The Flexlab Approach To Realistic Evaluation of Networked Systems

Robert Ricci, Jonathon Duerig, Pramod Sanaga, Daniel Gebhardt, Mike Hibler, Kevin Atkinson, Junxing Zhang, Sneha Kasera, and Jay Lepreau

NSDI 2007
April 12, Cambridge, MA
Emulators

Examples: ModelNet and Emulab

The Good: Control, repeatability, wide variety of network conditions

The Bad: Artificial network conditions
Overlay Testbeds

- Examples: RON and PlanetLab
- The Good: Real network conditions, deployment platform
- The Bad: Overloaded, few privileged operations, poor repeatability, hard to develop/debug on
Evaluating Networked Systems: Flexlab
Goal: Real Internet within Emulator
The Flexlab Approach

Measure
The Flexlab Approach

Measure

Model
The Flexlab Approach

Measure → Model → Emulate
The Flexlab Approach

Measure → Model → Emulate
The Flexlab Approach

- Measure
- Model 2
- Emulate
Key Points

• Software framework for pluggable network models

• Application behavior can drive measurements & model in real-time

• Application-Centric Internet Modeling
  – High fidelity measurement/emulation technique
  – Includes new techniques for ABW measurement
More in the Paper

- Flexible network measurement system
- Network stationarity results
- Two straightforward network models
- Shared bottleneck analysis
- PlanetLab scheduling delay measurements
Flexlab Architecture
Flexlab: Application
Flexlab: Application Monitor
Flexlab: Network Model

- Offered Load Model
- Network Model
- Emulab Host
- Application
- App Monitor
- Application Traffic
- Application
- App Monitor
- Emulab Host
Flexlab: Measurement Repo.
Flexlab: Path Emulator

Network Model

Measurement Repository

Offered Load Model

Emulab Host

Path Emulator

Network Characteristics

Application

App Monitor

Application Traffic

App Monitor
ACIM: Application-Centric Internet Modeling
Imagine Ideal Fidelity

Application Traffic

PlanetLab Host

Internet

PlanetLab Host

Emulab Host

Application

Emulab Host

Application
ACIM Architecture
ACIM Design Challenges

- Determining when to drop packets
- Finding relationship between throughput and ABW
- Extension to UDP
- CPU starvation on PlanetLab
  - Host artifacts in throughput
  - Packet loss in libpcap
ACIM Path Emulator Parameters

Available bandwidth

Packets enter

Queuing delay

Packets leave

All other delay

Available bandwidth

23
All Other Delay

- Base RTT: Smallest RTT seen recently [Vegas 95]
- Packets saw little or no queueing delay
Packet Loss

- Caused by full queue at bottleneck link
  - Difficult to measure directly
- So measure queue length in time: Max recent RTT - Base RTT
Throughput and ABW
Throughput and ABW
Throughput and ABW

- Offered load
- Available bandwidth
- Measured throughput

Time

Delay
Throughput and ABW

<table>
<thead>
<tr>
<th>Bandwidth</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Offered load</td>
<td>Available bandwidth</td>
</tr>
<tr>
<td></td>
<td>Measured throughput</td>
</tr>
<tr>
<td>Delay</td>
<td></td>
</tr>
<tr>
<td></td>
<td>RTT</td>
</tr>
</tbody>
</table>
Throughput and ABW

Bandwidth

- Offered load
- Available bandwidth
- Measured throughput

Delay

RTT
Throughput and ABW

![Graph showing Throughput and ABW](image-url)

- **Bandwidth**
  - Offered load
  - Available bandwidth
  - Measured throughput

- **Time**

- **RTT**

- **Delay**
Throughput and ABW

- **Throughput** and **Available Bandwidth (ABW)** are key performance indicators in network communications.

In the graph:
- **Offered load** increases and then decreases, indicating the load placed on the network.
- **Available bandwidth** remains relatively constant, illustrating the network's capacity.
- **Measured throughput** shows a corresponding decrease as the load increases, reflecting the network's performance under varying loads.
- **RTT (Round-Trip Time)** remains relatively stable, suggesting consistent network latency.

These metrics help in understanding the network's efficiency and capacity handling load changes.
Throughput and ABW
Throughput and ABW

• If (throughput > last ABW measurement), use new value

• Else, look for indications that throughput has reached ABW
  – Socket buffer is filling up AND
  – Recent RTTs have been increasing
  – Using linear regression
ACIM Features

- Precise: assesses only relevant parts of the network
  - Scales in nodes and paths

- Complete: automatically captures all relevant network behavior
  - Simpler to measure e2e effects than find causes
  - Detects rare and transient effects
  - Evokes all reactive network behaviors (except content-based)
  - Rapidly tracks conditions
ACIM Accuracy

• Is ACIM path emulation accurate?
• Is it accurate at fine granularity?
Methodology

- iperf runs in Emulab
- Measurement Agent runs on PlanetLab at UT Austin and AT&T Research
- We added transient TCP cross traffic between these sites
TCP iperf Throughput

Throughput (Mbps)

Time (seconds)
A Real Application

- Does ACIM give accurate results for a real, complicated application?
A Real Application

• Does ACIM give accurate results for a real, complicated application?

• ... does PlanetLab?
A Real Application

• Does ACIM give accurate results for a real, complicated application?

• … does PlanetLab?

• Can we discover ground truth?
Methodology: BitTorrent

- Two simultaneous instances of reference BitTorrent:
  - One on PlanetLab
  - One in Flexlab
- Eight nodes in US and Europe: One seed, seven clients
- We reduced randomness in BT ... but some still remains
BitTorrent w/ CPU Reservation
BitTorrent w/ CPU Reservation

**PlanetLab:** 5.2 Mbps average

**Flexlab:** 5.4 Mbps average
BitTorrent w/o CPU Reservation

PlanetLab: 2.3 Mbps average

Flexlab: 5.8 Mbps average
BitTorrent Bottom Line

- Conclusion: For this experiment, both Flexlab and PlanetLab with CPU reservations give accurate results
  - PlanetLab alone does not
- CPU availability on PlanetLab hurts BitTorrent
- ACIM reduces host resource needs on PlanetLab for this experiment
  - BitTorrent: 36-76% CPU
  - ACIM Agent: 2.6% CPU
  - Factor of 15 - 30 CPU
  - Factor of 4 memory
The Future?

• No need to perfect in PlanetLab:
  – Full resource isolation
  – Total control over hosts
  – Orthogonal control network

• ... use in the emulators that already have them

• Use PlanetLab nodes as NICs

• Conserve resources for deployed services with end users
Conclusion

• New approach to evaluating networked systems
• Separates the network model
• Designed to leverage vibrant measurement and modeling community
• Couples an emulator to an overlay testbed
• ACIM high fidelity emulation technique
• Contact testbed-ops@emulab.net to use
Backup Slides
Why not just add more nodes to every PlanetLab site?

- Remaining problems:
  - Poor repeatability
  - Hard to develop/debug
  - No privileged operations

- Some malicious traffic cannot be tested

- Some Flexlab network models reduce network load

- Emulab node pool stat muxed and shared more efficiently than per-site pools

- Overload can (will?) still happen with PL's pure shared-host model

- Major practical barriers: admin, cost
Flexlab and VINI

• Entirely different kinds of realism and control

• Flexlab: passes "experiment" traffic over shared path
  – Real Internet conditions from other traffic on same path, but app. traffic is not from real users
  – Control: of all software
  – Environment: friendly local dev. environ, dedicated hosts

• VINI: can pass "real traffic" over dedicated link
  – Real routing, real neighbor ISPs, potentially traffic from real users, but network resources are not realistic/representative
  – Dedicated pipes with dedicated bandwidth, that insulate experiment from normal Internet conditions
  – Control: restricted to VINI's APIs (Click, XORP, etc.)
  – Environment: distributed environ; shared host resources
## Change Point Analysis

<table>
<thead>
<tr>
<th>Path</th>
<th>High</th>
<th>Low</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>Asia to Asia</td>
<td>39%</td>
<td>15%</td>
<td>12%</td>
</tr>
<tr>
<td>Asia to Commercial</td>
<td>0.13%</td>
<td>2.9%</td>
<td>0.5%</td>
</tr>
<tr>
<td>Asia to Europe</td>
<td>0.59%</td>
<td>3.4%</td>
<td>0.02%</td>
</tr>
<tr>
<td>Asia to I2</td>
<td>56</td>
<td>56</td>
<td>0</td>
</tr>
<tr>
<td>Commercial to Commercial</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Commercial to Europe</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Commercial to I2</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>I2 to I2</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>I2 to Europe</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Europe to Europe</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Europe to I2</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>I2 to Asia</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>I2 to Commercial</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>I2 to Commercial</td>
<td>0%</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>

56
Simple Static Model
Simple Dynamic Model

Dynamic Network Model

Application Network Model from Monitor

Datapository
Flexmon Architecture

- Shared
- Reliable
- Safe

- Adaptive
- Controllable
- Accommodates high-performance data retrieval
CPU Starvation on PlanetLab

- **Host Artifacts**
  - Long period when agent can't read or write
  - Empty socket buffer or full receive window
  - Solution: Detect and ignore

- **Packet loss from libpcap**
  - Long period without reading libpcap buffer
  - Many packets are dropped at once
  - Solution: Detect and ignore
Reverse Path Congestion

- Can cause ack compression
- Throughput Measurement
  - Throughput numbers become much noisier
  - We abuse the TCP timestamp option
  - PlanetLab: homogeneous OS environment
  - Extending it would require hacking client
- RTT Measurement
  - Future work
Initial Conditions

- Needed to bootstrap ACIM
  - ACIM uses traffic to generate conditions
  - But conditions must exist for first traffic

- We created a measurement framework
  - All pairs of sites are measured
  - Put data into measurement repository
  - Set initial conditions to latest measurements
Simultaneous TCP iperf

Throughput (Mbps)

Time (seconds)

PlanetLab

Flexlab with ACIM
Repeatability vs. Fidelity

Emulab

Static

Dynamic

PlanetLab

ACIM

General Internet Model
Throughput and ABW

Agent `write()`s

Packets On The Wire

Data

ACK

ACK

Data

ACK

Data

ACK

Data

ACK

Throughput

Avail-BW?

Time

Application Offered Load
Currently available for Beta Testing

http://www.flux.utah.edu/flexlab
UDP Streaming Video

Throughput (Mbps) vs. Time (seconds) chart.
Opens Up New Questions

- Further validation
- Accuracy tests at runtime
  - Similar in spirit to Emulab's linktest
- Use to compare models
  - Find which models most appropriate for different classes of applications
- Replay for ACIM
- Study fidelity of different software combinations
  - Different TCP implementation or OS in Emulab