NetServ on GENI
GEC15 Houston, TX

Roberto Francescangeli
Columbia University
roberto@cs.columbia.edu

Outline

- What is NetServ
- Why GENI?
- What did I do with GENI?
- How I started using GENI
- Typical Experiment
- GENI learning curve

What is NetServ

- 4 year NSF-funded project led by Columbia University (http://www.cs.columbia.edu/irt/project/netserv)
- In-network service container
- Java-programmable, signal-driven router
- Supports both traditional network services and server services:
  - Packet interception, inspection, and modification
  - Sockets can be opened to listen to incoming connections

Why GENI? 1/2

- NetServ aims to be everywhere and we needed an at-scale testbed
- NetServ was a GENI Alpha project
  - Plenary session of GEC9
  - Demos and tutorials at GEC11, GEC12, and GEC13
- GENI gave us the resources needed for testing and also to make available our system to other researchers
  - VMs, physical machines, OF switches, WiMAX, etc.

Why GENI? 2/2

- You have full control over your machine (i.e. kernel, drivers, startup scripts)
- Integration of NetServ in GENI as a “GENI-lite” version (Java JARs vs VMs)
- NetServ on GENI for classrooms
  - GENI as an educational tool

What did I do with GENI? 1/2

- Platform development and debug
  - NetServ Container
  - NetServ Control Plane
  - NetServ Apps
- Evolution of NetServ control plane
  - Testing of a new gossip-based protocol tested on a large scale network over GENI.
- Evolution of NetServ transport plane
  - From Linux kernel to OpenFlow based switching
  - From L3 processing to L2 processing

What did I do with GENI? 2/2

- NetServ demos for other conferences (i.e. IEEE LCN 2011)
- NetServ apps development
  - ActiveCDN
  - Autonomous Management
  - Media Relay
  - SIP KeepAlive responder
- Custom image for L2 NetServ in progress
- Integration with I&M tools in progress

How I started using GENI

- Started by creating my account on one of the GENI sites
  - In my case it was Emulab, GPO, and Clemson
  - I used both Planetlab and ProtoGENI
  - Downloaded SSH keys and certificates
- Made myself comfortable with the tools that let you create GENI experiments (Omni, Flack)
  - Wiki and tutorials
- Asked on the mailing lists, always very helpful and super fast support!

Typical Experiment

- Create the topology
  - Flack if only IP is needed
  - Flack and Omni if I need OpenFlow resources
- Create startup scripts
- Launch the experiment by starting the machines (I don't need synchronization)
- Results collection by saving results into files
- SCP to retrieve results
- Release resources in GENI
- Offline processing of experimental results for stats and charts

GENI learning curve

- Credential system
  - One for each site
  - One for each testbed type (ProtoGENI, PL, etc)
- RSPEC files
  - Could be a barrier for newcomers
- GENI wiki Trac-based search engine
- Abundance of tools may be intimidating
- Users need to understand that GENI is an ongoing effort