

The Utah ProtoGENI Project

Jay Lepreau

University of Utah

www.emulab.net www.flux.utah.edu

1st GENI Engineering Conference

October 11, 2007

"Utah ProtoGENI Project" ...expanded, word by word

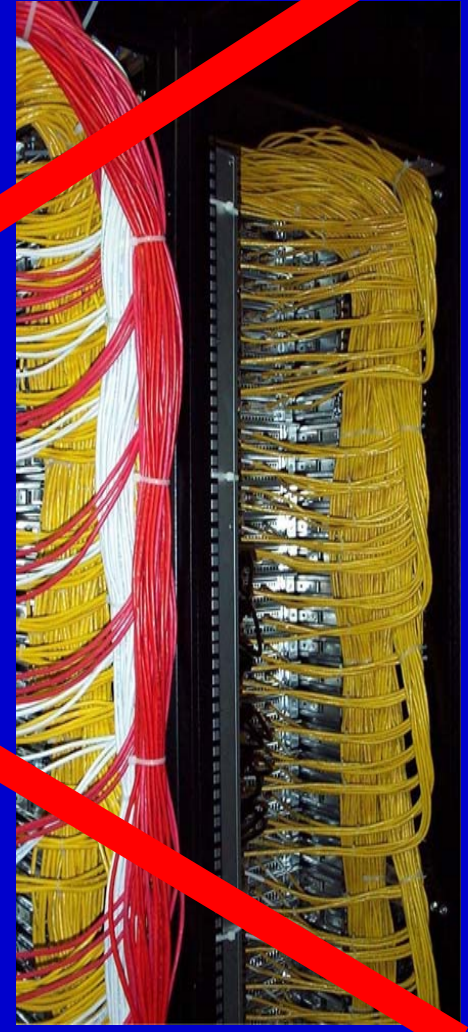
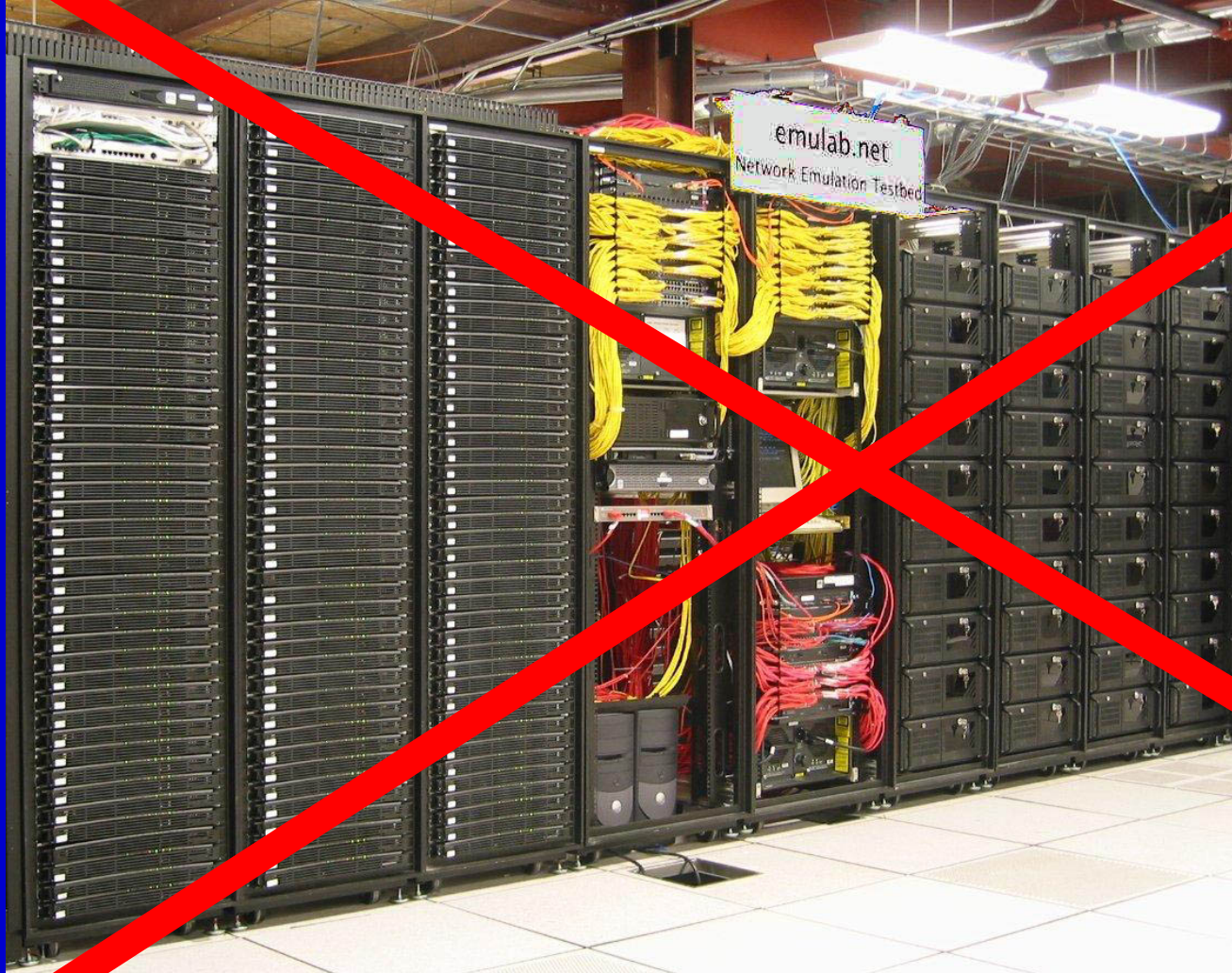
- This is a *Project*
 - Funded by NSF, Sep'07, 3 years
 - \$1.7M + \$200K from an MRI grant + ???
- To *build a GENI prototype*
 - Software
 - on Hardware, deployed around the world
 - Aim for much of it working within 18 months
 - Run it!
 - For experimenters
 - Real users?
- By *Utah*
 - Emulab software as primary base
 - Appropriate pieces from PlanetLab and others...

Sounds Ridiculous... Why Doable?

- Build on existing software, most of it *proven*
 - Emulab
 - Narrow waist, services, most substrate, most O&M
 - PlanetLab
 - Some substrate, some O&M
 - WUSTL's Programmable Router with node software VINI (probably)
 - Programmable core node
 - SWORD (maybe)
 - Wireless and wide-area resource allocation
 - Datapository
 - Measurement repository
 - OML (maybe)
 - ORBIT Wireless measurement library
 - CoBlitz/DOT/other (probably)
 - One to many data transfer service

Why is Emulab Good for This?

What is Emulab, anyway?



It's Software!

It's Software!

A rich system for managing experimentation in networks and distributed systems, across diverse hardware.

It Manages Hardware

- Lots of it
 - 1280 nodes run by the Utah Emulab site
- Lots of *kinds* of it
 - 13 node/link types
- That's successfully evolved!
 - Originally just *one* kind: PCs and VLANs
- at lots of places
 - 20 Emulabs around the world today

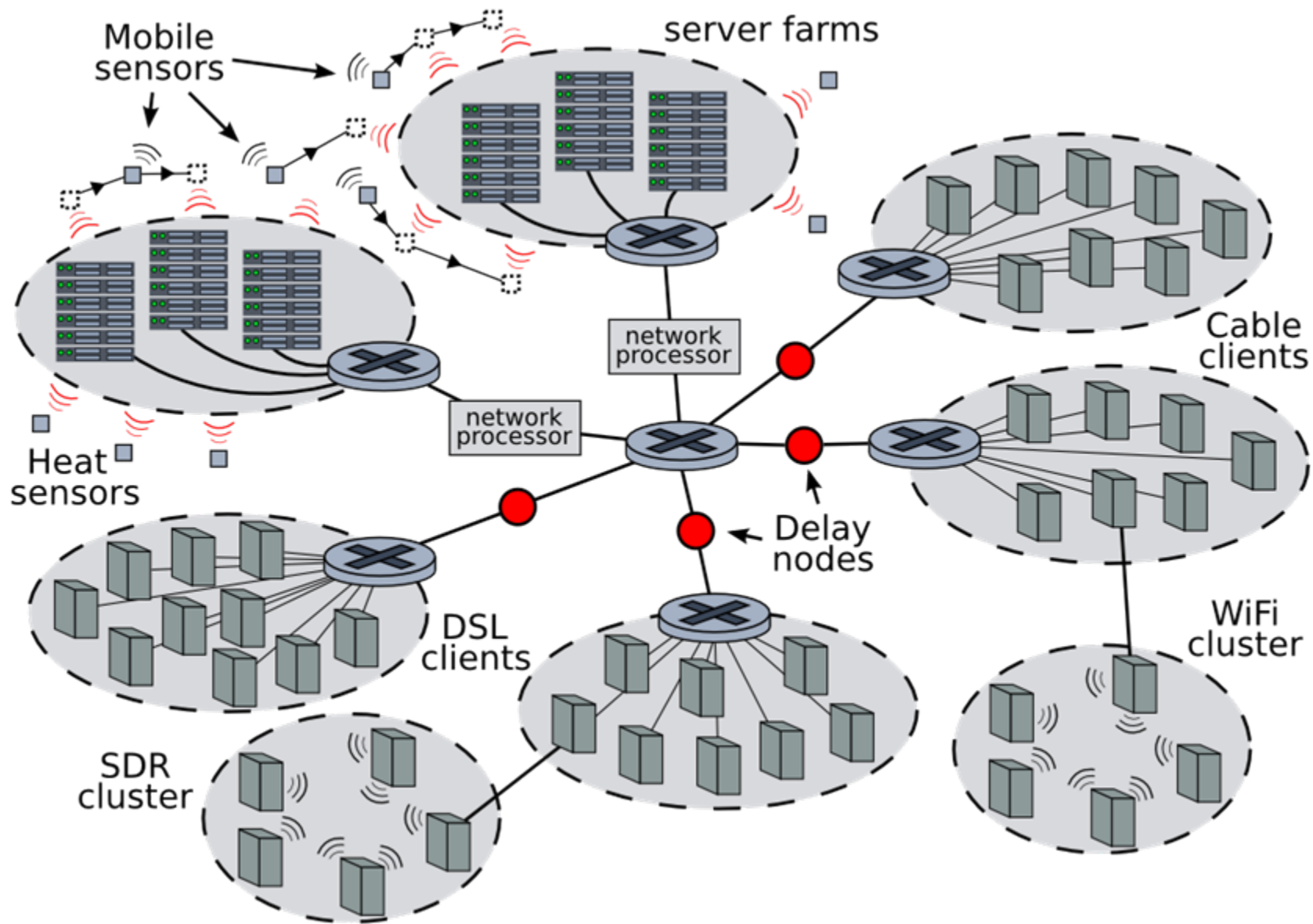
It Transparently Handles Heterogeneity

- Cluster nodes and switches
- Wireless: 802.11, software radio
- PlanetLab vservers
- Sensor motes: fixed, mobile on robots
- Special network processors (Intel IXPs)
- ...many more

- It knows about *networks*
 - Topologies, virtual networks, max flows, ...

Example: the "GHETE" Emulab Experiment

- Emulates a distributed datacenter
 - Two service clusters
 - Many client node types: wired/wifi PCs, virtual nodes, SDR nodes, sensors, mobile robots, IXP network processors
- Clients stream traffic to emulate, say, a distributed backup service
 - Traffic is load-balanced between servers
- Fixed and mobile sensors monitor clusters for overheating
 - Overheating clusters are shutdown and clients are routed to another cluster



Lotsa Hardware Types...

- PCs, VLANs
- Wide-area FreeBSD nodes (RON, Utah)
 - Internet paths
- Multiplexed VMs and links/LANs (virtual nodes & links)
- Simulated nodes and links/LANs (nse)
- PlanetLab virtual machines
- Intel IXP1200 network processors
- 802.11 a/b/g wireless
- Software Radio: "Universal Software Radio Peripheral"
- CMU FPGA-based wireless emulator (soon)
- "HomeNet" nodes
- Cisco routers: preconfigured scenarios (Wisconsin)
- Intel Stargates (ipaq-like)
- Motes: emotes, Stargate motes
- Robots: Garcia

It Serves Experimenters

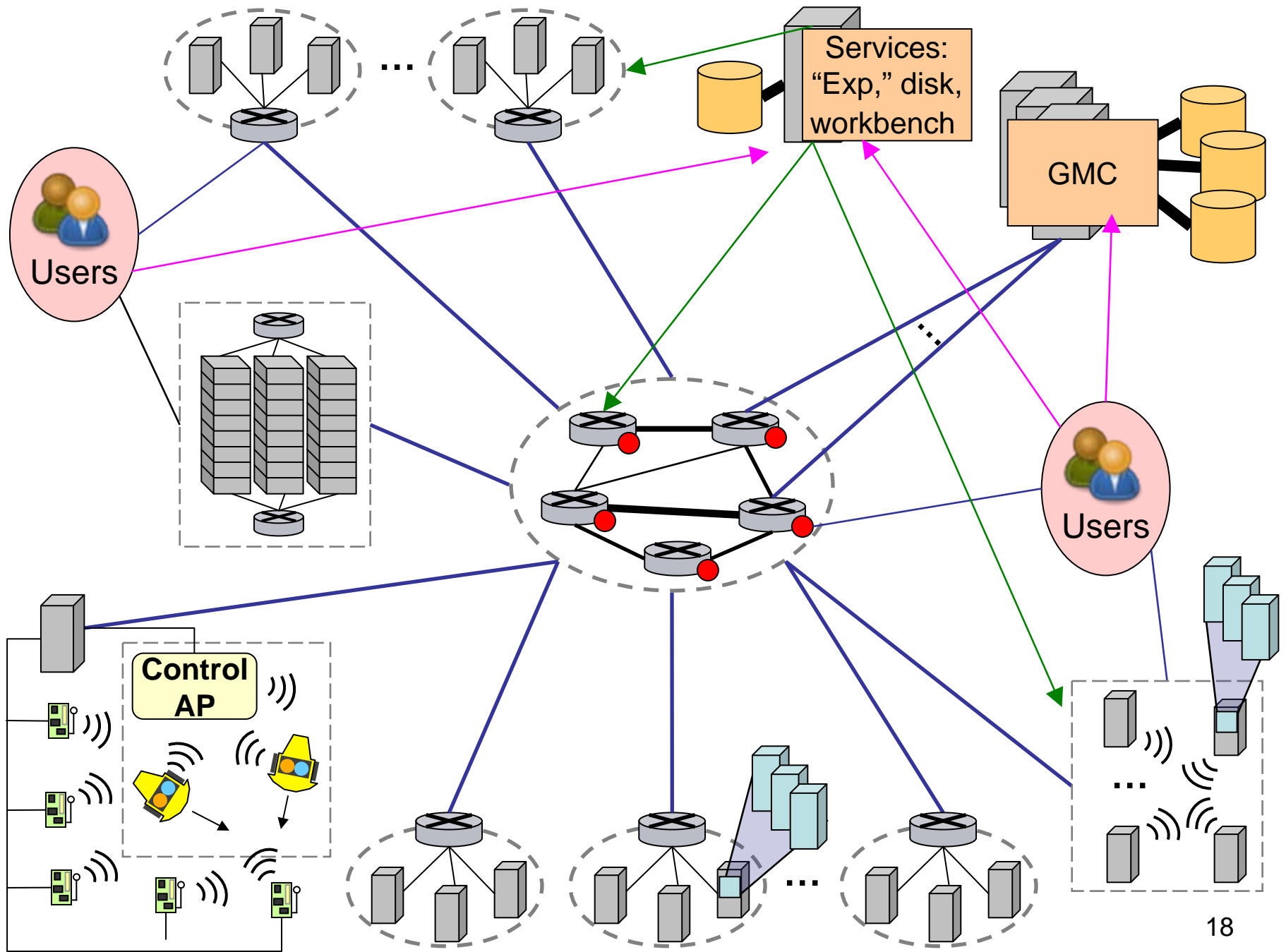
- For 7 years
- Thousands of users
- Hundreds of projects
- 18,450 experiments last year

What's inside the software that makes it good for this?

- Builtin first-class abstractions for key concepts in
 - Networking
 - Experimentation
 - Principal management
- Narrow waist, sure... but also:
- Lots of *Services*

Common Abstractions, Map to Diverse Mechanisms

- Nodes
 - Machines, VMs, Accts, Slivers, Boards, ...
- Addresses
 - IPv4, ns nodeid, ...
- Links
 - VLANs, tunnels, Internet paths, multiplexed link, virtual link
- Topology
- Queues
- Queuing disciplines
- Routing
- Applications
- Traffic generators
- Control channel/net
- Sync, startup, replay
- Events
- Viz of topology and traffic
- Monitors on links, nodes
- "Experiment" and life-cycle
- Admin entities



The Software Plan

Narrow Waist

- Component: Emulab "node" and "link" abstractions; add CM interface
- Component Aggregate: Emulab itself; add CM interface
- Management Authorities: Emulab, PlanetLab
- Slice: Emulab "experiment--"

Narrow Waist (cont'd)

- Sliver:
 - Unshared nodes: have
 - Shared nodes: have lots of the abstraction and impl (BSD jails++ and Xen).
Use PlanetLab OS software to get Linux vservers.
Need to extend Emulab physical-node based security model.
- Slice Authority: An Emulab instance
- GIDs, Certificates: Add; small steps done
 - GENI security architecture is in flux. TBD
- Federation: slowly. small steps so far. TBD

Narrow Waist APIs

- Slice operations

 - Component operations

 - Similar but not exactly the same interfaces are in Emulab: adapt/add

- Aggregate operations

 - Mostly private; mostly exist

- CM/component interface

 - Private: Emulab tmcc/tmcd protocol

Services

- Slice embedding service:
 - Emulab 'assign', converted to Rspec, + support for new features
 - Observe component-specific policies
 - Possibly SWORD for wireless & PL-like embeddings
 - ProtoGENI will support both virtualized and space-shared resources.
 - Latter can run custom OS software on bare metal
- Storage
 - Data push: have now; add optimized one->many service (CoBlitz, or DOT, or ...)
 - Logging and data pull: Emulab 'loghole' with improvements
 - Per-expt DB: have now
 - Per-node DB: add
 - Wide-area filesystem: possibly revive Emulab's bit-rotted SFS support, but probably omit

Services (cont'd)

- Control plane
 - For both internal operations & experimenters, across all components, Emulab uses a publish-subscribe system. Extend using RON-like overlay for additional robustness in wide-area.
- Resource allocation and policies
 - Science board interface: Emulab has coarse grained way to grant types of hardware to projects; will add a better interface.
 - Resource broker: not addressed
 - (Slice embedding service handles normal resource discovery and allocation for experimenters)
- Measurement
 - Emulab has a little builtin support now; only a little more is planned right now, mostly for wireless.

Services (cont'd)

- Experiment management/support
 - Emulab has rich support, incl. specifying topologies, installing and updating software, controlling execution, deploying in heterogeneous envs. Refine as needed.
 - Monitoring testbed resource state, publishing it
 - Emulab has much, and uses PlanetLab's where it can
 - Add more, using PL's sensors when possible
 - (Ongoing separate R&D work on "Experimenters' Workbench", now in alpha (NSDI '07). Should be production quality within 12 months.)

Substrates

- Virtual server: Emulab 'node' type with PL-like sharing. Use PL Node Manager and OS.
- Virtual router: Emulab 'node' + support for layer 3 devices
- Virtual switch: Emulab 'node' + support for layer 2 devices
- Virtual access point: Specialization of Virtual Server

Substrates (cont'd)

- Programmable edge nodes
 - ~200 PCs at 100 PlanetLab and RON sites
 - ~30 PCs in Pittsburgh apartments, via CMU Emulab MA
- Programmable core nodes
 - WUSTL programmable router (2), supporting PL node mgr interface, provided by WUSTL
 - ~60 PCs with netFPGA boards in I2/NLR colo centers
 - VINI code in PCs, mostly unchanged
- Programmable Edge Clusters
 - Emulab instances (Utah, probably others)
 - Generic clusters (probably)
- National backbone facility
 - I2/NLR likely, layer 2. Layer 1?

Substrates (cont'd)

- Programmable Wireless Nodes and Cognitive Radio Subnet
 - 120 node Utah wireless testbed, in offices and labs
 - Dual 802.11a/b/g
 - GNU radio (USRP and USRP2)
 - 8-16 node CMU wireless channel emulator testbed
- Urban 802.11 mesh network
 - ~30 Pittsburgh PCs in apartments
- Sensor net
 - 30 Mica2's in Utah office building
- Emulation Grid
 - Emulab today

Substrates (cont'd)

- <Your component or component aggregate here!>

We're looking for...

- People who want to develop missing or poor parts like
 - Resource brokers
 - Better virtualization technology (versver reservations, KVM, Xen, ...)
 - Measurement support/services
 - User opt-in technology support
 - ... <talk to Jay or Rob>
- Partners with substrate instances
 - Regional optical networks
 - Sensor nets
 - Clusters
 - Maybe more wireless
 - Optical, with optical expertise!

Summary

- We're building a real end-end prototype of GENI
- Both software and hardware, many kinds
- And going to run it
- Soon
- The key is building on robust mature software

Acknowledgements: Rob Ricci, many other Utah staff and students, demanding Emulab users, and all who worked so hard in the GENI design working groups, especially the facility architecture group, and especially Larry Peterson and John Wroclawski.