McCormick
Northwestern Engineering
Department of Electrical Engineering and Computer Science

SatelliteLab: Adding Heterogeneity to Planetary-Scale Network Testbeds

Marcel Dischinger, Andreas Haeberlen, Ivan Beschastnikh, Krishna P. Gummadi and Stefan Saroiu

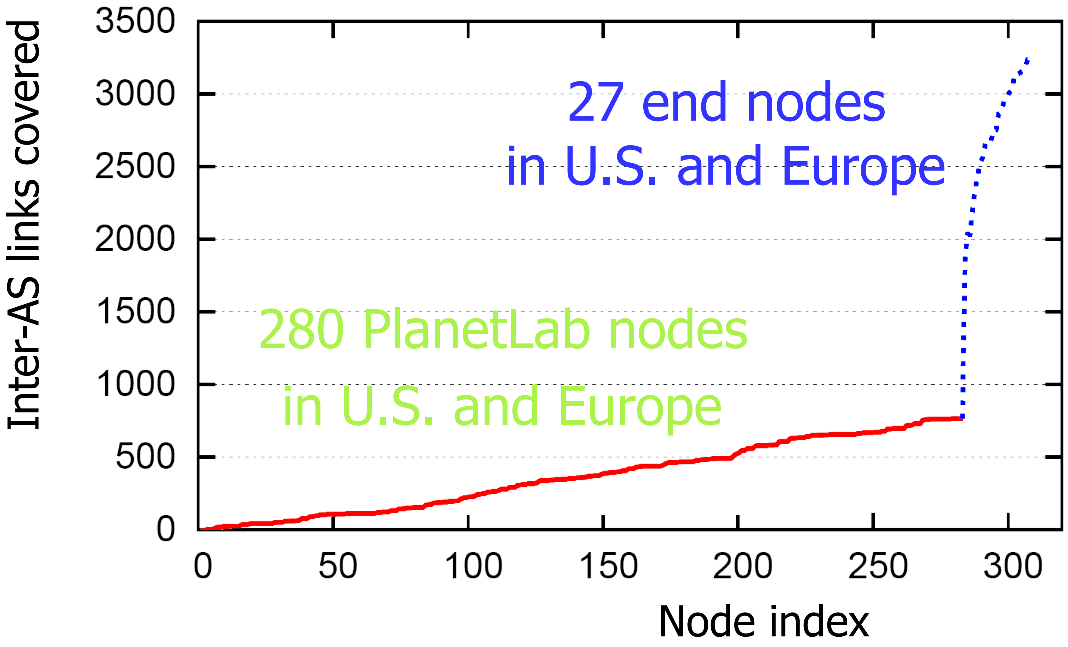
Presented by: Mario Sanchez
Instructor: Fabian Bustamante
Date: 01/13/2012

* Animations and graphs “borrowed” from SIGCOMM08 original slides

Internet Testbeds

- Indispensable for evaluating prototypes of distributed systems:
  - Expose systems to realistic traffic conditions
- Main problem:
  - Nodes connected to well-provisioned research networks
  - Lack the heterogeneity that characterizes the commercial Internet
- Most Internet nodes are in edge networks
  - Internet bottlenecks are in access networks
  - Studies show that most queueing, loss etc. happens there

Growing the testbed is not enough



- More nodes from acad. networks do not improve heterogeneity
- Nodes from the commercial Internet are much more effective
- **Need nodes from the commercial Internet**

Challenges

- Security and accountability concerns
  - Arbitrary experiment code on end-user machines discouraged
  - Administrative control not possible
- Assumptions about capabilities of participating edge nodes are not possible
  - Limited storage and processing resources
  - Uniform OS architecture
- Accessibility concerns
  - Edge nodes are often located behind middle boxes

WHAT ABOUT INCENTIVE CONCERNS?

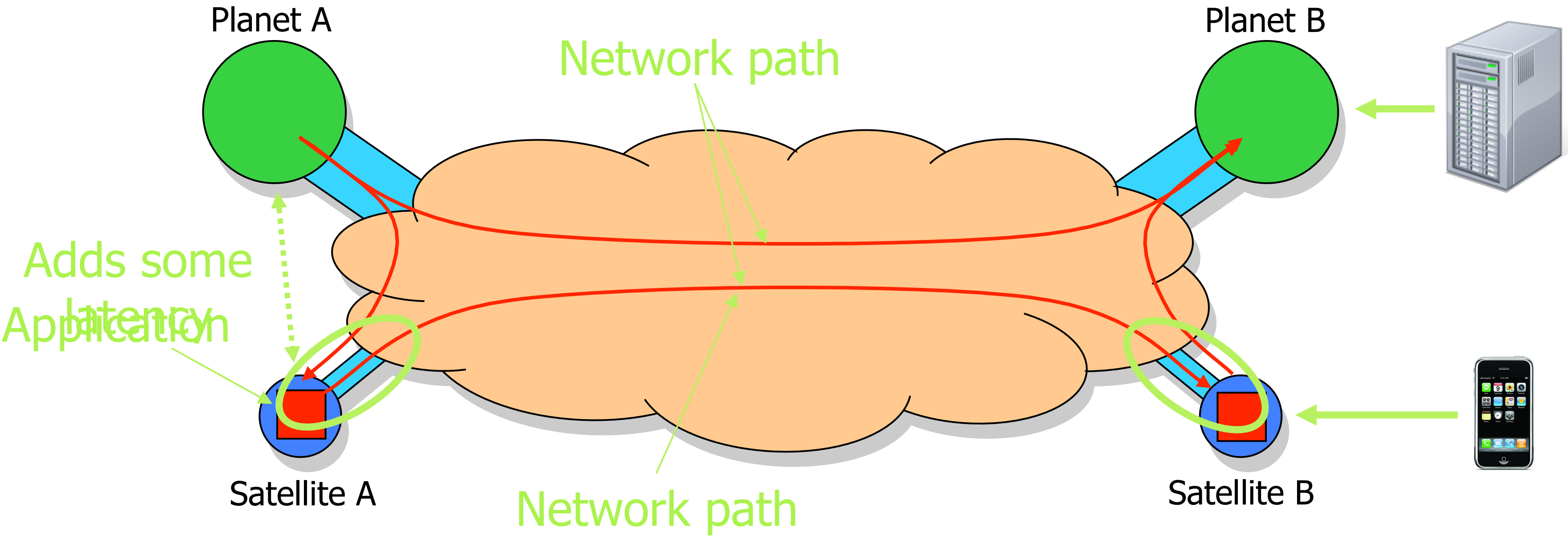
SatteliteLab

- Improve the heterogeneity of current testbeds by recruiting nodes from the Internet edges
- Separate computation and communication
  - Tier 1 (Planets): Powerful, well-connected nodes (from existing testbed)
    Contribute computation power, memory, storage...
  - Tier 2 (Sattelites): Light-weight nodes
    Contribute "just their access-network links"

Goals

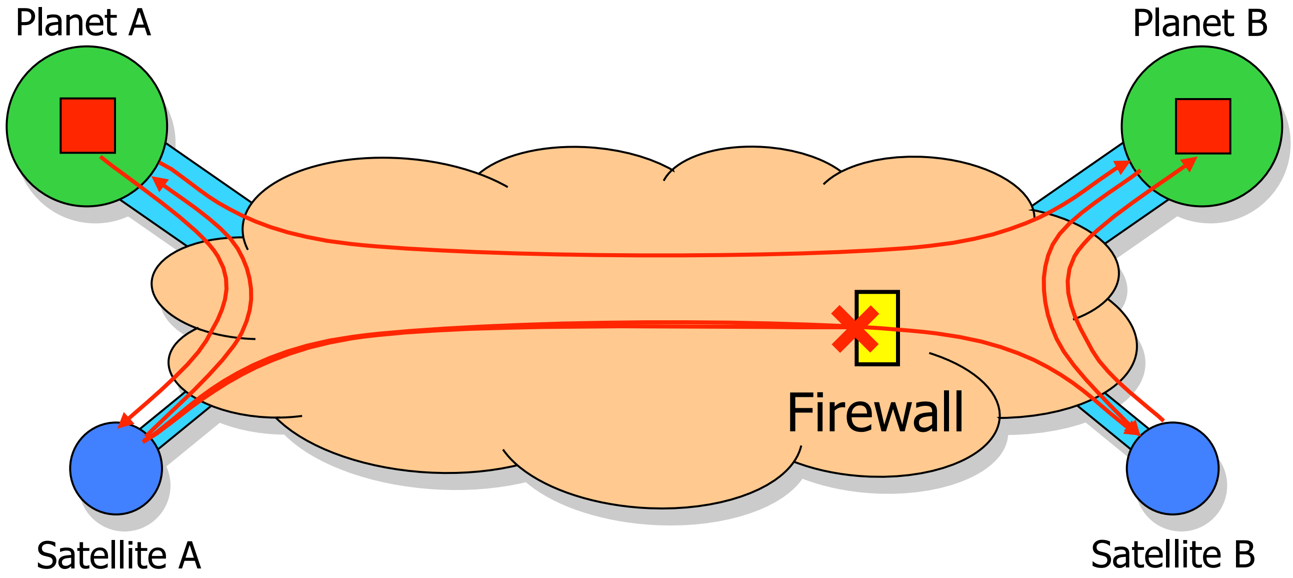
- Preserve the benefits of existing testbeds:
    - Stable software environment,
    - Management of private virtual slices,
    - Extensive API on top of which useful distributed services can be built

Delegation



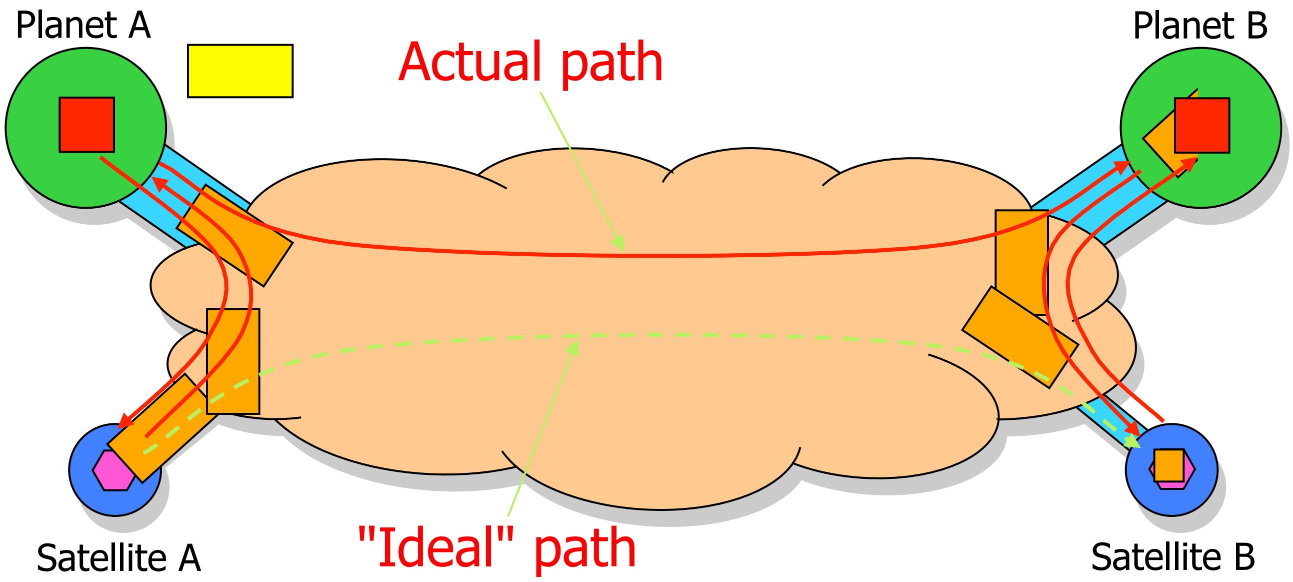
- Delegate code execution to the planets
  - Problem: Changes the network path
  - Misses the access links
- Solution: Send traffic through the satellites

Detour routing



- Two challenges remain:
  - Satellites are sending traffic that may trigger complaints
  - NATs and firewalls may prevent connectivity
- Solution: Detour traffic through the planets

The SatelliteLab detour path



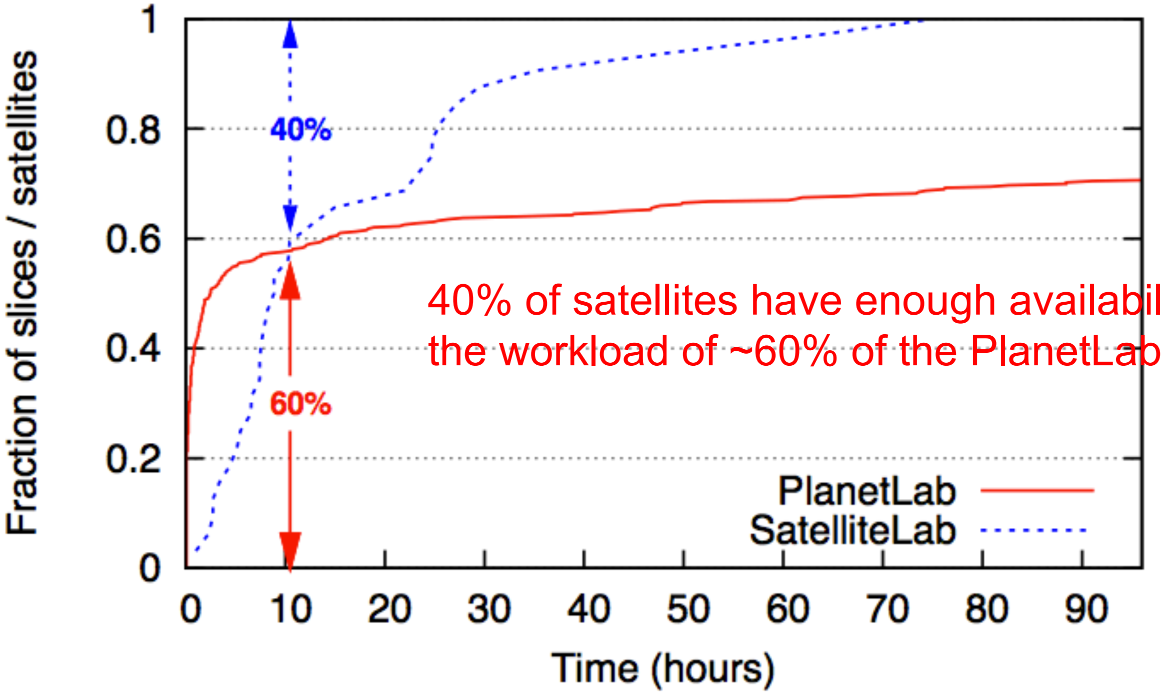
- To send a packet from planet A to planet B:
  1. A intercepts the data packet
  2. A exchanges a probe packet with its satellite
  3. A sends the data packet to B
  4. B exchanges a probe packet with its satellite
  5. B delivers the data packet

Implementation

- The planet proxy
  - Linux daemon with ~2,400 lines of C++ code
  - Creates virtual Ethernet device configured with private subnet
  - Traffic sent by the application is intercepted by local planet proxy
- The satellite helper
  - Implemented in Java with 118 lines of code
  - Registers with closest planet (network latency)
  - Waits for / responds to incoming UDP probes from planet
  - Heartbeat mechanism: (a) inform planet of it’s public IP, (b) prevent address translation rules in the NAT from expiring
  - Does not communicate with any other node

EVALUATION

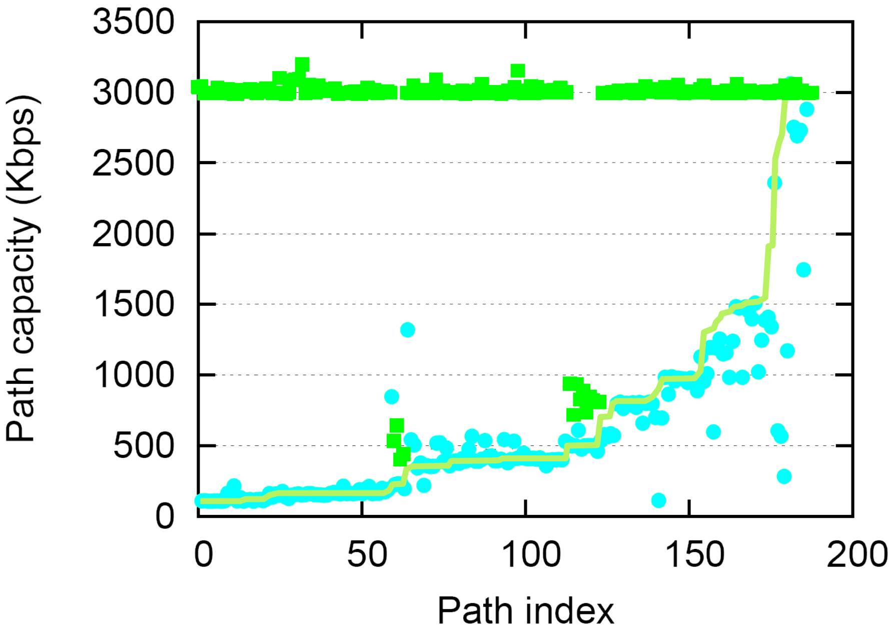
Evaluation: Satellites Availability

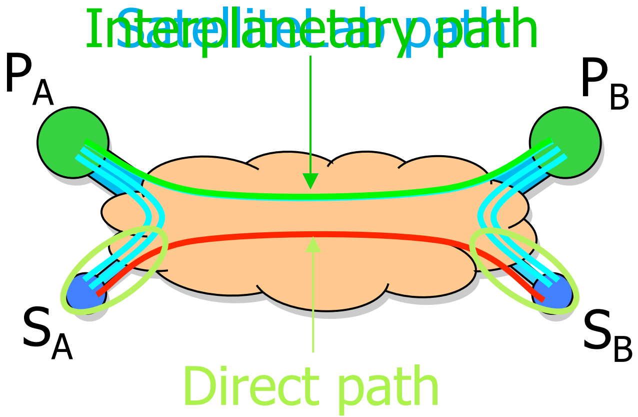


40% of satellites have enough availability to support the workload of ~60% of the PlanetLab slices

Availability of SatelliteLab nodes is sufficient to support potential experiments

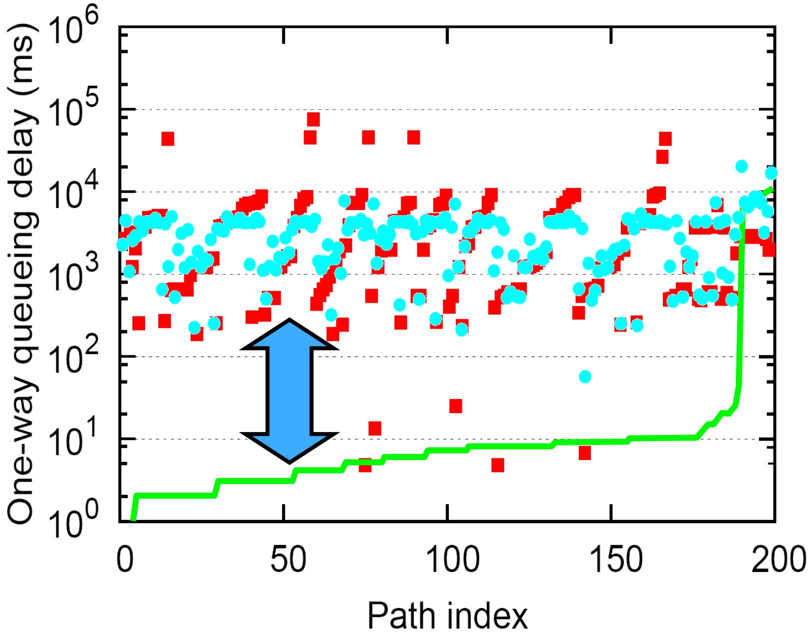
Evaluation: SatelliteLab preserves path capacity

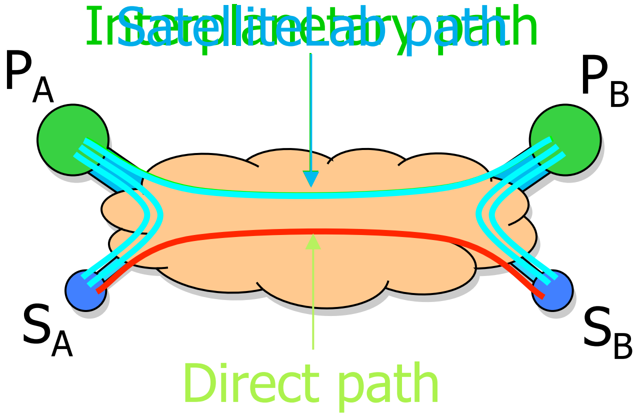




- Direct path vs. interplanetary path
  - Large difference because bottlenecks are in access networks
- Detour routing preserves the path capacity

Evaluation: SatelliteLab preserves queueing delay

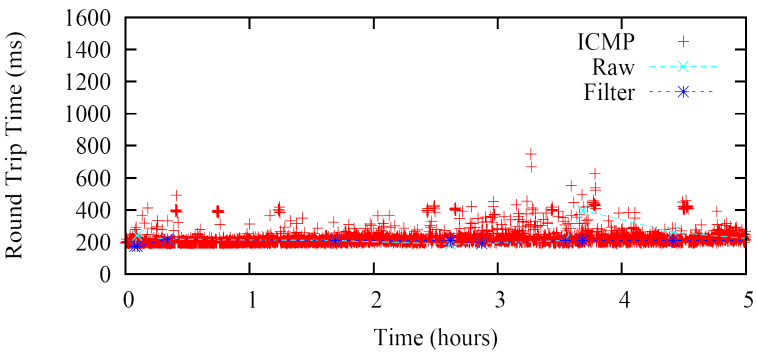


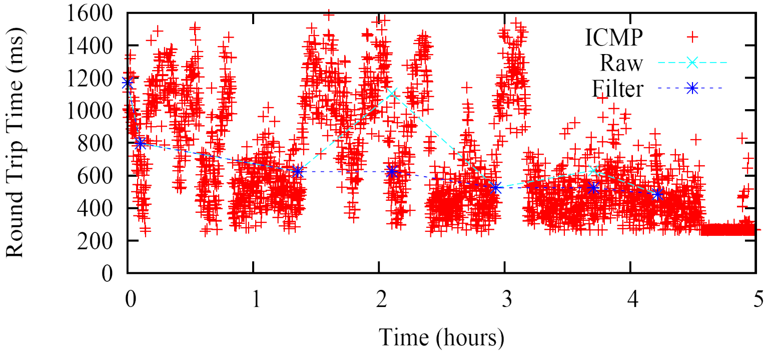


- Same comparison for queueing delay
  - Almost all the queueing is in the access networks
- Detour routing preserves the queueing delay

APPLICATIONS

Application: Network coordinates



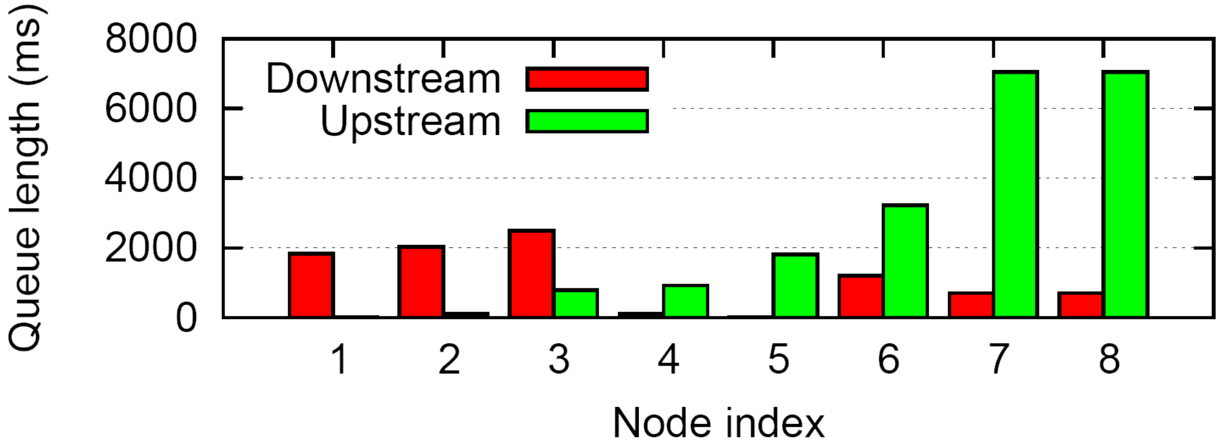


- Need to measure RTT between nodes
- Surprising observation (in previous work):
  - Discarding outliers improves network coordinates on PlanetLab [NSDI'06] but not when deployed with Azureus BitTorrent client [NSDI'07]
  - Reason: Huge RTT variance
- PlanetLab did not predict this observation

Figures taken from [NSDI'07]

Application: Network coordinates

- Experiment with several Azureus nodes on SatelliteLab
  - Observed same effect as in [NSDI'07]
  - Measurements revealed extremely long queues on the broadband access links
  - Azureus downloads fill up the queues → high RTT variance!



McCormick

Northwestern Engineering

Department of Electrical Engineering and Computer Science

DIMES: Let the Internet Measure Itself

Yuval Shavitt and Eran Shir

Presented by: Mario Sanchez
Instructor: Fabian Bustamante
Date: 01/13/2012

DIMES

- The Problem:
  - Main handicap of measurement projects is their limited number of measurement nodes and their location
- Suggested Solution:
  - Move away from small sets of dedicated nodes to a large community of host nodes, running lightweight agents as a background process

DIMES

- Highly distributed, global Internet measurement infrastructure
- Deployment of thousands of light weight measurement agents around the globe
- Aims to measure the structure and evolution of the Internet using a large set of interacting agents
- Goal is to take full snapshots of the Internet at several levels of granularity:
  - At the AS level
  - At the PoP level
  - At the Router Level

DIMES

- Written in Java
- Scripting language, subset of Java called PENny
- Timed scripting is permitted for coordination
- Agent initiated communication through HTTPS
- Open source
- Constraint network resource usage
- High flexibility in the agent, allowing the easy deployment of new types of measurements, and the ability to create complex experiments
- Auto-update mechanism
- Capabilities: TCP/UDP ping, traceroutes

DIMES

- Two types of incentives:
  - **User experience incentives** consist of features such as dynamic visualization of Internet map segments and competition for data contribution
  - **Knowledge incentives** will consist of relevant information to the users, such as reports on ISP performance measures, reports on ease of access to user’s web resources from other parts of the globe, and interfacing with web browsers and P2P applications to help them to optimize.

Comparing DIMES vs. BGP Topologies

- The Route Views project
  - Gather BGP updates from many BGP speakers around the world.
  - AS topologies are inferred from this data

- Methodology
  - Collect traceroute measurements from DIMES agents
  - Traceroutes are integrated in time to produce periodic AS topologies
  - Map IP to ASes based on longest-prefix match from BGP
  - Augment results with ‘whois’ data resolution

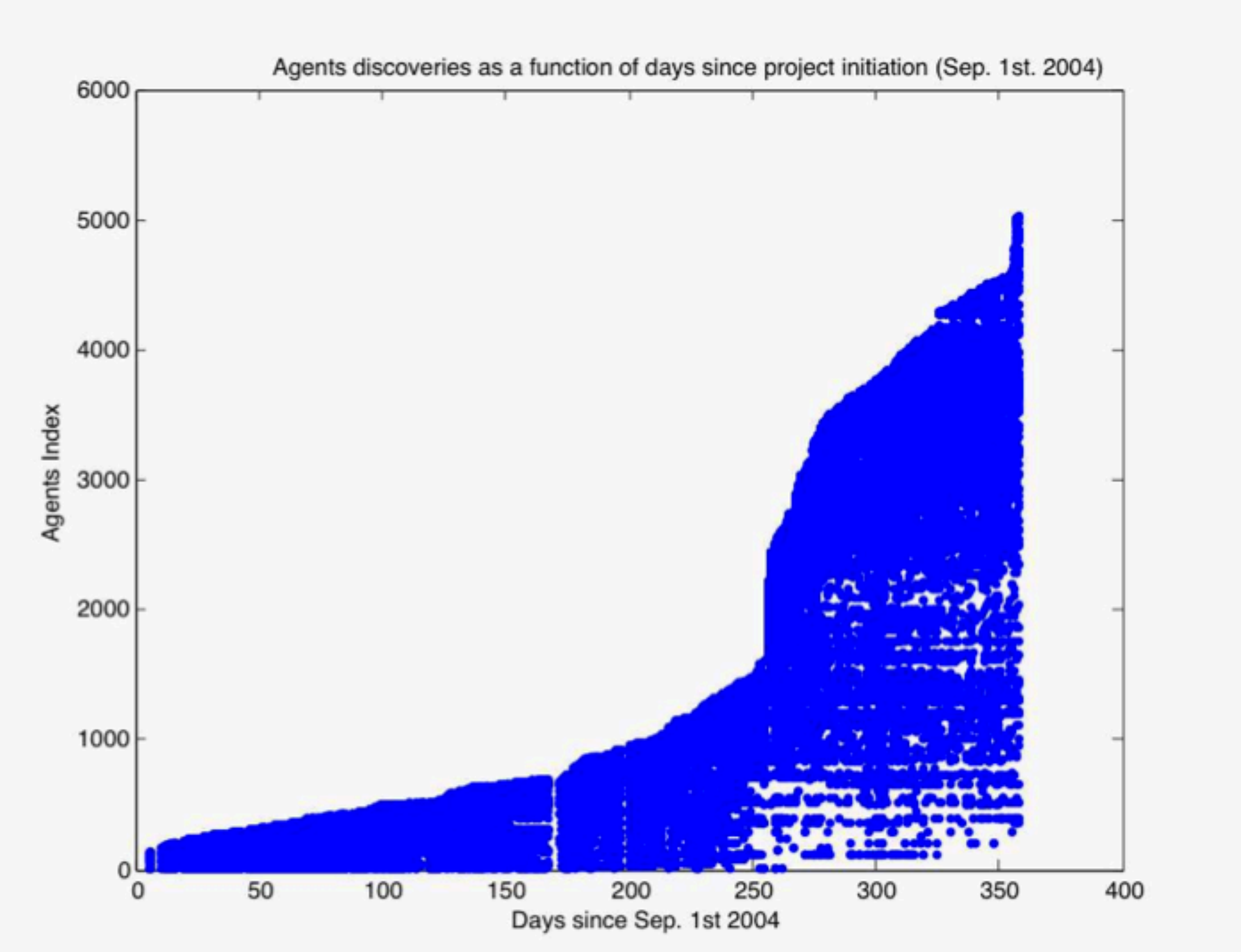
Comparing DIMES vs. BGP Topologies

- Compare four different topologies created from the set of measurements defined above:
  - DIMES topology
  - BGP Topology
  - Complete Topology: unification of above two
  - BGPinDIMES Topology: Topology subgraph which spans only AS nodes that belong to the DIMES topology

Takeaways

- Many of the new links found by DIMES are periphery peer links
- Comparing the degree of nodes in the BGP topology vs. their degree in the Complete topology: about one third of the nodes has a higher degree than perceived by BGP data

Agents contribution



Discussion