Emulab Introduction

Shiv Kalyanaraman
shivkuma@ecse.rpi.edu
Google: “Shiv RPI”
Recall: Simulation vs Measurement/Implementation

1. Real system is more credible, but more complex – lot of auxiliary concerns & murphy’s law strikes often!
   - But simulation must be thought of as a first step to real implementation
   - (I.e. to get a stable design that must be validated by implementation and/or analysis)

2. Measurement of Internet traffic may not be the same as measurement tomorrow (real, but still random samples!)
   - Representative measurement traces can be used to drive simulation (I.e. trace-driven simulation)

3. New emulation platforms: Utah’s emulab (next slide)
   - Takes out the configuration complexity from small/medium sized real experiments!

4. Bottom line: mix and match both tools depending upon the problem at hand
Utah Emulab and Click: Emulation and Modular Implementation Platforms

Utah’s Emulab Testbed: control & interconnect the kernels of 100s of machines just by using ns-2 scripting!!!

MIT’s Click Modular Router
On Linux:
Forwarding Plane Implns

Modular  Router
This is done! You just write ns-2 scripts!
What is Emulab?

- A configurable Internet emulator in a room
  - Today: 200 nodes, 500 wires, 2x BFS (switch)
  - virtualizable topology, links, software
- Bare hardware with lots of tools
- An instrument for experimental CS research
- Universally available to any remote experimenter
- Simple to use: just write ns-2 scripts
Why?

- "We evaluated our system on five nodes."
  - job talk from university with 300-node cluster
- "We evaluated our Web proxy design with 10 clients on 100Mbit ethernet."
- "Simulation results indicate ..."
- "Memory and CPU demands on the individual nodes were not measured, but we believe will be modest."
- "The authors ignore interrupt handling overhead in their evaluation, which likely dominates all other costs."
- "Resource control remains an open problem."
Why 2

- “You have to know the right people to get access to the cluster.”
- “The cluster is hard to use.”
- “<Experimental network X> runs FreeBSD 2.2.x.”
- “October’s schedule for <experimental network Y> is…”
- “<Experimental network Z> is tunneled through the Internet”
Key Design Aspects

- Allow experimenter complete control
- … but provide *fast* tools for common cases
  - OS’s, disk loading, state mgmt tools, IP, traffic generation, batch, …
- Virtualization
  - of *all* experimenter-visible resources
  - node names, network interface names, network addresses
  - Allows swapin/swapout
Design Aspects (cont’d)

- Flexible, extensible, powerful allocation algorithm

- Persistent state maintenance:
  - none on nodes
  - all in database
  - leverage node boot time: only known state!

- Separate control network

- Familiar, powerful, extensible configuration language: *ns*
Some Unique Characteristics

- User-configurable control of “physical” characteristics: shaping of link latency/bandwidth/drops/errors (via invisibly interposed “shaping nodes”), router processing power, buffer space, ...

- Node breakdown (old):
  - 40 core, 160 edge
More Unique Characteristics

- Capture of low-level node behavior such as interrupt load and memory bandwidth
- User-replaceable node OS software
- User-configurable physical link topology
- Completely configurable and usable by external researchers, including node power cycling
An “Experiment”

- emulab’s central operational entity
- Directly generated by an ns script,
- … then represented entirely by database state

- Steps: Web, compile ns script, map, allocate, provide access, assign IP addr, host names, configure VLANs, load disks, reboot, configure OS’s, run, report
Mapping Example
Automatic mapping of desired topologies and characteristics to physical resources

- Algorithm goals:
  - minimize likelihood of experimental artifacts (bottlenecks)
  - “optimal” packing of multiple simultaneous experiments
  - Extensible for heterogeneous hardware, software, new features
- Randomized heuristic algorithm: simulated annealing
- May move to genetic algorithm
Virtual Topology
Mapping into Physical Topology

Based upon slides from Jay Lepreau, Utah
For more info....