Enhanced EDF Scheduling Algorithms for Orchestrating Network-wide Active Measurements

Prasad Calyam, Chang-Gun Lee

Phani Kumar Arava, Dima Krymskiy

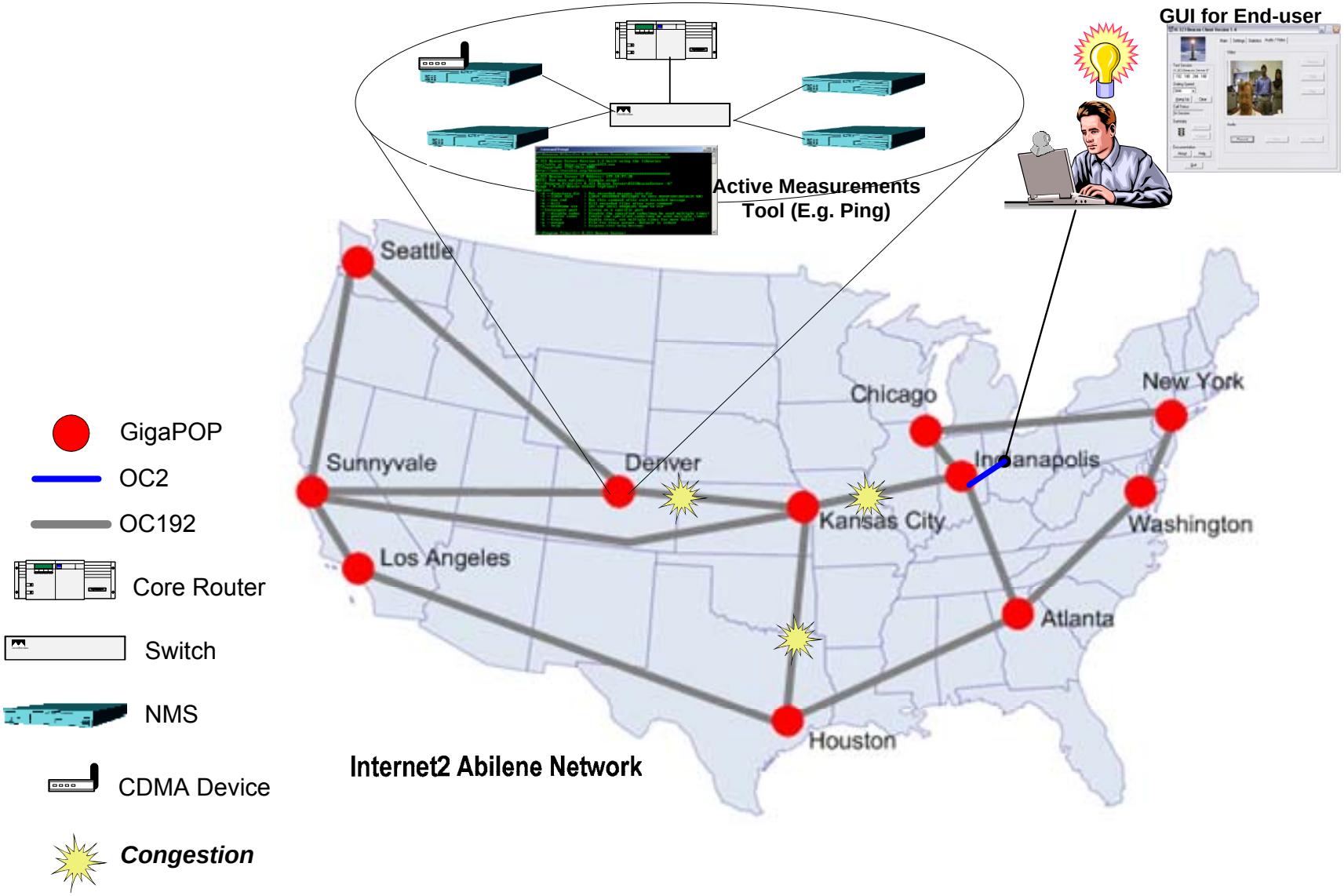
OARnet, The Ohio State University

IEEE RTSS, Miami, December 2005

Active Measurements

- "Active Measurements" involve injecting test packets into network paths to determine network status in terms of -
  - Topology, Bandwidth, Delay, Jitter, Loss, …
- It has become a common practice for ISPs to instrument networks so as to support *network-wide active measurements* to help -
  - **Researchers**
    - Want to study the characteristics of networks that could be adopted in simulation models to develop new network protocols for advanced end-applications
  - **ISPs**
    - Determine end-to-end performance bottlenecks and trends of network
      - Helpful for resource capacity planning and detection of DDoS attacks
  - **End users**
    - Would like to know about the network performance they are getting at their computer
      - *"Why is my video quality so poor in the videoconference?"*
      - Bandwidth?, …

Network-wide Active Measurements

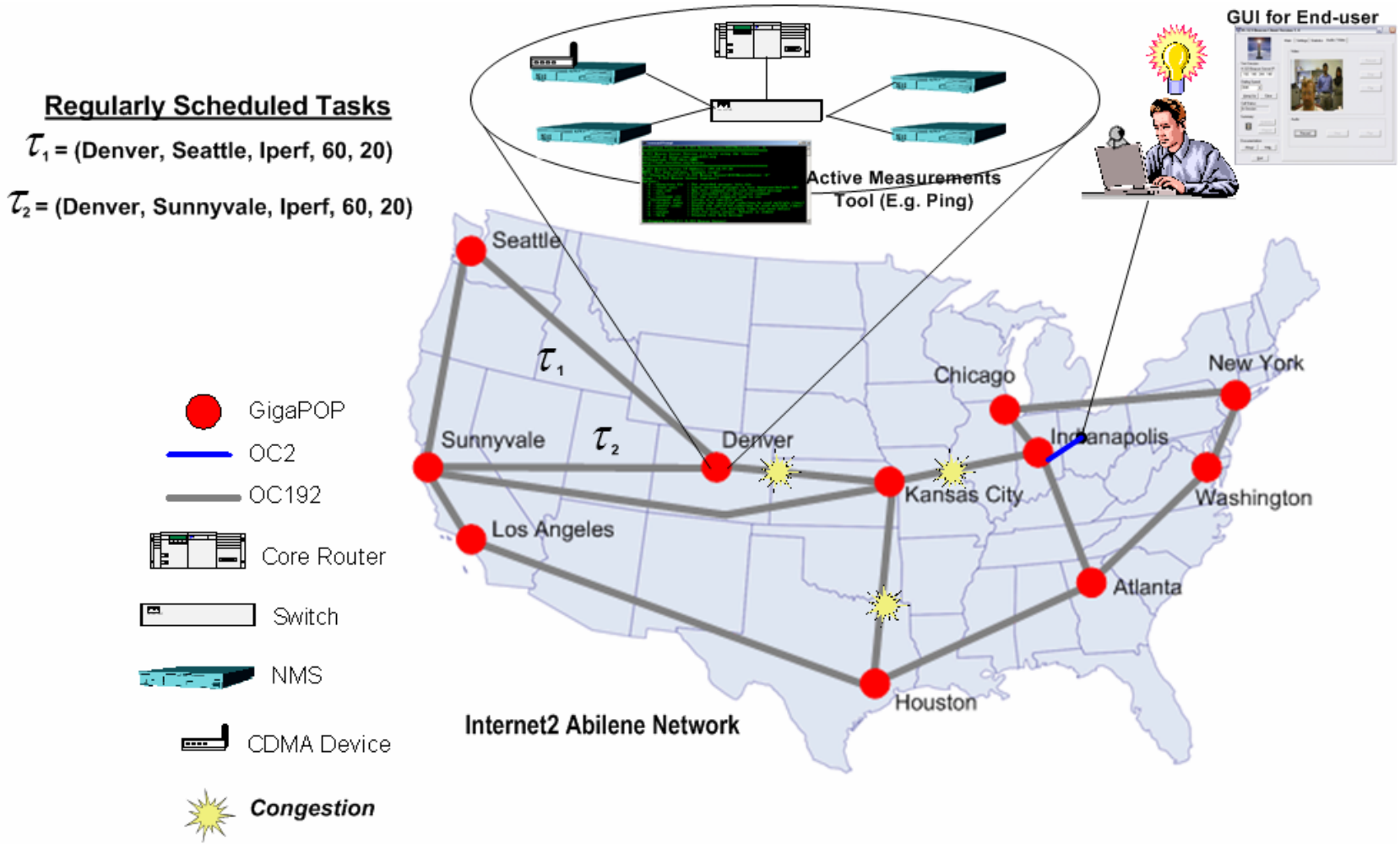


Types of Active Measurements

- Regularly Scheduled Active Measurements
  - Different kinds of network status measurements (e.g. delay, bandwidth, …) with definite ***periodicity*** requirements *(in the order of minutes)*
- On-demand Active Measurements
  - One-off measurements that need *to be executed as soon as possible* without disrupting the regularly scheduled measurements

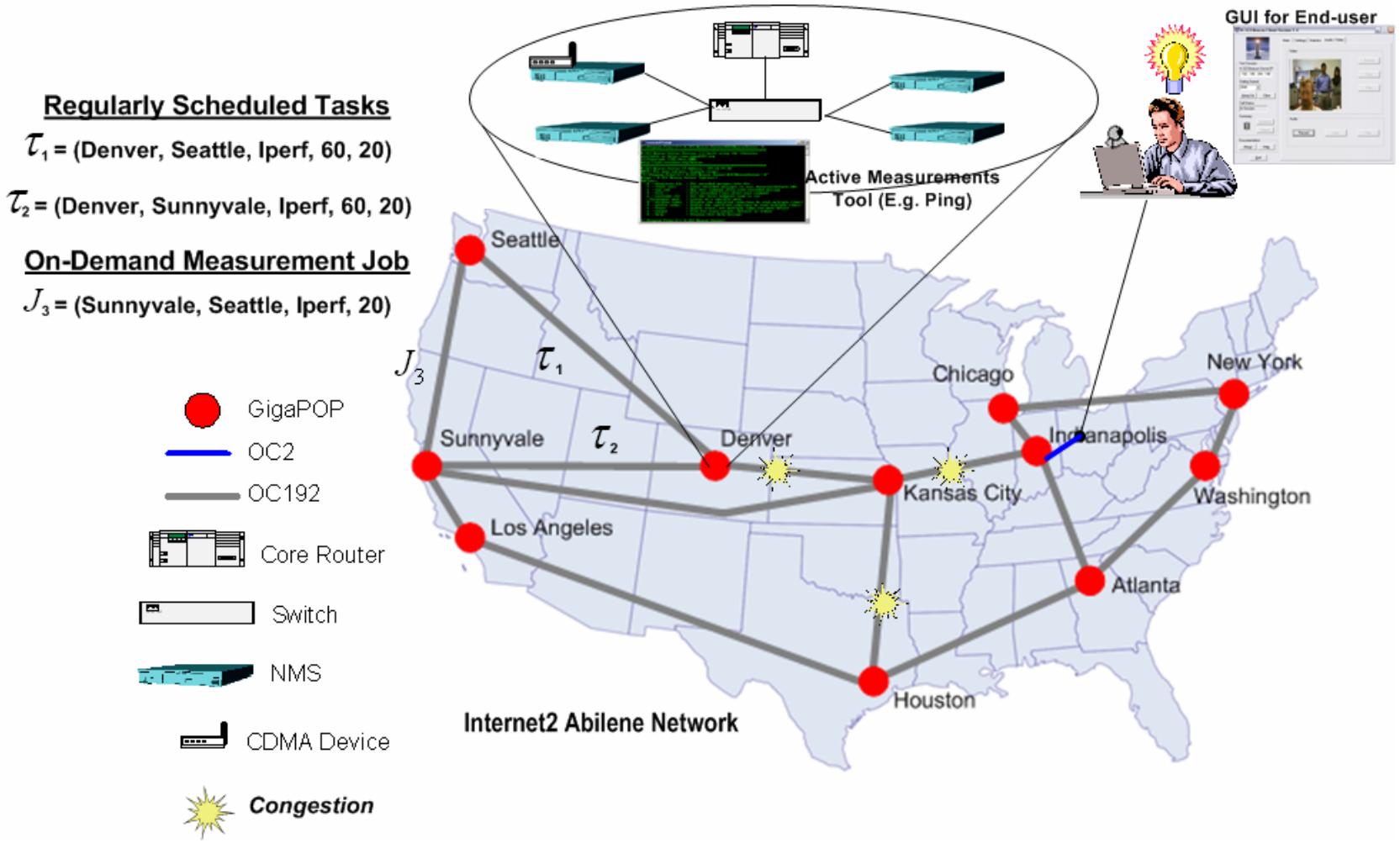
Network-wide Active Measurements

NOTE: **Task Format**
(*<src>*, *<dst>*, *<tool>*, *<period>*, *<execution time>*)

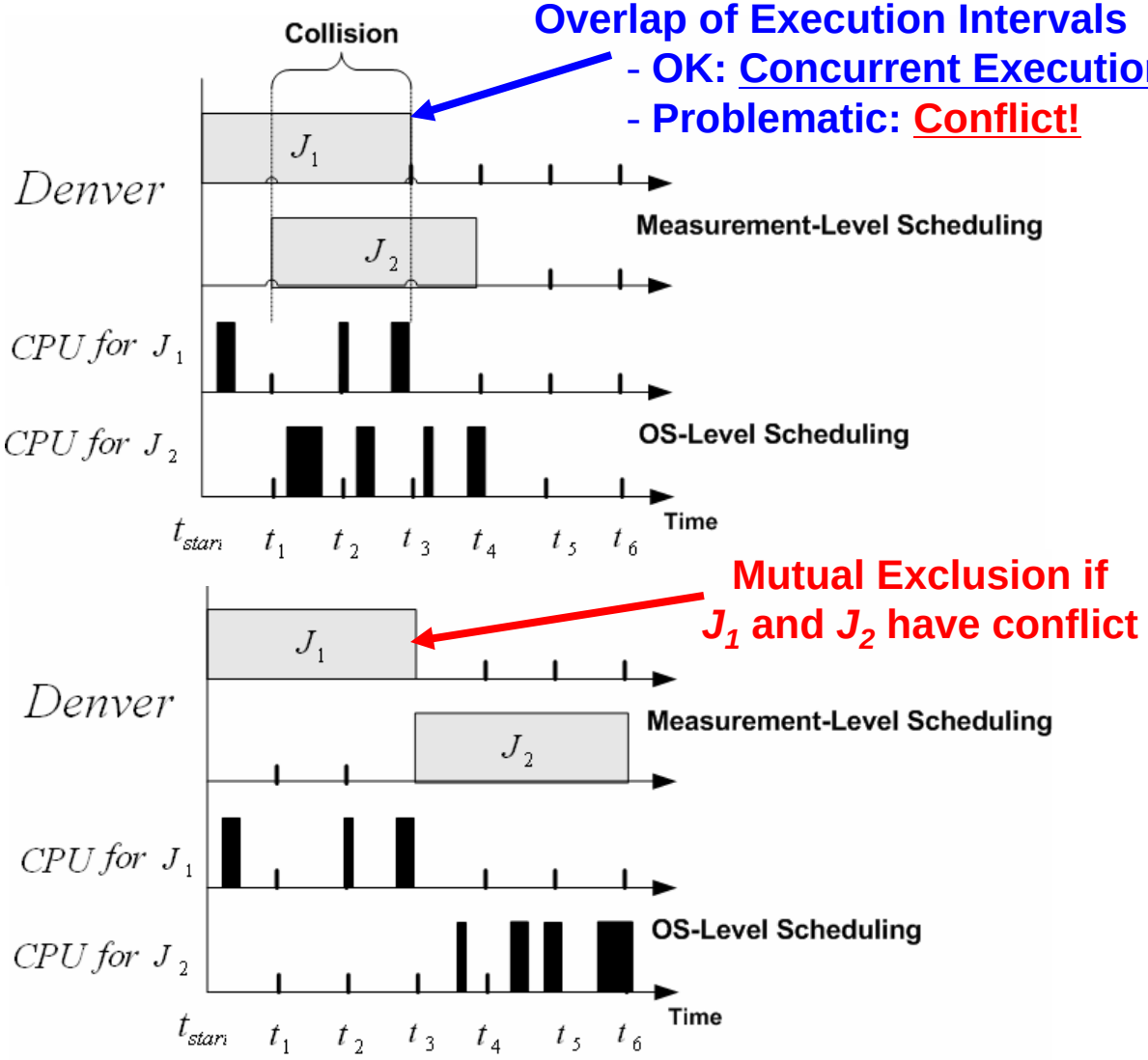


Network-wide Active Measurements

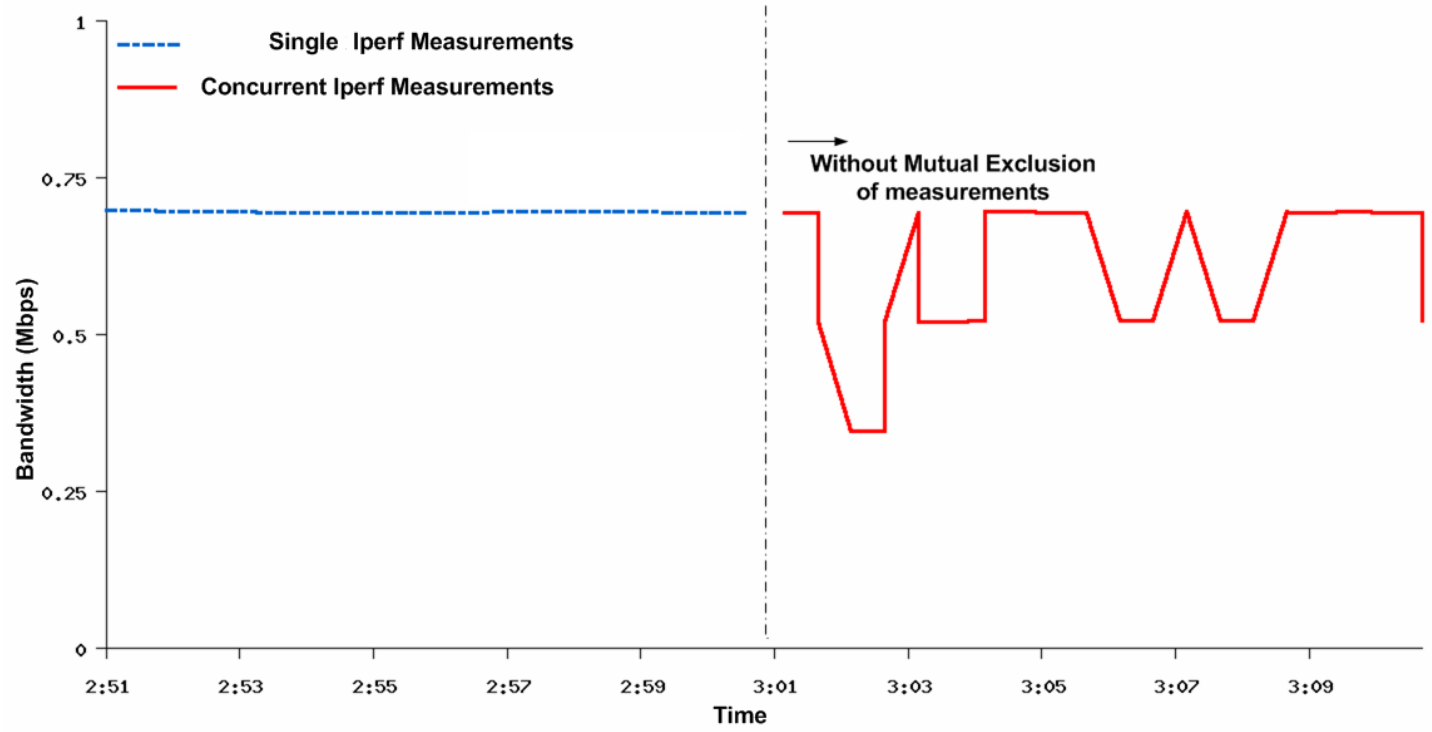
NOTE: **Task Format**
(*<src>*, *<dst>*, *<tool>*, *<period>*, *<execution time>*)



OS-Level Scheduling (Pre-emptive)
Vs
Measurement-Level Scheduling (Non Pre-emptive)



“Measurement Conflict” Illustration



- Iperf bandwidth tests in a LAN testbed with 1500Kbps bandwidth
- Background traffic of a Videoconferencing session using approx. 768Kbps bandwidth in the LAN testbed

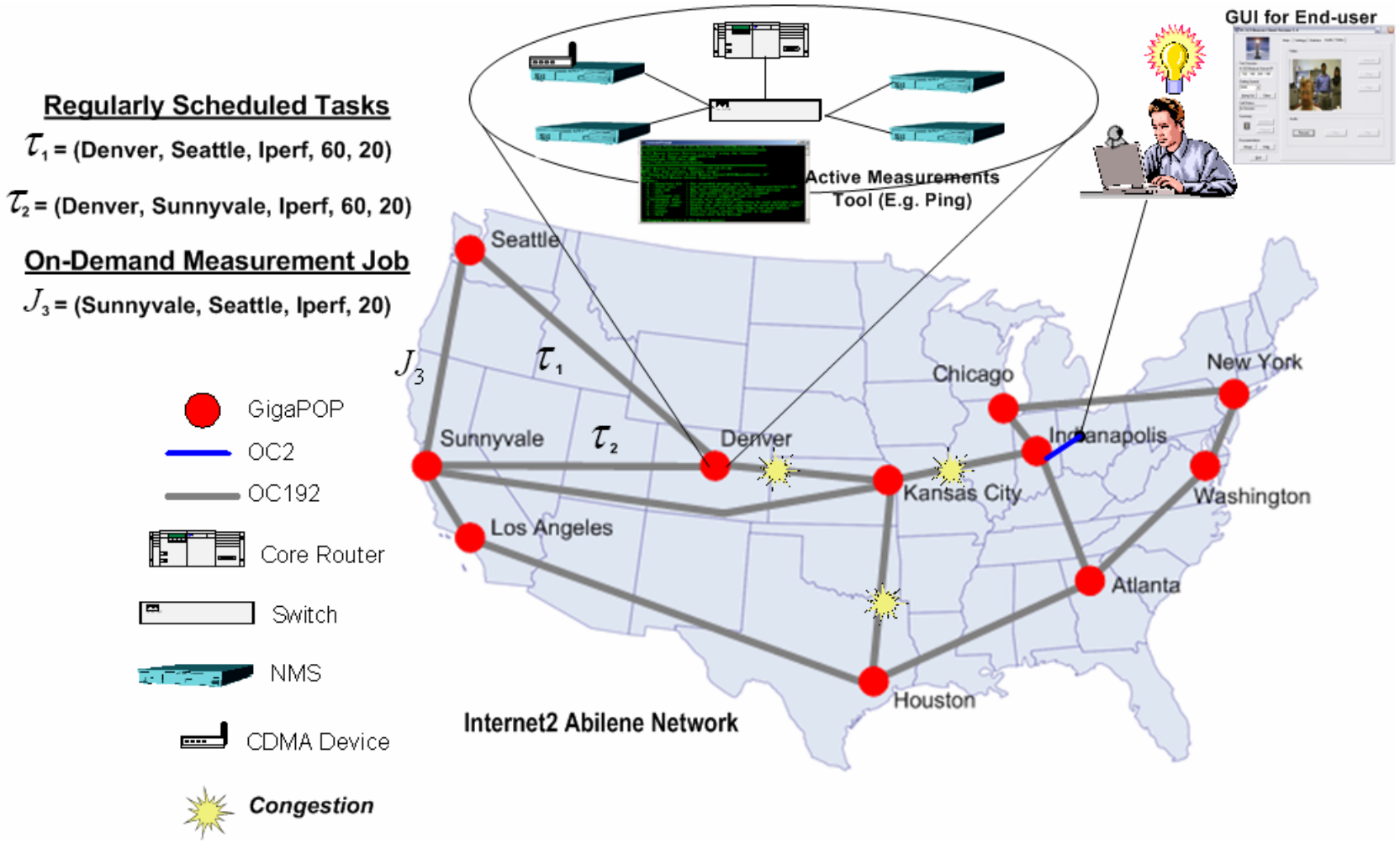
Measurement Conflicts

Measurement Tool	CPU Intensive	Channel Intensive
Iperf	Yes	Yes
H.323 Beacon	No	Yes
Pathchar	Yes	No
Ping	No	No

- Executing multiple concurrent active measurements could result in misleading reports of network performance
  - Caused due to CPU and Channel resource sharing
  - Concurrent execution of tools that are either channel or CPU intensive (e.g. Iperf, H.323 Beacon) needs to be avoided!
    - Conflict arises when concurrent execution is on same measurement server or on the same network path
  - Concurrent execution of tools that are neither channel or CPU intensive (e.g. Ping, Traceroute) could be provisioned
    - To allow for a faster repetition of measurement schedules and thus permit more frequent sampling i.e. better understanding of network health status

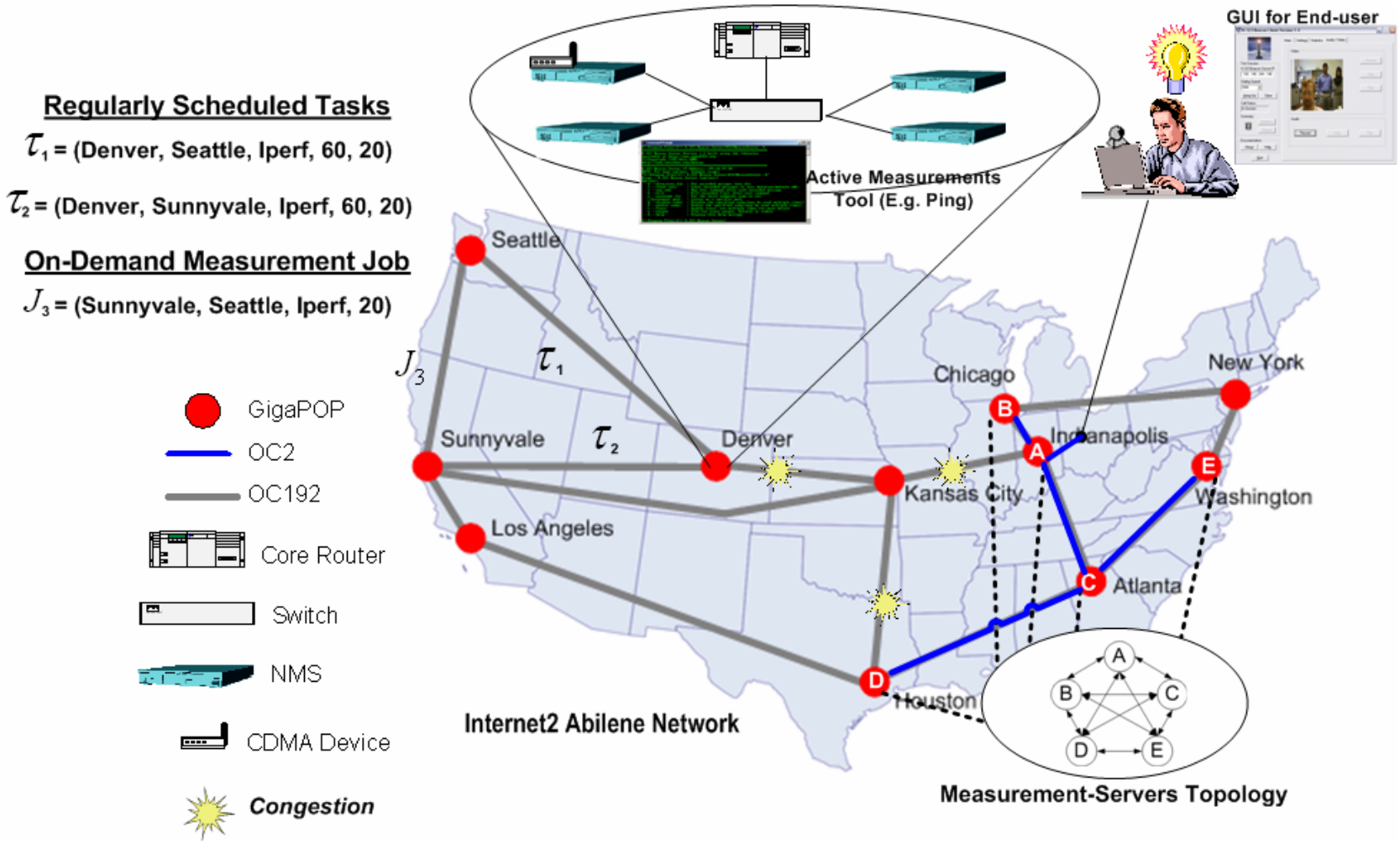
Network-wide Active Measurements

NOTE: **Task Format**
(*<src>*, *<dst>*, *<tool>*, *<period>*, *<execution time>*)



Measurement-Servers Topology

NOTE: **Task Format**
(*<src>*, *<dst>*, *<tool>*, *<period>*, *<execution time>*)



Problem Description

- Given:
  - *N* ={*A,B,C,D, …*} is the set of measurement servers
  - *E* is the set of edges between a pair of servers
  - *G* = *(N, E)* measurement-servers topology
  - $\zeta = \{\tau_1, \tau_2, \tau_3, ...\}$ corresponds to a measurement task set
    - $\tau_i = (src_i, dst_i, tool_i, p_i, e_i)$
  - *ψ* refers to a "Measurement Level Agreement" (MLA)
    - E.g. Only (1-2) Mbps or (1-5) % of active measurement traffic permitted
- Problem:
  - ***Offline Scheduling*** – For a G measurement-servers 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*

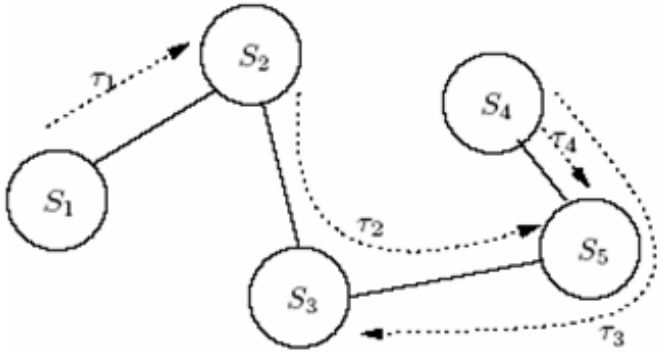
Related Work

- No Orchestration
  - Used in Traditional
  - They did not notice measurement conflicts
- Single-Processor-Like Scheduling
  - Simple Round-robin Scheduling
    - Used in NIMI and other Common NMIs
  - Resource broker Scheduling
    - Used in Internet2 E2Epi PIPES
  - Token-passing Protocol
    - Used in Network Weather Service

1. None of them leverage Concurrent Execution when possible

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

“Concurrent Execution” (CE) Principle



(a) Measurement Topology

$\tau_1 = (S_1, S_2, Pathchar, 60, 10)$

$\tau_2 = (S_2, S_5, Iperf, 40, 10)$

$\tau_3 = (S_4, S_3, H.323Beacon, 20, 5)$

$\tau_4 = (S_4, S_5, Ping, 30, 5)$

(b) Task Set

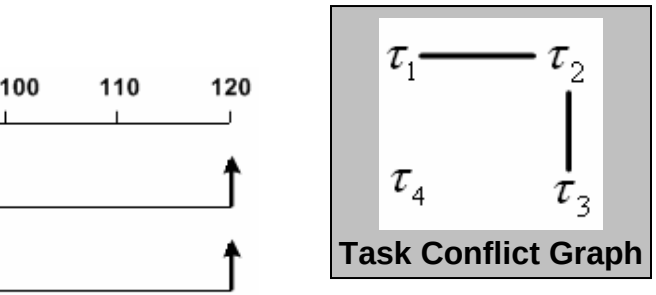
	Iperf	H.323Beacon	Pathchar	Ping
Iperf	1	1	1	0
H.323Beacon	1	1	1	0
Pathchar	1	1	1	0
Ping	0	0	0	0

(c) Tool Conflict Matrix

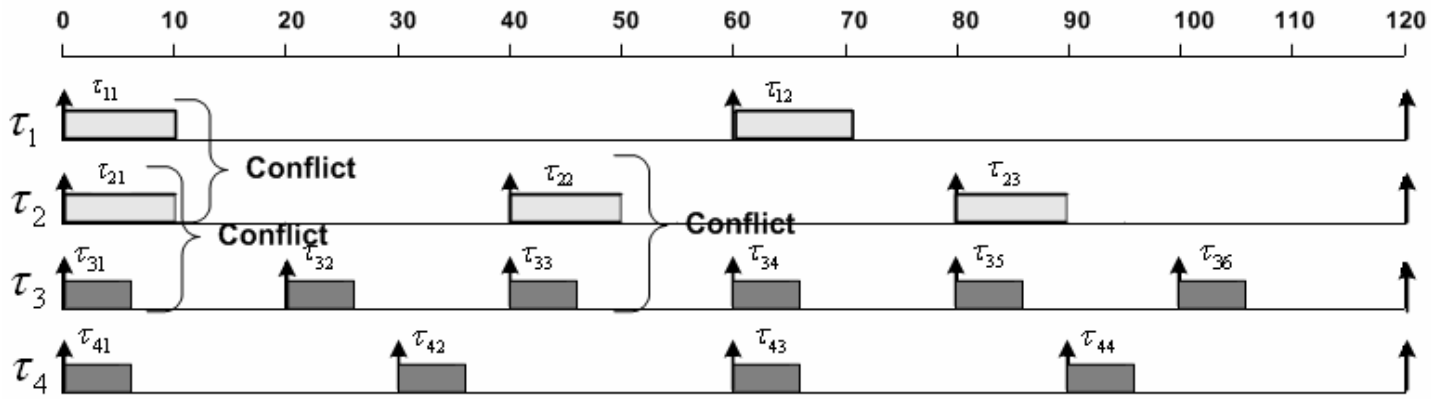
(d) Task Conflict Graph

- We 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!

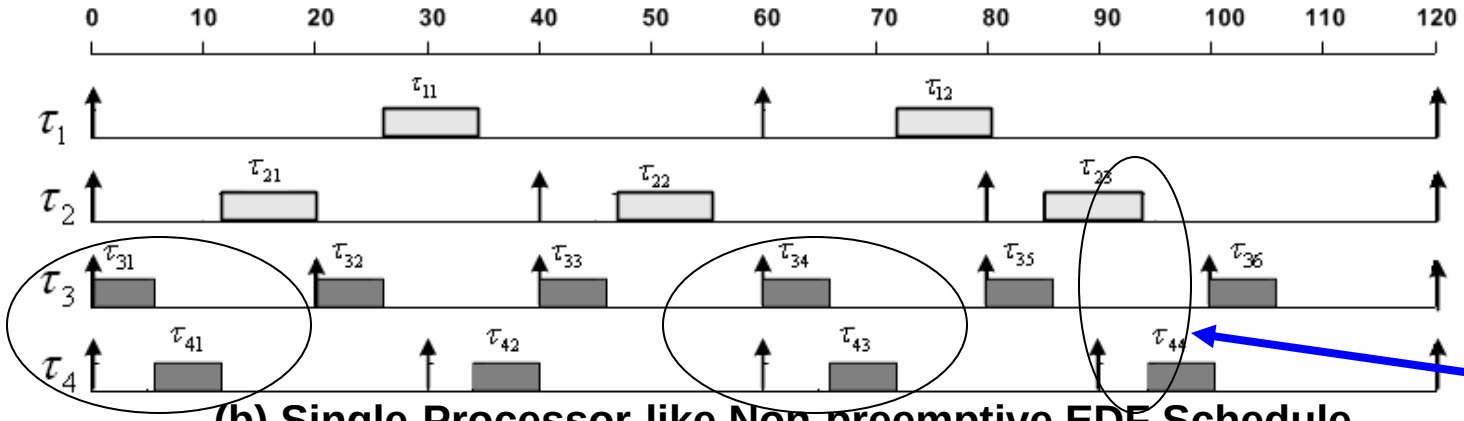
Offline Scheduling



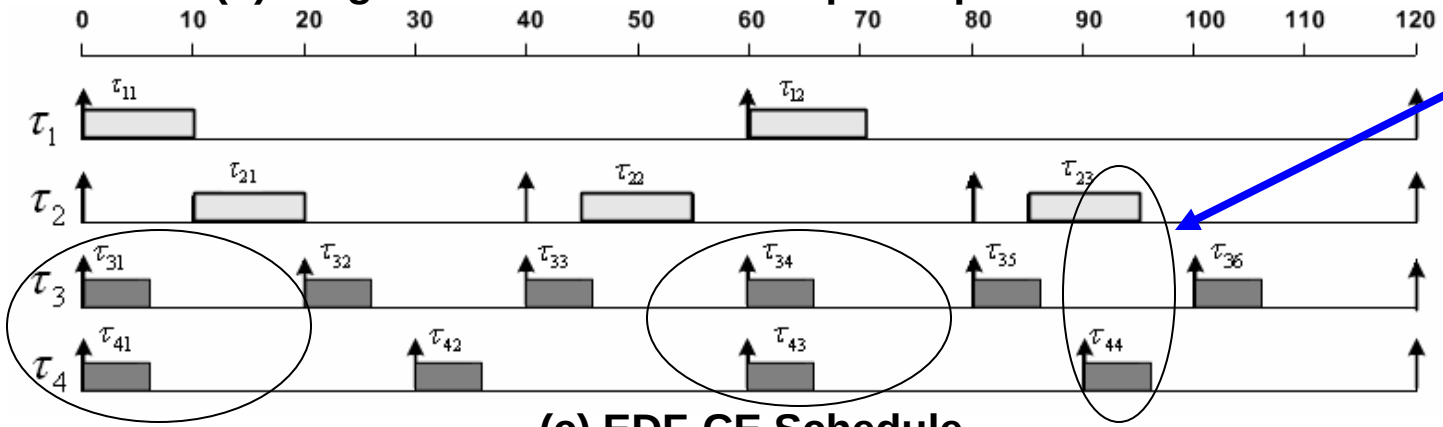
Task Conflict Graph



(a) No Orchestration



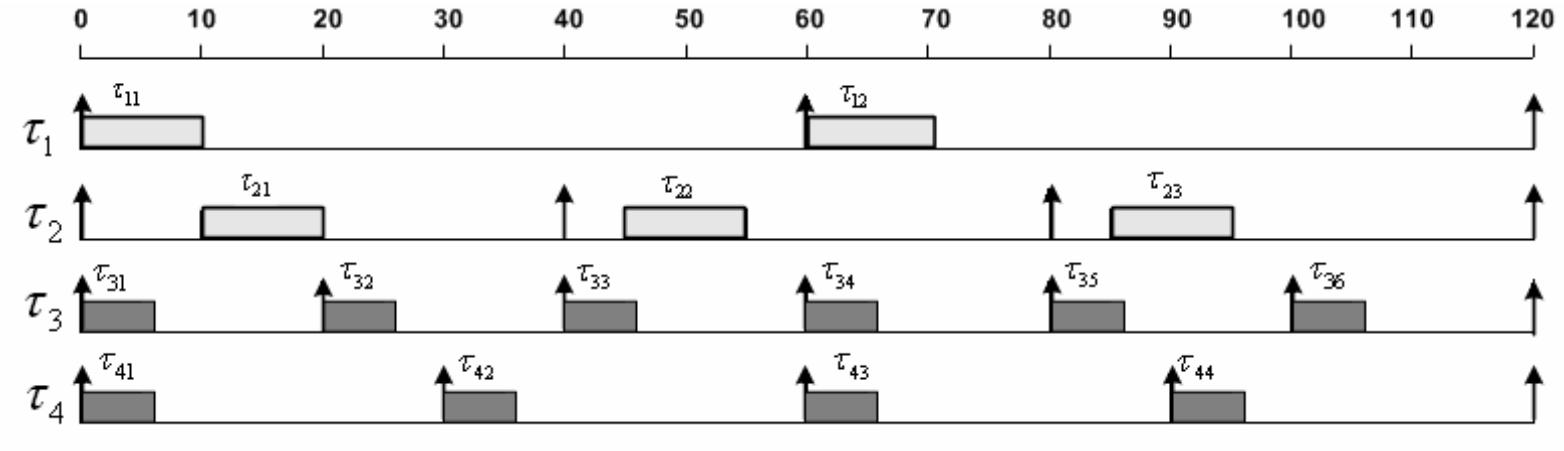
(b) Single-Processor-like Non-preemptive EDF Schedule



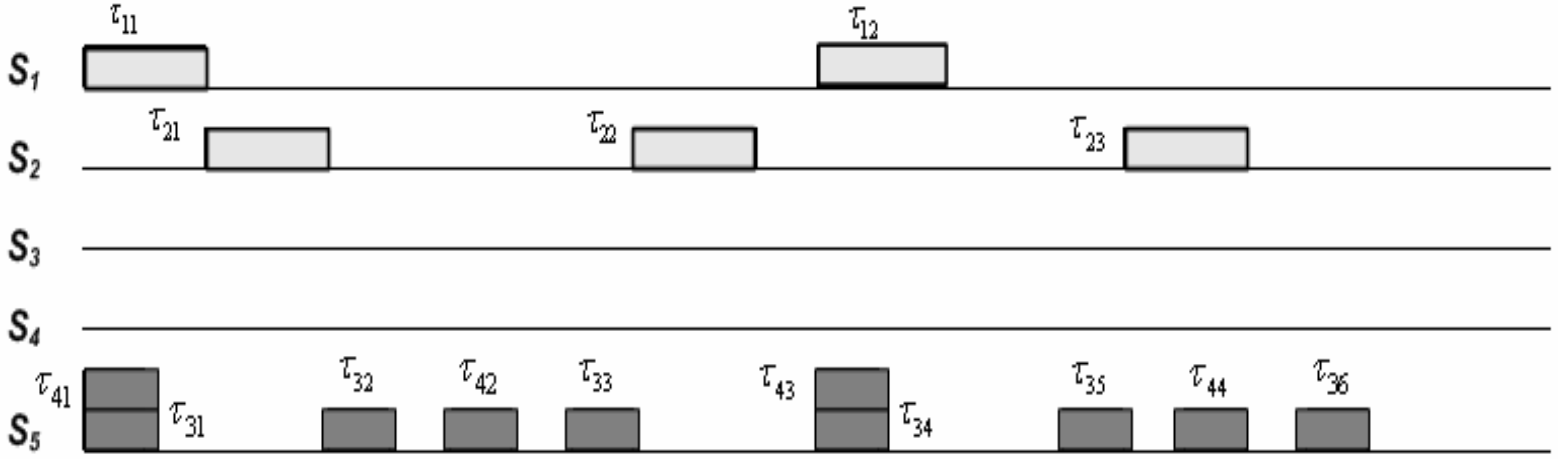
(c) EDF-CE Schedule

Concurrent Execution Effect

Measurement Schedules Tables for Servers

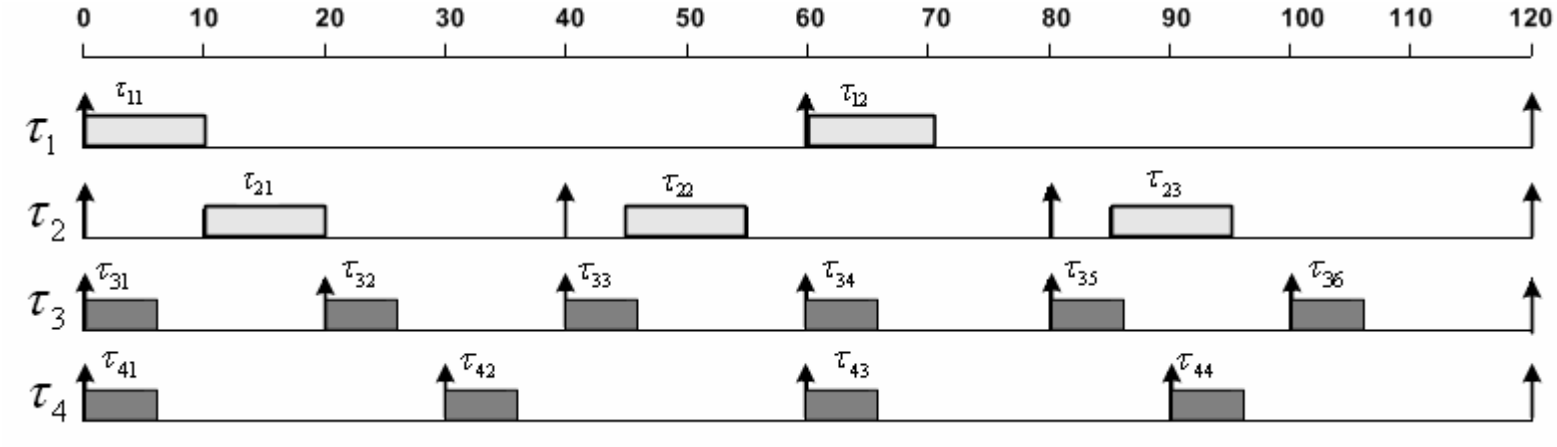


(a) EDF-CE Schedule



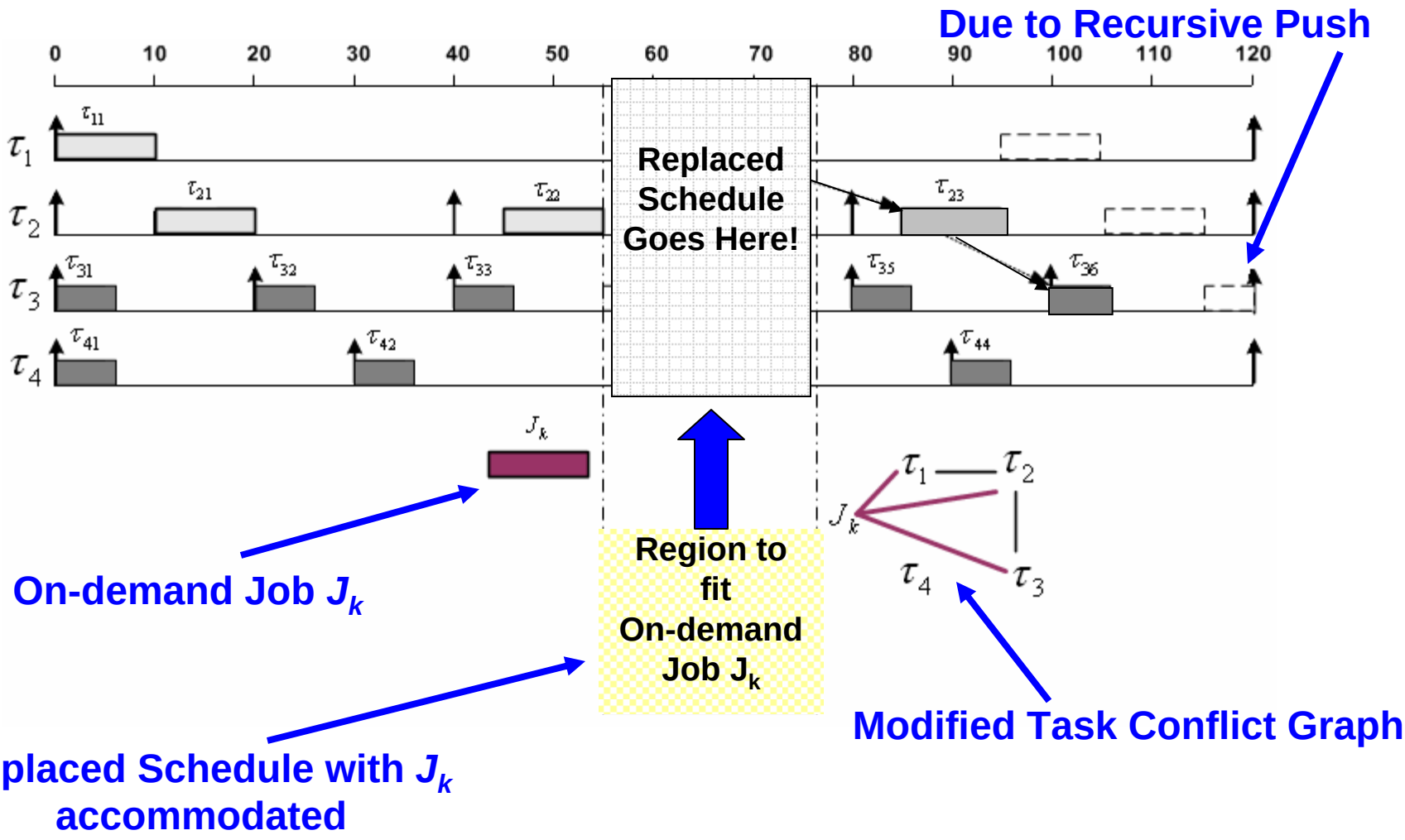
(b) Time Table for Jobs Execution on each Measurement-Server

Online Scheduling of On-demand Measurement Requests



Recursive Pushing for Maximum Early Slack Calculation

Online Scheduling of On-demand Measurement Requests

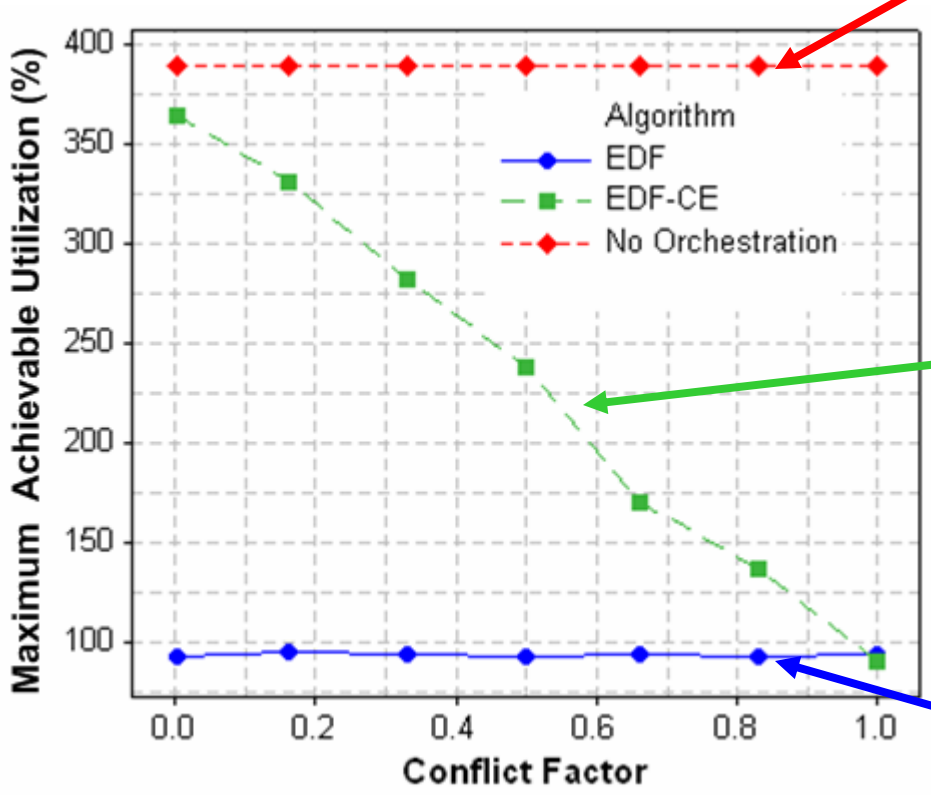


Recursive Pushing for Maximum Early Slack Calculation

Experiments 1: Synthetic Tasks

- Synthetic Task set comprised of 4 periodic measurement tasks – $\tau_1, \tau_2, \tau_3, \tau_4$
- Period p_i of each Task τ_i is randomly generated in the range [1000 – 10000]
- Execution time e_i of each Task τ_i is randomly generated in the range [100 – 999]
- The task conflict graph of the four tasks is also randomly created using a parameter called a "*conflict factor*"
  - The conflict factor represents the probability that there is a conflict edge between any two tasks
  - i.e. when the conflict factor is 1, the task conflict graph is fully connected; If the conflict factor is 0, there is no edge between tasks

Maximum Utilization



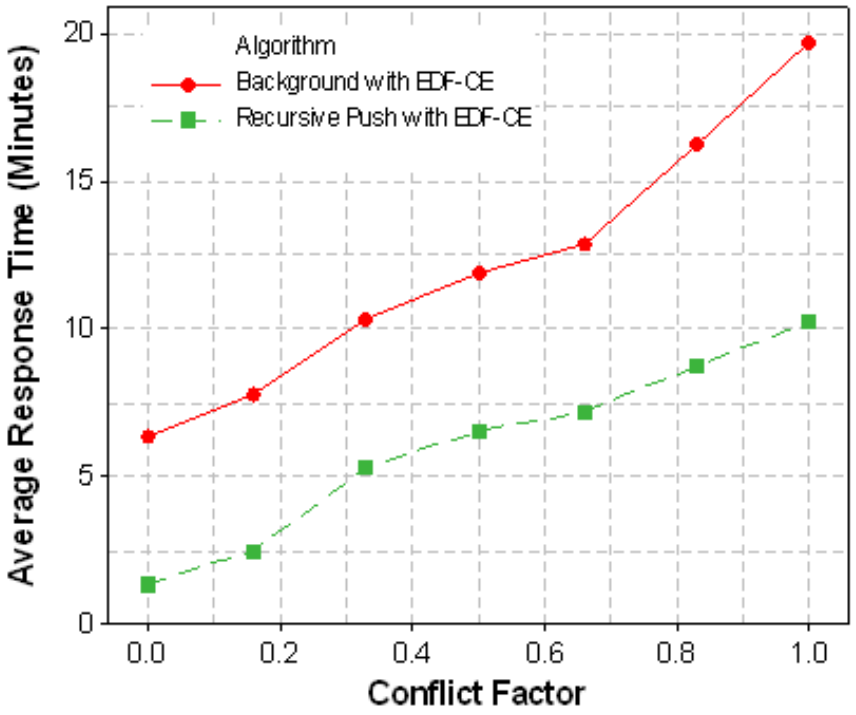
Produces misleading reports of network performance

EDF-CE same/better than EDF!

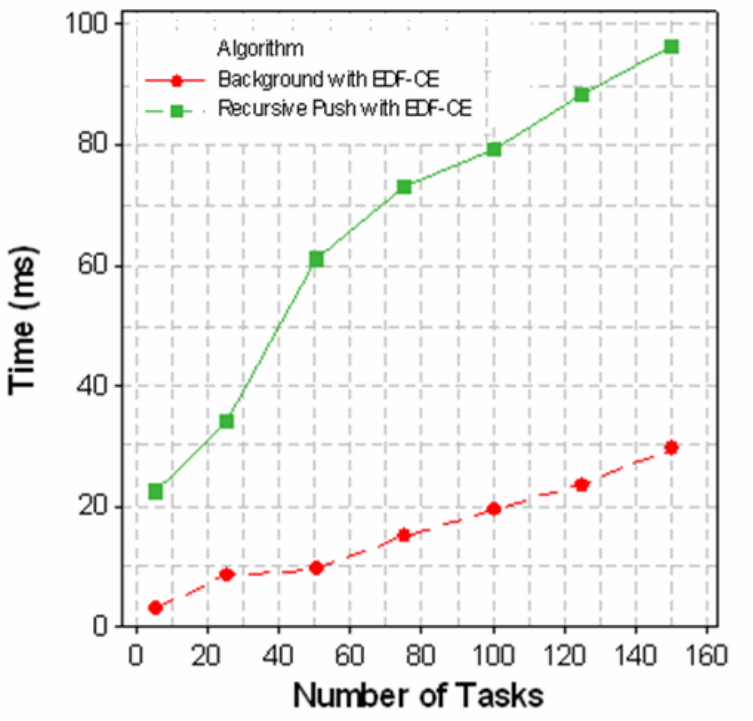
EDF or any Single-Processor-Like Scheduling

- EDF in above Fig. is representative of Single Processor Scheduling Schemes such as Round Robin, Resource Broker and Token Ring Scheduling

Average Response Time and Scheduling Overhead



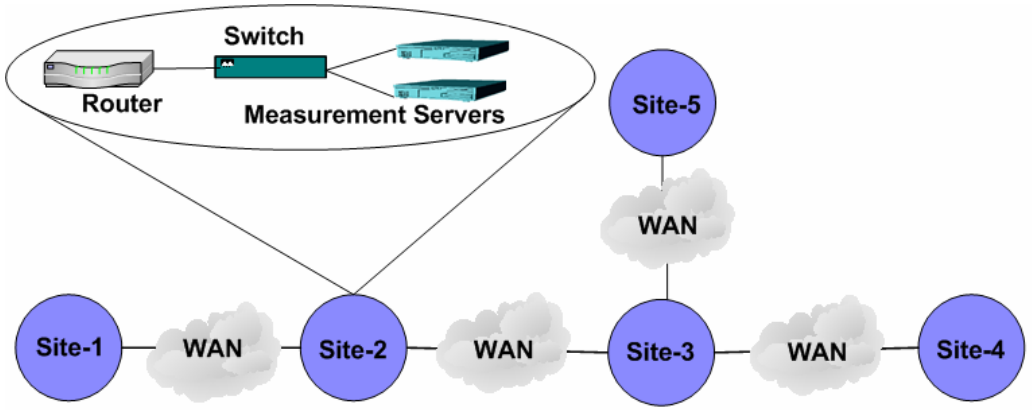
(a) Comparing Average Response Time



(b) Comparing Scheduling Overhead

- "Background" approach schedules an on-demand job in the earliest gap present in the offline EDF-CE schedule within which it can execute to completion

Experiment 2: Internet Testbed



(a) Measurement Servers Setup

τ_1 = (Site-3, Site-2, Iperf, 60, 20)

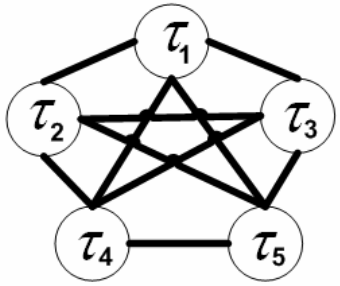
τ_2 = (Site-3, Site-4, H.323 Beacon, 10, 5)

τ_3 = (Site-3, Site-1, Pathchar, 60, 20)

τ_4 = (Site-3, Site-5, Pathload, 60, 5)

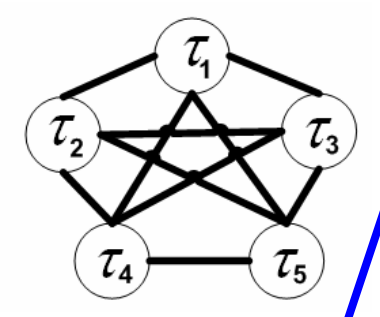
τ_5 = (Site-1, Site-4, Iperf, 60, 20)

(b) Task Set



(c) Task Conflict Graph

Amount of Conflict



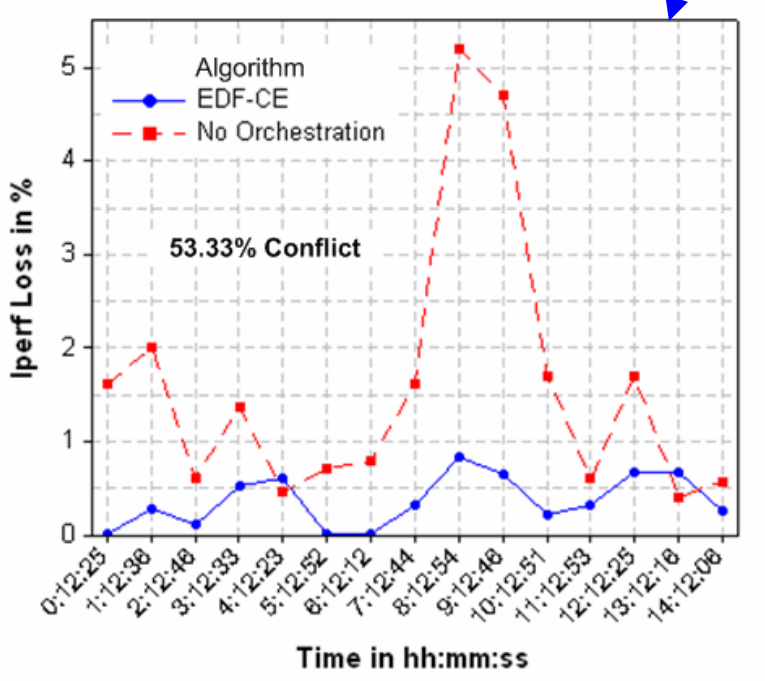
τ_1 = (Site-3, Site-2, Iperf, 60, 20)

τ_2 = (Site-3, Site-4, H.323 Beacon, 10, 5)

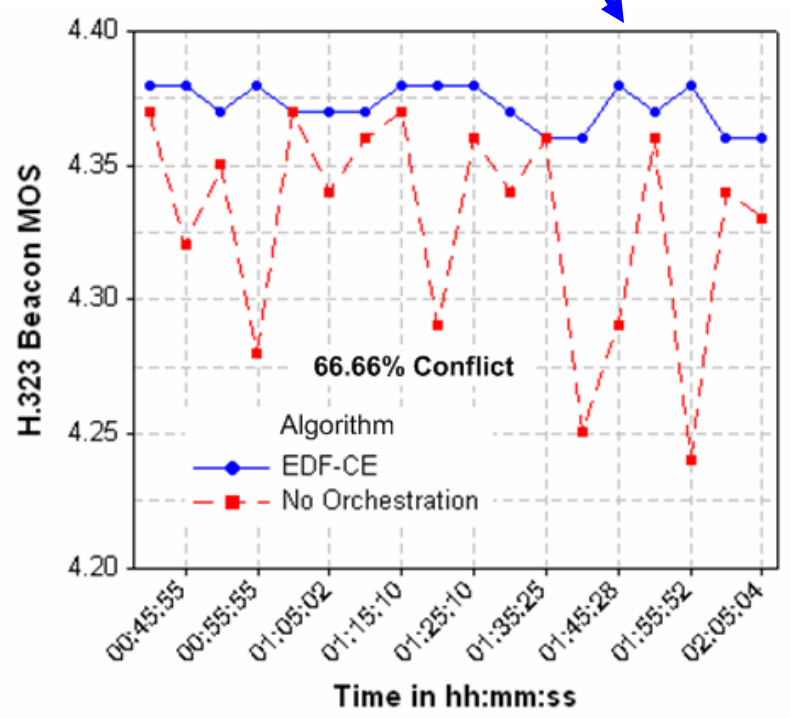
τ_3 = (Site-3, Site-1, Pathchar, 60, 20)

τ_4 = (Site-3, Site-5, Pathload, 60, 5)

τ_5 = (Site-1, Site-4, Iperf, 60, 20)



(a) Between Site-3 and Site-2



(b) Between Site-3 and Site-4

Conclusion

- We formulated network-wide active measurements scheduling as a “real time scheduling” problem
- We proposed an offline scheduling algorithm (EDF-CE) that leverages “concurrent execution” of active measurements to improve schedulability
- We proposed an online recursive pushing algorithm that has the least response time when handling on-demand measurement requests without affecting the offline measurement schedules
- We demonstrated the performance of our algorithms using synthetic task simulations and using actual Internet Testbed measurements

Questions?

Effects of MLA