OnTimeMeasure-GENI: Centralized and Distributed Measurement Orchestration Software

Prasad Calyam, Ph.D. (PI)

Paul Schopis, (Co-PI)
Weiping Mandrawa (Network Software Engineer)
Chris Hartley (Network Testbed Engineer)

6th GENI Engineering Conference, Salt Lake City, UT
November 2009

Spiral 2 Project Objectives

- Develop a measurement service for GENI to timely cater *on-going* and *on-demand* active measurement requests
  - Extend existing OnTimeMeasure software [Calyam-TC] to enable monitoring, forecasting, anomaly detection, and fault-location diagnosis for GENI experiments (ProtoGENI) and operations (GMOC)

- Year-1: Prototype service for handling on-going measurements
  - Define technical and policy requirements for IP-layer active measurements, and design OnTimeMeasure software extensions
  - Multi-federation measurement testbed setup involving 4-sites on OARnet backbone, and any relevant ProtoGENI and GMOC sites
  - On-going measurements orchestration software package development and release for early experimenters to perform *centralized* and *distributed* active measurement scheduling
  - Integrate web-services to support: (i) measurement request/response schemas that allow initiation/termination/status-query, (ii) movement of measurement data between databases, instrumented-nodes and users
  - Document validation experiences from testbed deployments and study integration with other GENI clusters comprising of multiple aggregates

[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.

Challenges

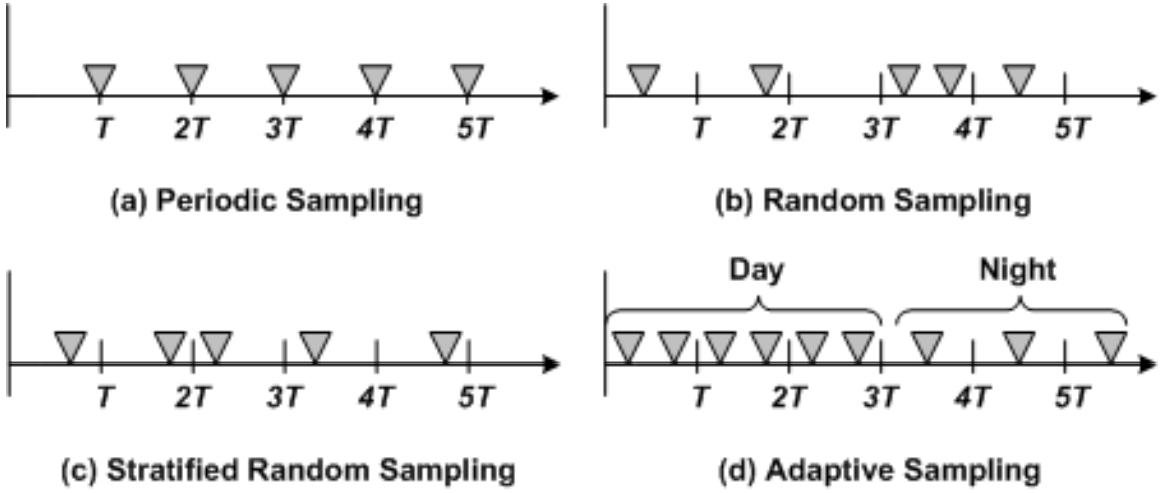
- Identifying nature of measurement requests that will come to the measurement service from users
  - Sampling requirements (e.g., periodic, poisson, exponential, adaptive)
  - Tools to be supported (e.g., Ping, Traceroute, Iperf, custom?, commercial?)
- Identifying policies for measurements scheduling within specific clusters and across cluster federations
  - Semantic priorities (e.g., intra vs. inter-domain, superGENI-er vs. GENI-er!)
  - Measurement level restrictions (e.g., allowable measurement bandwidth and measurement flow duration for different nodes/paths/user-roles)
  - Security enforcement frameworks (e.g., application-level, network-level)
- Identifying analysis objectives of measurement requests
  - 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, verify repeatability of experiments in a slice, troubleshoot a network bottleneck)

Proposed Approach

- Design inputs from ProtoGENI team, GMOC team, RSpec efforts, other instrumentation and measurement project efforts
- Demo at GEC7 – showcase basic prototype of OnTimeMeasure-GENI measurement service and get user community feedback
- Release v1.0 software package in end of May 2010, and actively engage potential ProtoGENI and GMOC users to use the software
- Demo at GEC8 – showcase efforts with early experimenters using the v1.0 software on the multi-federation testbed
- Based on the v1.0 deployment and validation experiences, work with users of other clusters and develop plans for Year-2 and Year-3

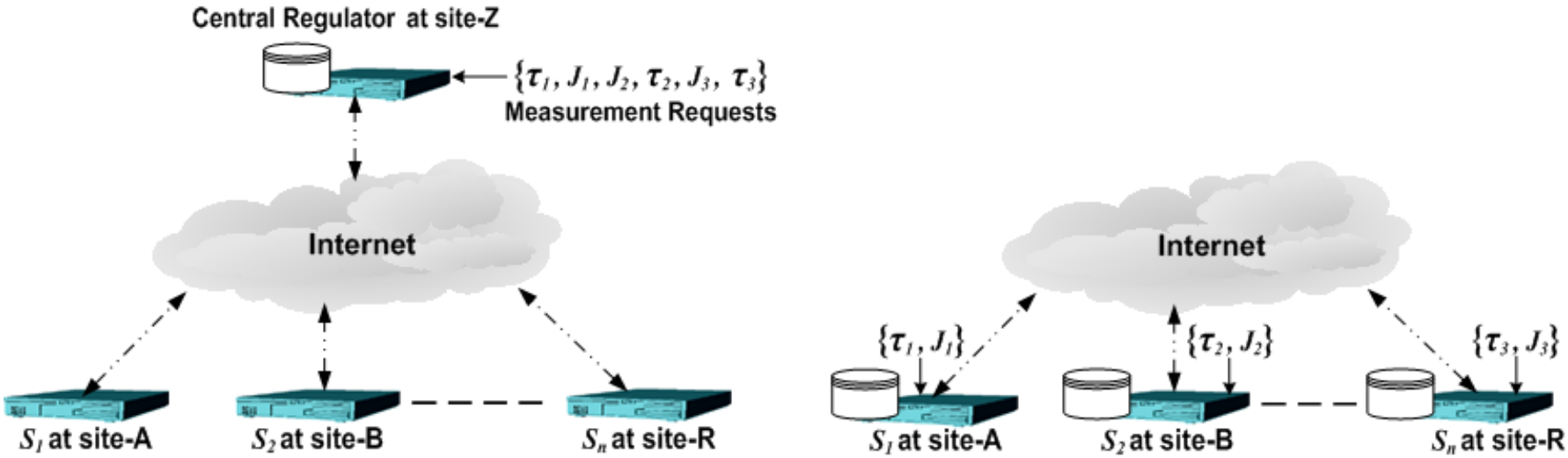
Thank you for your attention ! ☺

Network Status Measurement Scheduling



- Ongoing Measurements
  - Frequent sampling for anomaly detection, strict periodicity for network weather forecasting, random (e.g., poisson, exponential)
    - Sampling time interval pattern chosen depends on the monitoring accuracy objectives (e.g. least variance between actual and estimated)
- On-demand Measurements
  - One-off measurements with quick response times (e.g. to traceback a DDoS attack in a network segment)

Centralized versus Distributed Scheduling



(a) Centralized scheduling **(b) Distributed scheduling**

- Centralized scheduling maintains a global schedule
  - E.g., suitable for “network weathermaps”
- Distributed scheduling avoids single point of failure and provides greater flexibility in data collection and analysis
  - E.g., suitable for multi-federation measurements