How To Use It ...

- Submit ns script via web form
- Relax while emulab …
  - Generates config from script & stores in DB
  - Maps specified virtual topology to physical nodes
  - Provides user accounts for node access
  - Assigns IP addresses and host names
  - Configures VLANs
  - Loads disks, reboots nodes, configures OSs
  - Yet more odds and ends …
  - Runs experiment
  - Reports results
- Takes ~3 min to set up 25 nodes

Who's Using It?

- 22 projects have used it (18 external)
  - Plus several class projects
- Two OSDI'00 and three SOSP'01 papers
  - 20% SOSP general acceptance rate
  - 75% SOSP acceptance rate for emulab users!
- More emulabs:
  - Planned or in progress: Kentucky, CMU, Duke, Cornell
  - Others considering
- Available today at www.emulab.net

emulab:
A Public Emulation Platform for Research in Distributed Systems and Networks

Jay Lepreau*, Chris Alfeld, Shashi Guruprasad*,
Mike Hibler, Mac Newbold*, Rob Ricci,
Leigh Stoller, Brian White*

University of Utah

http://www.emulab.net

(* = here at SOSP)

Why?

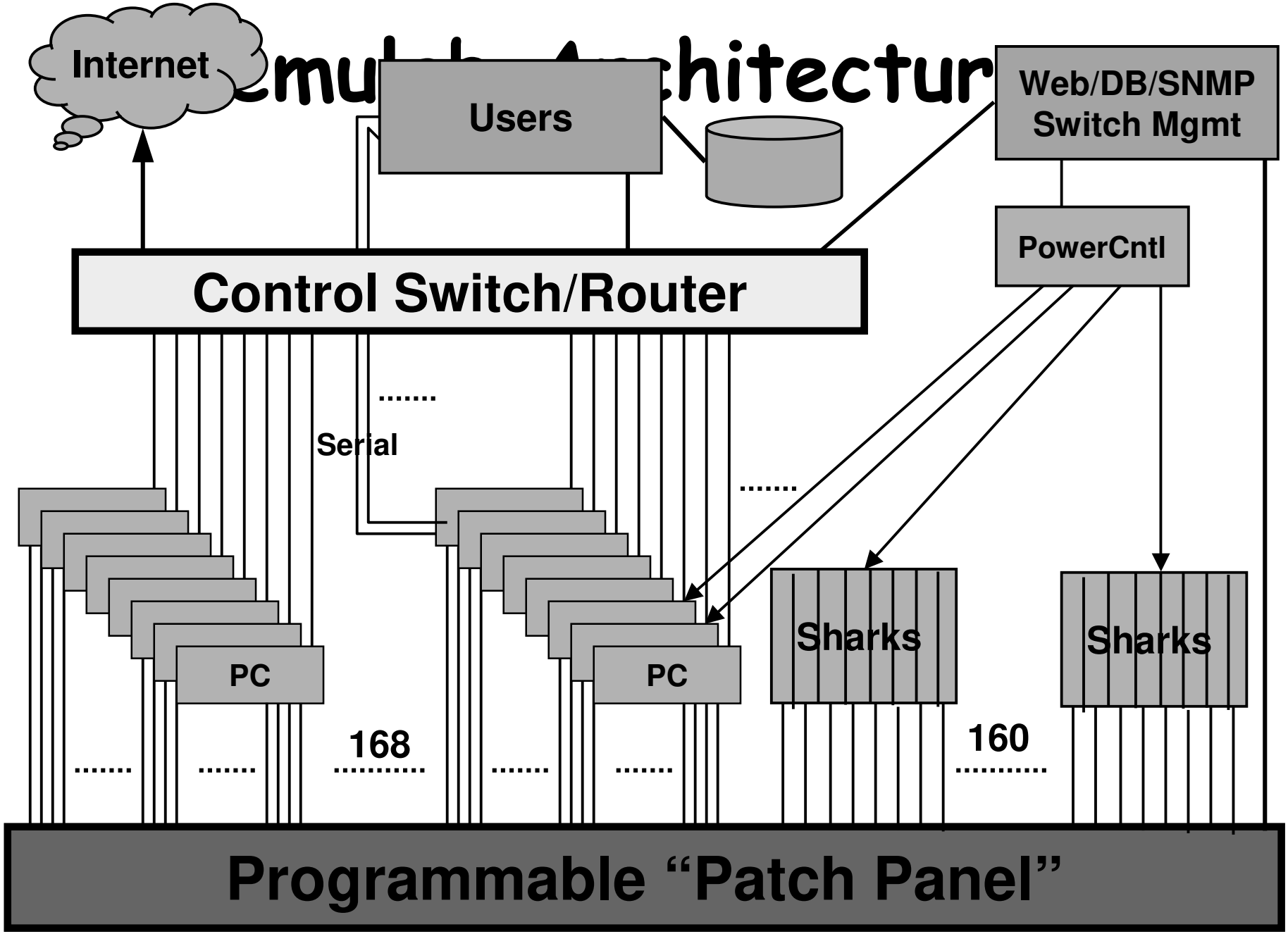
- "We evaluated our system on five nodes."
  -job talk from university with 300-node cluster
- "We evaluated our Web proxy design with 10 clients on 100Mbit ethernet."
- "Simulation results indicate ..."
- "Memory and CPU demands on the individual nodes were not measured, but we believe will be modest."
- "The authors ignore interrupt handling overhead in their evaluation, which likely dominates all other costs."
- "You have to know the right people to use the cluster."
- "The cluster is hard to use."
- "<Experimental network X> runs FreeBSD 2.2.x."
- "February's schedule for <Testbed Y> is..."
- "<Network Z> is tunneled through the Internet"

328 nodes, 3 big Ciscos

1646 cables (25,000 ft/5 miles)

What?

- An instrument for experimental CS research
- A *completely configurable* "Internet emulator" in a room
  - At its core, it's bare hardware, with…
  - … complete remote access and control
- But, also *simple to use*
  - Lots of fast tools for common case
    - Automatic topology, node and link configuration
    - Automatic traffic generation
  - Universally available
    - Universities, research labs, companies
  - Zero-penalty for remote research


Internet
Users
Web/DB/SNMP
Switch Mgmt
PowerCntl
Control Switch/Router
Serial
PC
PC
Sharks
Sharks
168
160
Programmable “Patch Panel”

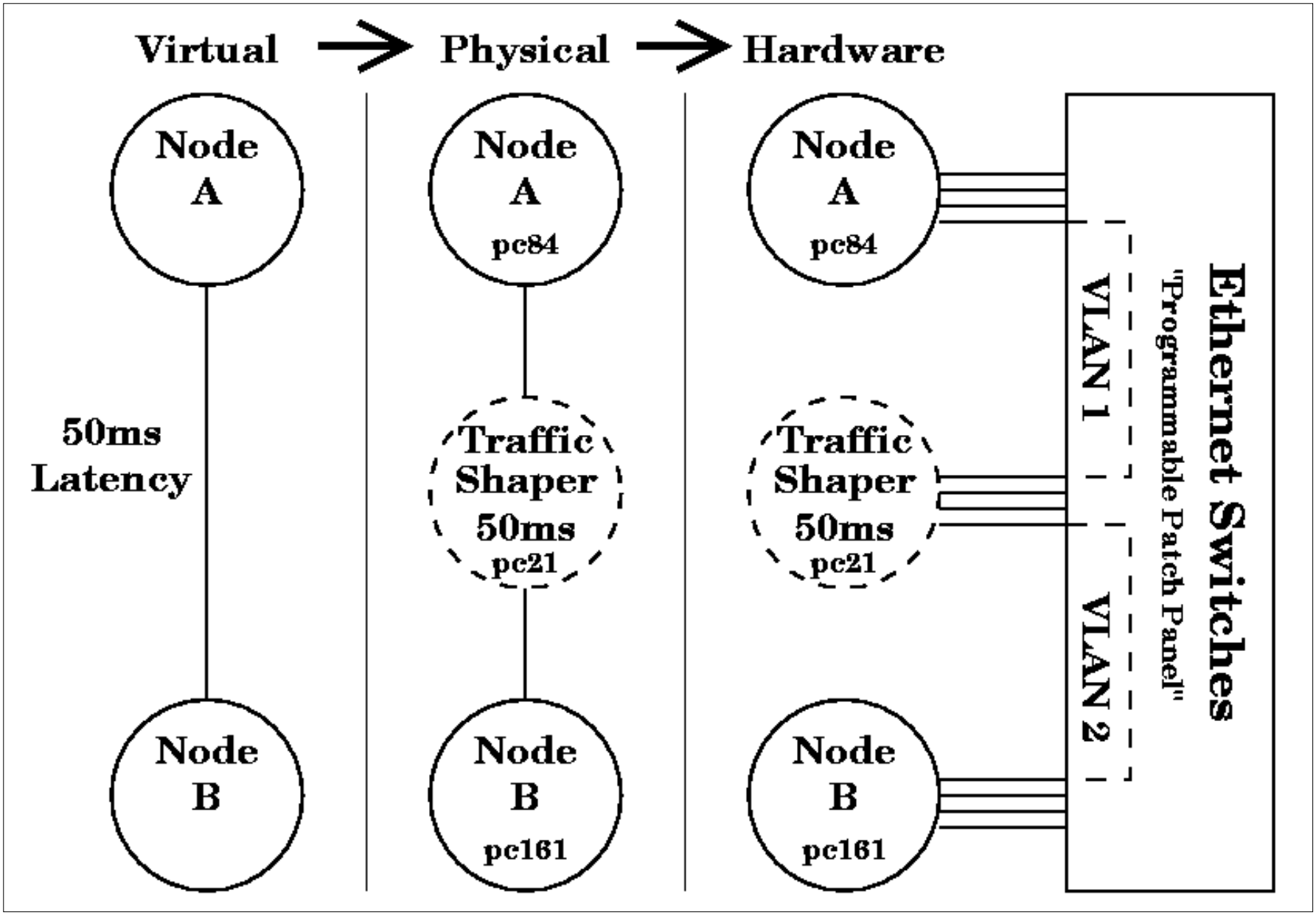
Key Design Aspects

- Allow experimenter complete control
  - Configurable link bandwidth, latency, and loss rates, via transparently interposed "traffic shaping" nodes that provide WAN emulation
- … but provide *fast* tools for common cases
  - OS's, state mgmt tools, IP, batch, …
  - Disk loading – 6GB disk image FreeBSD+Linux
    - Unicast tool: 88 seconds to load
    - Multicast tool: 40 nodes simultaneously in < 5 minutes
- Virtualization
  - of *all* experimenter-visible resources
  - node names, network interface names, network addrs
  - Allows swapin/swapout, easily scriptable

Key Design Aspects (cont'd)

- Flexible, extensible, powerful allocation algorithm
  - Matches desired "virtual" topology to currently available physical resources
- Persistent state maintenance:
  - none on nodes, all in database
  - work from known state at boot time
- Familiar, powerful, extensible configuration language: *ns*
- Separate, isolated control network

Experiment Creation Process



Some Issues and Challenges

- Network management of unknown and untrusted entities
- Security (users get root on nodes!)
- Scheduling of experiments
- Calibration, validation, and scaling
- Artifact detection and control
- NP-hard virtual --> physical mapping problem
- Providing a reasonable user interface
- ….

Automatic mapping of desired topologies and characteristics to physical resources

- NP-hard problem: graph to graph mapping
- Algorithm goals:
  - Minimize likelihood of experimental artifacts (bottlenecks)
  - "Optimal" packing of many simultaneous experiments
  - Extensible for heterogeneous hardware, software, new features
- Randomized heuristic algorithm: simulated annealing
- Typically completes in < 1 second

Ongoing and Future Work

- Wireless nodes, mobile nodes
- IXP1200 nodes, tools, code fragments
  - Routers, high-capacity shapers
- Simulation/emulation transparency
- Event system
- Scheduling system
- Linked set of testbeds
- Data capture, logging, and visualization tools
- Microsoft OSs, high speed links, more nodes!

New Stuff: Extending to Wireless and Mobile

- Problems with existing approaches
  - Same problems as wired domain
  - But worse (simulation scaling, ...)
  - And more (no models for new technologies, ...)

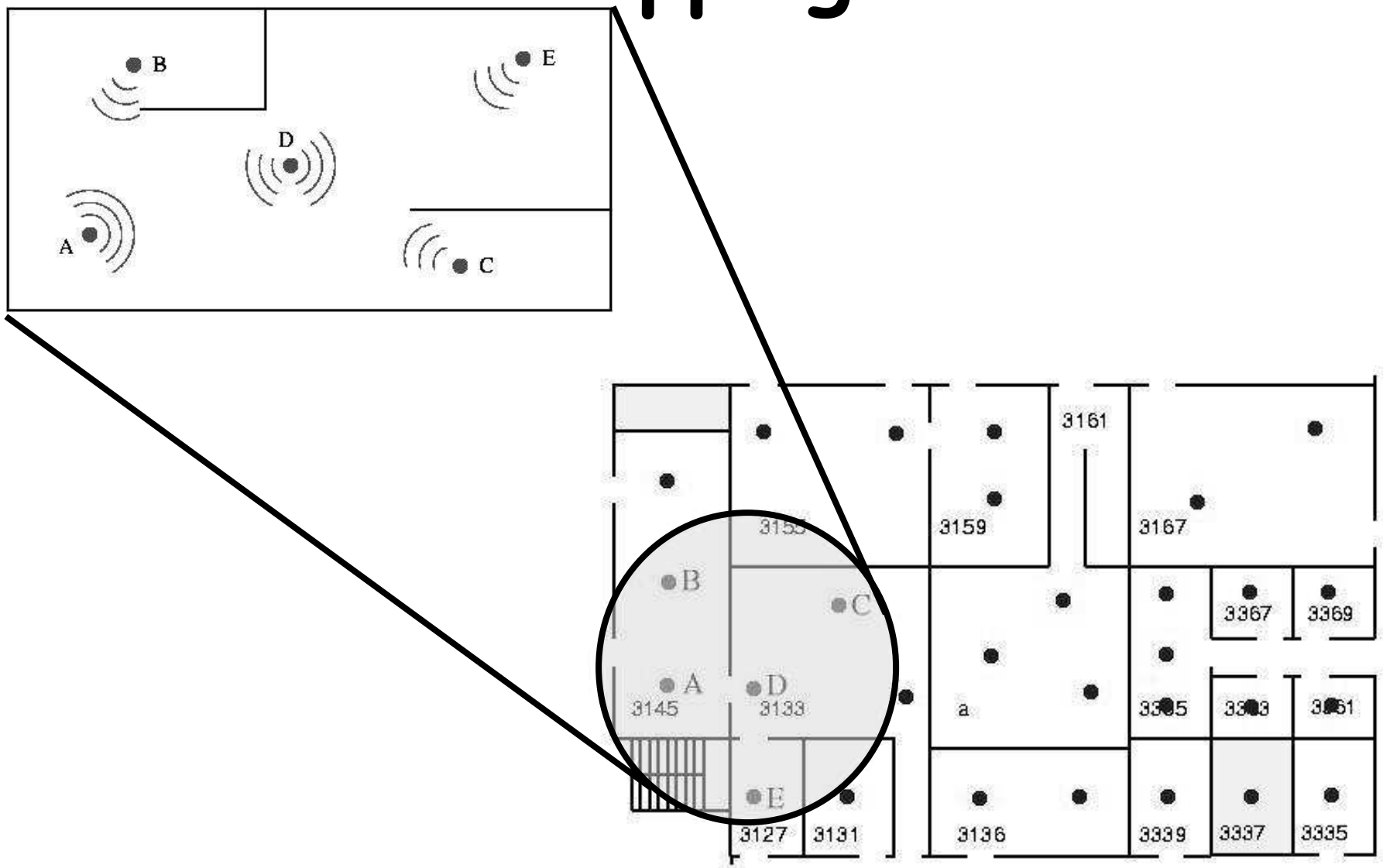
Our Approach: Exploit a Dense Mesh of Devices

- Density enables broad range of emulation
- Wireless
  - Deploy devices throughout campus
  - Measure NxN path characteristics
    (e.g. power, interference, bit error rate)
  - Employ diversity: 900 MHz, Bluetooth, IEEE 802.11
- Mobile
  - Leverage passive "couriers"
    - Assign PDAs to students walking to class
    - Equip public transit system with higher-end devices
  - Provides a realistic, predictable mobile testbed

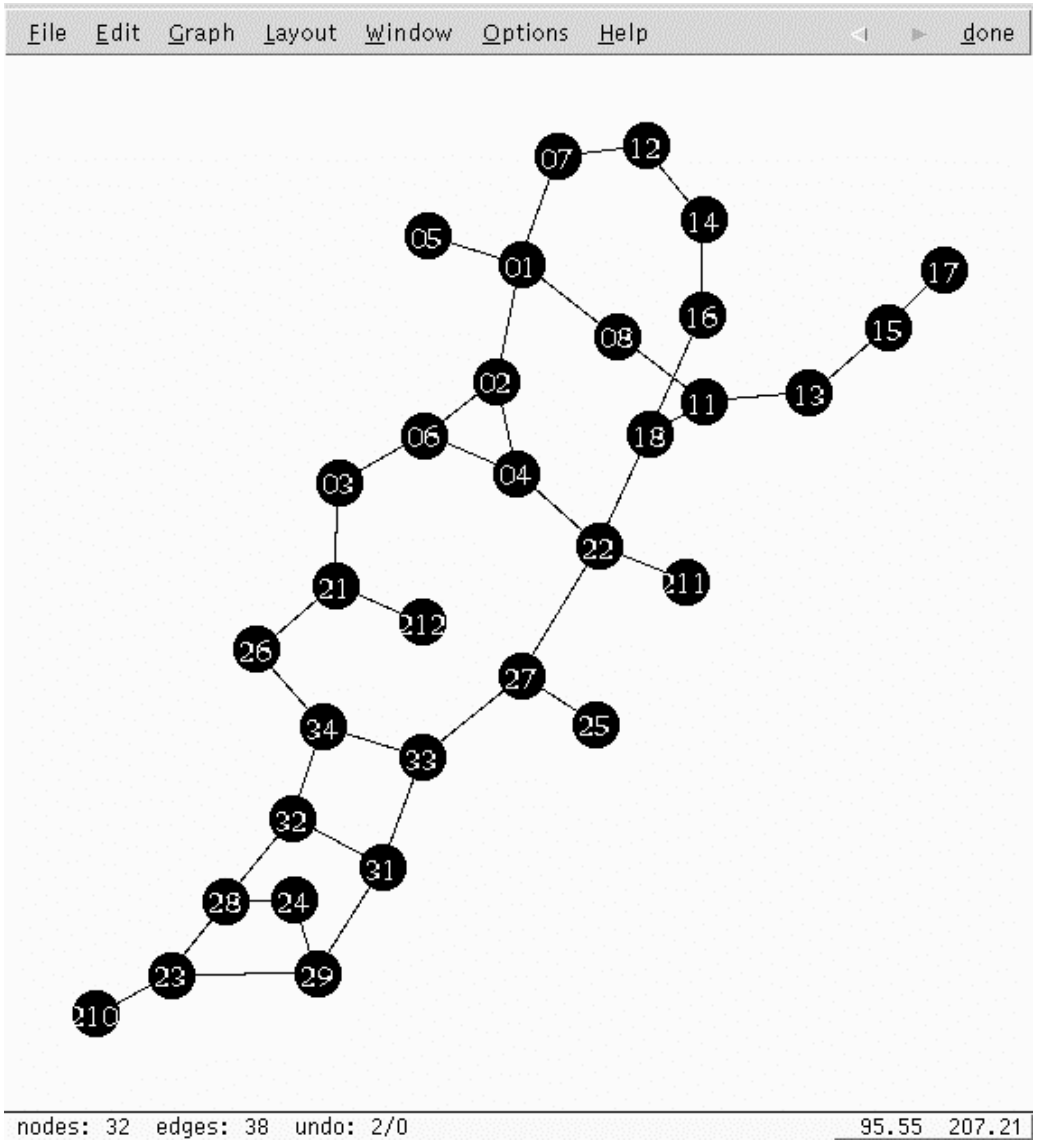
Possible User Interfaces

- Specify desired device and path properties
  - emulab selects closest approximation
- Specify desired spatial layout
  - emulab selects closest mapping
- Manually select from deployed devices

Wireless Virtual to Physical Mapping



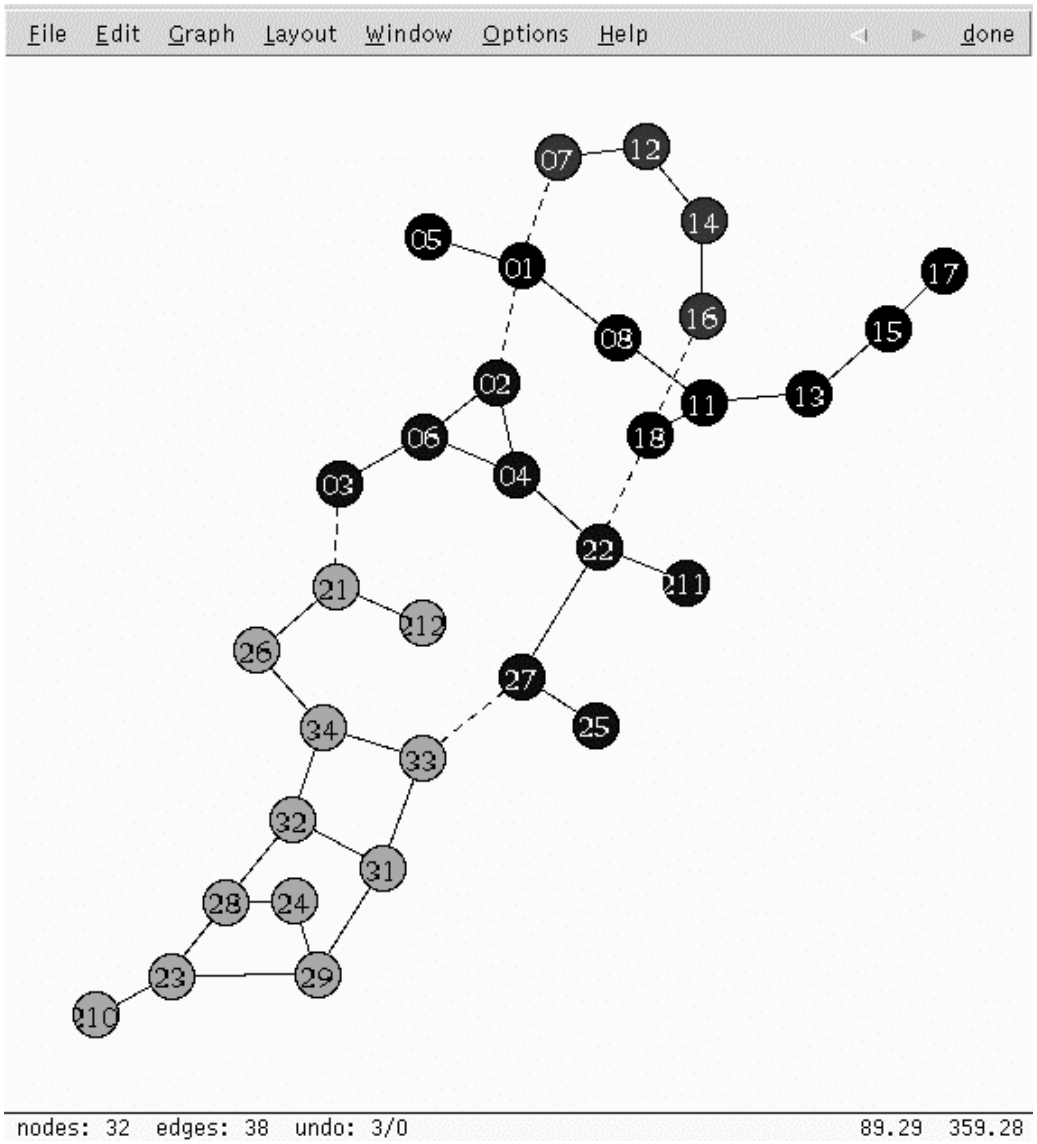
Virtual Topology



Constraints on Mapping

- Must map each virtual node and link to available physical resources
- Multiple switches usually have limited interconnect bandwidth
- In our example, there are four switches, each with 400 Mbps interconnect
- More than 4 links mapped onto a given interconnect would produce an artifact because of the bottleneck
- In solution, node color indicates its switch

Mapping into Physical Topology



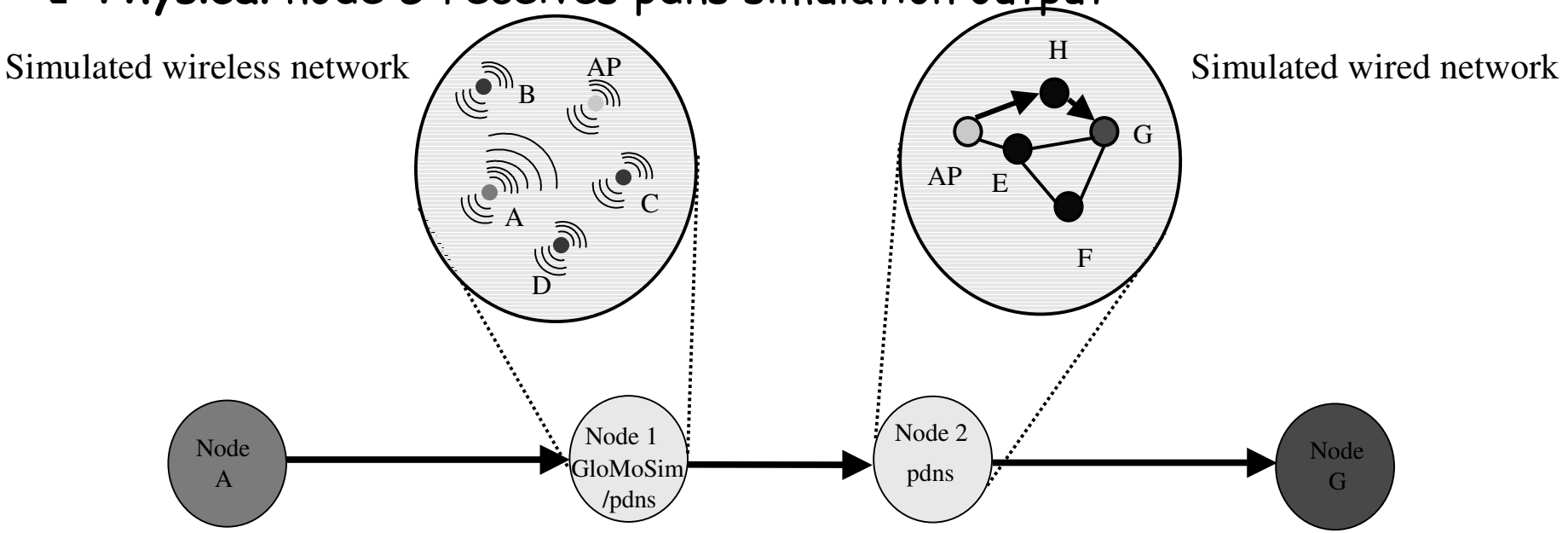
Complementary to Other Experimental Environments

- Simulation
  - Fast prototyping, easy to use, but less realistic
- Small static testbeds
  - Real hardware and software, but hard to configure and maintain, and lacks scale
- Live networks
  - Realistic, but hard to control, measure, or reproduce results

Emulab complements and also helps validate these environments

Wireless Virtual to Physical Mapping

- Exacerbated with increased diversity of devices
  - Select subset of mesh to effect mapping
  - Configure programmable physical properties (e.g. antenna gains)
- Alternate approaches
  - Genetic algorithms
  - Parallel algorithms

Integrated Simulation & Emulation

- Leverages flexible allocation algorithm to optimize integration of simulation and emulation
- Example scenario: traffic routed from node *A* to *G*
  - **Physical** node *A* injects traffic into simulation
  - **Simulated** wireless node *A* sends to multi-homed access point *AP*
  - GloMoSim simulation outputs to pdns
  - **Simulated** *AP* node sends to **simulated** *H*, then *G*
  - **Physical** node *G* receives pdns simulation output