

**OnTimeMeasure: Centralized and Distributed Measurement
Orchestration Software**

*High-Level Design Document for Global Environment for
Network Innovations (GENI) Spiral 2 Project # 1764*

Prasad Calyam, Kunpeng Zhu, Yingxiao Xu, Alex Berryman

**Ohio Supercomputer Center/OARnet,
The Ohio State University**

This material is based upon work supported by the National Science Foundation under Grant No. CNS-0940805. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of BBN Technologies, Corp., the GENI Project Office, or the National Science Foundation.

Table of Contents

I. Purpose of this document	3
II. Revision History.....	3
III. Project Overview	3
IV. OnTimeMeasure Software.....	3
A. Node Beacon	4
B. Root Beacon	4
V. Measurement Service Capabilities.....	5
A. Researcher Portal	6
B. Policy Authority	7
C. Measurement Instance.....	7
D. Publish Authority	7
E. Web Services	7
i. Measurement Request Submission	8
ii. Measurement Service Control	8
iii. Measurement Results Query.....	9
VI. OnTimeMeasure-GENI Deployment Architectures	10
A. Centralized Orchestration	10
B. Distributed Orchestration	10
VII. Conclusion.....	11
Acknowledgements.....	11
References.....	11

I. Purpose of this document

This document presents the high-level design of the “OnTimeMeasure: centralized and distributed measurement orchestration software” and its integration into the GENI control frameworks and GENI Meta Operations (GMOC).

II. Revision History

Version ID	Revised Sections	Date	Remarks
OnTimeMeasure-GENI_HLD_Dec09	All	December 2009	First version of the OnTimeMeasure-GENI software high-level design details
OnTimeMeasure-GENI_HLD_Dec09	IV, V, VI	March 2010	Minor revisions based on the prototype development experiences and services terminology discussions in the I&M Working Group
OnTimeMeasure-GENI_HLD_Dec09	All	March 2011	Updated design based on Spiral 2 development of OnTimeMeasure

III. Project Overview

The goal of our project is to provide the GENI community with a measurement service for provisioning on-going and on-demand measurement requests. The measurement service can be deployed on testbeds aimed at “clean-slate” re-design of the Internet to overcome limitations of the current Internet. For example, the measurement service can be deployed on ProtoGENI [1], PlanetLab [16] and CRON testbed [17] for Internet-scale research experiments. The measurement service can also be deployed on a separate testbed for GMOC [2] that monitors the GENI facility status.

The expected outcomes of our project are:

- OnTimeMeasure software to perform centralized and distributed measurement orchestration and provisioning of active measurements
- Measurement service that uses OnTimeMeasure in GENI experiments to enable:
 - Network paths monitoring
 - Network weather forecasting
 - Network performance anomaly detection
 - Network-bottleneck fault-location diagnosis

IV. OnTimeMeasure Software

The OnTimeMeasure software was originally developed at the Ohio Supercomputer Center/OARnet through seed funding from the Ohio Board of Regents. It was deployed as a measurement service for orchestration of active measurements on the Ohio’s regional network backbone, OARnet. The software service code-named “OnTimeMeasure” is part of OSC/OARnet’s ActiveMon [3] framework for network health status monitoring. The algorithms used in OnTimeMeasure software have been extensively peer-reviewed and the software has been deployed on numerous network testbeds involving both regional and national partners [4 - 8].

There are two main modules in OnTimeMeasure: (i) Node Beacon, and (ii) Root Beacon. The system setup for the node and Root Beacons to provision on-demand and on-going active measurements is shown in Figure 1.

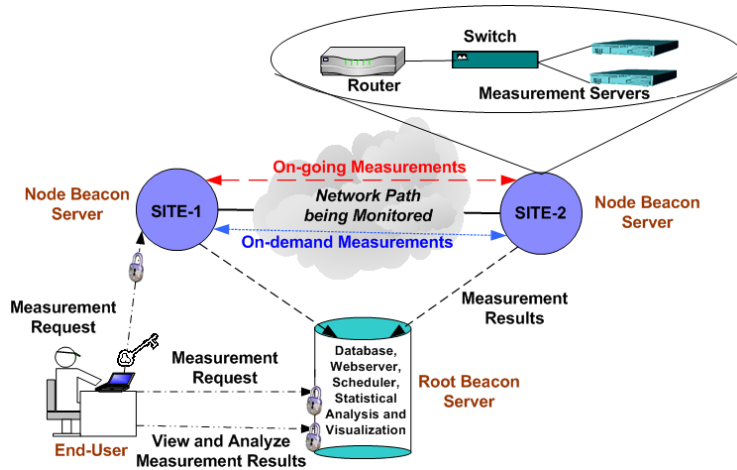


Figure 1: System Setup for Node and Root Beacons

A. Node Beacon

The Node Beacon software installs active measurement tools on a measurement server and measures network health in terms of metrics such as: route changes, delay, jitter, loss, and bandwidth. The active measurement tools supported include: TCP/UDP Iperf, Traceroute, Ping, Pathrate, Pathload, Pathchar, and OWAMP. Other tools can be easily supported in the OnTimeMeasure framework if there is user demand. The Node Beacon initiates measurements based on a conflict-free and policy-compliant measurement schedule, and generates measurement results in “raw” and “processed” formats. The processed formats are for storing measurements in a MySQL database for visualization and statistical analysis of measurement results.

B. Root Beacon

The Root Beacon software installs Apache webserver and MySQL database. It also creates database tables, generates measurement schedules for Node Beacons, collects measurement results and provides dashboard visualization, as well as statistical analysis (i.e., anomaly detection and network weather forecasting) coupled with alarm generation capabilities.

Table 1 below lists the various services offered by the Node and Root Beacons as specified in the GENI Instrumentation and Architecture (GIMA) document [15].

Service Name	Functions	Module
Measurement Orchestration (MO)	Part of Experiment Control service, uses a language to orchestrate I&M services	Root Beacon
Measurement Point (MP)	Instrumentation that taps into a network and/or systems, links and/or nodes, to capture measurement data and format it using a standardized schema	Node Beacon
Measurement Collection (MC)	Programmable systems that collect, combine, transform and cache measurement data	Root Beacon
Measurement Analysis and Presentation (MAP)	Programmable systems that analyze and then present measurement data	Root Beacon
Measurement Data Archive (MDA)	Measurement data repository, index and portal	Root Beacon

Table 1: OnTimeMeasure Modules that caters to GIMA defined Service Functions

V. Measurement Service Capabilities

- Measurement request handling
 - Path-based active measurements (e.g., delay, jitter, loss, throughput, route changes)
 - Measurement topology (e.g., full-mesh, tree, hybrid)
 - Sampling requirements (e.g., periodic, stratified random, random, adaptive)
 - Host-based system performance (e.g., CPU, memory)
- Enforce policies for measurements scheduling
 - Measurement level restrictions for probing tools (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 data or graph output to human/component-service, processed output of multiple tools to a dashboard)
 - Measurement use context hooks (e.g., verify network path(s) performance in a new slice, network-awareness in an experiment to develop a novel network control scheme)
 - OnTimeMeasure instance metadata (e.g., needed by GMOC, NetKarma)

Figure 2 shows the architecture for OnTimeMeasure software integration in GENI.

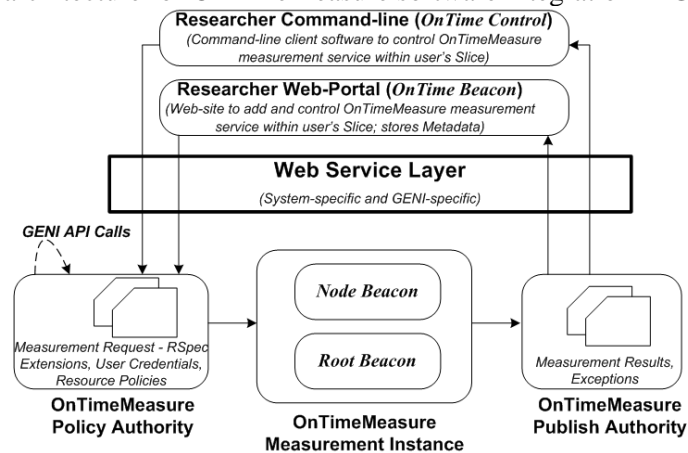


Figure 2: Architecture for OnTimeMeasure Software integration in GENI

User Workflow for OnTimeMeasure is as follows:

1. User creates an experiment slice using ProtoGENI or PlanetLab control framework tools
2. User registers at the "Researcher Web-Portal" (<http://ontime.oar.net>)
3. User installs OnTimeMeasure measurement instance in experiment slice
 - Slice RSpec should include reservation of any required measurement resources
 - Node/Root Beacons need to be installed as slivers based on the instructions provided in Step-1 for the specific aggregate
4. Valid login to the "Researcher Web-Portal" allows user to:
 - Provide slice RSpec information for the measurement service
 - Interact with the measurement service in user's experiment slice
 - i. Submit measurement requests
 - ii. Control the measurement service
 - iii. Query measurement data

A. Researcher Portal

The researcher portal refers to the “OnTime Beacon” web-portal (<http://ontime.oar.net>) that enables a GENI users to submit measurement requests to the OnTimeMeasure measurement service. The user first registers with the OnTime Beacon web-portal, and later provides the RSpec information. This information will be used by the OnTime Beacon to communicate with the OnTimeMeasure software modules running within the user’s GENI slice. Subsequently, the user can use the OnTime Beacon web-portal to invoke and control the measurement service functions within the user’s slice.

The user can submit a measurement request by specifying the network status sampling desired (see Figure 3 for sampling patterns), the active measurement tools (e.g., Ping, Traceroute, Iperf) to be installed on the Node Beacons and the measurement topology that determines the measurement matrix (e.g., full-mesh, tree, hybrid) of the Node Beacons. We can refer Figure 4 that shows the difference between physical topology and measurement topology in an experiment slice. The full-mesh topology will add measurement tasks for each Node Beacon to measure paths to all the other Node Beacons, the tree topology (shown in Figure 4 example) will add measurement tasks for each Node Beacon to measure paths between neighboring Node Beacons, and the hybrid topology will add measurement tasks that have portions of full-mesh and portions of hybrid network paths.

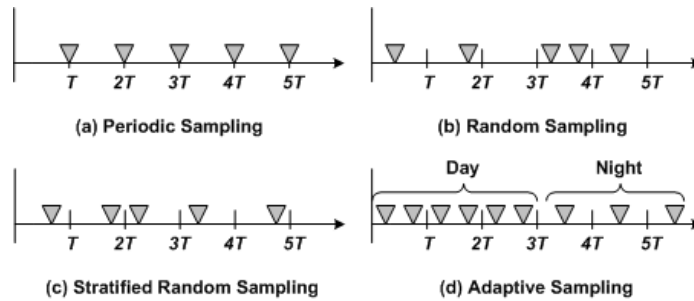


Figure 3: Network Status Measurement Sampling Patterns

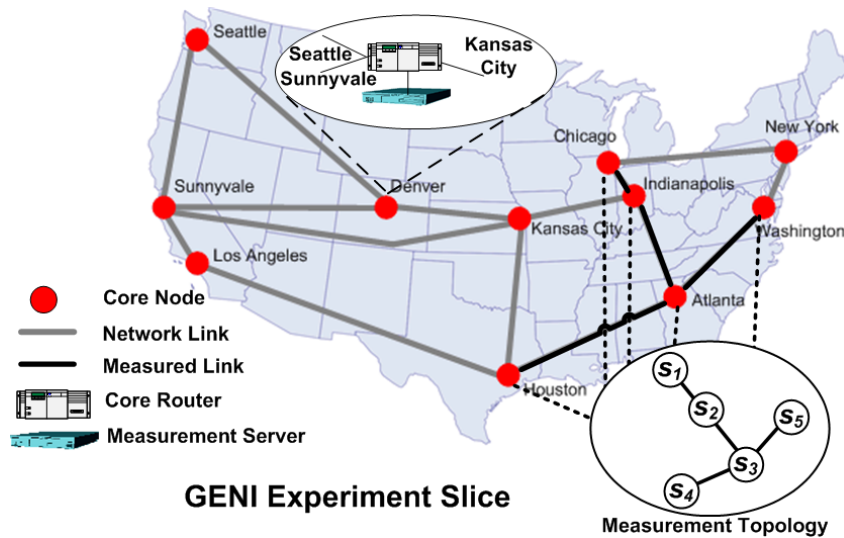


Figure 4: Measurement Topology in a GENI Experiment Slice

B. Policy Authority

OnTimeMeasure Policy Authority is part of the OnTime Beacon web-portal that determines the list of measurement capabilities that a user is permitted to request in an experiment slice. The measurement capabilities permitted are based on pre-defined policy specifications of the resources in the measurement topology. The policies enforced for measurements scheduling include: (a) semantic priorities based on user roles (e.g., superGENI-er vs. GENI-er) to resolve measurement scheduling conflicts, and (b) measurement level restrictions (e.g., allowable measurement bandwidth and measurement flow duration for different nodes/paths/user-roles) to regulate the amount of network measurement traffic permitted on network paths. Once a measurement request has been cleared by the Policy Authority, the measurement request will be processed to generate slivers in experiment slices with installs of Node and Root Beacon software. In addition, the user can add/ remove a set of measurement tasks for the OnTimeMeasure scheduler to process and generate measurement timetables for Node Beacons. Further, the user can start, stop and also check status of the various measurement service components. The measurement service components whose status is reported include: (a) Slice Accessibility, (b) Root Beacon Scheduler, (c) Node and Root Beacon Communications, (d) Measurement Data Collector, (e) Analysis and Publish Authority, and (f) Measurement Data Visualization.

C. Measurement Instance

Each experiment that uses the OnTimeMeasure measurement service gets its own measurement instance comprising of a Root Beacon and/or multiple Node Beacons. Consequently, when the slice expires, the measurement instance also expires and the collected measurements will be deleted. If the user requires all the experiment slice measurements, the measurement service will provide a copy of the measurement archive (e.g., MySQL dump, graph plots, time series data files) before the slice expires.

D. Publish Authority

The OnTimeMeasure Publish Authority is part of the OnTime Beacon web-portal that provides raw and processed measurement results back to the user. The processed measurement results could correspond to: (a) time series of active measurements over a time range, (b) time series of active measurements over a time range with associated network performance anomaly events, and (c) time series of active measurements over a time range with associated network weather forecasts. The user can also request measurement data from the Publish Authority using a web-service client ([OnTime Beacon web-portal](#) or [OnTime Control command-line client](#)). Before a slice expires, the Publish Authority will keep the measurements metadata in the Root Beacon database, and also will transfer a copy of the measurements metadata of an experiment slice to GMOC (i.e., to the NetKarma [14] metadata database).

E. Web Services

OnTimeMeasure web services will allow users to interact with the measurement service without requiring them to know the software and database internals. The web services will also enable users to use any GENI compliant measurement request/response schemas. Consequently, we will develop a web services API that has both “system-specific” and “GENI-specific” layers. The web services API will have 3 managers based on the functionality they provide: (i) Measurement Request Submission, (ii) Measurement Service Control, and (iii) Measurement Results Query.

i. Measurement Request Submission

Name	AddCentralizedMeasurementServiceToSlice
Type	System-specific
Description	Adds measurement topology to physical topology of experiment slice and requests for installation of a Root Beacon and one or more Node Beacons. Here, we can assume that that the user has previously submitted a valid RSpec of the user's reserved GENI slice to the OnTimeBeacon, and has credentials for the OnTime Beacon to access the user's GENI slice

Name	AddDistributedMeasurementServiceToSlice
Type	System-specific
Description	Adds measurement topology to physical topology of experiment slice and requests for installation of one or more Node Beacons. Here also, we can assume that that the user has previously submitted a valid RSpec of the user's reserved GENI slice to the OnTimeBeacon, and has delegated credentials for the OnTime Beacon to access the user's GENI slice

Name	CheckPolicyConstraints
Type	System-specific
Description	Returns a list of measurement capabilities that a user is permitted to request in an experiment slice; the capabilities permitted are based on the pre-defined policy specifications of the resources in the measurement topology

Name	AddMeasurementTasks
Type	System-specific
Description	Adds a set of measurement tasks for the OnTimeMeasure scheduler to process and generate measurement timetables for Node Beacons

Name	DeleteMeasurementTasks
Type	System-specific
Description	Clears one or more of the measurement tasks being processed by the OnTimeMeasure scheduler

Name	SubmitMeasurementRequest
Type	System-specific
Description	Invokes the OnTimeMeasure scheduler to process the measurement tasks and generate measurement timetables, thus initializing the measurement service

ii. Measurement Service Control

Name	CheckMeasurementServiceStatus
Type	System-specific
Description	Notifies status of the various measurement service components that include: (a) Slice Accessibility, (b) Root Beacon Scheduler, (c) Node and Root Beacon Communications, (d) Measurement Data Collector, (e) Analysis and Publish Authority, and (f) Measurement Data Visualization.

Name	<code>StartMeasurementService</code>
Type	System-specific
Description	Initiates communications between Root Beacons and/or Node Beacons to start the active measurements data collection

Name	<code>StopMeasurementService</code>
Type	System-specific
Description	Terminates communications between Root Beacons and/or Node Beacons to stop the active measurements data collection

iii. Measurement Results Query

Name	<code>QueryRawMeasurementResults</code>
Type	GENI-specific
Description	Returns raw tool outputs of active measurements

Name	<code>QueryMeasurementResultsForTimeRange</code>
Type	GENI-specific
Description	Returns time series of active measurements over a time range

Name	<code>TransferMeasurementRequestMetadata</code>
Type	GENI-specific
Description	Transfers metadata of active measurements of an experiment slice to GMOC (NetKarma database)

Name	<code>QueryPerformanceAnomaliesForTimeRange</code>
Type	System-specific
Description	Returns time series of active measurements over a time range with associated network performance anomaly events

Name	<code>QueryPerformanceForecastsForTimeRange</code>
Type	System-specific
Description	Returns time series of active measurements over a time range with associated network weather forecasts

Name	<code>RequestMeasurementResultsArchive</code>
Type	System-specific
Description	Returns measurement results archive (e.g., MySQL dump, graph plots, time series data files) of the experiment slice

Based on GENI user requirements, more measurement analysis API functions can be included in the future such as: (a) `ReportsSliceNetworkStatus` that checks delay, loss, jitter, bandwidth of all the network paths of the experiment slice, (b) `EnableNetworkAwareness` that initiates a complex sampling and analysis profile to provide network-awareness in an experiment to develop for e.g., a novel network control scheme, and (c) `DiagnoseNetworkBottleneck` that attempts to troubleshoot a network bottleneck that is affecting an advanced application e.g., a bulk data transfer using a custom TCP protocol.

VI. OnTimeMeasure-GENI Deployment Architectures

The OnTimeMeasure-GENI measurement service can be deployed either using a “Centralized Orchestration” architecture, or a “Distributed Orchestration” architecture. The centralized orchestration allows measurement scheduling for continuous monitoring, persistent measurements storage in an experiment slice and processed network measurement feeds. Such functionality will generally be useful for “network weathermaps” and long-standing experiments with advanced measurement analysis capabilities. In comparison, the distributed orchestration allows measurement scheduling of on-demand measurement requests without need for persistent measurements storage. Such functionality will generally be useful for users or helper apps needing one-off or occasional raw measurement tool outputs.

A. Centralized Orchestration

Figure 5 shows the deployment architecture for centralized orchestration. We can see that the GENI user interacts with the Policy Authority and Publish Authority in GENI using the Internet. The Root Beacon does not rely on experimental network to interact with the Node Beacons, but instead uses the GENI Control Network. The active measurements collected will be on the experimental network paths to which the Node Beacons will be connected. We assume that the Root Beacon has sufficient storage to support the measurement service requirements of a GENI user.

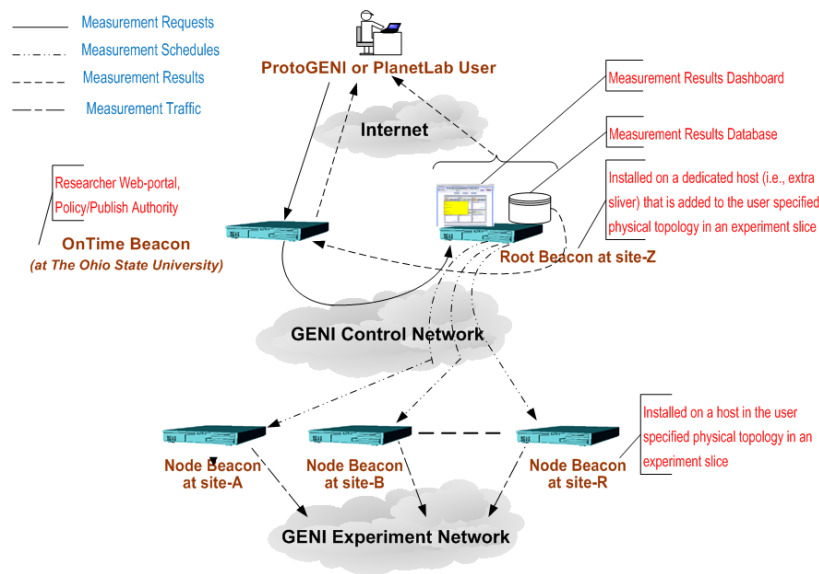


Figure 5: Deployment Architecture for Centralized OnTimeMeasure Software

B. Distributed Orchestration

Figure 6 shows the deployment architecture for distributed orchestration. We can see that the GENI user interacts with the Policy Authority and Publish Authority in GENI using the Internet. No Root Beacon will be installed when a user demands a distributed orchestration architecture for the measurement service. The Node Beacons do not rely on experimental network to interact with each other for exchanging measurement schedules, but instead use the GENI Control Network. The active measurements collected will be on the experimental network paths to which the Node Beacons will be connected. Given that there is no persistent measurement data storage necessity for distributed orchestration, the Node Beacons need not have large amounts of storage to support the measurement service requirements of a GENI user.

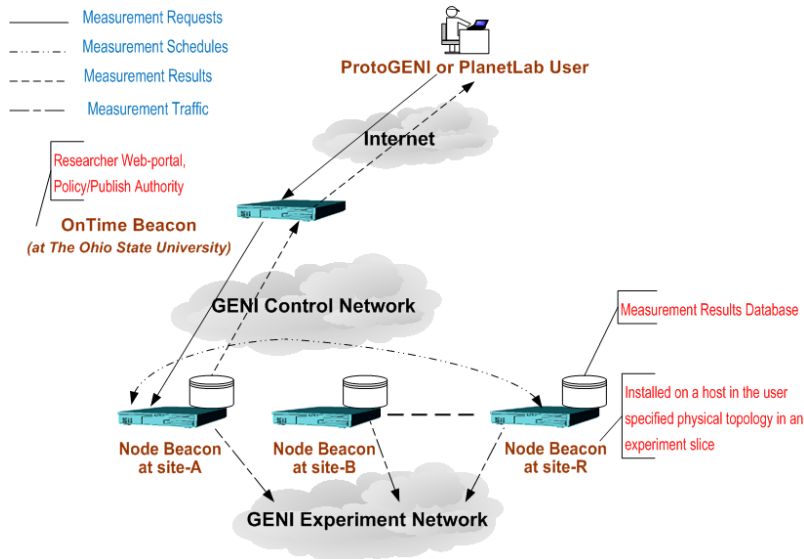


Figure 6: Deployment Architecture for Distributed OnTimeMeasure Software

VII. Conclusion

In our high-level design presented in this document, we have made several assumptions of user requirements and system capabilities in GENI. The design will thus need to be refined as our and GENI software development progresses, and as early adopters in the GENI community continue to provide their measurement service requirements.

Acknowledgements

We would like to thank the following individuals who have helped in the high-level design review by providing valuable suggestions:

- Robert Ricci – University of Utah
- Chris Small – Indiana University
- Camille Davis – Indiana University
- Vicraj Thomas – GENI Project Office
- Thomas Bitterman – Ohio Supercomputer Center
- Mukundan Sridharan – The Ohio State University
- Amruta Joshi – The Ohio State University
- James Griffioen – University of Kentucky
- Hussamuddin Nasir – University of Kentucky

References

- [1] ProtoGENI – <http://www.protogeni.net>
- [2] GMOC - <http://gmoc.grnoc.iu.edu>
- [3] ActiveMon - <http://www.oar.net/initiatives/itecoho/activemon>
- [4] P. Calyam, et. al., “Orchestrating Network-wide Active Measurements for Supporting Distributed Computing Applications”, IEEE Transactions on Computers Journal (TC), 2006.
- [5] P. Calyam, et. al., “Enhanced EDF Scheduling Algorithms for Orchestrating Network-wide Active Measurements”, IEEE Real-Time Systems Symposium (RTSS), 2005.

- [6] P. Calyam, P. Schopis, et. al., "Active and Passive Measurements on Campus, Regional and National Network Backbone Paths", IEEE International Conference on Computer Communications and Networks (ICCCN), 2005.
- [7] P. Calyam, et. al., "OnTimeMeasure: A Scalable Framework for scheduling active measurements", IEEE/IFIP End-to-End Monitoring Workshop (E2EMON), 2005.
- [8] P. Calyam, P. Schopis, et. al., "TBI: End-to-End Network Performance Measurement Testbed for Empirical-bottleneck Detection", IEEE Testbeds and Research Infrastructures for the Development of Networks and Communities (TRIDENTCOM), 2005.
- [9] ProtoGENI Backbone - <http://www.protogeni.net/trac/protogeni/wiki/Backbone>
- [10] ProtoGENI RSpec - <http://www.protogeni.net/trac/protogeni/wiki/RSpec>
- [11] ProtoGENI RSpec Extensions - <http://www.protogeni.net/trac/protogeni/wiki/RSpecExtensions2>
- [12] LAMP - <http://groups.geni.net/geni/wiki/LAMP>
- [13] OGF-NMWG - <http://nmwg.internet2.edu>
- [14] NetKarma - <http://groups.geni.net/geni/wiki/netKarma>
- [15] H. Mussman, "GENI Instrumentation and Measurement Architecture", *GENI project document ID: GENI-SE-IM-ARCH-0.1*, March 2010.
<http://groups.geni.net/geni/wiki/GeniInstrumentationandMeasurementsArchitecture>
- [16] PlanetLab - <http://groups.geni.net/geni/wiki/PlanetLab>
- [17] CRON Testbed - <http://groups.geni.net/geni/wiki/CRON>