

Enterprise-GENI Status Report

Report Q3 2009

Milestones achieved / Deliverables made

Outreach and deployments (Ticket #178, #194):

We organized an OpenFlow campus deployment workshop for the Universities that have conditionally agreed to deploy OpenFlow in their production network. We presented status of OpenFlow deployment at Stanford University, along with guidelines and good practices, to jumpstart deployments at the respective campuses.

Aggregate Manager externally accessible (Ticket #143, #179):

The E-GENI aggregate manager can be accessed by pointing the E-GENI SFA server to `nfcml3.stanford.edu` port 7889

Integration of E-GENI and PlanetLab control framework (Tickets #144, #176):

We have completed integration of the E-GENI control framework with that of PlanetLab, and tested it between Stanford and Princeton. The online source code of the aggregate manager has been updated, and the necessary changes to the SFA server code base has been checked into `svn.planet-lab.org`

The integration feature 3 PL nodes connected to the OpenFlow network at Stanford, as well as a 2 PL nodes tunneled in from Princeton using a Soekris box. At the Stanford side, we use the experimental Stanford OpenFlow deployment, comprising of about 8 OpenFlow switches and 30 OpenFlow Wireless APs, as the substrate network. At the Princeton side, we partitioned two Soekris boxes into 4 OpenFlow switches, with each running the software reference implementation (0.8.9_ver 4) of OpenFlow.

All switches at Princeton, and Stanford communicate with the same FlowVisor at Stanford University. The portion of the flowspace, to be used to the experiments, are marked as unroutable in the FlowVisor policy, until a slice explicitly requests for that flowspace (or a subset). The aggregate manager is used to manage access to this flowspace. The slices in the FlowVisor will be created by the Aggregate Manager after each call from the ClearingHouse.

We use the SFI tool (developed by the PlanetLab consortium) as the Clearinghouse for the integration test. We use SOAP/sfa.wsdl for communication between SFI and the E-GENI aggregate manager¹. The E-GENI aggregate manager is open on `nfcml3.stanford.edu:7889` and responds to 7 main function calls (`create_slice`, `start_slice`, `stop_slice`, `delete_slice`, `reset_slices`, `get_slices`, `get_resources`).

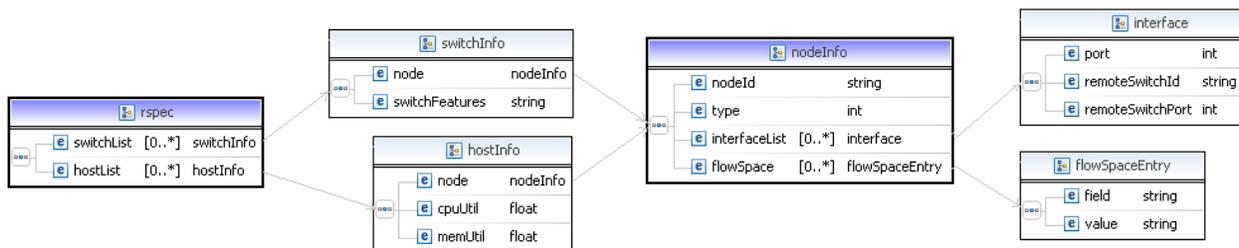


Fig.1: Format of the Enterprise-GENI RSpec

¹ All authentication issues and registry issues are reserved as future work, and ignored for the purpose of this integration test

When a *get_resources* call is placed, the aggregate manager queries the network state and returns the list of resources in the form of the RSpec shown in Fig.1. The experimenters use the same RSpec format during a *create_slice* call to request for resources from the aggregate.

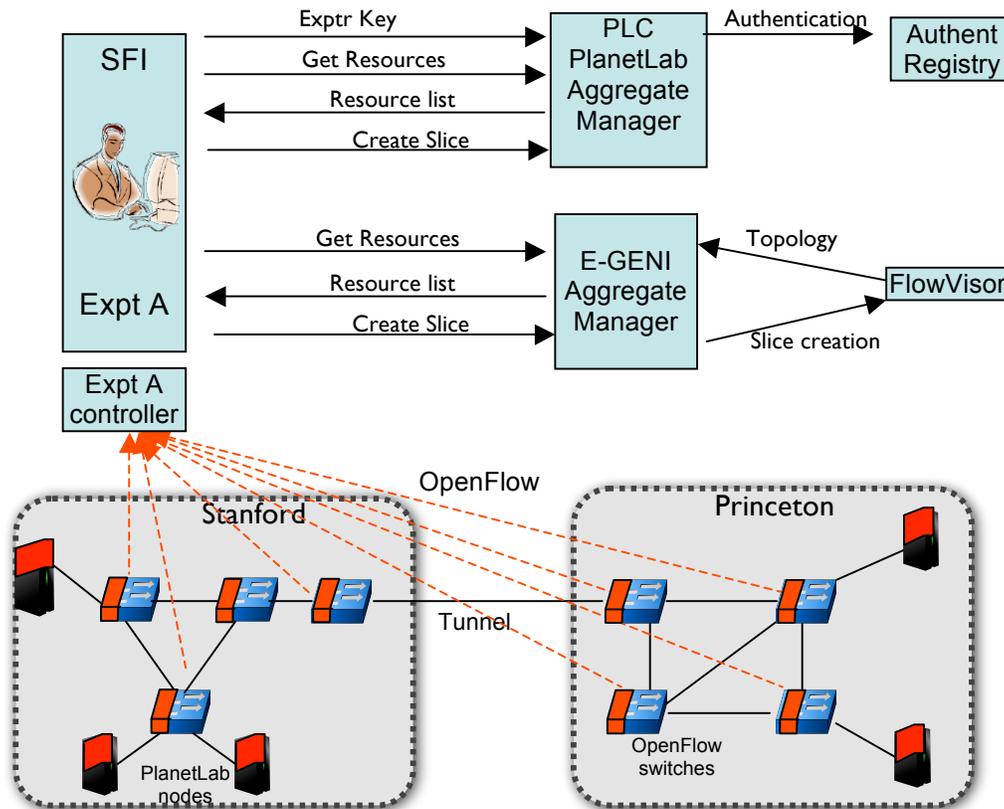
Summary of demonstration (Ticket #175):

We illustrated the integration of the two control frameworks by creating slices on both PL and E-GENI substrate using the same SFI tool, and having the two slivers provide an integrated feel of an end-to-end slice. In this slice, PL nodes offer the computing resources, while E-GENI provides the networking resources. We conducted a simple test that uses the OpenFlow substrate to send TCP traffic between two PlanetLab nodes using the *netcat* tool over several TCP ports.

In the demonstration, the experimenter first creates a slice in the PlanetLab substrate using SFI and initiates *netcat* send/receive calls. The PL nodes identify the *bind* call on that TCP port, and use that to demultiplex the traffic belonging to that slice. Next, the experimenter issues a slice creation call to the E-GENI aggregate manager, with the flowspace requested in the RSpec corresponding to the TCP port number of interest.

The aggregate manager creates the necessary FlowVisor policies, after which the experimenter’s controller receives control over the requested flowspace and topology. The experimenter’s controller (running the NOX “routing” module) then undertakes the necessary switching decisions, and makes the necessary flow table entries in the network, thereby causing the *netcat* traffic to start flowing.

Following is the logical sequence of events that occur in the demonstration:



More details at: http://www.openflowswitch.org/wk/index.php/GENI_Control_Integration_Sept09

Publications (individual and organizational)

- KK. Yap, M. Kobayashi, D. Underhill, S. Seetharaman, P. Kazemian, N. McKeown, “*The Stanford OpenRoads Deployment*,” ACM Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterization (WiNTECH), September, 2009.