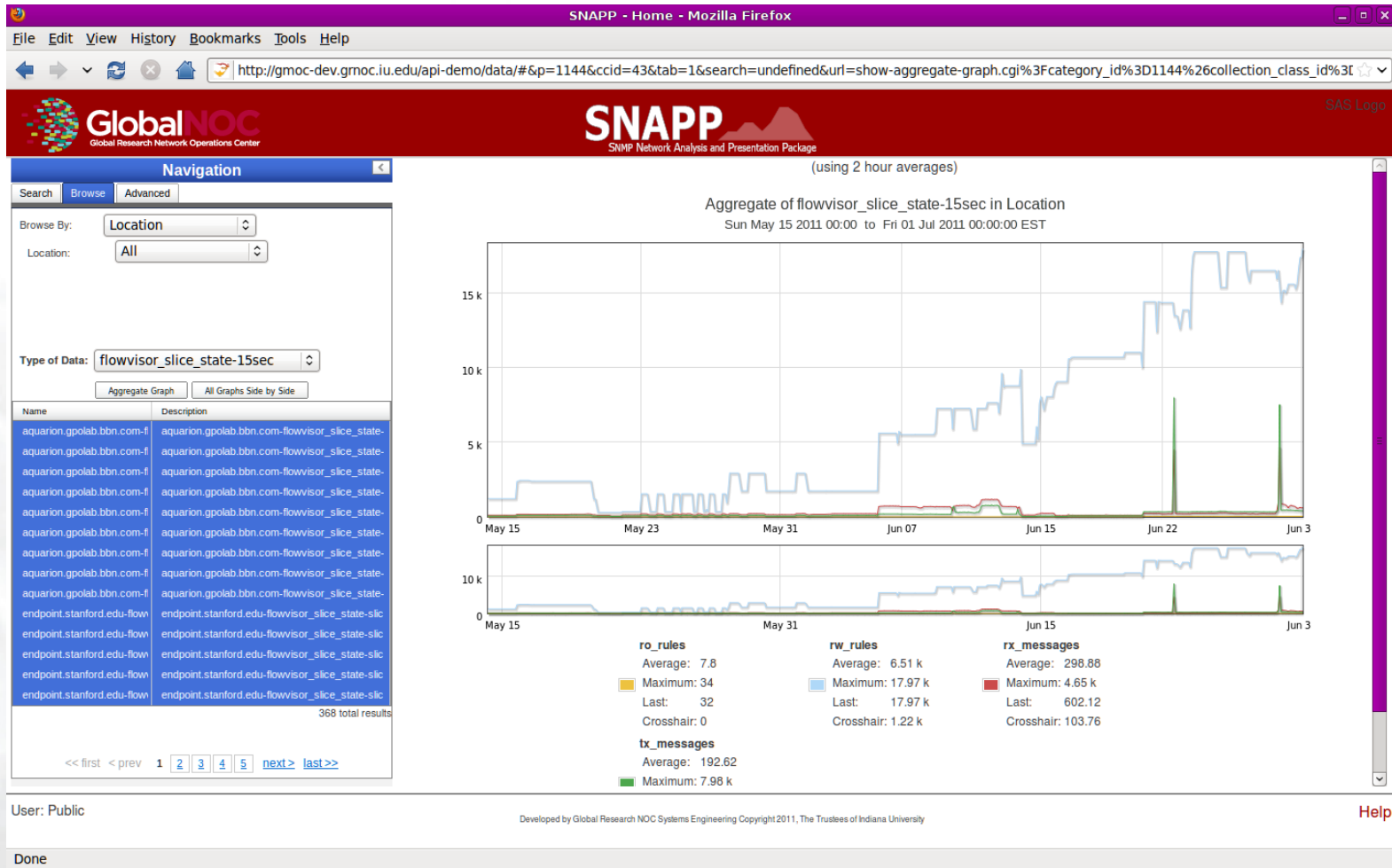
(Clemson's OpenFlow switch diagram. Thanks, Clemson! Other campuses are structurally similar.)

<http://groups.geni.net/geni/wiki/TangoGENI#ParticipatingAggregates>

- All mesoscale campus send data to GMOC
 - NTP is essential for correlating data between sites
 - GMOC has an interface for browsing (SNAPP)
 - Anyone can download/analyze the raw data
 - BBN downloads data and publishes graphs
- Data for both ops and experimenters
 - Per-aggregate, per-host, per-NIC, etc
 - Also some per-slice info
 - Not fully granular, e.g. not per-slice-per-NIC
- More in-slice monitoring is an active area of development

<http://groups.geni.net/geni/wiki/PlasticSlices/MonitoringRecommendations>

Monitoring example - SNAPP



The screenshot shows the SNAPP (SNMP Network Analysis and Presentation Package) interface. On the left is a navigation sidebar with search and browse options. The main area displays an aggregate graph for 'flowvisor_slice_state-15sec' in 'Location' from May 15, 2011, to June 3, 2011. The graph shows two data series: a blue line representing 'ro_rules' and a red line representing 'rx_messages'. The blue line shows a significant increase starting in late May, peaking at approximately 17.97k in early June. The red line shows a smaller peak of about 4.65k around the same time. Summary statistics for these series are provided below the graphs.

Series	Average	Maximum	Last	Crosshair
ro_rules	7.8	34	32	0
rx_messages	298.88	4.65 k	602.12	103.76
tx_messages	192.62	7.98 k		

(GMOC's SNAPP interface, showing the total number of flowspace rules in all mesoscale FlowVisors.)

<http://gmoc-db.grnoc.iu.edu/api-demo/>

- Ten slices, plastic-101 through plastic-110
 - A sliver on MyPLC plnodes at each campus
 - An OpenFlow sliver controlling an IP subnet (10.42.X.0/24)
 - A simple OpenFlow controller (NOX ‘switch’)
- Odd-numbered on VLAN 3715, evens on 3716
- Various subsets of the eight campuses:
 - Two with all sites
 - Two at the VLAN endpoints
 - Two including campuses who share a FrameNet switch
 - Two with five sites
 - Two with six sites

<http://groups.geni.net/geni/wiki/PlasticSlices/SliceStatus>

Plastic Slices: Slivers per Slice - Mozilla Firefox

http://monitor.gpolab.bbn.com/plastic-slices/slivers-per-slice.html

Plastic Slices: Slivers per Slice

Last update: 2011-07-22 11:24:59.060597

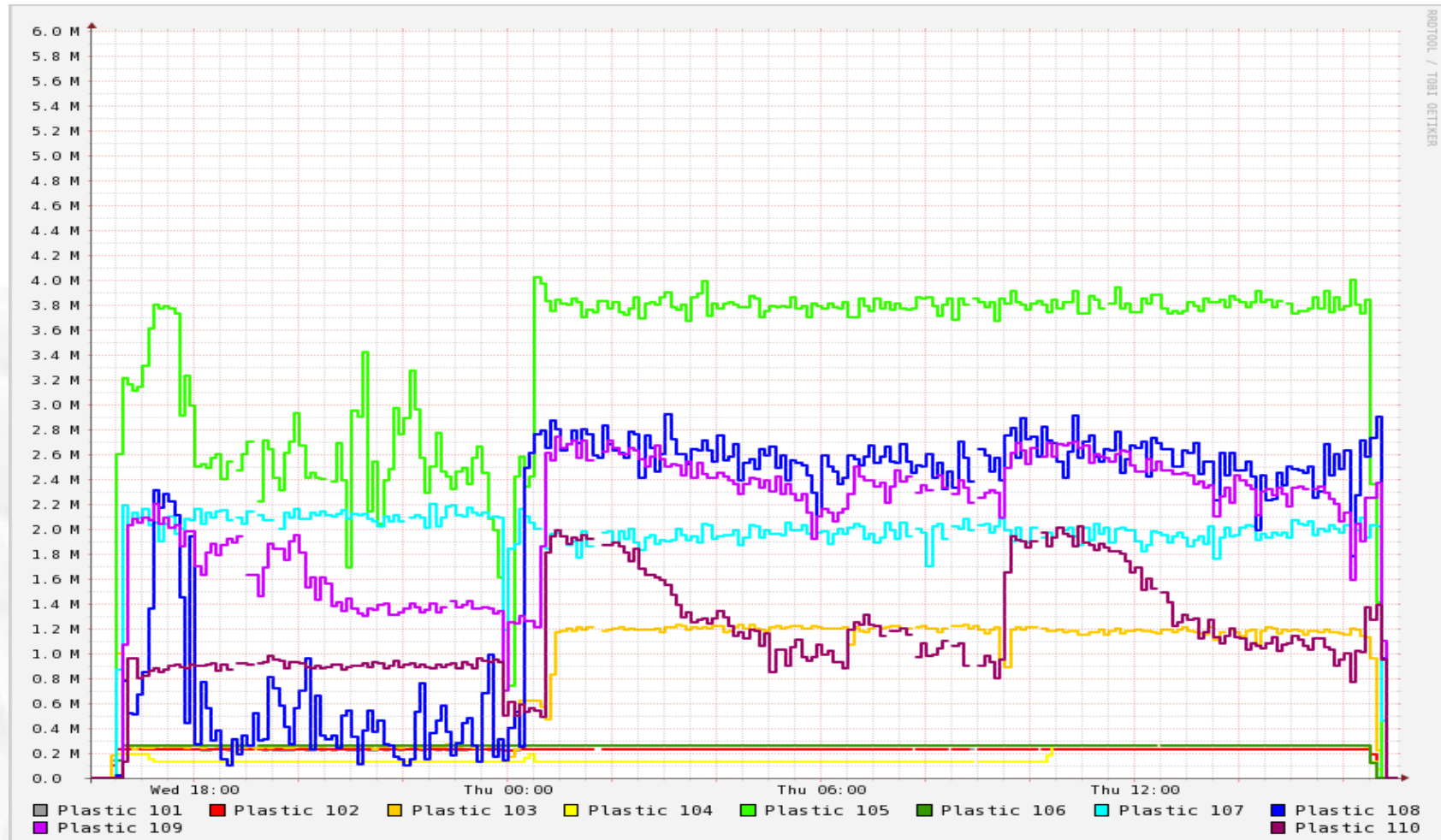
Slice	OpenFlow slivers in slice	MyPLC slivers in slice
plastic101	naxos-33101 ID endpoint stanford edu 75@endpoint.stanford.edu	pgenigpolabbbncom_plastic101@hegen.gpolab.bbn.com
	naxos-33101 ID expedient clemson edu 8@expedient.clemson.edu	pgenigpolabbbncom_plastic101@myplc.cip.gatech.edu
	naxos-33101 ID expedient oflow cip gatech edu 2@flowvisor.oflow.cip.gatech.edu	pgenigpolabbbncom_plastic101@myplc.clemson.edu
	naxos-33101 ID nox_orbit-lab org 6@nox.orbit-lab.org	pgenigpolabbbncom_plastic101@myplc.grnoc.iu.edu
	naxos-33101 ID tulum_gpolab_bbn_com 60@tulum.gpolab.bbn.com	pgenigpolabbbncom_plastic101@myplc.stanford.edu
	naxos-33101 ID wings-openflow-1 wail_wisc.edu 7@wings-openflow-1.wail.wisc.edu	pgenigpolabbbncom_plastic101@plc.orbit-lab.org
plastic102	naxos-33102 ID endpoint stanford edu 76@endpoint.stanford.edu	pgenigpolabbbncom_plastic102@hegen.gpolab.bbn.com
	naxos-33102 ID expedient clemson edu 9@expedient.clemson.edu	pgenigpolabbbncom_plastic102@myplc.cip.gatech.edu
	naxos-33102 ID expedient oflow cip gatech edu 5@flowvisor.oflow.cip.gatech.edu	pgenigpolabbbncom_plastic102@myplc.clemson.edu
	naxos-33102 ID nox_orbit-lab org 8@nox.orbit-lab.org	pgenigpolabbbncom_plastic102@myplc.grnoc.iu.edu
	naxos-33102 ID tulum_gpolab_bbn_com 58@tulum.gpolab.bbn.com	pgenigpolabbbncom_plastic102@myplc.stanford.edu
	naxos-33102 ID wings-openflow-1 wail_wisc.edu 5@wings-openflow-1.wail.wisc.edu	pgenigpolabbbncom_plastic102@plc.orbit-lab.org
plastic103	naxos-33103 ID endpoint stanford edu 77@endpoint.stanford.edu	pgenigpolabbbncom_plastic103@hegen.gpolab.bbn.com
	naxos-33103 ID nox_orbit-lab org 7@nox.orbit-lab.org	pgenigpolabbbncom_plastic103@myplc.stanford.edu
	naxos-33103 ID tulum_gpolab_bbn_com 84@tulum.gpolab.bbn.com	pgenigpolabbbncom_plastic103@plc.orbit-lab.org
plastic104	naxos-33104 ID expedient clemson edu 10@expedient.clemson.edu	pgenigpolabbbncom_plastic104@hegen.gpolab.bbn.com
	naxos-33104 ID nox_orbit-lab org 9@nox.orbit-lab.org	pgenigpolabbbncom_plastic104@myplc.clemson.edu
	naxos-33104 ID tulum_gpolab_bbn_com 86@tulum.gpolab.bbn.com	pgenigpolabbbncom_plastic104@myplc.grnoc.iu.edu
	naxos-33104 ID wings-openflow-1 wail_wisc.edu 12@wings-openflow-1.wail.wisc.edu	pgenigpolabbbncom_plastic104@plc.orbit-lab.org
plastic105	naxos-33105 ID expedient clemson edu 11@expedient.clemson.edu	pgenigpolabbbncom_plastic105@hegen.gpolab.bbn.com
	naxos-33105 ID expedient oflow cip gatech edu 6@flowvisor.oflow.cip.gatech.edu	pgenigpolabbbncom_plastic105@myplc.cip.gatech.edu

(A monitoring page at BBN showing the slivers in each slice.)

<http://monitor.gpolab.bbn.com/plastic-slices/slivers-per-slice.html>

- Five experiments, with different types of traffic
 - ping: ICMP (1500 byte packets at different rates)
 - netcat: Unencrypted TCP
 - wget (HTTPS): Encrypted TCP
 - iperf TCP: TCP, with performance stats
 - iperf UDP: UDP, with performance stats
- Simple and widely available, with some variation
- Similar to traffic sent by real mesoscale GENI experiments
- Not intended to measure performance

<http://groups.geni.net/geni/wiki/PlasticSlices/Experiments>

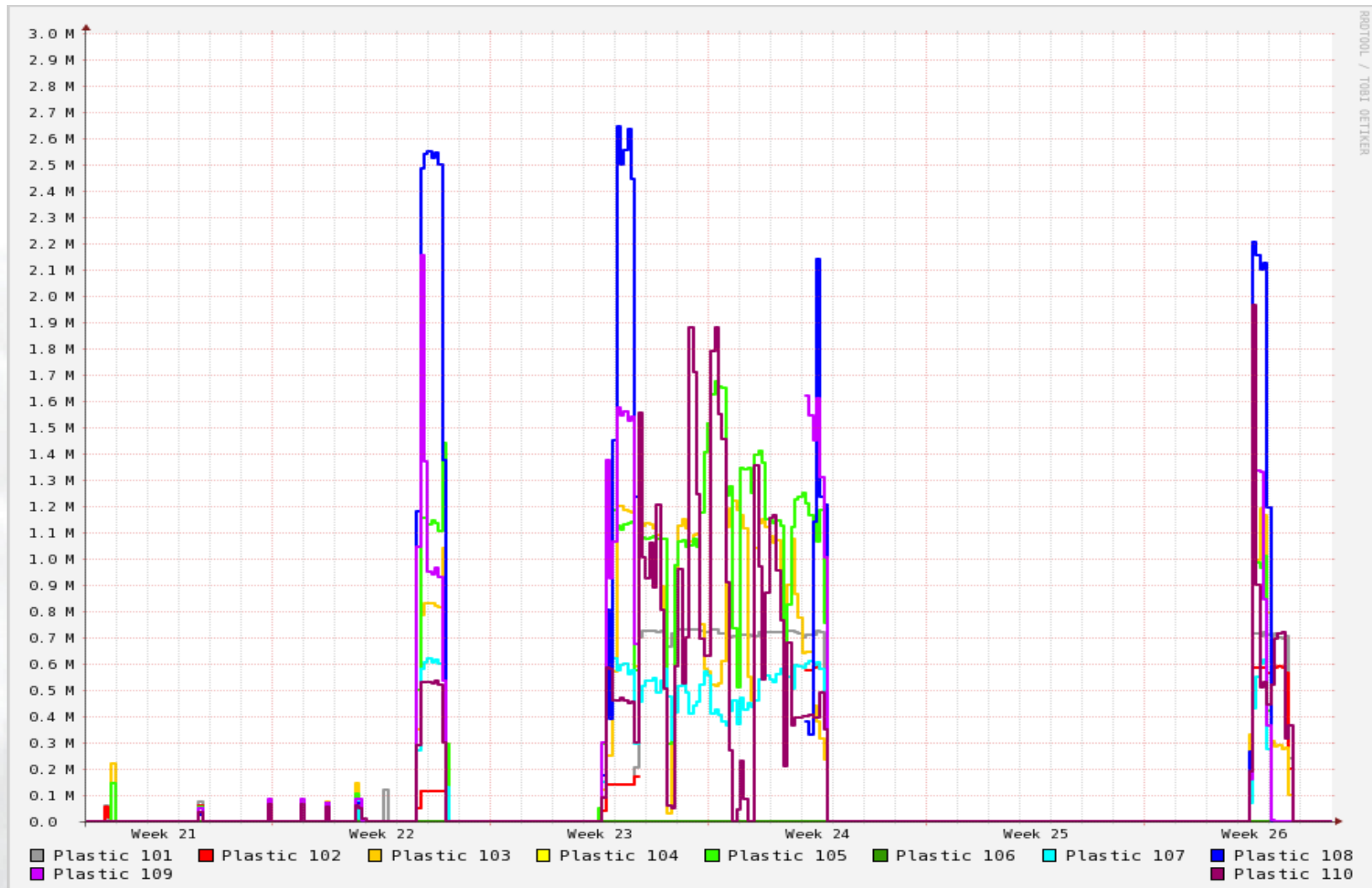


(Traffic overview graphs from Baseline 5; each different colored line is a different slice.)

<http://groups.geni.net/geni/wiki/PlasticSlices/BaselineEvaluation/Baseline5Traffic>

- Confirm basic functionality and stability
 - Baseline 1: At least 1 GB per day, for 24 hours
 - Baseline 2: At least 1 GB per day, for 72 hours
 - Baseline 3: At least 1 GB per day, for 144 hours
- Send continuous traffic
 - Baseline 4: At least 1 Mb/s for 24 hours
 - Baseline 5: At least 10 Mb/s for 24 hours
 - Baseline 6: At least 10 Mb/s for 144 hours
- Exercise procedures at larger scale
 - Baseline 7: Perform an Emergency Stop test
 - Baseline 8 : Create many slices very quickly

<http://groups.geni.net/geni/wiki/PlasticSlices/BaselineEvaluation>



(A graph of total bytes transmitted by all slices over the duration of the project.)

- Simplistic command-line tools:
 - Subversion directories full of rspecs
 - Omni (to manage slices and slivers)
 - Files with lists of logins (for input to rsync/shmux)
 - rsync (to copy files to/from plnodes)
 - shmux (to run commands on all plnodes)
 - screen (to log in to all slivers, and capture logs)
 - Common dotfiles for all plnodes
- More sophisticated tools can also do these things
 - Gush, Raven, et al
 - A little more overhead in setting them up
 - ...especially when we first started

<http://groups.geni.net/geni/wiki/PlasticSlices/Tools>

- Most things worked as expected
- Long-running experiments are more vulnerable to:
 - Infrastructure hardware/software bugs and upgrades
 - Outages
 - Large log file (filling disks, hard to analyze, etc)
- GENI is different! And gives you lots of flexibility
 - The way you design your experiment can produce different results than you'd get on a “regular” network
 - ...and our experiments clearly show this

<http://groups.geni.net/geni/wiki/PlasticSlices/Reports>

<http://groups.geni.net/geni/wiki/PlasticSlices/BaselineEvaluation>

- Question: Why so much packet loss?
 - e.g. 8% loss from BBN to Clemson with UDP in a 40-second test
- Answer: Startup delays while the OpenFlow switches across the country each contact their controller in Boston
 - As the packet hits each switch in the path, each has to phone home *in turn*, and this can take a few seconds
 - So, these stats are saying more like “the first 8% of packets failed”, not “every hundred packets, eight of them failed”
- Alternatives:
 - We were using a simplistic learning-switch controller
 - Smarter (experiment-specific) controllers can add flows in advance
 - Or the experimenter can send a little seed traffic

- Client log:

```
[ 3] Server Report:
[ 3] 0.0-38.5 sec 461 MBytes 100 Mbits/sec 0.067 ms 27877/356658 (7.8%)
[ 3] 0.0-38.5 sec 208 datagrams received out-of-order
```

- Server log:

```
[ 3] local 10.42.104.52 port 5104 connected with 10.42.104.104 port 39958
[ 3] 0.0- 1.0 sec 12.1 MBytes 101 Mbits/sec 0.053 ms 27604/36219 (76%)
[ 3] 0.0- 1.0 sec 128 datagrams received out-of-order
[ 3] 1.0- 2.0 sec 12.0 MBytes 101 Mbits/sec 465.109 ms 6/ 8523 (0.07%)
[ 3] 1.0- 2.0 sec 60 datagrams received out-of-order
[ 3] 2.0- 3.0 sec 12.0 MBytes 100 Mbits/sec 0.038 ms 11/ 8519 (0.13%)
[ 3] 2.0- 3.0 sec 19 datagrams received out-of-order
[ 3] 3.0- 4.0 sec 11.9 MBytes 100 Mbits/sec 0.043 ms 9/ 8524 (0.11%)
[ 3] 4.0- 5.0 sec 12.0 MBytes 100 Mbits/sec 0.038 ms 10/ 8547 (0.12%)
[ 3] 5.0- 6.0 sec 12.0 MBytes 100 Mbits/sec 0.031 ms 12/ 8546 (0.14%)
[ 3] 6.0- 7.0 sec 12.0 MBytes 100 Mbits/sec 0.029 ms 4/ 8539 (0.047%)
[ 3] 7.0- 8.0 sec 11.9 MBytes 100 Mbits/sec 0.032 ms 6/ 8523 (0.07%)
```

<http://www.gpolab.bbn.com/plastic-slices/baseline-logs/baseline-3/round-2/plastic-104-screen-0.log>
<http://www.gpolab.bbn.com/plastic-slices/baseline-logs/baseline-3/round-2/plastic-104-screen-1.log>

GENI is different – Topology and latency

- Question: Why such low throughput?
- Answer: TCP throughput is greatly affected by latency
 - Not all network paths are optimized for distance (on purpose, since some experiments want long links)
 - e.g. you can take ten thousand miles to get from BBN to Rutgers
- Alternatives:
 - Ye cannae change the laws of physics
 - ...but you can pick shorter or longer paths in the current topology
 - ...or design and engineer a totally different topology if need be

Results – A closer look at latency

- **BOS - CHIC - ATLA - DC - NJ (BBN to Rutgers via NLR 3715) - 74.3 ms**

```
PING 10.42.101.111 (10.42.101.111) 56(84) bytes of data.  
64 bytes from 10.42.101.111: icmp_seq=1 ttl=64 time=74.3 ms  
64 bytes from 10.42.101.111: icmp_seq=2 ttl=64 time=74.3 ms  
64 bytes from 10.42.101.111: icmp_seq=3 ttl=64 time=74.3 ms
```

- **BOS - NY - LA - HOUS - ATLA - DC - NJ (I2 3715) – 152 ms**

```
PING 10.42.103.111 (10.42.103.111) 56(84) bytes of data.  
64 bytes from 10.42.103.111: icmp_seq=1 ttl=64 time=152 ms  
64 bytes from 10.42.103.111: icmp_seq=2 ttl=64 time=152 ms  
64 bytes from 10.42.103.111: icmp_seq=3 ttl=64 time=152 ms
```

- **BOS - CHIC - DENV - SEAT - SUNN - ATLA - DC - NJ (NLR 3716) – 179 ms**

```
PING 10.42.102.111 (10.42.102.111) 56(84) bytes of data.  
64 bytes from 10.42.102.111: icmp_seq=1 ttl=64 time=179 ms  
64 bytes from 10.42.102.111: icmp_seq=2 ttl=64 time=179 ms  
64 bytes from 10.42.102.111: icmp_seq=3 ttl=64 time=179 ms
```

- **BOS - NY - DC - NJ (I2 3716) – 14.8 ms**

```
PING 10.42.104.111 (10.42.104.111) 56(84) bytes of data.  
64 bytes from 10.42.104.111: icmp_seq=1 ttl=64 time=14.8 ms  
64 bytes from 10.42.104.111: icmp_seq=2 ttl=64 time=14.8 ms  
64 bytes from 10.42.104.111: icmp_seq=3 ttl=64 time=14.8 ms
```


- The formal part of the project is now complete
 - We plan to keep running experiments and tests
 - We’ ll publish plans and results on the GENI wiki
- Keep data flowing continuously
- Run experiments that are less artificial
- Dig deeper into things that we didn’ t have time for
- Improve ops procedures and practices

Send us your ideas! help@geni.net

What next – Specific baselines

- Emergency Stop tests with every campus
- More tests with high throughput (UDP and TCP)
- More tests of high user volume (like Baseline 8)
- Dynamic ARP (for IP-based experiments)
- More iterations of long-running experiments
 - Including some more sophisticated experimental tools

Send us your ideas! help@geni.net

- If you're already a mesoscale campus:
 - Continue to support the mesoscale GENI resources
 - Write and/or maintain your aggregate info pages
 - Set and measure uptime goals
 - Communicate (esp w/ GMOC) about issues and outages
 - The GMOC is now supporting mesoscale campuses
 - Other mailing lists, chatrooms, etc, are also good for keeping in touch
 - Encourage brave experimenters at your campus
 - Tutorial about mesoscale resources in the experimenter track
 - All of this is documented in the GENI wiki, and reproducible
- If you're not a mesoscale campus:
 - Let us know if you're interested in connecting! help@geni.net

Thanks to all who
supported the project!

- Campuses: Clemson, Georgia Tech, Indiana, Rutgers, Stanford, Washington, Wisconsin
- Regionals: NoX, SoX, Indiana GigaPoP, MAGPI, CENIC, PNWGP, WiscNet
- Backbones: Internet2, NLR
- Monitoring: GMOC and GPO staff
- Software: Developers at Stanford, Princeton, Utah, GMOC, and GPO



- Any questions?

Thanks for coming!



Final report: <http://groups.geni.net/geni/wiki/PlasticSlices/Reports>