DOE Award # DE-SC0001331:
Sampling Approaches for Multi-Domain Internet Performance Measurement

PI: Prasad Calyam, Ph.D.
pcalyam@osc.edu

Project Website: http://www.oar.net/initiatives/research/projects/multidomain_sampling

Progress Report

October 25, 2010

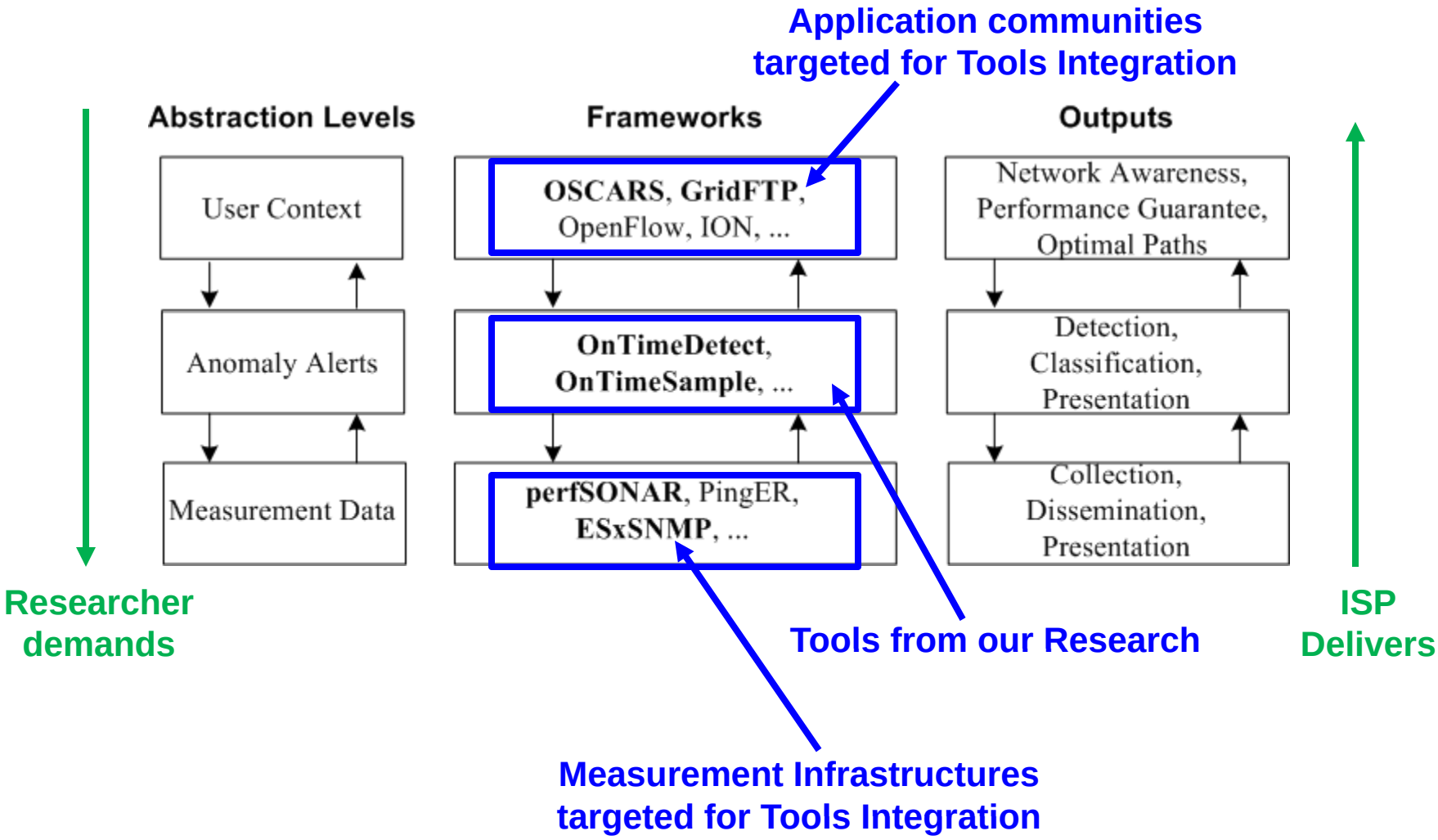
Topics of Discussion

- Project Overview
- Workplan Status
- Accomplishments
  - Part I: perfSONAR Deployments' Measurements Analysis
    - Major Activities, Results and Findings
  - Part II: Multi-domain Measurement Scheduling Algorithms
    - Major Activities, Results and Findings
  - Part III: Outreach and Collaborations
- Planned Next Steps

Project Overview

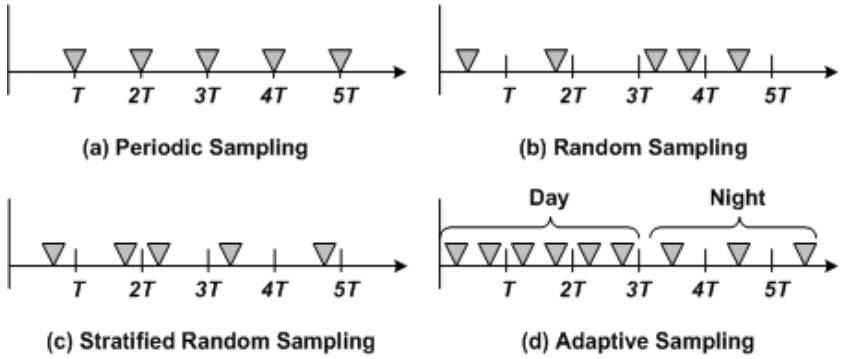
- DOE ASCR Network Research Grant
  - PI: Prasad Calyam, Ph.D.
  - Team: Mukundan Sridharan (Software Engineer), Lakshmi Kumaraswamy (Graduate Research Assistant), Pu Jialu (Undergraduate Research Assistant), Thomas Bitterman (Software Engineering Consultant)
- Goal: To develop *multi-domain* network status sampling techniques and tools to measure/analyze multi-layer performance
  - To be deployed on testbeds to support networking for DOE science
  - E.g., perfSONAR deployments for E-Center network monitoring, Tier-1 to Tier-2 LHC sites consuming data feeds from CERN (Tier-0)
- Collaborations: LBNL, FermiLab, Bucknell U., Internet2
- Expected Outcomes:
  - Enhanced scheduling algorithms and tools to sample multi-domain and multi-layer network status with active/passive measurements
  - Algorithms validation with measurement analysis tools for network weather forecasting, anomaly detection, fault-diagnosis

Context of our Research



Sampling and Analysis Requirements

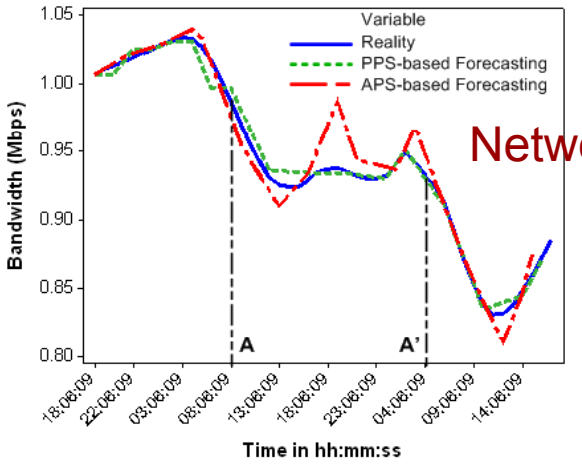
- Applications need *precisely timed* measurements across multiple network domains for analysis and consequent adaptation
  - Ongoing Measurement Requirement
    - Strict periodicity for network weather forecasting
    - Frequent random (e.g., poisson) sampling for anomaly detection
    - Stratified random sampling for routine network monitoring
    - Adaptive sampling to regulate probing bandwidth and measurement storage



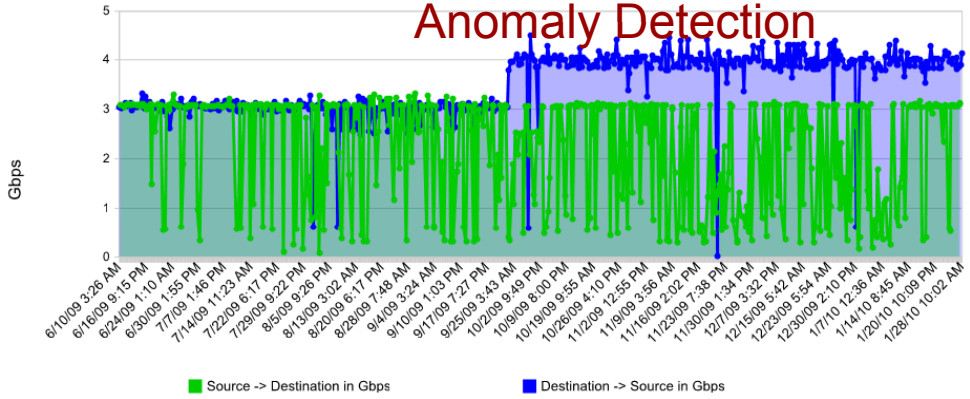
Sampling time interval pattern chosen depends on the monitoring accuracy objectives (i.e., pattern should produce least variance between actual and estimated)

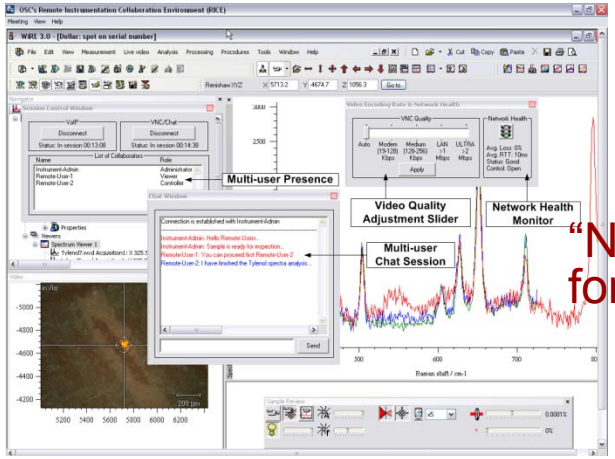
  - On-demand Measurement Requirement
    - One-off measurements with quick response times for e.g., to traceback a DDoS attack in a network segment

Examples to show Inter-sampling timing needs



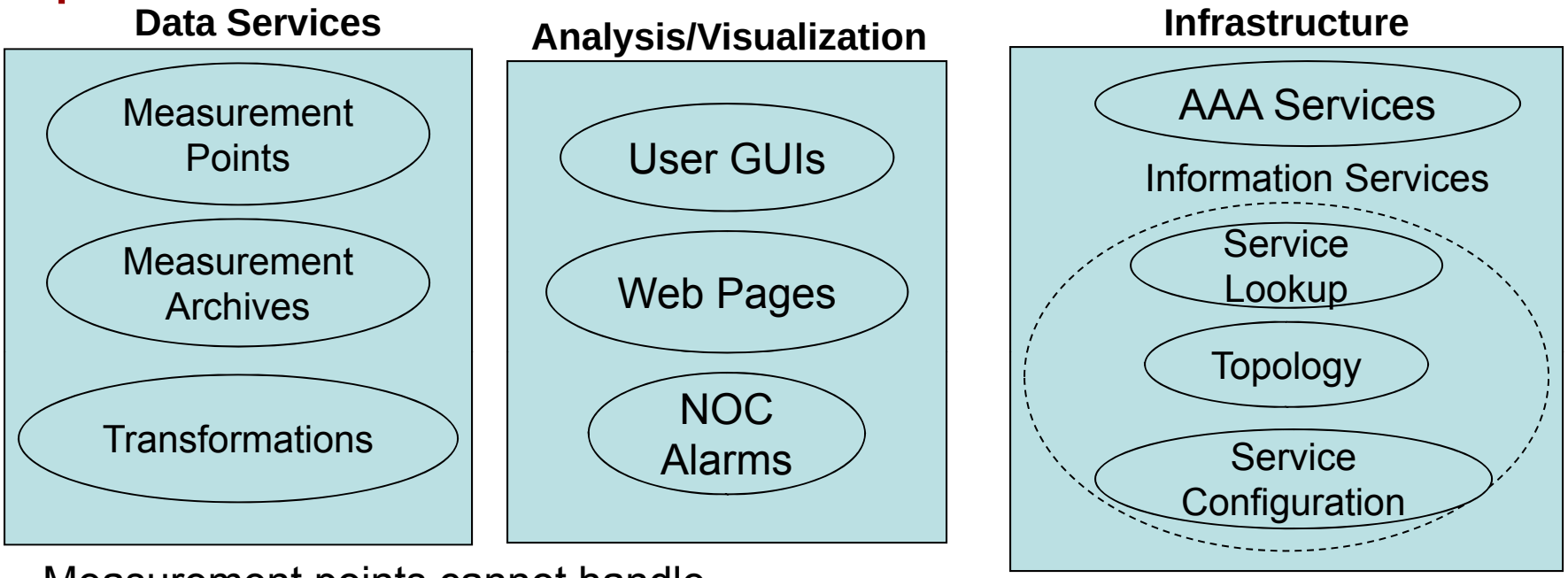
Network Weather Forecasting





"Network-awareness" for Real-time Control

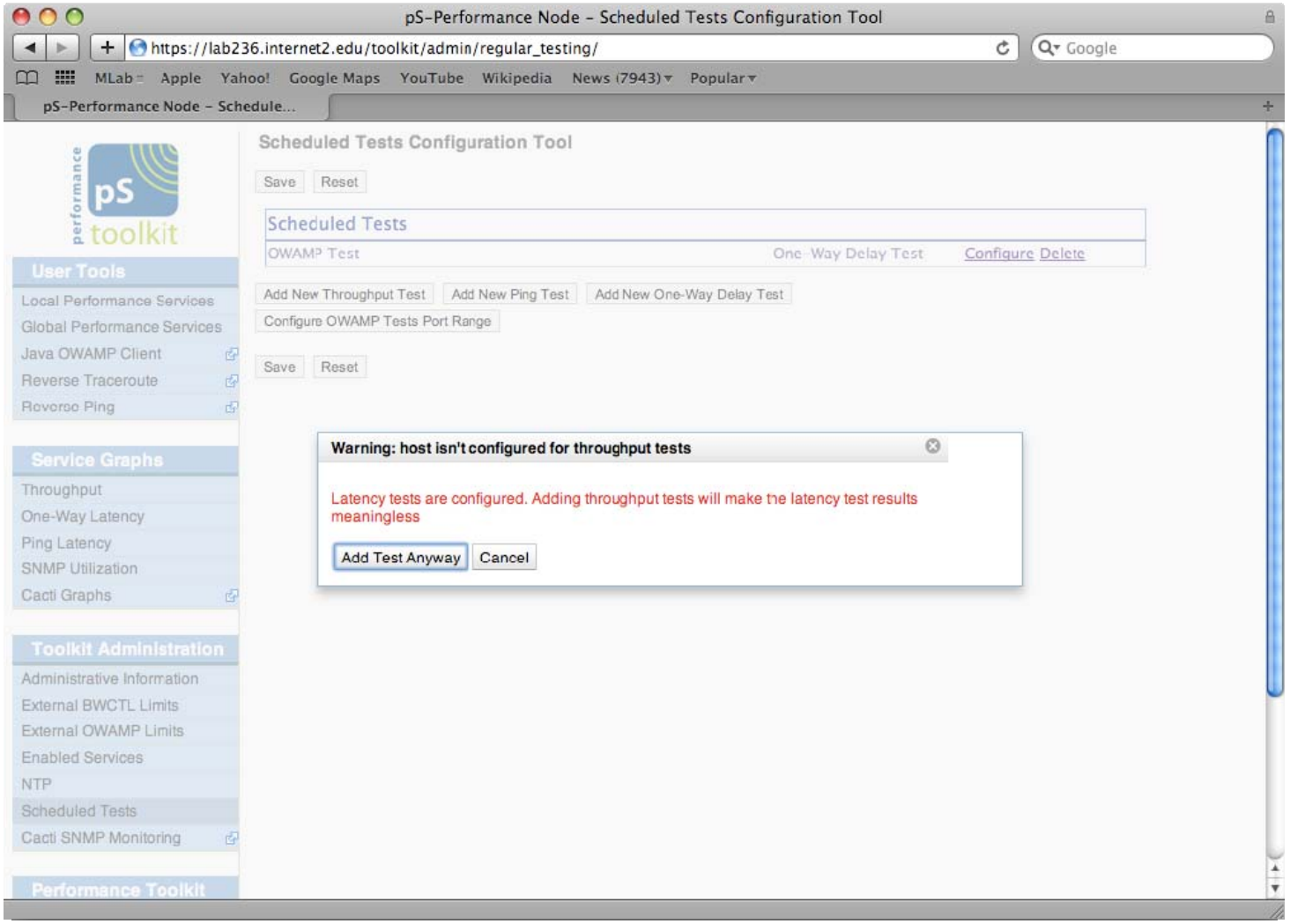
perfSONAR Architecture and some Limitations



- Measurement points cannot handle diverse sampling requirements
  - Full mesh periodic and best-effort on-demand measurements only
- Meta-scheduler to control measurement points is not developed
  - Current set of 3 tools (Ping, Traceroute, Iperf) will conflict if another tool is added (e.g., pchar)
  - Policies for regulation and semantic priorities cannot be enforced

- Measurement archives have large data sets but lack automated analysis techniques and tools
  - Anomaly detection and notification, weather forecasting, and automated fault diagnosis tools are needed along with easy-to-use GUIs
  - Integration with other measurement frameworks for important events correlation needs improvement

Measurement Conflict Resolution in perfSONAR



Workplan Status

Legend:
- Behind Schedule
- Planned for later
- In Progress
- Work finished

Phase		Timeline							
No.	Description	Qtr-1	Qtr-2	Qtr-3	Qtr-4	Qtr-5	Qtr-6	Qtr-7	Qtr-8
I	Investigate Technical and Policy Requirements								
II	Multi-domain Measurement Scheduling Algorithms								
III	Algorithms Validation with Measurements Analysis								
IV	Measurement Level Agreement Policies								
V	Measurement Framework Development								
VI	Measurement Framework Deployment & User Support								
VII	Outreach – talks, demos, papers								

Progress and Accomplishments Summary

- Conducted the “first” study to analyze worldwide perfSONAR measurements (480 paths, 65 sites) to detect network anomaly events
  - Developed an adaptive anomaly detection algorithm that is more accurate than existing static schemes (e.g., NLANR/SLAC plateau detector)
  - Demonstrated how a novel adaptive sampling scheme can reduce anomaly detection times from several days to only a few hours in perfSONAR deployments
  - Paper with results published in 2010 IEEE MASCOTS conference
- Released algorithms and toolkit for network anomaly notification to perfSONAR users/developers (http://ontimedetect.oar.net)
  - GUI tool and Command-line tools with web-interfaces developed
  - Tools have been developed to leverage perfSONAR web-service interfaces for BWCTL, OWAMP and SNMP measurements
  - Demonstrated need for “ground truth” correlation (e.g., NetAlmanac, logs) with detected network anomaly events in perfSONAR community
  - Receiving user feedback for additional features, analysis collaboration and integration into ESnet operations

Progress and Accomplishments Summary

- Developed semantic scheduling algorithms that will allow end-users (not just operators) at DOE Labs and collaborator sites to control measurement sampling in perfSONAR deployments
  - To resolve measurement resource contention when measurement requests exceed the amount of measurement resources:
    - (i) developed ontologies and an inference engine to prioritize measurement requests, and
    - (ii) weather forecasting based solver to lower priority of measurement requests that are oversampling
  - Developed a combined deterministic and heuristic scheduling algorithm to address sampling requirements for meeting monitoring objectives
  - Paper with initial results published in the 2010 IEEE CNSM conference
- Presented research findings and demos in major national and international conferences and workshops in 2010
  - Winter ESCC/Joint Techs, Summer ESCC/Joint Techs, perfSONAR workshop, IEEE MASCOTS, IEEE CNSM, Internet2 Spring Member meeting, (SC10)

Technical Challenges

Grey Area!

(in Multi-domain Measurement Federations)

- Intra-domain and Inter-domain measurement probes access
- Measurement conflicts avoidance
- Measurement request/response protocols
- Measurement sampling frequency guarantees
- Measurement orchestration flexibility (e.g., centralized and distributed)
- Data fusion of multi-metric/layer/timescale measurements
- Expert-systems for “network-aware” applications

Policy Challenges

Grey Area!

(in Multi-domain Measurement Federations)

- Measurement Level Agreements
  - Share topologies, allowed duration of a measurement, permissible bandwidth consumption for measurements, …
- Semantic Priorities
  - Some measurement requests have higher priority than others
- Authentication, Authorization, Accounting
  - Determine access control and privileges for users or other federation members submitting measurement requests
- Measurement Platform
  - Operating system, Hardware sampling resolution, TCP flavor for bandwidth measurement tests, fixed or auto buffers, …

Topics of Discussion

- Project Overview
- Workplan Status
- Accomplishments
  - Part I: perfSONAR Deployments' Measurements Analysis
    - Major Activities, Results and Findings
  - Part II: Multi-domain Measurement Scheduling Algorithms
    - Major Activities, Results and Findings
  - Part III: Outreach and Collaborations
- Planned Next Steps

PART - I
perfSONAR Deployments Measurement Analysis

- Activity:
  - Evaluated a network performance “plateau-detector” algorithm used in existing large-scale measurement infrastructures (e.g., NLANR AMP, SLAC IEPM-BW)
  - Analyzed anomaly detection performance for both synthetic and ESnet perfSONAR measurements data, identified limitations in existing implementations’ “sensitivity” and “trigger elevation” configurations
  - Developed “OnTimeDetect” v0.1 GUI and command-line tools based on evaluations
- Significance:
  - perfSONAR data web-service users need automated techniques and intuitive tools to analyze anomalies in real-time and offline manner
  - Network anomaly detectors should produce minimum false alarms and detect bottleneck events quickly
- Findings:
  - Nature of network performance plateaus that affect sensitivity and trigger elevation levels for low false alarms
  - Dynamic scheme for “sensitivity” and “trigger elevation” configuration based on the statistical properties of historic and current measurement samples

Topics of Discussion

- **Related Work**
- Plateau Anomaly Detection
- Adaptive Plateau-Detector Scheme
- OnTimeDetect Tool
- Performance Evaluation
- Conclusions

Related Work

- Recent network anomaly detection studies utilize various statistical and machine learning techniques
- User-defined thresholds are employed to detect and notify anomalies (e.g., Cricket SNMP)
  - Network path's inherent behavior is often not considered
- Mean ± Std Dev (MSD) methods in Guok et. al., calculate thresholds via moving window summary measurements
  - Not robust to outliers

Related Work (2)

- Soule et. al., created traffic matrix of all links in an enterprise and used a Kalman-filter based anomaly detection scheme
- Several related studies use machine learning techniques for unsupervised anomaly detection (Thottan et. al.,)
- Plateau-detector algorithm (McGregor et. al.,) used in the predecessors (NLANR AMP/SLAC pingER) effectively
  - Widely-used due to simplicity in the statistics involved
  - Easy to configure and interpret for network operators
  - Has limitations for perfSONAR users
    - Static configurations of the salient threshold parameters such as sensitivity and trigger elevation
    - Embedded implementations that are not extensible for web-service users who can query perfSONAR-ized measurement data sets

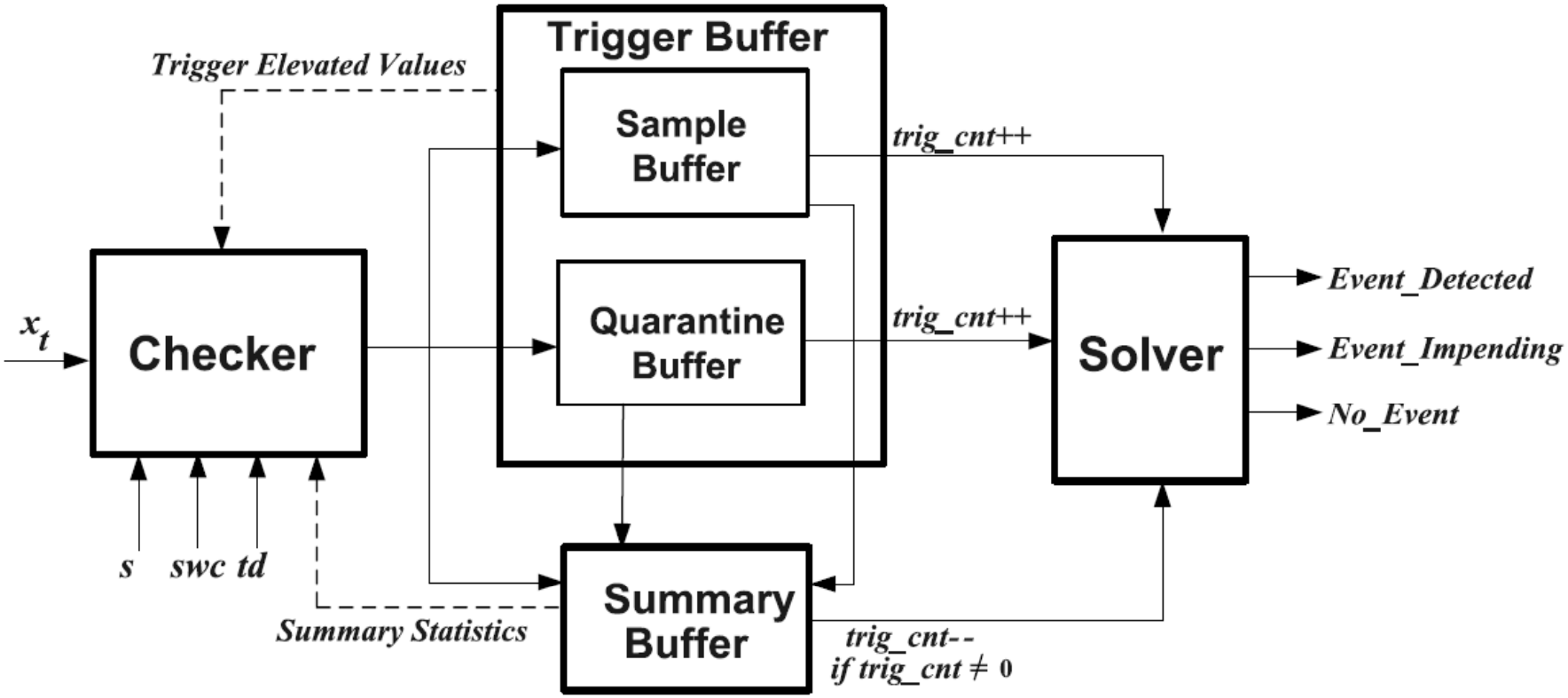
Topics of Discussion

- Related Work
- **Plateau Anomaly Detection**
- Adaptive Plateau-Detector Scheme
- OnTimeDetect Tool
- Performance Evaluation
- Conclusions

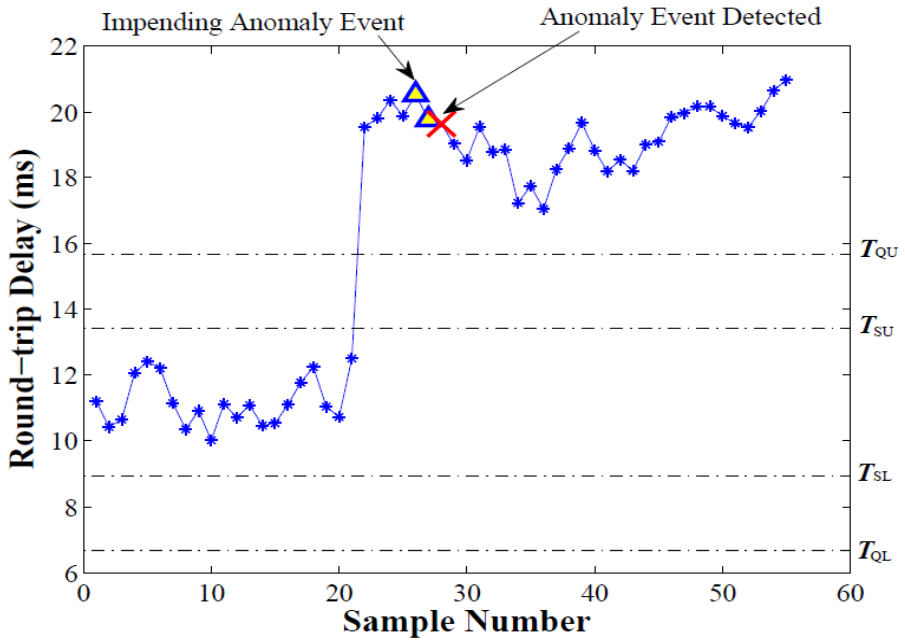
Plateau Anomaly Detection

- Enhanced mean ± standard deviation (MSD) algorithm
- Plateau detector uses two salient thresholds
  - Sensitivity (s) which specifies magnitude of plateau change that may result in anomaly
  - Trigger duration (t_d) specifies duration of the anomaly event before a trigger is signaled
- Network health norm is determined by calculating mean for a set of measurements sampled recently into “summary buffer”
  - The number of samples in “summary buffer” is user defined and is called summary window count (swc)

Plateau-Detector Block Diagram



Plateau-Detector Thresholds



ts(.)

$$
\begin{aligned}
T_{SU} &= \mu + s * \sigma \\
T_{QU} &= \mu + 2 * s * \sigma \\
T_{SL} &= \mu - s * \sigma \\
T_{QL} &= \mu - 2 * s * \sigma
\end{aligned}
$$

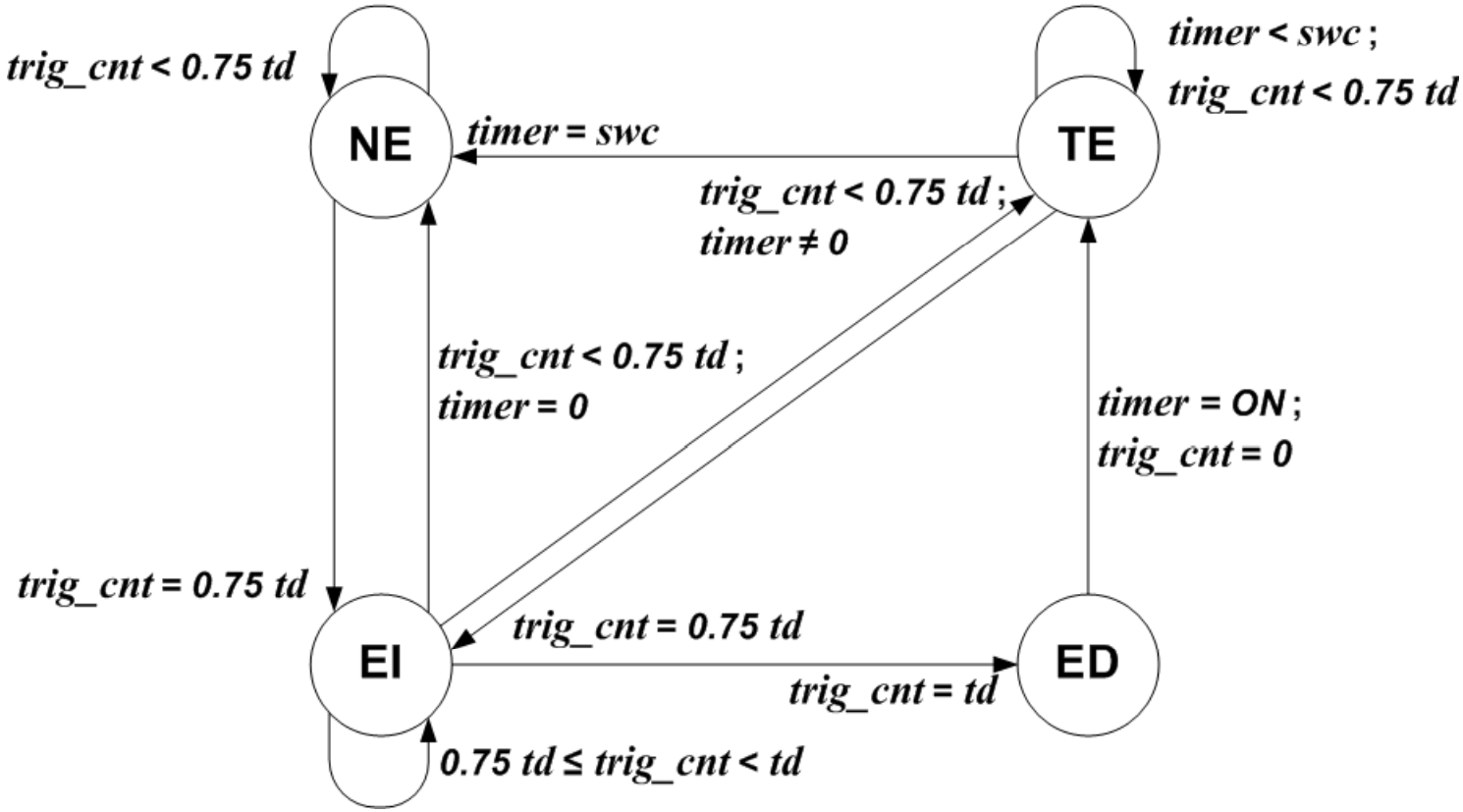
Upper and lower threshold limits

ts'(.)

$$
\begin{aligned}
T'_{SU} &= 1.2 * max(x_t) \text{ in } trigbuff \\
T'_{QU} &= 1.4 * max(x_t) \text{ in } trigbuff \\
T'_{SL} &= 0.8 * min(x_t) \text{ in } trigbuff \\
T'_{QL} &= 0.6 * min(x_t) \text{ in } trigbuff
\end{aligned}
$$

Upper and lower threshold limits in trigger elevated state

Plateau-Detector State Transitions



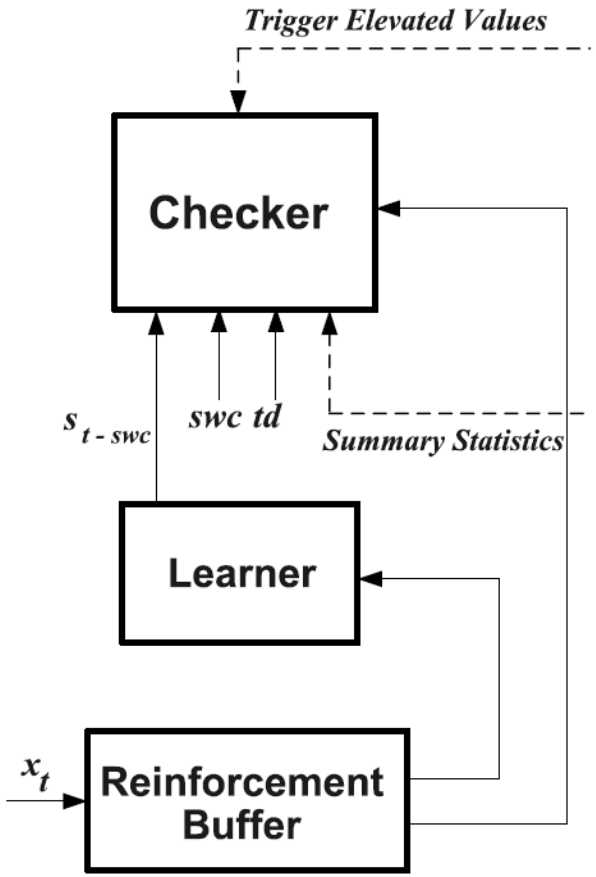
Topics of Discussion

- Related Work
- Plateau Anomaly Detection
- **Adaptive Plateau-Detector Scheme**
- OnTimeDetect Tool
- Performance Evaluation
- Conclusions

Need for Dynamic Plateau-detector Thresholds

- Minor differences in s and $ts'(.)$ parameters selection using "Static Plateau-Detector" (SPD) scheme greatly influence anomaly detection accuracy
  - Evidence from analysis of real and simulated traces
  - Increasing s form 2 to 3 reduces false positives but causes false negative
  - Increasing s to 4 minimizes false positives but false negatives remains
  - Static $ts'(.)$ settings do not detect consecutive anomalies of similar nature occurring within swc in trigger elevated state

Dynamic Threshold Parameters Selection



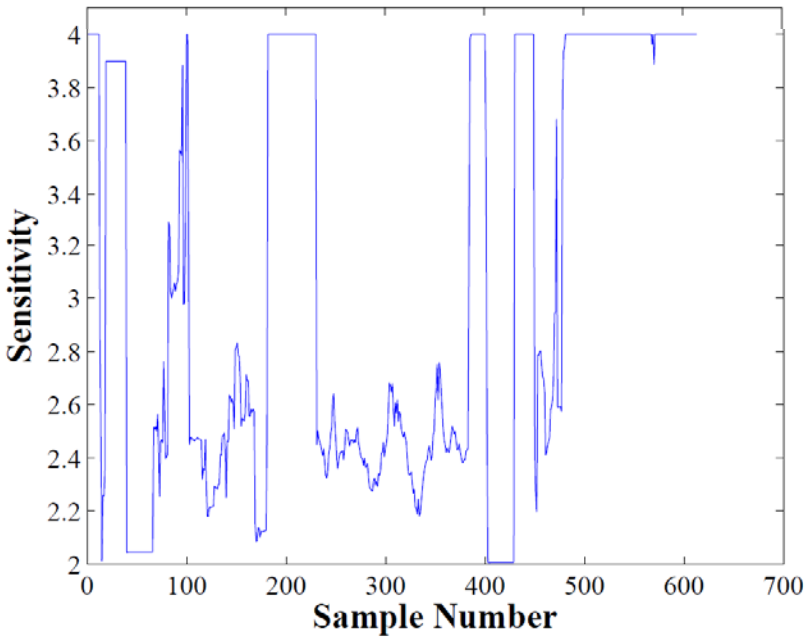
- Our goal is to avoid manual calibration of sensitivity s and trigger elevation $ts'(.)$ threshold parameters in the SPD scheme
- We apply reinforcement learning that guides the learning process for anomaly detection in our "Adaptive Plateau-Detector" (APD) scheme
  - APD scheme is based on a study of anomaly events in real and synthetic measurement traces, and derived closed-form expressions
  - APD scheme achieves low false alarm rates at the cost of a fractional increase in online detection time for the reinforcement learning

Dynamic Sensitivity Selection

- "Ground truth" challenge – difficult to decide what kind of events are to be notified as "anomaly events"
  - Plateau anomalies – they could affect e.g., data transfer speeds
  - The events we mark as anomalies are based on:
    - Our own experience as network operators
    - Discussions with other network operators supporting HPC communities (e.g., ESnet, Internet2)
- From our study of anomaly events in real and synthetic measurement traffic:
  - We observed that false alarms are due to persistent variations in time series after an anomaly event is detected
  - We concluded that leveraging variance of raw measurements just after an anomaly event for reinforcement learning makes anomaly detection more robust

Dynamic Sensitivity Selection (2)

- Sensitivity s needs to be re-evaluated at each time step
- We use $\frac{\sigma_f^2}{\sigma_c^2}$ relation in our APD scheme to determine sensitivity dynamically at a time step



Sensitivity variations when using APD Scheme for a trace

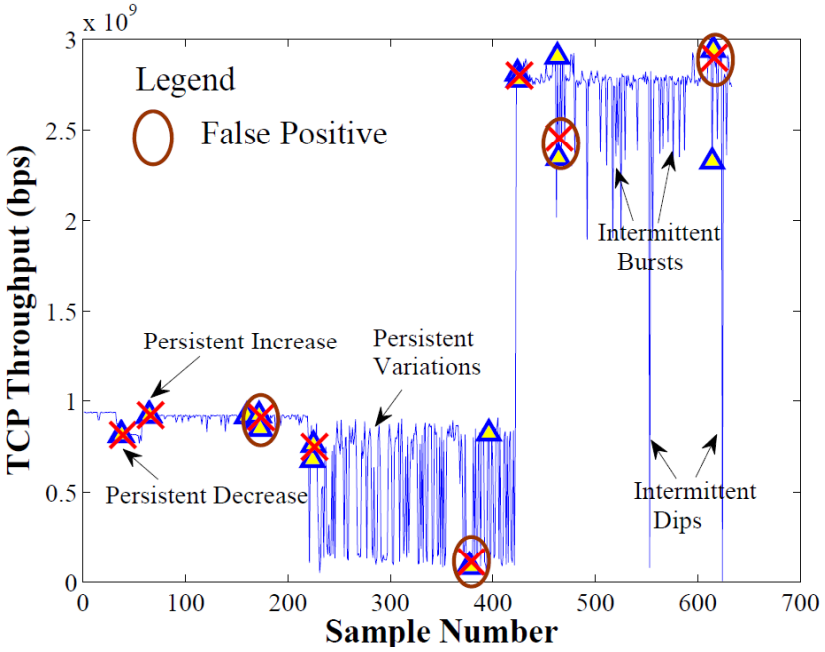
$$\mu_c = \frac{1}{swc}\sum_{i=1}^{swc} swd[i]$$

$$\sigma_c = \sqrt{\frac{1}{swc-1}\sum_{i=1}^{swc}(swd[i]-\mu_c)^2}$$

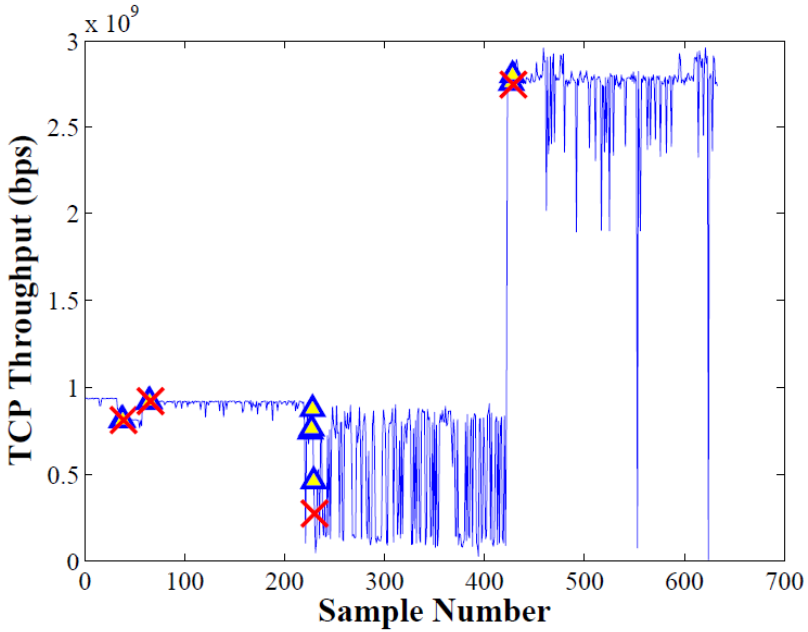
$$\mu_f = \frac{1}{swc}\sum_{i=1}^{swc} rbd[i]$$

$$\sigma_f = \sqrt{\frac{1}{swc-1}\sum_{i=1}^{swc}(rbd[i]-\mu_f)^2}$$

Anomaly Detection Comparison



Static Sensitivity in SPD Scheme



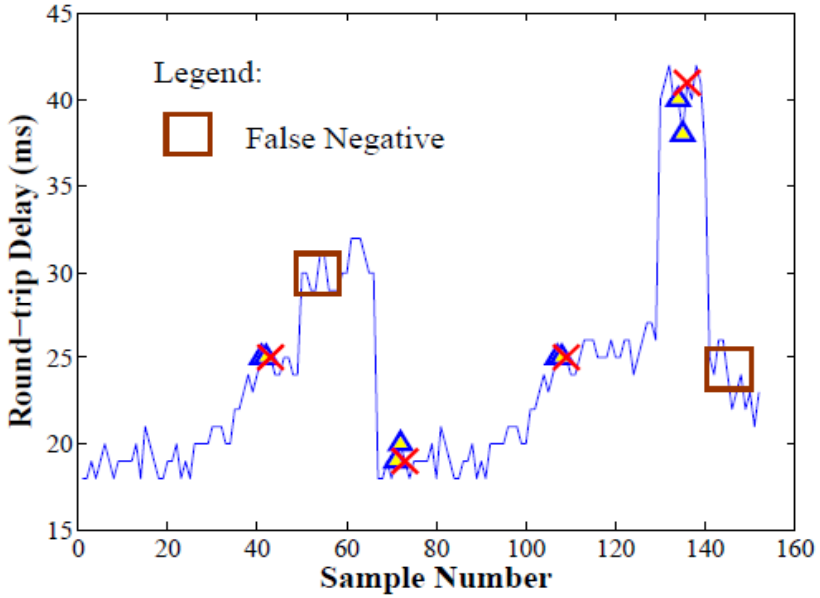
Dynamic Sensitivity in APD Scheme

Dynamic Trigger Elevation Selection

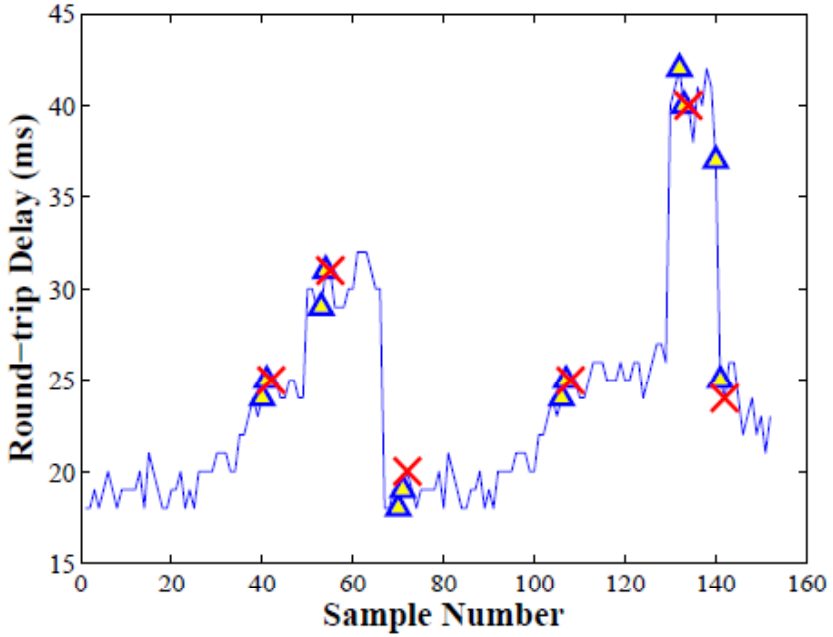
- Using static $ts'(.)$ settings based on max(x_t) and min(x_t) in SPD scheme resulted in false alarms
- x_d is the measurement sample arriving at the time instant when an anomaly event is detected (cross mark X annotation in graphs)
- Using x_d as network norm in trigger elevated state for calculating thresholds avoids false alarms

$$
\begin{aligned}
T'_{SU} &= x_d + s_t * \sigma_c \\
T'_{QU} &= x_d + 2 * s_t * \sigma_c \\
T'_{SL} &= x_d - s_t * \sigma_c \\
T'_{QL} &= x_d - 2 * s_t * \sigma_c
\end{aligned}
$$

Dynamic Trigger Elevation Comparison



Static Trigger Elevation in SPD Scheme



Dynamic Trigger Elevation in APD Scheme

Topics of Discussion

- Related Work
- Plateau Anomaly Detection
- Adaptive Plateau-Detector Scheme
- **OnTimeDetect Tool**
- Performance Evaluation
- Conclusions

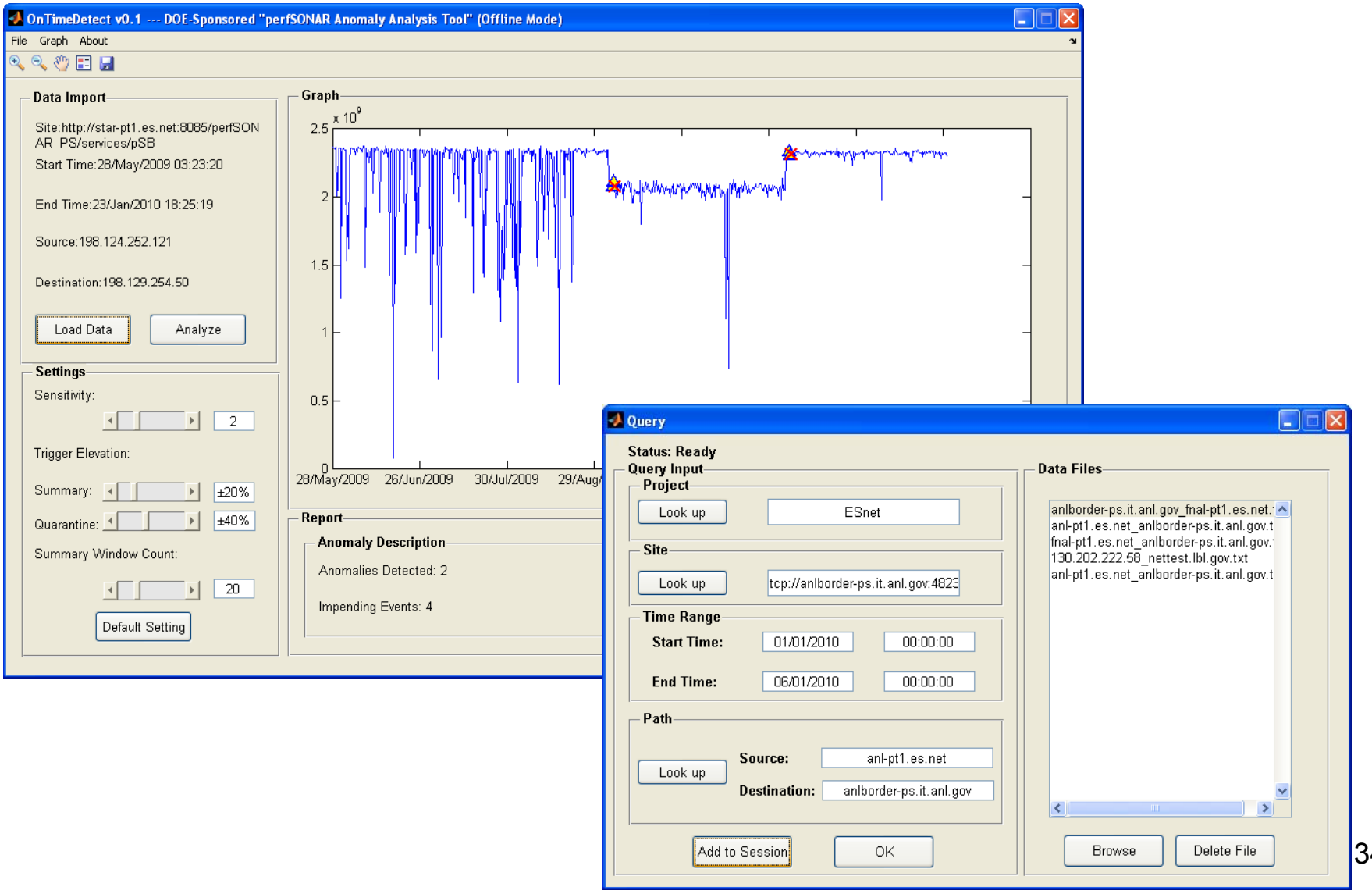
OnTimeDetect Tool Features

- GUI Tool (Windows/Linux) and Command-line Tool (Linux)
- Offline Mode
  - Query perfSONAR web-services based on projects, site lists, end-point pairs, and time ranges
    - BWCTL, OWAMP and SNMP data query and analysis capable
  - Drill-down analysis (Zoom-in/Zoom-out, Hand browse) of anomaly events in path traces at multi-resolution timescales
  - Modify plateau-detector settings to analyze anomalies
  - Save analysis sessions with anomaly annotated graphs
- Online Mode
  - Real-time anomaly monitoring for multiple sites
  - Web-interface for tracking anomaly events
    - Interactive web-interface, Twitter feeds, Monitoring Dashboard
- Software downloads, demos, manuals are at -

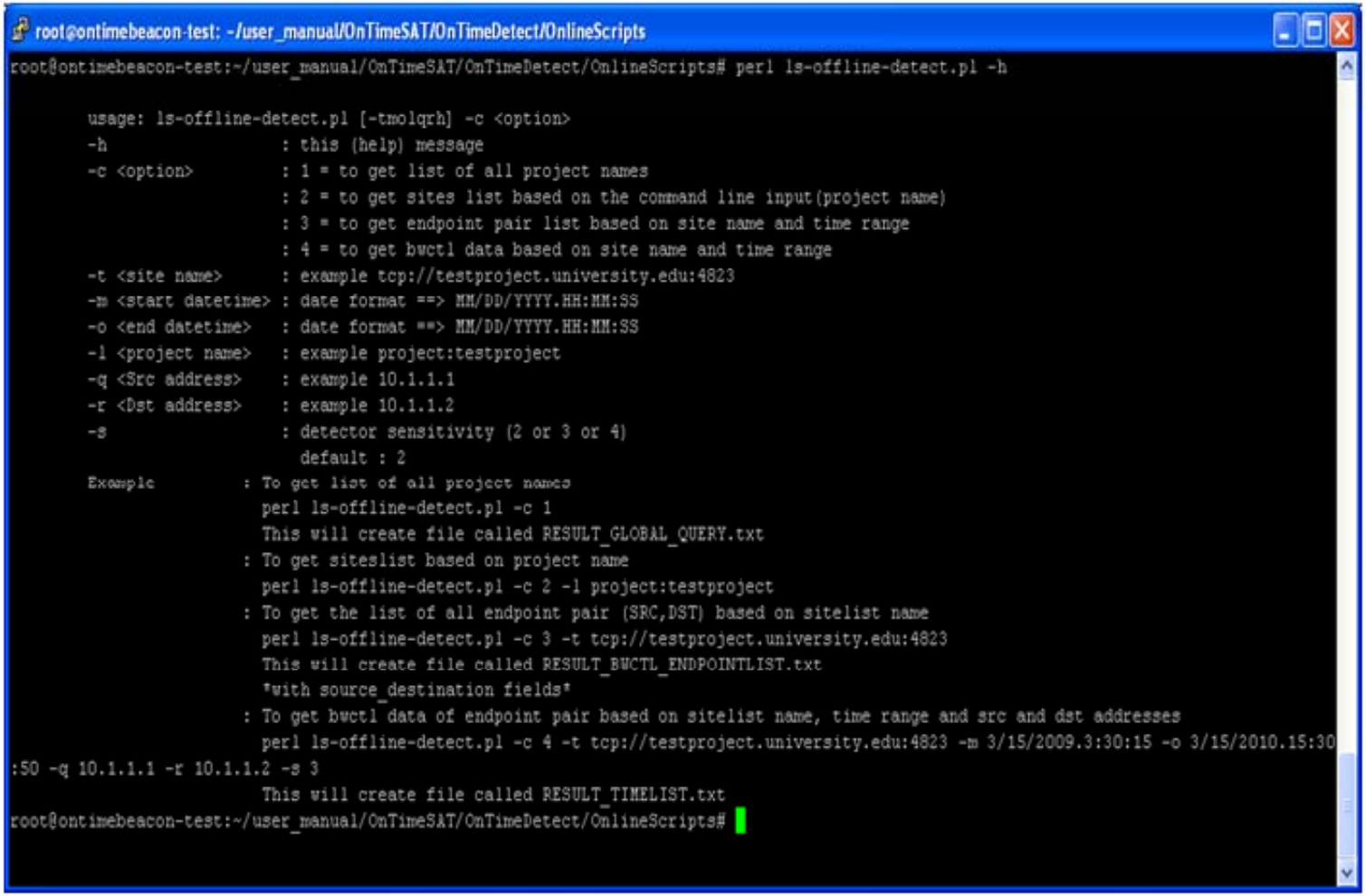
http://ontimedetect.oar.net

http://www.perfsonar.net/download.html

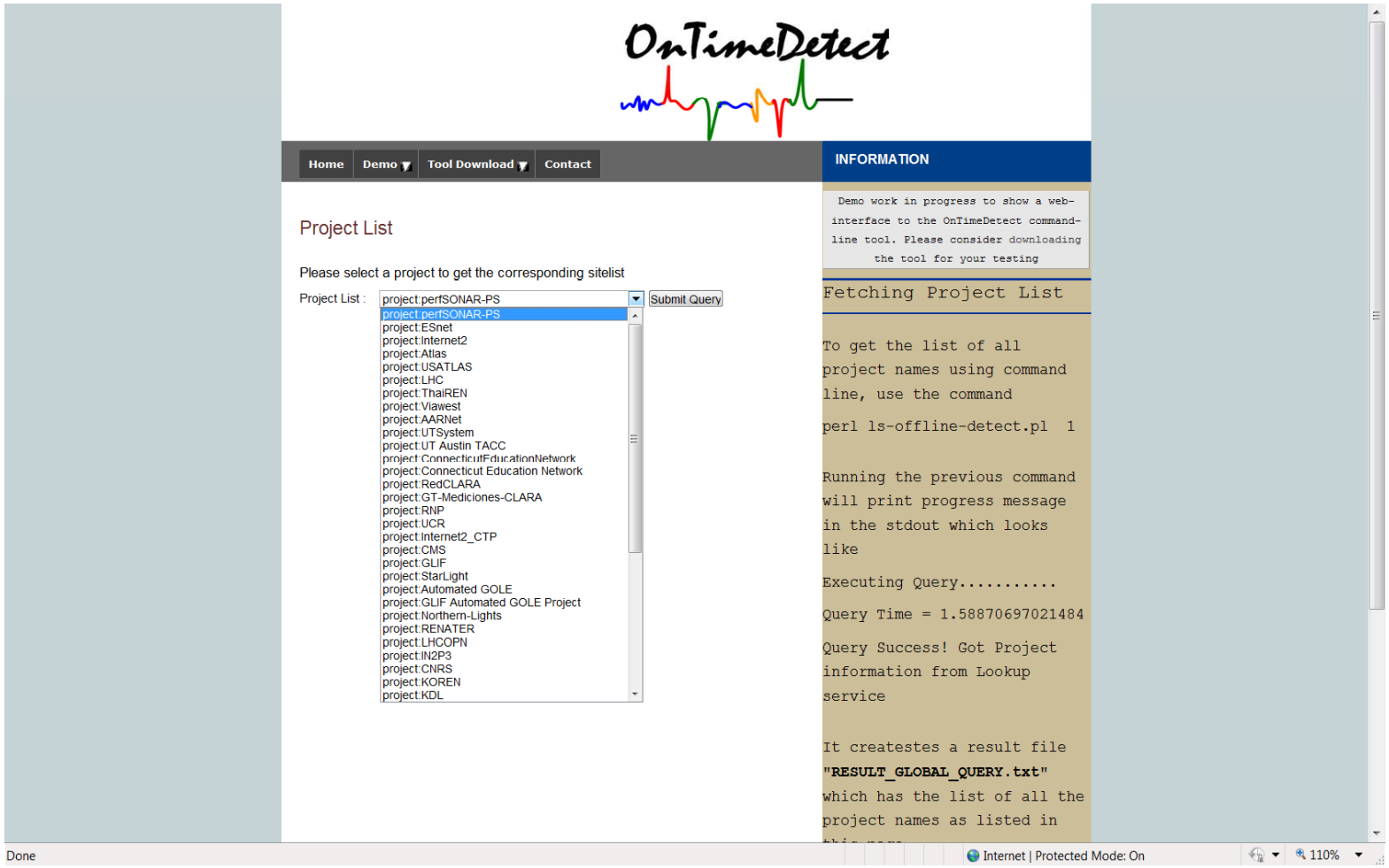
OnTimeDetect GUI Tool



OnTimeDetect Command-line Tool



Interactive OnTimeDetect Web-interface

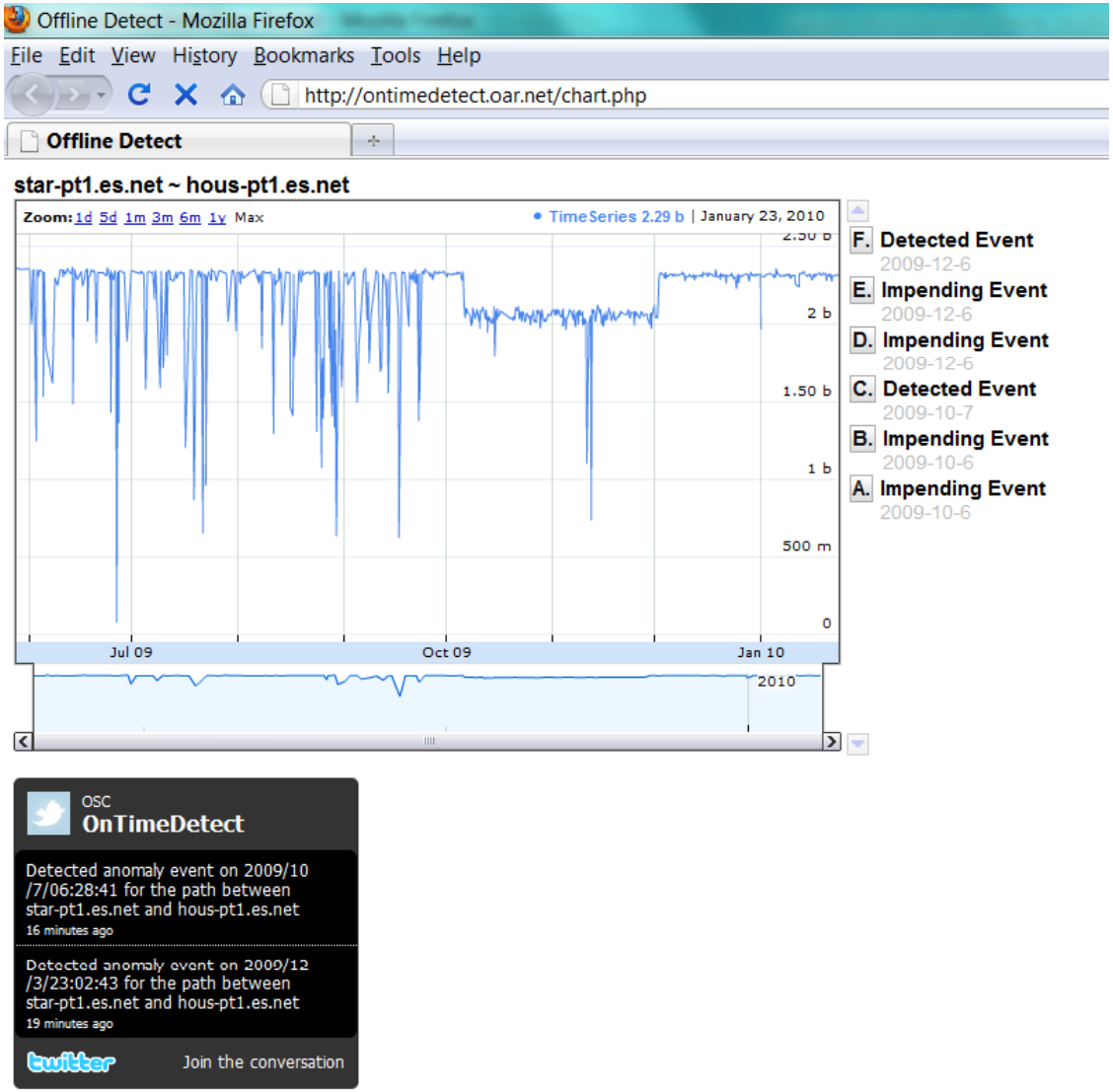


Interactive OnTimeDetect Web-interface (2)

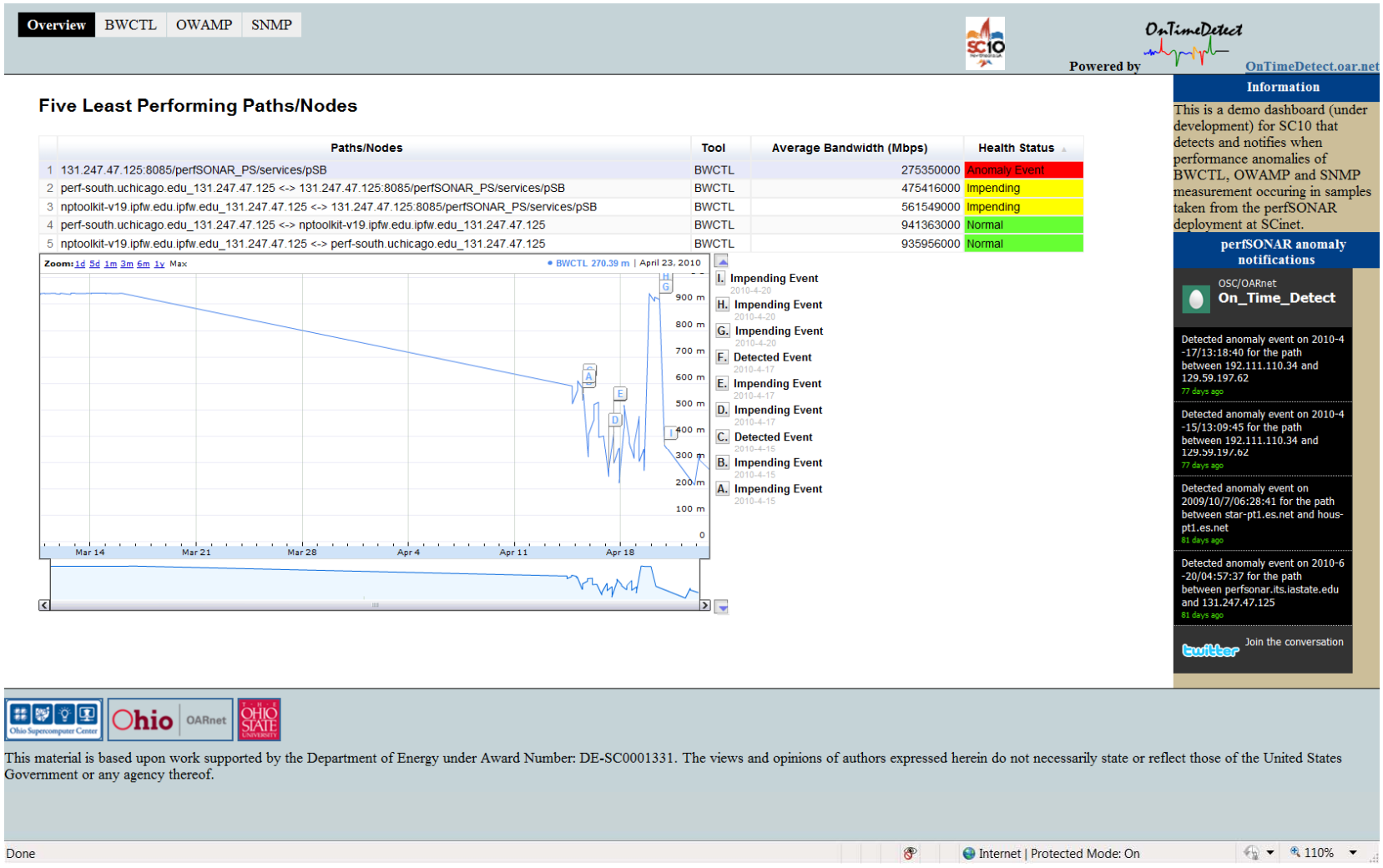
Interactive OnTimeDetect Web-interface (3)

Interactive OnTimeDetect Web-interface (4)

Interactive OnTimeDetect Web-interface (5)



SC10 SCinet Demo Dashboard
(Work in progress…)



Topics of Discussion

- Related Work
- Plateau Anomaly Detection
- Adaptive Plateau-Detector Scheme
- OnTimeDetect Tool
- **Performance Evaluation**
- Conclusions

Tool Deployment Experiences

- OnTimeDetect tool has been used to analyze BWCTL measurements from perfSONAR-enabled measurement archives at 65 sites

- Anomalies analyzed on 480 network paths connecting various HPC communities (i.e., universities, labs, HPC centers) over high-speed network backbones that include ESnet, Internet2, GEANT, CENIC, KREONET, LHCOPN, …

- Evaluation performed in terms of *accuracy*, *scalability* and *agility* of anomaly detection

Accuracy Evaluation Metrics

Success Ratio $$R_s = \frac{number\ of\ true\ triggers\ detected}{number\ of\ true\ triggers}$$

False Positive Ratio $$R_f+ = \frac{number\ of\ false\ triggers\ detected}{number\ of\ true\ triggers}$$

False Negative Ratio $$R_f- = \frac{number\ of\ true\ triggers\ missed}{number\ of\ true\ triggers}$$

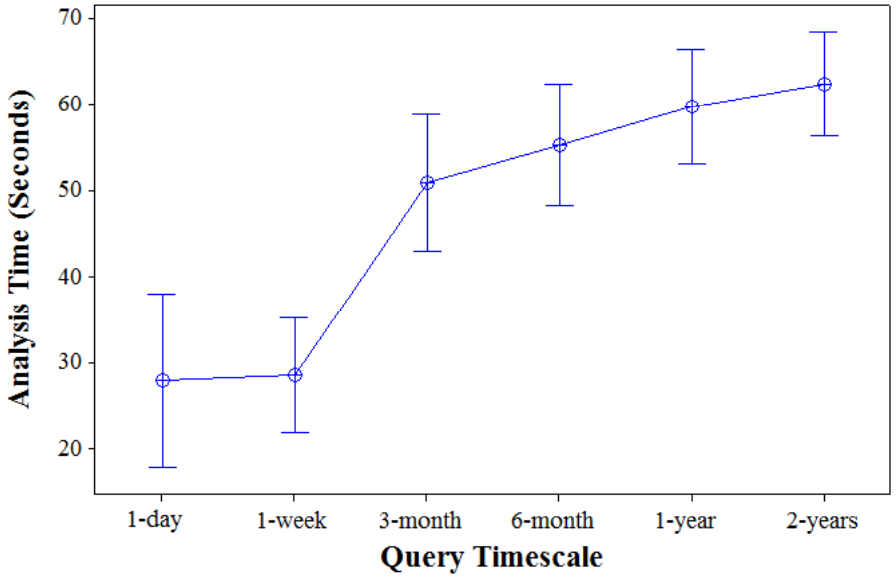
Employed Traces Description

Trace ID	Source ↔ Destination	Time Range (Start - End)	Time Series Characteristics
1	psmsu02.aglt2.org ↔ psum02.aglt2.org	2009-10-9 15:03:19 - 2010-4-7 17:28:05	Persistent Decrease, Burst Decrease, Intermittent Dips
2	bwctl.ucsc.edu ↔ bwctl.atla.net.internet2.edu	2010-1-16 06:51:22 - 2010-4-7 20:36:05	Persistent Decrease, Persistent Increase, Intermittent Dips
3	bwctl.ucsc.edu ↔ bwctl.wash.net.internet2.edu	2010-1-16 08:50:36 - 2010-4-7 20:37:43	Persistent Decrease, Persistent Increase, Intermittent Bursts, Intermittent Dips
4	wtg248.otctest.psu.edu ↔ perfsonar.dragon.maxgigapop.net	2010-2-8 14:08:31 - 2010-4-7 21:25:57	Persistent Variations
5	chic-pt1.es.net ↔ anl-pt1.es.net	2009-7-2 20:04:41 - 2010-1-9 12:32:48	Persistent Increase, Persistent Decrease, Persistent Variations
6	nersc-pt1.es.net ↔ wash-pt1.es.net	2009-5-18 22:48:13 - 2010-1-9 16:46:47	Persistent Increase, Intermittent Bursts, Intermittent Dips
7	hous-pt1.es.net ↔ pnwg-pt1.es.net	2009-5-19 04:05:12 - 2010-4-7 13:39:31	Persistent Increase, Persistent Variations, Intermittent Dips
8	nettest.boulder.noaa.gov ↔ wtg248.otctest.psu.edu	2009-10-6 20:41:22 - 2010-4-7 21:27:05	Persistent Decrease, Persistent Increase, Intermittent Bursts, Intermittent Dips

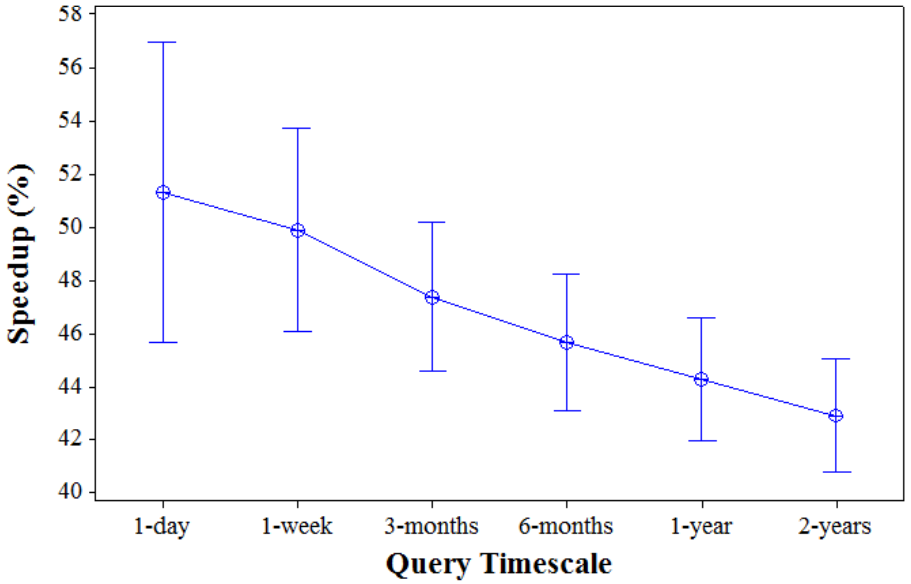
Accuracy Results from Traces

Trace ID	$SPD_{s=2}$			$SPD_{s=3}$			$SPD_{s=4}$			$APD_{s=2...4}$		
No.	R_s	R_{f+}	R_{f-}	R_s	R_{f+}	R_{f-}	R_s	R_{f+}	R_{f-}	R_s	R_{f+}	R_{f-}
1	1	0	**2**	1	0	1	1	0	0	1	0	0
2	1	0	**1.5**	**0.5**	**0.5**	**0.5**	**0.5**	**0.5**	0	1	0	0
3	1	0	0	**0.67**	**0.33**	**0.33**	1	0	0	1	0	0
4	**0.5**	**0.5**	**5**	1	0	0	1	0	0	1	0	0
5	1	0	**0.5**	1	0	0	**0.5**	**0.5**	0	1	0	0
6	**0**	0	**3**	1	0	**2**	1	0	0	1	0	0
7	1	0	**0.5**	**0.5**	**0.5**	0	**0.5**	**0.5**	0	1	0	0
8	1	0	**0.5**	1	0	0	1	0	0	1	0	0

Scalability Results

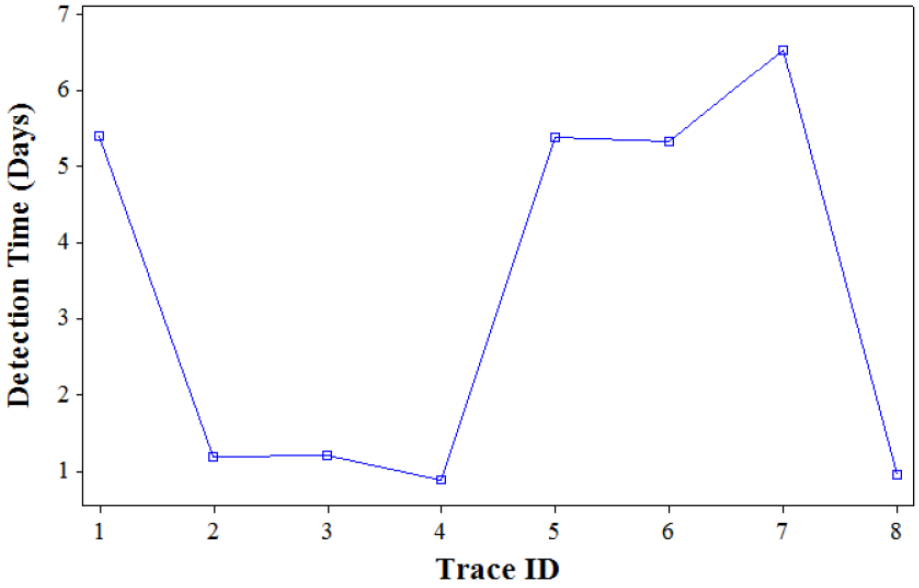


Sequential Query

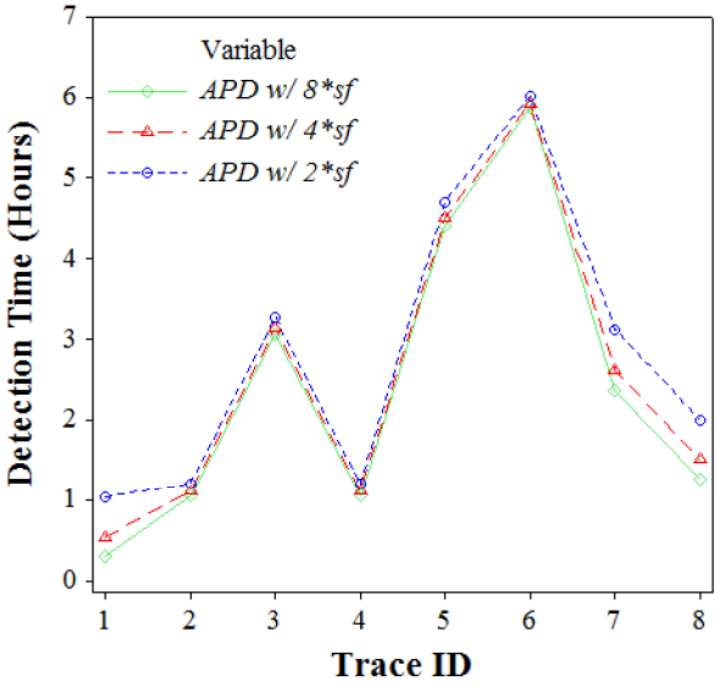


Parallel Query

Agility Results



Average Periodic Sampling



Adaptive Sampling

Conclusions

- Effort to extend the NLANR/SLAC implementations of a network performance “plateau-detector” for perfSONAR deployments
- Evaluated anomaly detection performance for both actual perfSONAR and synthetic measurement traces
- Developed a dynamic scheme for “sensitivity” and “trigger elevation” configuration based on the statistical properties of historic and current measurement samples
  - Produces low false alarm rate and can detect anomaly events rapidly when coupled with adaptive sampling
- Developed “OnTimeDetect” tool from evaluation experiences
  - Tool with APD scheme and intuitive usability features for detecting and notifying network anomalies for the perfSONAR community

Topics of Discussion

- Project Overview
- Workplan Status
- Accomplishments
  - Part I: perfSONAR Deployments' Measurements Analysis
    - Major Activities, Results and Findings
  - Part II: Multi-domain Measurement Scheduling Algorithms
    - Major Activities, Results and Findings
  - Part III: Outreach and Collaborations
- Planned Next Steps

PART - II
Multi-domain Measurement Scheduling Algorithms

- Activity:
  - Evaluated an offline Heuristic Bin Packing algorithm and Earliest Deadline First (EDF) based deterministic scheduling algorithm for scheduling active measurement tasks in large-scale network measurement infrastructures
  - Developed a combined deterministic and heuristic scheduling algorithm to address sampling requirements for meeting monitoring objectives
  - Analyzed scheduling output for various sampling time patterns (e.g., periodic, random, stratified random, adaptive) and comparing with monitoring objectives
- Significance:
  - Measurement schedulers should handle diverse sampling requirements of users to assist in their measurement analysis objectives
  - Efficient scheduling algorithms should allow more users (e.g., network operators, researchers) to sample network paths, handle semantic priorities and can also better support on-demand measurement sampling with rapid measurement response times
- Findings:
  - Effects of scheduling measurement tasks with mixtures of sampling pattern requirements – context of full-mesh, tree and hybrid topologies for increasing number of measurement servers, measurement tools and MLA bounds
  - Potential for tuning sampling frequency and limiting oversampling measurement requests using network weather forecasting

Measurements Provisioning Meta-scheduler

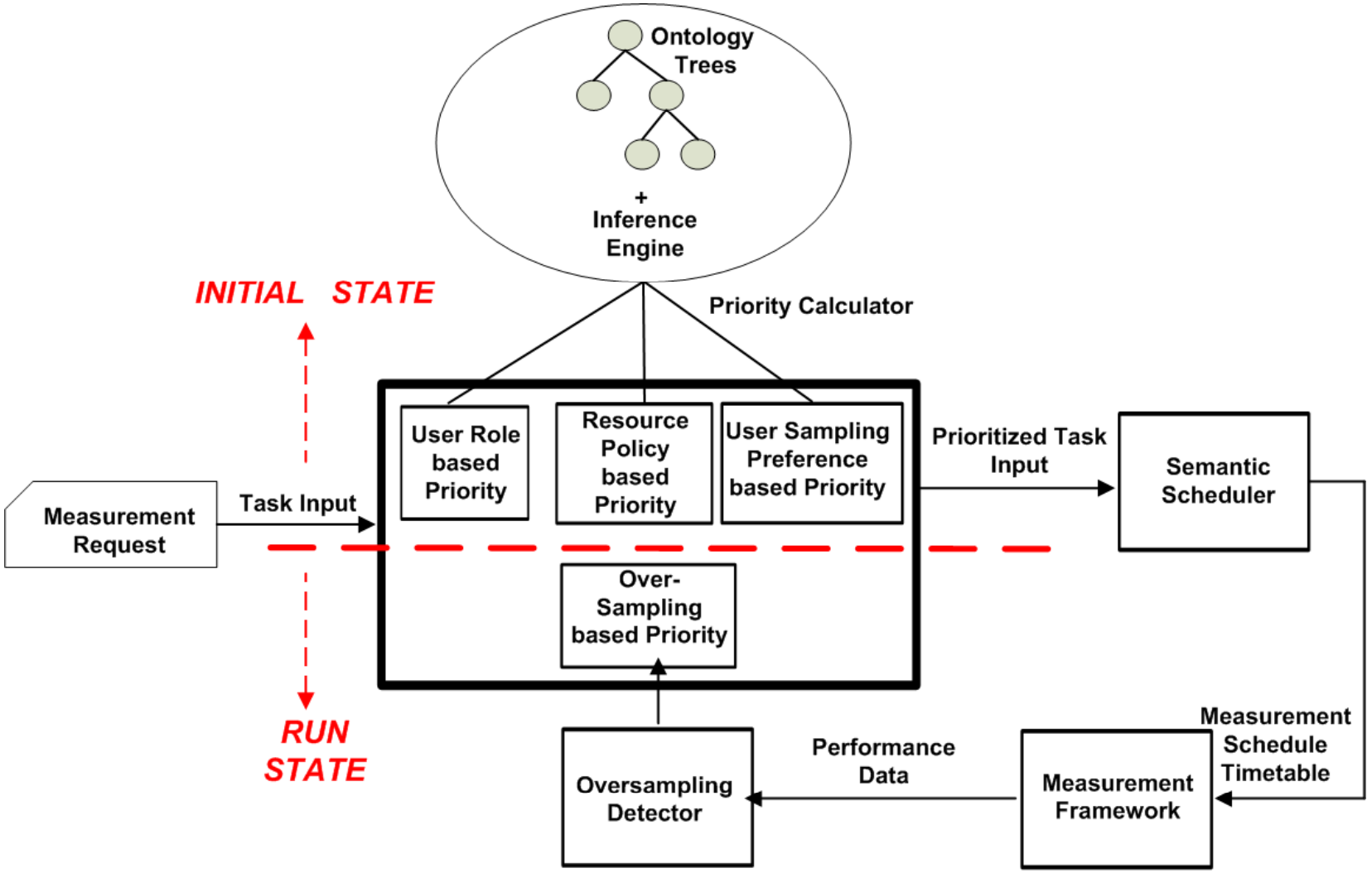
- Meta-scheduler for provisioning perfSONAR measurements
  - Benefit is that measurement collection can be targeted to meet network monitoring objectives of users (e.g., adaptive sampling)
  - Provides scalability to perfSONAR framework
    - If more tools are added, it allows for conflict-free measurements
    - On-demand measurement requests served with low response times
  - Can enforce multi-domain policies and semantic priorities
    - Measurement regulation; e.g., Only (1-5) % of probing traffic permitted
    - Intra-domain measurement requests may have higher priority, and should not be blocked by inter-domain requests
    - Measurement requests from users with higher credentials (e.g., backbone network engineer) may need higher priority than other users (e.g., casual perfSONAR experimenter)

Dynamic Predictor on ESnet perfSONAR Data

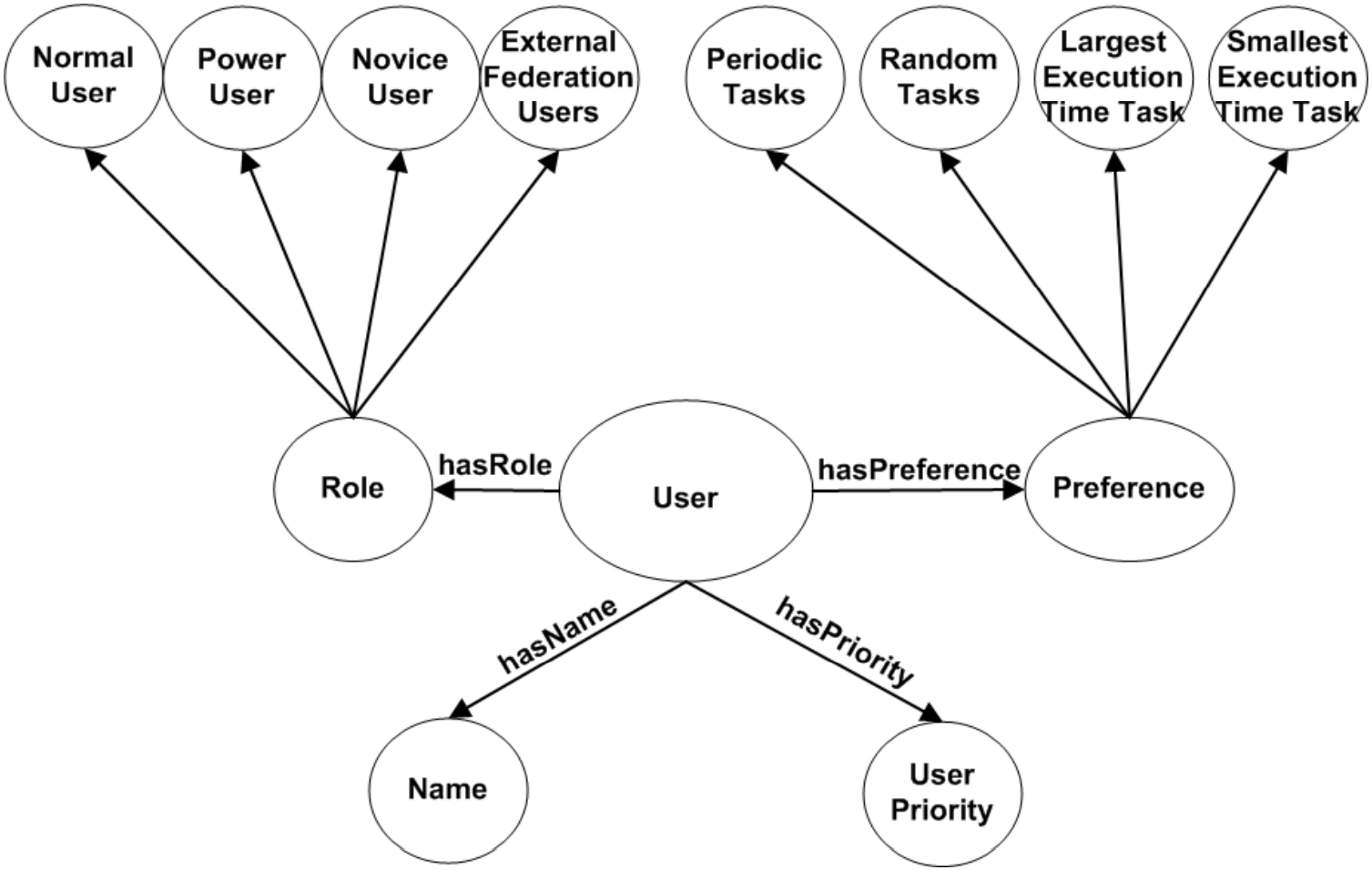
Trace ID	1	2	3	4	5	6	7	8	9	10
10% Exp Smooth	42	13	51	19	42	29	55	49	28	27
15% Exp Smooth	44	37	42	18	30	25	28	41	16	25
20% Exp Smooth	48	24	51	12	47	28	20	48	25	24
30% Exp Smooth	44	31	60	13	41	34	33	35	21	29
30% Trimmed Median Window 31	25	47	78	23	54	62	51	86	29	37
30% Trimmed Median Window 51	51	65	110	34	76	75	67	98	45	56
40% Exp Smooth	39	28	57	14	51	28	49	36	15	23
5% Exp Smooth	36	40	90	29	96	38	106	48	51	45
50% Exp Smooth	73	21	101	19	55	44	54	66	31	31
75% Exp Smooth	121	29	182	28	81	77	83	119	39	38
Adaptive Median Window 21	43	57	247	37	81	105	138	155	29	34
Adaptive Median Window 5	60	62	237	39	107	113	152	161	46	59
Last Value	342	176	513	94	204	172	350	362	93	114
Median Window 31	26	49	238	30	65	82	135	133	45	37
Median Window 5	65	87	276	35	74	107	163	195	53	61
Running Mean	133	48	334	26	77	127	180	122	37	25
Sliding Window Avg	91	179	319	54	201	139	147	213	84	95

Clear evidence of oversampling in perfSONAR deployments

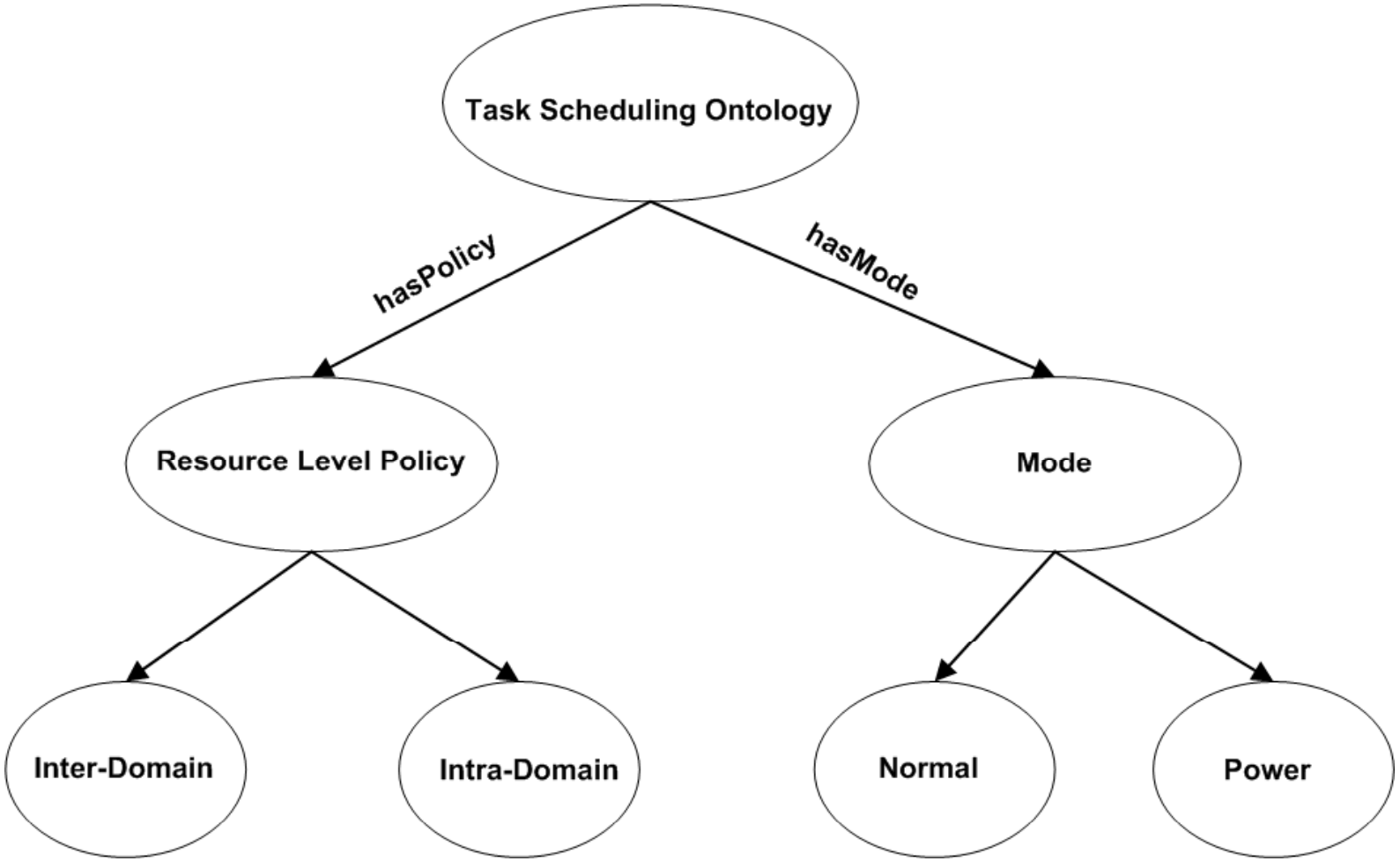
Ontology-based Semantic Meta-scheduler



Personal Ontology



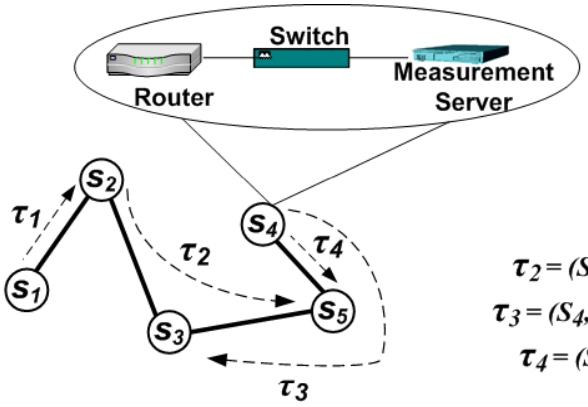
Task Scheduling Ontology



Basic Sampling Problem Overview

- Given:
  - $N = \{S_1, S_2, S_3, S_4, \ldots\}$ is the set of measurement servers
  - E is the set of edges between a pair of servers
  - $G = (N, E)$ measurement topology
  - $\zeta = \{\tau_1, \tau_2, \tau_3, \ldots, \tau_n\}$ corresponds to a measurement task set
  - ψ refers to a "Measurement Level Agreement" (MLA)
- Problem:
  - ***Offline Scheduling*** – For a G measurement topology, find the schedule of measurement jobs such that all deadlines (equal to periods) can be met for all tasks in ζ , while *maximizing concurrent execution,* but *preventing conflicts* and adhering to MLA constraint ψ
  - ***Online Scheduling*** - For an on-demand measurement request J_k, schedule it *as early as possible* without violating deadlines of tasks in ζ, but *preventing conflicts* and adhering to MLA constraint ψ

"Concurrent Execution" (CE) Principle



(a) Measurement Topology

$\tau_1 = (S_1, S_2, Iperf, 30, 10, Low)$

$\tau_2 = (S_2, S_5, Iperf, f_t(\text{AdaptivePeriodic}), 5, Medium)$

$\tau_3 = (S_4, S_3, Pathrate, f_t(\text{RandomPoisson}), 15, Medium)$

$\tau_4 = (S_4, S_5, Ping, f_t(\text{RandomExponential}), 5, Low)$

(b) Task Set

	Iperf	Iperf	Pathrate	Ping
Iperf	1	1	1	0
Iperf	1	1	1	0
Pathrate	1	1	1	0
Ping	0	0	0	0

(c) Tool Conflict Matrix

(d) Task Conflict Graph

- Construct a "Task Conflict Graph" based on a "Tool Conflict Matrix" obtained from empirical observations
- Concurrent execution decision during scheduling is based on "Task Conflict Graph" edges
  - Edge implies conflict exists!

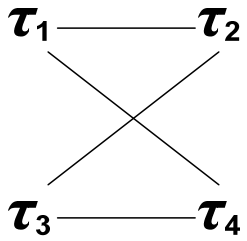
Offline Scheduling Algorithms

- Goal: To schedule on-going measurements maximizing concurrent execution
- Algorithms based on real-time systems scheduling principles; two preliminary algorithms we developed are:
  - Heuristic bin packing
    - Simple and effective for routine network monitoring, but is rigid to handle on-demand measurement requests
    - Causes job starvation problems
    - Provides schedule with deadline misses that is sometimes sufficient
  - Earliest Deadline First with CE (EDF-CE)
    - Caters measurement periodicity and flexible for on-demand measurements
    - Infeasible schedules for deadline misses

Heuristic Bin Packing Illustration

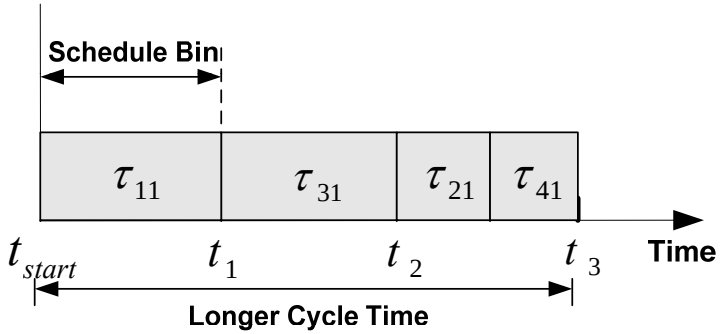
$\tau_1 = (S_1, S_2, Pathchar, 20)$
$\tau_2 = (S_2, S_3, Iperf, 10)$
$\tau_3 = (S_3, S_5, H.323Beacon, 20)$
$\tau_4 = (S_5, S_1, Iperf, 10)$

(a) Task Set

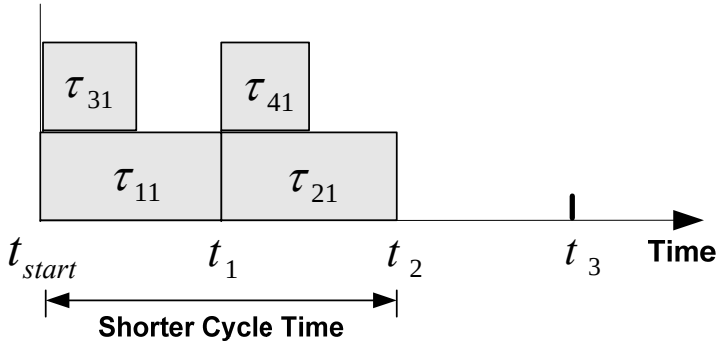


(b) Task Conflict Graph

(c) Round Robin Packing (RRP)



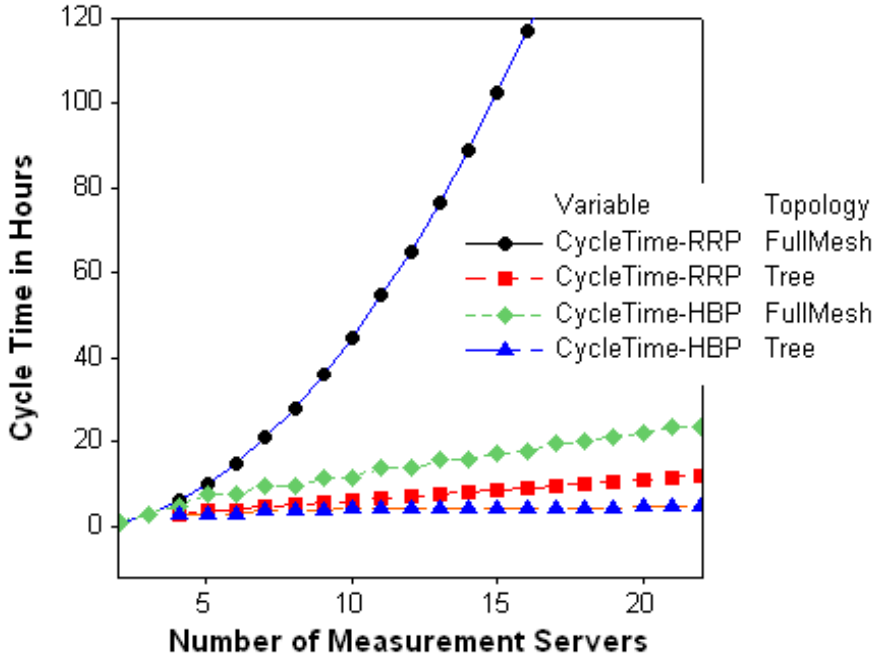
(d) Heuristic Bin Packing (HBP)



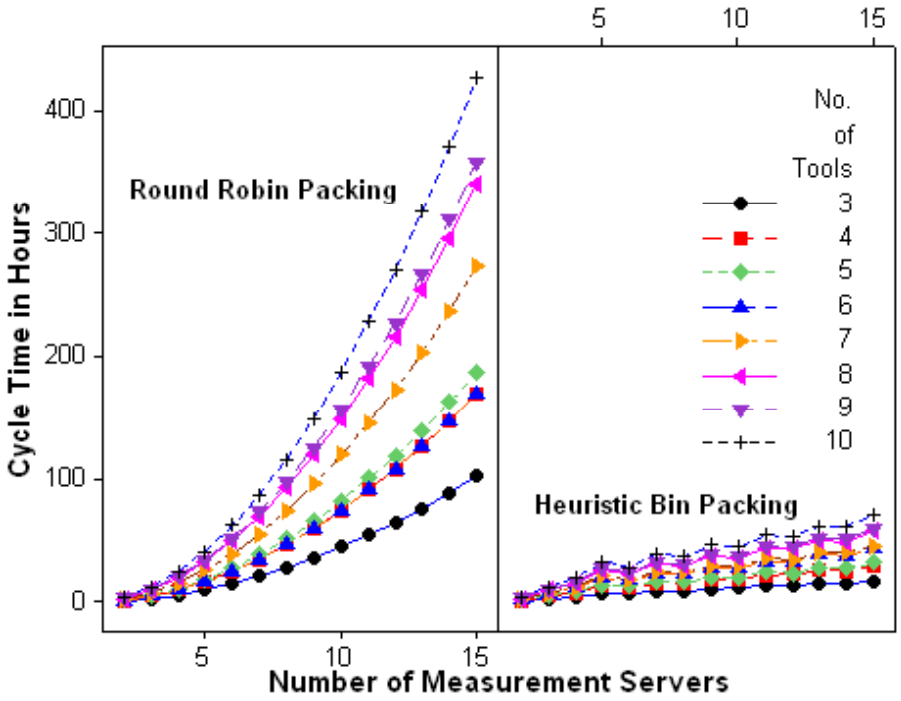
- Cycle time – measurement schedule completion time in NMI

Performance of Heuristic Bin Packing

- Synthetic task set simulation with 5 and 20 minute execution time jobs; fixed bin size of 20 minutes
- Measurement topologies: Full-mesh, Tree, Hybrid
- Significantly shorter cycle times for HBP compared to RRP
  - For any measurement topology, For any number of tools
- Developed an optimum bin size selection scheme that can improve cycle times and minimize “job starvation”

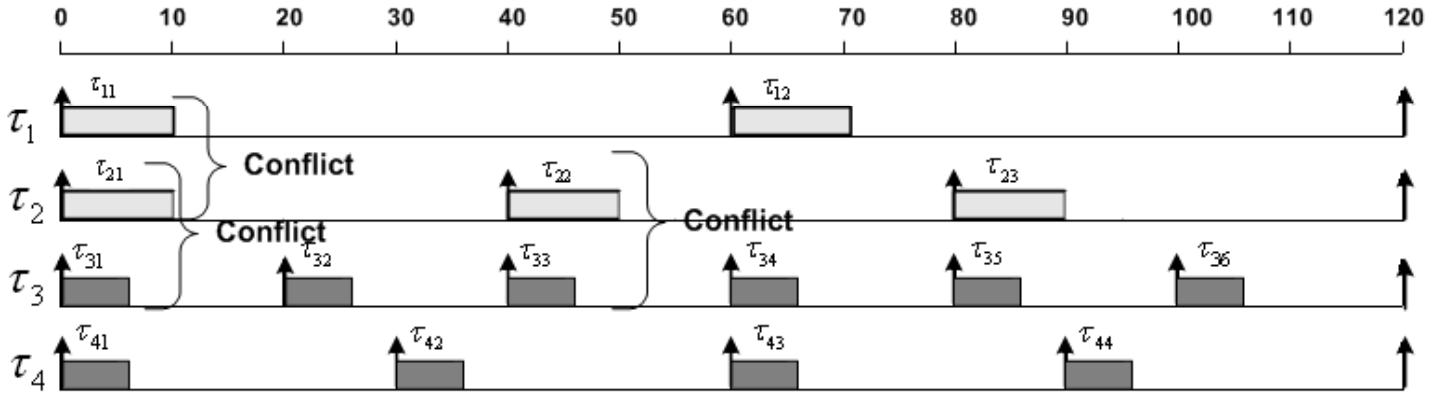


(a) Comparison with Round-robin for different measurement topologies

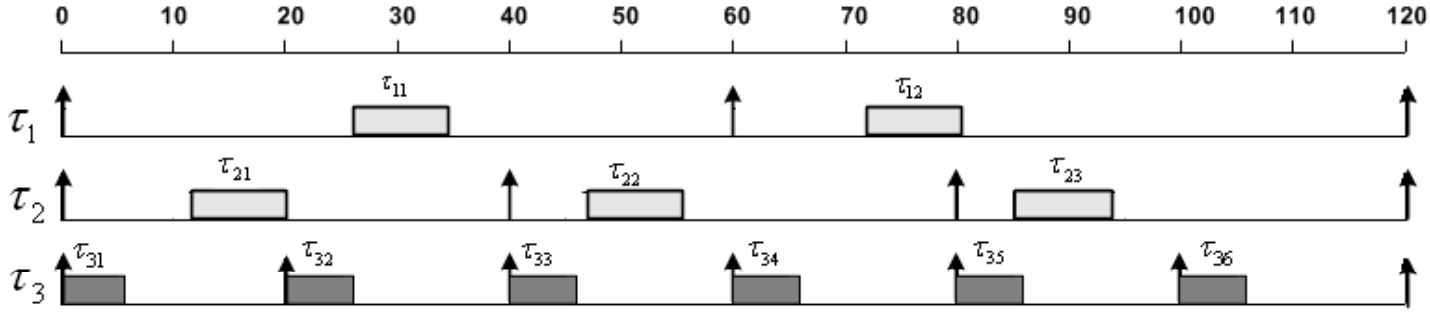


(b) Comparison with Round-robin for increasing number of tools

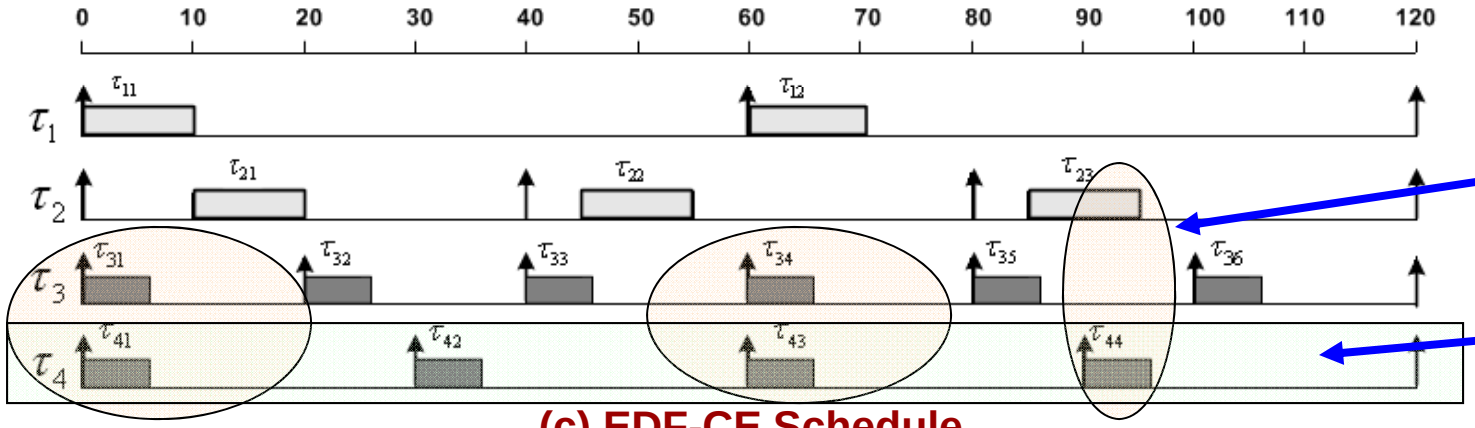
EDF-CE Illustration



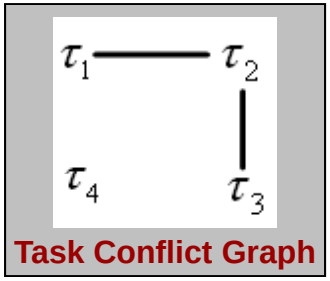
(a) No Orchestration



(b) Single-Processor-like EDF Schedule



(c) EDF-CE Schedule

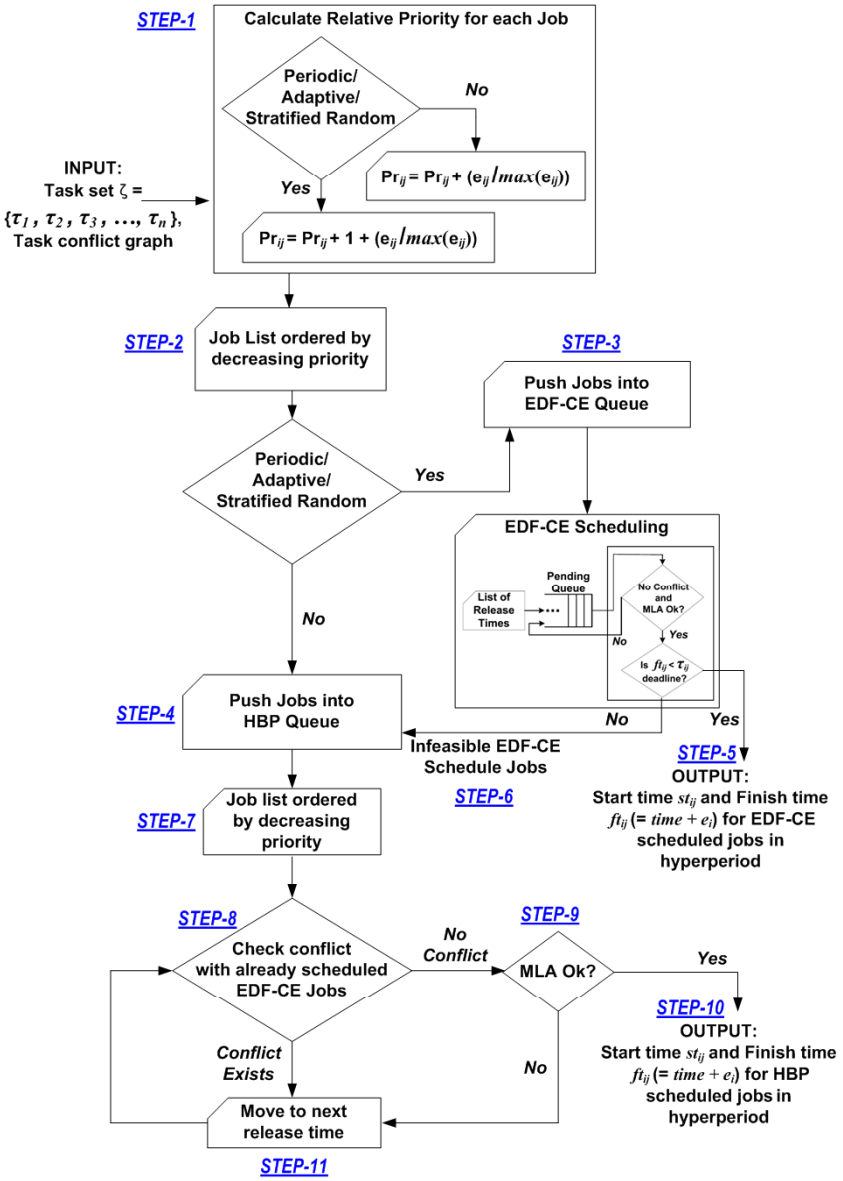


Task Conflict Graph

Higher schedulability due to CE

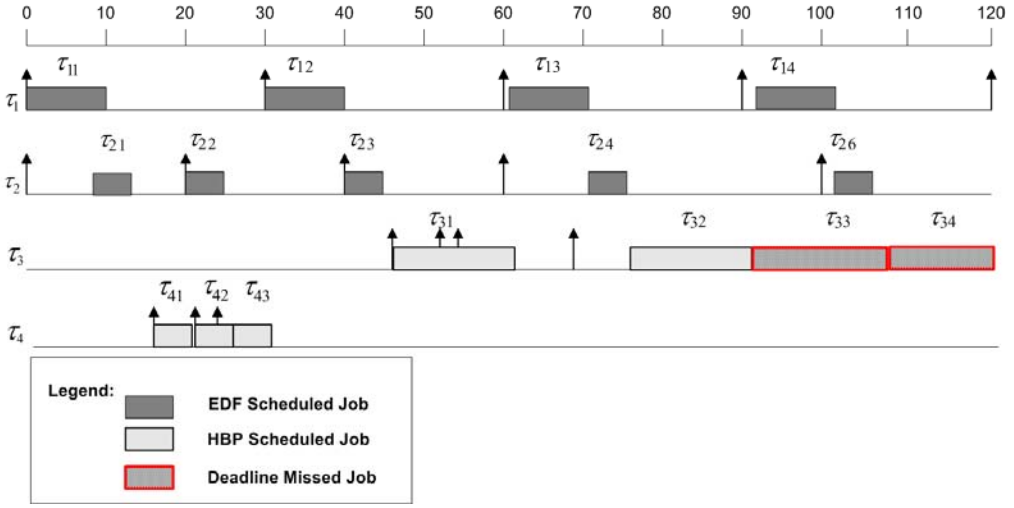
Can schedule an additional task

Semantic Scheduling Scheme

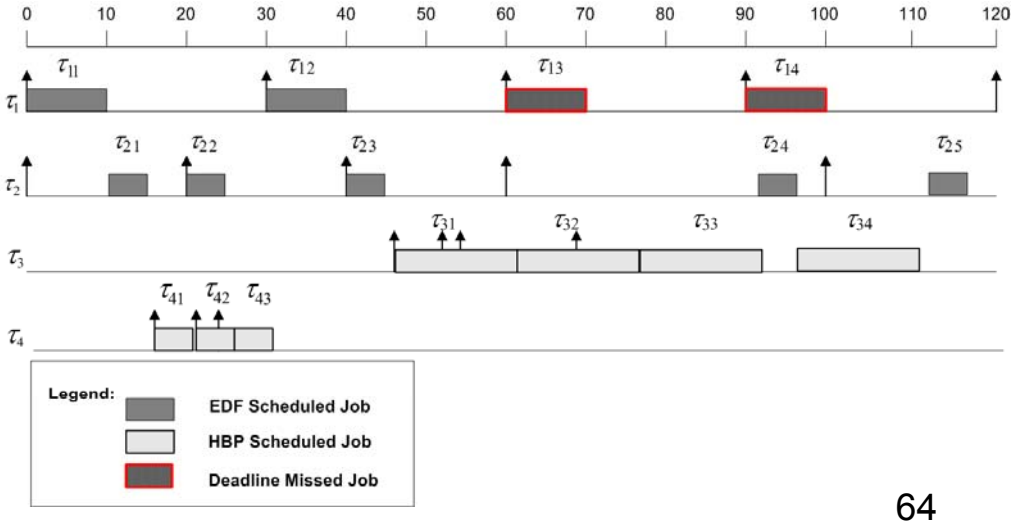


Semantic Grade Influence on Schedules

- *SPS-PE algorithm*: High execution time periodic tasks get higher priority, and have the best chance to be scheduled before deadline

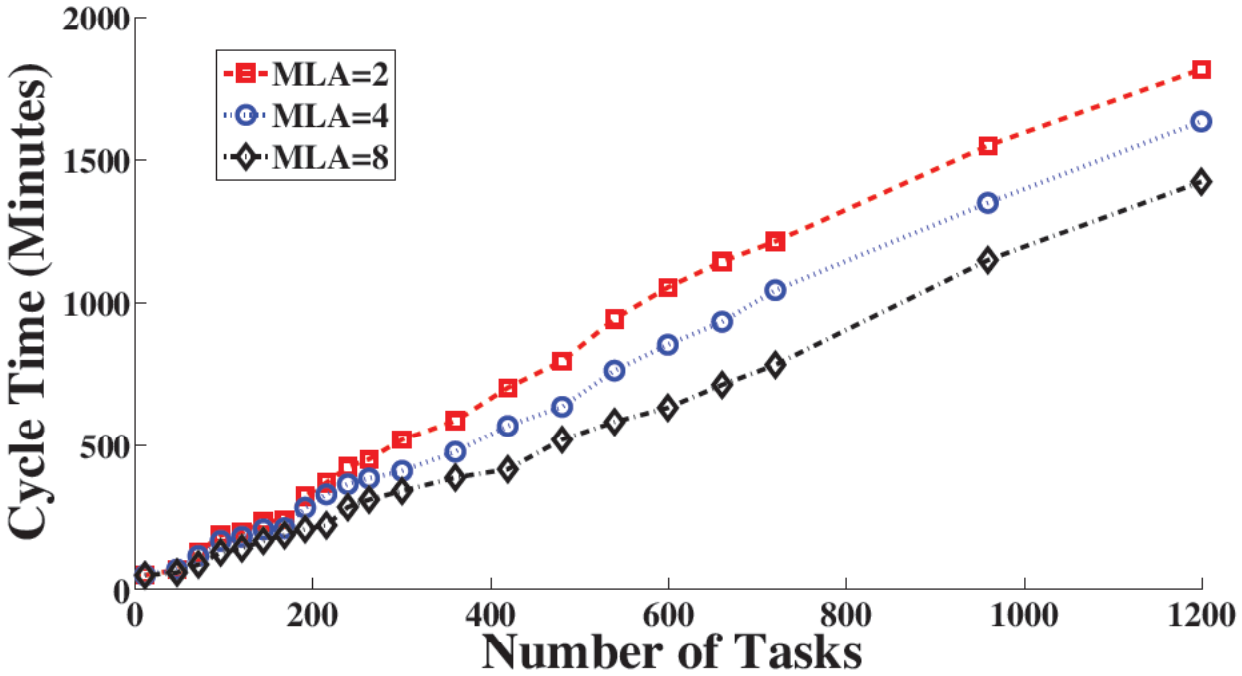


- *SPS-GPE algorithm:* Semantic grade overrides priorities based on period and execution time



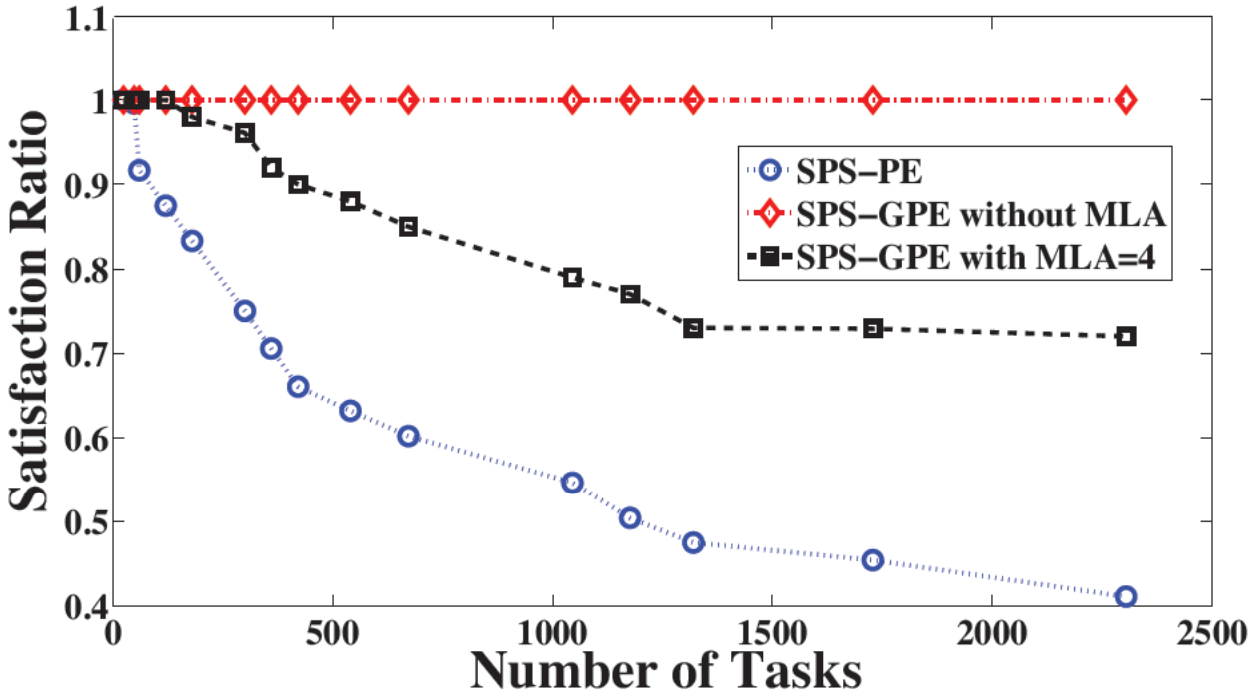
Cycle Time Performance

- As the MLA constraint value (given by maximum number of concurrent execution jobs permitted network-wide) increases, the cycle time decreases



Satisfaction Ratio Performance

- Satisfaction ratio using the SPS-GPE algorithm is always equal to 1 even with large number of task inputs



Topics of Discussion

- Project Overview
- Workplan Status
- Accomplishments
  - Part I: perfSONAR Deployments' Measurements Analysis
    - Major Activities, Results and Findings
  - Part II: Multi-domain Measurement Scheduling Algorithms
    - Major Activities, Results and Findings
  - Part III: Outreach and Collaborations
- Planned Next Steps

PART - III
Outreach and Collaborations

- **Project Website**
- **Presentations**
  - "Experiences from developing analysis techniques and GUI tools for perfSONAR users", perfSONAR Workshop, Arlington, VA, 2010.
  - "Multi-domain Internet Performance Sampling and Analysis Tools", Internet2/ESCC Joint Techs, Columbus, OH, 2010.
  - "OnTime Detect Tool Tutorial", Internet2 Spring Member Meeting, Arlington, VA, 2010.
  - "Multi-domain Internet Performance Sampling and Analysis", Internet2/ESCC Joint Techs, Salt Lake City, 2010.
- **Peer-reviewed Papers**
  - P. Calyam, J. Pu, W. Mandrawa, A. Krishnamurthy, "OnTimeDetect: Dynamic Network Anomaly Notification in perfSONAR Deployments", *IEEE Symposium on Modeling, Analysis & Simulation of Computer & Telecommn. Systems (MASCOTS)*, 2010. [Poster]
  - P. Calyam, L. Kumarasamy, F. Ozguner, "Semantic Scheduling of Active Measurements for meeting Network Monitoring Objectives", *IEEE Conference on Network and Service Management (CNSM) (Short Paper)*, 2010. [Poster]
- **Software Downloads**
  - OnTimeDetect: Offline and Online Network Anomaly Notification Tool for perfSONAR Deployments [Web-interface Demo] [SC10 Demo] [Twitter Demo]
- **News Articles**
  - "Research seeks to improve service for users of next-generation networks", OSC Press Release, October 2009.

Planned Next Steps

- Multi-domain Measurement Scheduling Algorithms
  - Continue evaluation of semantic scheduling algorithms
  - Investigate multi-layer sampling approaches
  - Design and develop perfSONAR web-service extensions for multi-layer and multi-domain measurements
  - Integrate into perfSONAR resource protection service
- perfSONAR deployments' Measurements Analysis
  - Use Throughput, RTT, and SNMP data sets and compare with other anomaly detection methods (e.g., Kalman filter, PCA)
  - Release improved versions of "OnTimeDetect" tool to diverse users (e.g., network operators, researchers)
  - Integrate anomaly detection research into DOE operations and applications for analyzing measurement data sets

Thank you for your attention! ☺

“OnTime*” Toolkit

- “OnTime*”: OnTime Sampling and Analysis Toolkit
  - “OnTimeSample”, “OnTimeDetect”, “OnTimePredict”
- End-user toolkit that allows end-to-end performance sampling and analysis in DOE science community applications (e.g., OSCARS, GridFTP)
  - Allows users to specify monitoring objectives and provisions on-going and on-demand measurement samples on ESnet paths
  - Uses multi-layer measurements from ESnet perfSONAR deployments for analysis such as:
    - Network paths monitoring
    - Network weather forecasting
    - Network performance anomaly detection
    - Network-bottleneck fault-location diagnosis
  - Integrates into social networking forums
    - Forum examples - ESnet’s Net Almanac, Twitter

The "network-awareness" gap!

- **Network Researcher**
  - Bandwidth-on-demand
  - DDoS Traceback
  - Path Switching
  - …..

"Measurements Provider can provide measurements *when*, *where* and *how* ever I want!"

"Hey Measurements Provider, I need *pure periodic samples* of available bandwidth on *xyz paths* crossing *A, B and C domains* for my performance forecasting"

- **Measurements Provider**
  - Measurements collection
  - Measurement graphs
  - Measurement query
  - …..

"I am collecting *all* the measurements a network researcher would want!"

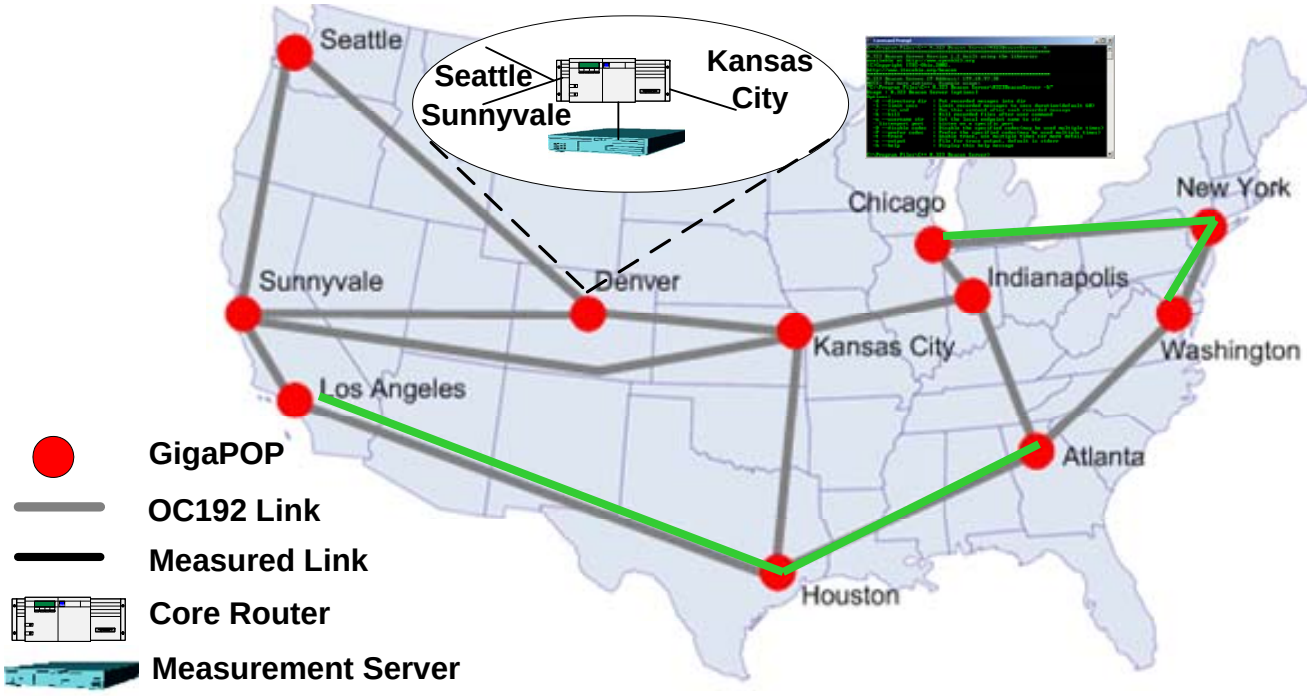
"Oh, I *don't collect pure periodic* samples, and *don't know if A, B and C* domains collect available bandwidth measurements!"

- This gap between "assumptions of theory" (researchers) and "delivering ability of reality" (ISPs) can be bridged by:
  - Efficient sampling techniques that meet measurement timing demands
  - Measurement federation policies to provision multi-domain measurements

Network Measurement Infrastructures

- NMIs monitor network paths for network weather forecasting, anomaly-detection and fault-diagnosis
- Measurement servers are deployed at strategic network points in a network domain
- Tools on measurement servers measure network QoS along end-to-end paths

Performance Measurement Tools



Measurement Requests

- On-going measurement task format

  $\tau_i = (<src_i>, <dst_i>, <tool_i>, <period\ p_i>, <execution\ time\ e_i>)$

  Example: $\tau_1 = (Denver, Seattle, Iperf, 60, 20)$

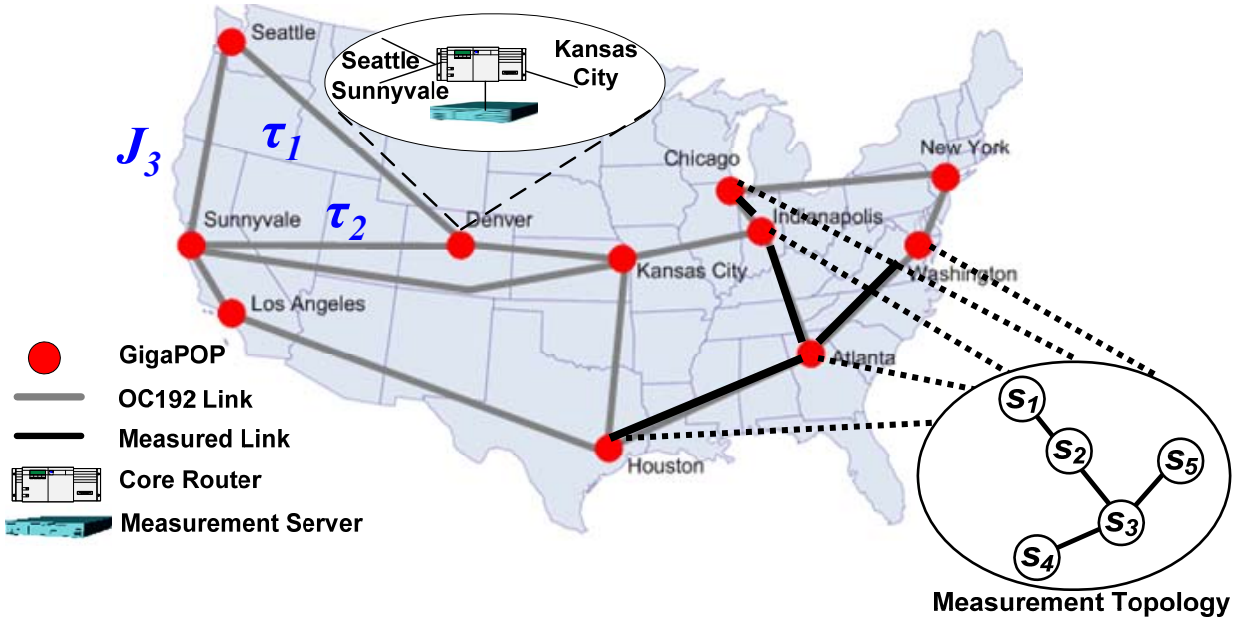
  $\tau_2 = (Denver, Sunnyvale, Iperf, 60, 20)$

- On-demand measurement job format

  $J_i = (<src_i>, <dst_i>, <tool_i>, <execution\ time\ e_i>)$

  Example: $J_3 = (Sunnyvale, Seattle, Iperf, 20)$

- Measurement topology

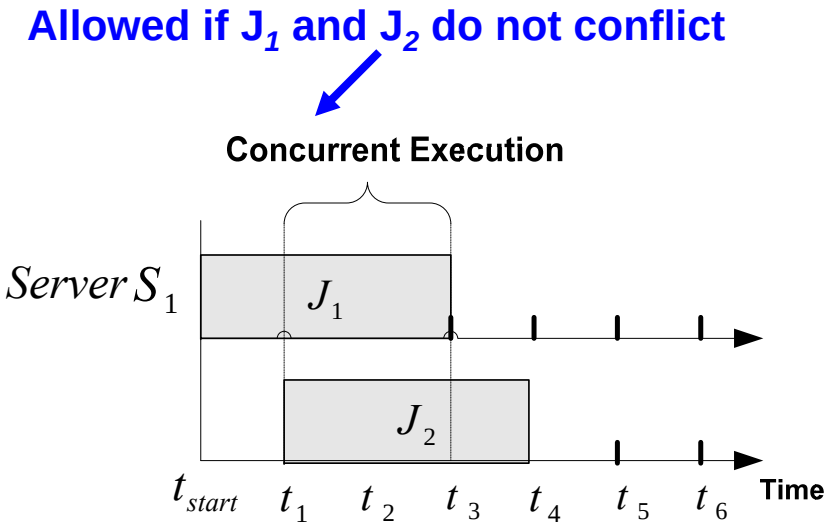


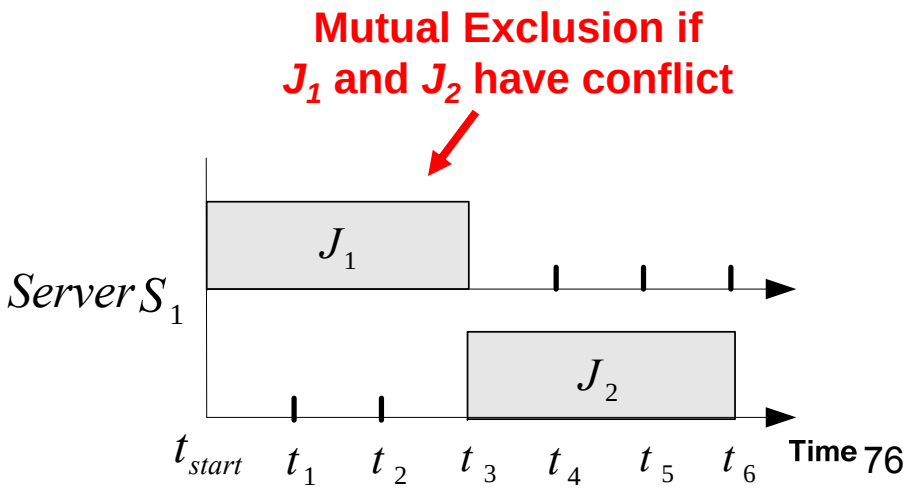
perfSONAR Overview

- A set of high level services for managing multi-domain measurement/monitoring infrastructures
- International community of developers
  - Implementing Open Grid Forum (OGF) Network Measurement (NM-WG) recommendations
- Deployed at ESnet, Internet2, GEANT, DOE Labs, Regional networks, University campuses, …

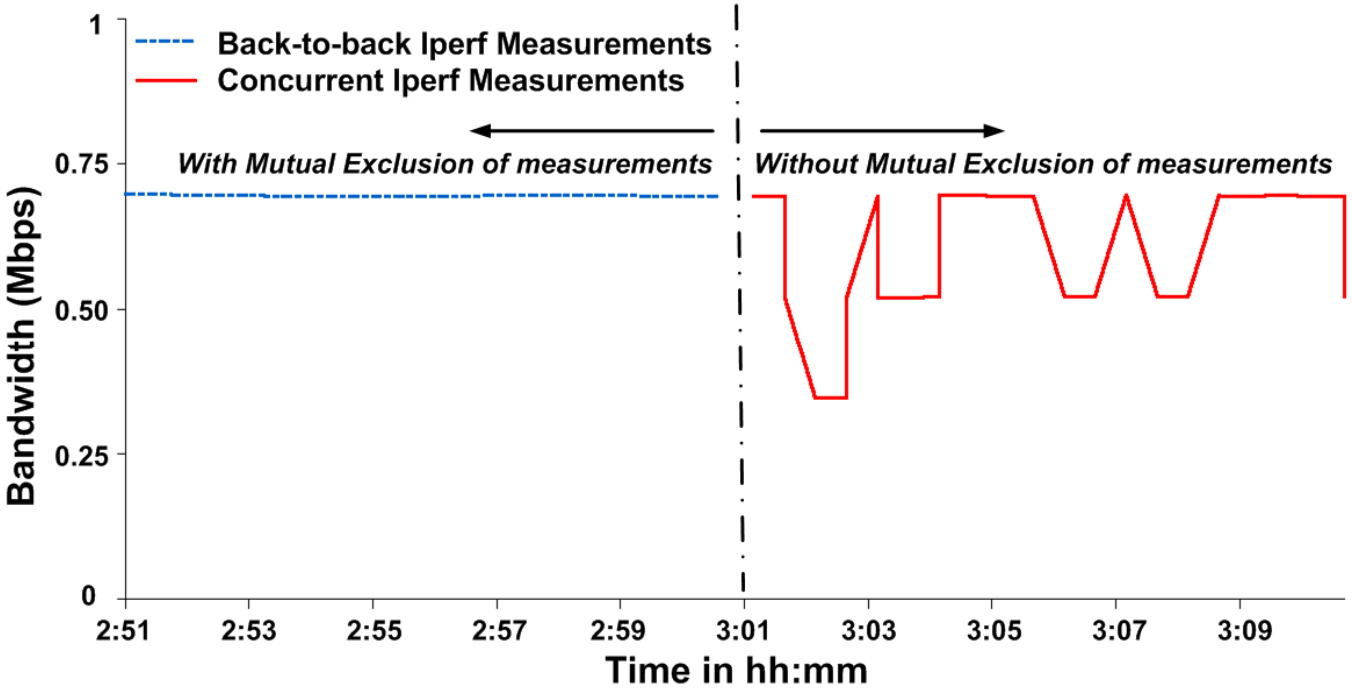
Measurement Conflict Problem

- Tools can intensively consume CPU and/or channel resources
  - Measurements of such tools conflict with one another if concurrently executed on same measurement server or path
    - Produce misleading reports of network status
    - Channel resource limitation is the main bottleneck
  - Concurrent execution allows more frequent sampling of network status (i.e., improves schedulability)
    - Measurements tools not on same path or non-conflicting on a server





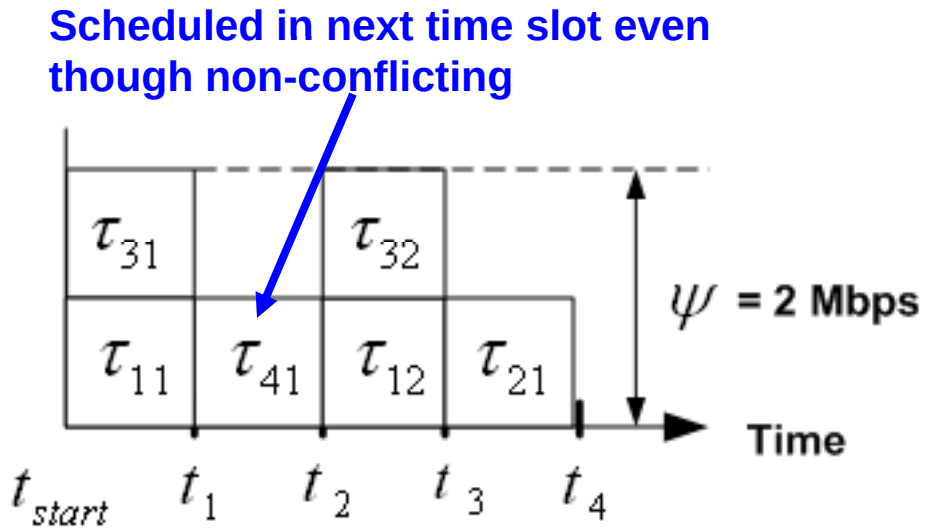
"Measurement Conflict" Illustration



- Iperf bandwidth tests in a LAN testbed with 1500Kbps bandwidth
- Background traffic (i.e., a Videoconference session) using ~768Kbps bandwidth in the LAN testbed

Measurement Regulation

- Measurement traffic consumes bandwidth of actual application traffic
- Regulation using Measurement Level Agreements (MLAs)
  - E.g. Only (1-2) Mbps or (1-5) % of active measurement traffic permitted
- MLA constraint restricts the amount of measurements on a path
  - Example below assumes each job consumes 1 Mbps bandwidth



Related Measurement Conflict Resolution Work

- No Orchestration
  - Used in traditional NMIs (e.g., pingER)
  - Measurement conflicts not an issue
- Single-processor-like Scheduling
  - Simple Round-robin Scheduling
    - Used in NLANR AMP [1]
  - Resource broker Scheduling
    - Used in Internet2 perfSonar [2]
  - Token passing Scheduling
    - Used in Network Weather Service [3]

1. None of them leverage Concurrent Execution when possible

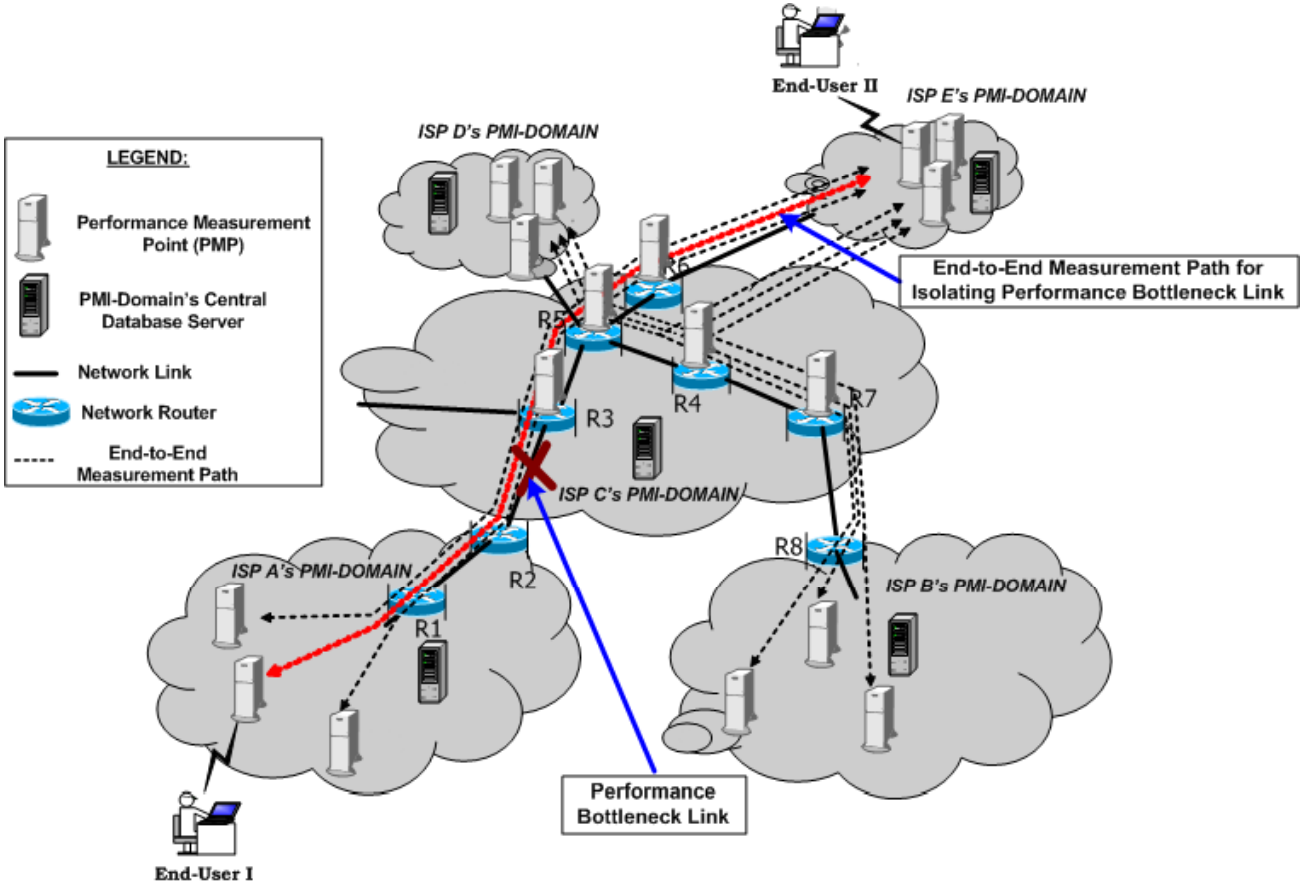
2. None of them handle on-demand measurement job requests

[1] T. McGregor, H.-W. Braun, J.Brown, “The NLANR Network Analysis Infrastructure", *IEEE Communications Magazine*, Pages 122-129, May 2000.

[2] E. Boyd, J. Boote, S. Shalunov, M. Zekauskas, “The Internet2 E2E piPES Project: An Interoperable Federation of Measurement Domains for Performance Debugging", *Internet2 Technical Report*, 2004.

[3] B. Gaidioz, R. Wolski, B. Tourancheau, “Synchronizing Network Probes to avoid Measurement Intrusiveness with the Network Weather Service", *Proc. of IEEE High-performance Distributed Computing Conference*, 2000.

Multi-domain Performance Measurement



- *Measurement Federations (e.g., ESnet, Internet2, GEANT)*
  - Sharing measurement topologies, MLAs, AAA, measurement data exchange formats, …