Measurement Service Requirements in a GENI Experiment Slice

(encompassing Internet2 backbone resources)

Prasad Calyam, Ph.D.

pcalyam@osc.edu

Internet2 Performance Working Group Presentation,
February 2010

Topics of Discussion

- What is different in GENI facility measurements?
- GENI Measurement Projects
- OnTimeMeasure-GENI Measurement Service
  - Overview
  - Architecture
- Measurement Service Capabilities
- Discussion

What is different in GENI facility measurements?

- GENI supports testbeds aimed at “clean-slate” re-design of the Internet to overcome limitations of current Internet
- Users have greater options/control on measurements
  - Measurement server software/hardware
  - Advanced open-source/commercial instrumentation
  - Measurement service providers (who may customize)
  - Measurements across wired/wireless aggregates
  - Internet-scale measurements with “interesting” cross-traffic

GENI Measurement Projects (Spiral 2)

- Instrumentation Tools – Uni. Of Kentucky
- Measurement System – Led by Uni. of Wisconsin
- LAMP – Uni. of Delaware, Internet2
- Scalable Monitoring – Purdue Uni., HP
- OnTimeMeasure – The Ohio State Uni., OARnet/OSC

For details, see - http://groups.geni.net/geni/wiki/SpiralTwo

OnTimeMeasure-GENI Overview

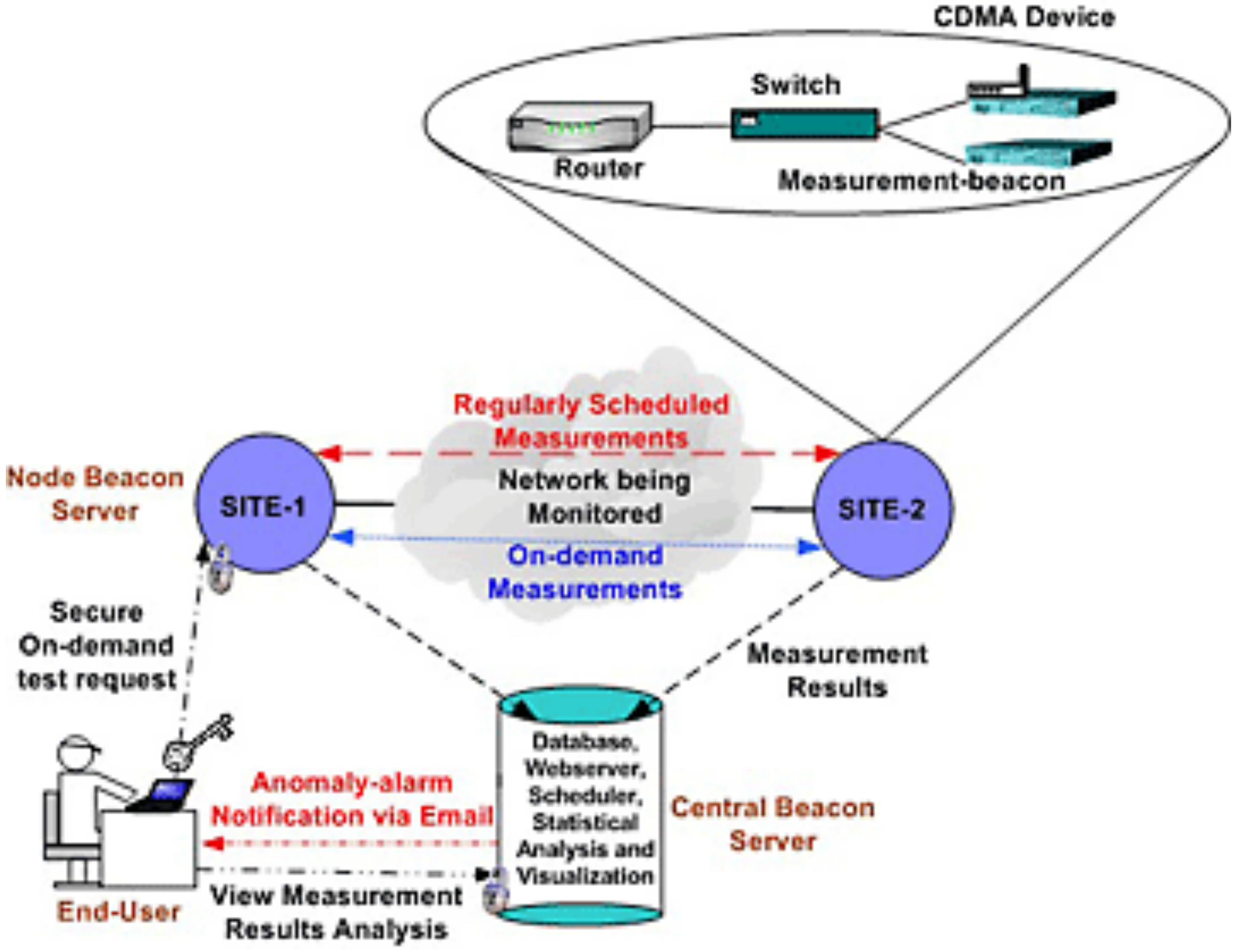
- Goal: Provide GENI community with a shared measurement service for provisioning on-going and on-demand measurement requests
  - To be deployed on ProtoGENI testbed (distributed over Internet2 – WASH, UTAH, KANS) for researchers and GENI Meta Operations
- Expected Outcomes:
  - OnTimeMeasure-GENI Software to perform centralized and distributed measurement orchestration and provisioning of active measurements
  - Measurement service that uses OnTimeMeasure-GENI in GENI experiments to enable:
    - Network paths monitoring
    - Network weather forecasting
    - Network performance anomaly detection
    - Network-bottleneck fault-location diagnosis

OnTimeMeasure Software

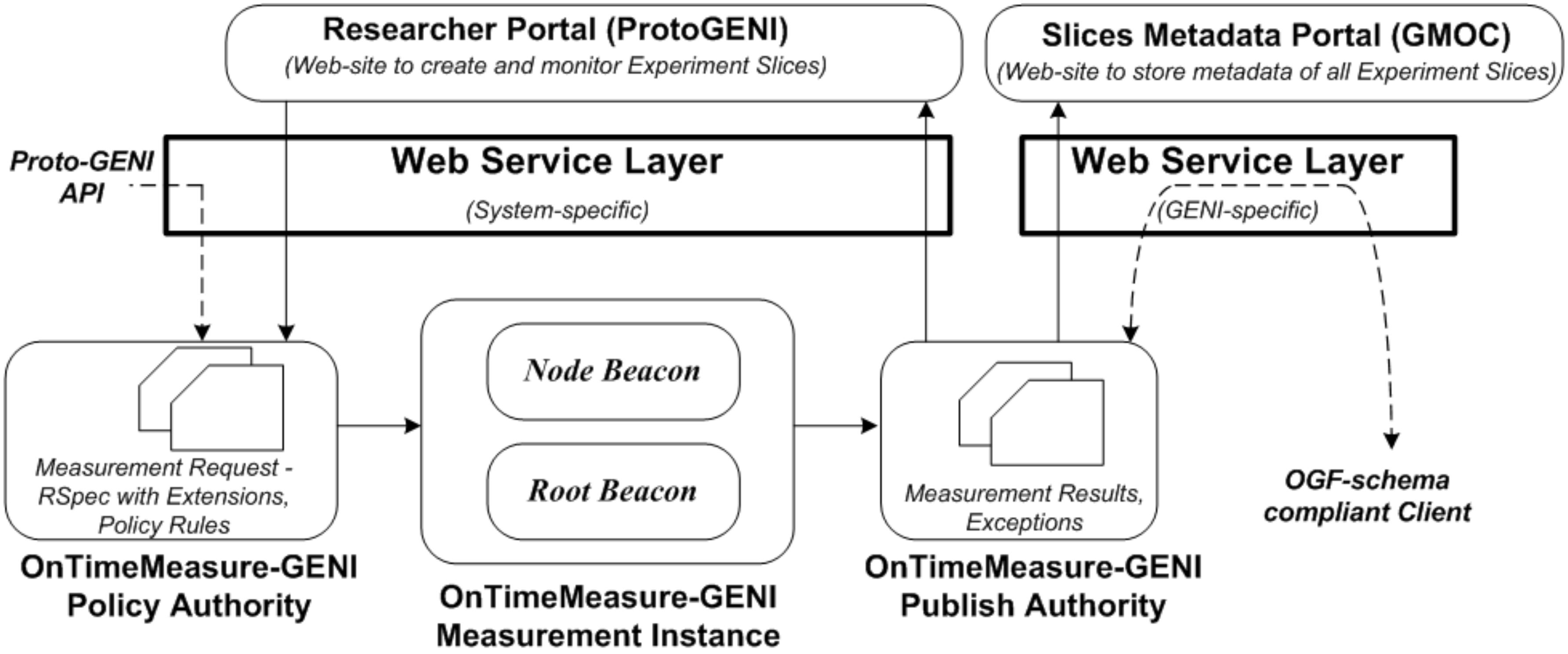
- Customizable software [Calyam-TC] developed at OSC/OARnet
- Two main modules installed within a GENI experiment slice as part of an active measurement service
  - Node Beacon
    - Installs tools that measure network health metrics such as: route changes, delay, jitter, loss, bandwidth
      - TCP/UDP Iperf, Traceroute, Ping, Pathload, OWAMP, etc.
    - Runs measurements based on a schedule and outputs in “raw” and “processed” formats
  - Root Beacon
    - Installs Apache, MySQL, creates database tables, generates measurement schedules for node beacons, collects data and provides dashboard visualization, statistical analysis (i.e., anomaly detection and weather forecasting) with alarm generation

[Calyam-TC] P. Calyam, C.-G.Lee, E. Ekici, M. Haffner, N. Howes, “Orchestrating Network-wide Active Measurements for Supporting Distributed Computing Applications”, *IEEE Transactions on Computers Journal (TC)*, 2006.

OnTimeMeasure Deployment


CDMA Device
Switch
Router
Measurement-beacon
Regularly Scheduled Measurements
Network being Monitored
On-demand Measurements
Node Beacon Server
SITE-1
SITE-2
Secure On-demand test request
Measurement Results
Database, Webserver, Scheduler, Statistical Analysis and Visualization
Central Beacon Server
Anomaly-alarm Notification via Email
View Measurement Results Analysis
End-User

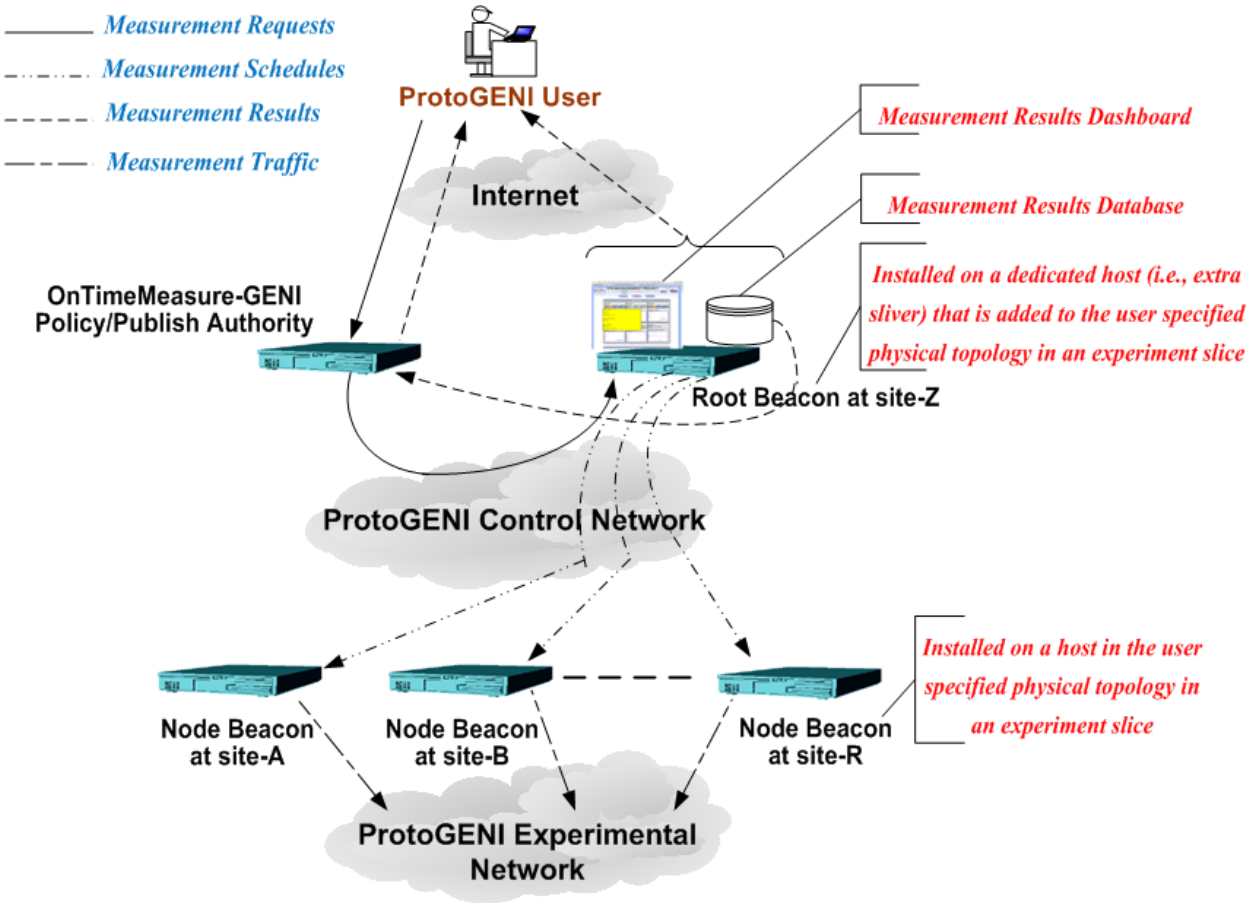
OnTimeMeasure-GENI Integration in ProtoGENI


Researcher Portal (ProtoGENI)
(Web-site to create and monitor Experiment Slices)
Slices Metadata Portal (GMOC)
(Web-site to store metadata of all Experiment Slices)
Proto-GENI API
Web Service Layer
(System-specific)
Web Service Layer
(GENI-specific)
Node Beacon
Root Beacon
Measurement Request - RSpec with Extensions, Policy Rules
OnTimeMeasure-GENI Policy Authority
OnTimeMeasure-GENI Measurement Instance
Measurement Results, Exceptions
OnTimeMeasure-GENI Publish Authority
OGF-schema compliant Client

Assumptions

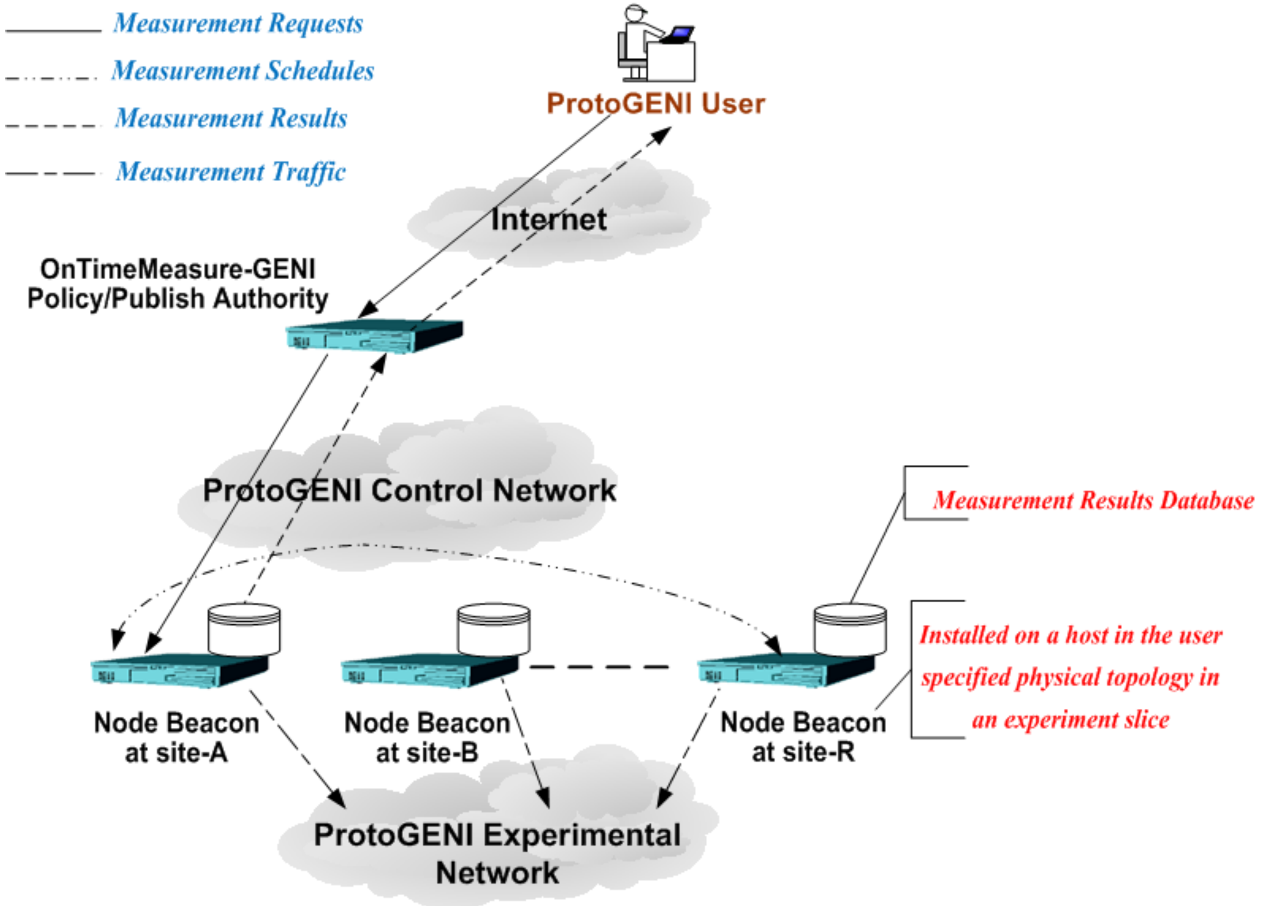
- All user requests for measurements are routed through the ProtoGENI control framework for slice requests
  - User requests measurement instances in an out-of-band fashion through ProtoGENI API or “My Experiments” web-page
- Each experiment slice needing measurement gets its own OnTimeMeasure-GENI instance
  - Rspec extensions could allow reservation of measurement resources
  - Node/Root Beacons get installed as slivers
- User sees centralized and distributed measurement data and related analysis display through separate OnTimeMeasure web-portal (can be linked from “My Measurements” web-page in ProtoGENI)
- User can also request measurement data using an OGF-NMWG schema compliant client with corresponding slice experimenter’s credentials
- Every OnTimeMeasure-GENI instance metadata gets stored at GMOC

Centralized Orchestration


Measurement Requests
Measurement Schedules
Measurement Results
Measurement Traffic
ProtoGENI User
Internet
Measurement Results Dashboard
Measurement Results Database
Installed on a dedicated host (i.e., extra sliver) that is added to the user specified physical topology in an experiment slice
OnTimeMeasure-GENI Policy/Publish Authority
Root Beacon at site-Z
ProtoGENI Control Network
Node Beacon at site-A
Node Beacon at site-B
Node Beacon at site-R
Installed on a host in the user specified physical topology in an experiment slice
ProtoGENI Experimental Network

- Centralized scheduling for continuous monitoring, persistent measurements storage and processed network measurement feeds
  - Useful for “network weathermaps” and long-standing experiments with advanced measurement analysis capabilities

Distributed Orchestration



- Distributed scheduling for on-demand measurement requests without need for persistent measurements storage
  - Useful for users or helper apps needing one-off or occasional raw measurement tool outputs

Measurement Service Capabilities

- Active measurement request handling
  - Sampling requirements (e.g., periodic, stratified random, random, adaptive)
  - Active measurement tools to be used (e.g., Ping, Traceroute, Iperf)
  - Measurement topology (e.g., full-mesh, tree, hybrid)
- Enforce policies for measurements scheduling
  - Semantic priorities (e.g., superGENI-er vs. GENI-er)
  - Measurement level restrictions (e.g., allowable measurement bandwidth and measurement flow duration for different nodes/paths/user-roles)
- Provide raw and processed measurement
  - Measurements provisioning interfaces (e.g., raw output of a tool to human/ component-service, processed output of multiple tools to a viz application)
  - Measurement use context (e.g., curiosity about network path(s) performance in a new slice, network-awareness in an experiment to develop a novel network control scheme, troubleshoot a network bottleneck affecting an advanced application)

Discussion

- How will researchers use GENI measurement services?
- How will national, regional and campus network operators take advantage of GENI measurement services?
- How will industry adapt their strategies to support GENI measurement services?
- How will home users benefit from GENI measurement services?

Thank you for your attention ! ☺