Towards a High Quality Path-oriented Network Measurement and Storage System

David Johnson, Daniel Gebhardt, Jay Lepreau

School of Computing, University of Utah

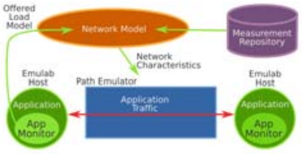
www.emulab.net

Different Goals for our NMS

- Many uses for Internet-scale path measurements:
  - Discover network trends, find paths
  - Building network models
  - Run experiments using models and data
- A different design point on the NMS spectrum:
  - Obtain highly accurate measurements
  - … from a ***resource-constrained***, ***unreliable*** network
  - … for ***multiple simultaneous users***
  - … sometimes at ***high frequency***
  - ... and return results ***fast*** and ***reliably***

Flexlab: a Motivating Use Case

- Problem: real Internet conditions matter, but can make controlled experiments difficult
- Flexlab [NSDI 07]: integrate network models into emulation testbeds (i.e., Emulab)
  - Example: network models derived from PlanetLab
- How it works:
  - Measure Internet paths in real time
  - Clone conditions in Emulab



Requirements

- Shareable
  - Anticipate multiple users
  - Frequent simultaneous probing can cause self-interference, and increase cost
  - ***Amortize cost of measurements*** by removing probe duplication across users
- Reliable
  - Reliably ***buffer, transmit, and store*** measurements
  - Probing & storage should continue when network partitions disrupt control plane

Requirements, cont'd

- Accurate
  - Need best possible measurements for models
- Safe
  - Protect resource-constrained networks and nodes from probing tools, and vice versa
    - And yet support high freq measurements
  - Limit BW usage, reduce probe tool CPU overhead
- Adaptive & controllable
  - Function smoothly despite unreliable nodes
  - Modify parameters of executing probes

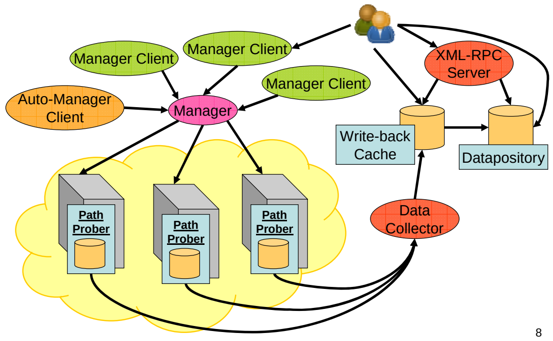
Hard System To Build!

- End-to-end reliability
  - Data transfer and storage, control
  - PlanetLab: overloaded nodes, sched delays
- Measurement accuracy vs resource limits
- => We're not all the way there yet

Flexmon

- A measurement service providing *shared, accurate, safe, reliable* wide area path-oriented measurements
  - ***Reliable*** probing and results transfer & storage atop unreliable networks and nodes
  - ***Accurate***, high freq measurements for multiple users despite network resource limits
  - Transfers and exports results quickly and safely
- Not perfect, but good start
- Deployed on an unreliable network, PlanetLab, for 2+ yrs
- Nearly 1 billion measurements
- Data available publicly via multiple query interfaces and the Web

Flexmon Overview



User Interface

- Authentication through Emulab
- Users request probes through manager clients
  - Type of probe, set of nodes, frequency and duration, and other tool-specific arguments
  - Users can “edit” currently executing probes to change parameters
- Get measurements from caching DB via SQL

Central Management

- Manager applies safety checks to client probe requests:
  - Reject if probe request is over frequency and duration thresholds
  - Can reject if expected bandwidth usage will violate global or per-user limits
    - Estimates future probe bandwidth usage based off past results in write-back cache

Background Measurements

- The ***Auto-manager Client*** requests all-pairs probing for one node at each PlanetLab site
  - Assumption: all nodes at a site exhibit “identical” path characteristics to other sites
  - Chooses least loaded node at each site to avoid latencies in process scheduling on PlanetLab
- Assesses node liveness and adjusts node set
- Uses low probe duty cycle to leave bandwidth for high-freq user probing

Probing

- A ***Path Prober*** on each node receives probe commands from the Manager
- Spawns probe tools at requested intervals
  - Newer (early) generic tool support, although safety not generalized
- Multiple probe modes to reduce overhead
  - One-shot: tool is executed once per interval, returns one result
  - Continuous: tool is executed once; returns periodic results

Probing, cont’d

- Probers maintain a queue of probe commands for each probe type and path, ordered by frequency
  - Serially execute highest-frequency probe
  - All users get at least what they asked for, maybe more
- Trust model: only allow execution of approved probing tools with sanity-checked parameters
- Currently use two tools
  - *fping* measures latency
    - Attempts to distinguish loss/restoration of connectivity from heavy packet loss by increasing probing frequency
  - Modified *iperf* estimates ABW

Collecting & Storing Measurements

- Probers send results to central data collector over UDP
  - Stable commit protocol on both sides
  - Collector drops duplicate results from retransmits
- Not perfectly reliable – i.e., cannot handle node disk failures
- Use write-back cache SQL DB for perf
- Newest results in write-back cache are flushed hourly to long-term storage in Datapository
  - Fast stable commit

Searching the Data

- “Write-back cache” SQL DB
  - Available to Emulab users by default
  - Fast but limited scope
- Datapository containing ***all*** measurements
  - Access upon request
  - Weekly data dumps to www
- XMLRPC server
  - Can query both DBs over specific time periods
  - More expressive query power (i.e., FullyConnectedSet, data filtering, etc)

Deployment & Status

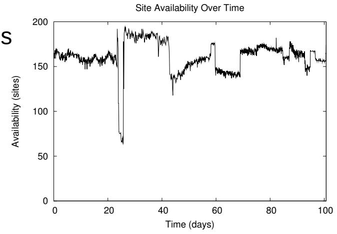
- Probers run in an Emulab experiment, using Emulab’s portal to PlanetLab
- Managers, clients, and data collectors run on a central Emulab server
  - Use secure event system for management
- Running on PlanetLab for over 2 years
  - Some architecture updates, but largely unchanged over past year
  - Some system “hiccups” – i.e., our slice has been bandwidth-capped by PlanetLab
  - Set of monitored nodes changes over time

Measurement Summary

- Many measurements of pairwise latency and bandwidth
- Latency measurements are 89% of total
  - 17% are failures (timeouts, name resolution failures, ICMP unreachable)
- Available bandwidth estimates are 11%
  - Of these, 11% are failures (mostly timeouts)

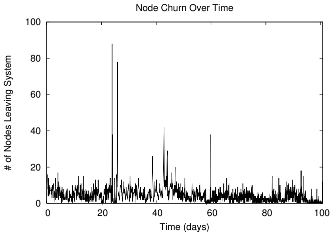
PlanetLab Sites

- Logfile snapshot of 100-day period
- Median of 151 sites
- System “restart” is the big drop

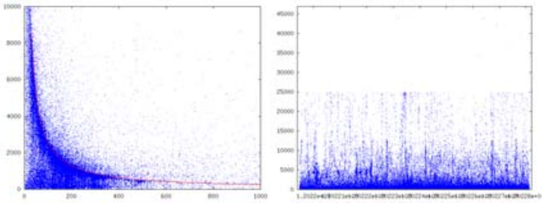


Node Churn

- Typically 250-325 nodes in slice
- Churn: number of newly unresponsive nodes at periodic liveness check



Brief Look at Some Data



- 24-hour snapshot from Feb
  - 100k+ ABW samples; 1M+ latency samples
- Latency vs bandwidth: curve approx BDP
  - Outliers due to method

Related Work

- S3: scalable, generic probing framework; data aggregation support
  - We need fast & reliable results path
  - Need support to limit probe requests when necessary
  - Also need adaptability for background measurements
- Scriptroute: probe scripts executed in safe environment, in custom language
  - No node-local storage, limited data output facilities
- Others that lack shareability or reliable storage path; see paper

More To Be Done…

- More safety
  - LD_PRELOAD, libpcap to track usage tool-agnostically at probe nodes
  - distributed rate limiting [SIGCOMM ’07]; scale probe frequency depending on use
- Add another user data retrieval interface (pubsub would be nice)
- Increase native capabilities of clients
  - Adaptability, liveness

Conclusion

- Developed an accurate, shareable, safe, reliable system
- Deployed on PlanetLab for 2+ years
- Accumulated lots of publicly-available data

Data!

- http://utah.datapository.net/flexmon
  - Weekly data dumps and statistical summaries
- Write-back cache DB available to Emulab users
- SQL Datapository access upon request; ask testbed-ops@flux.utah.edu