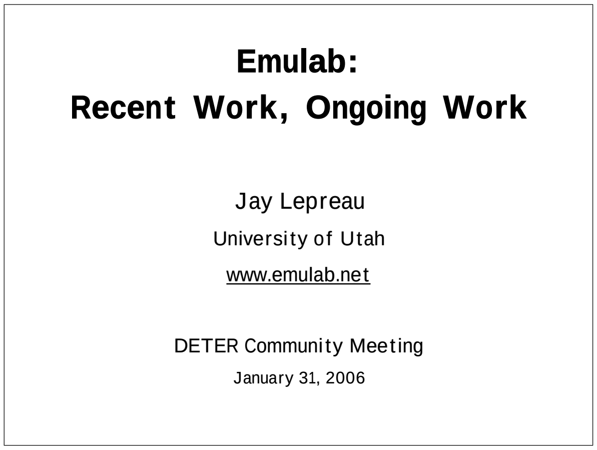


Emulab: Recent Work, Ongoing Work

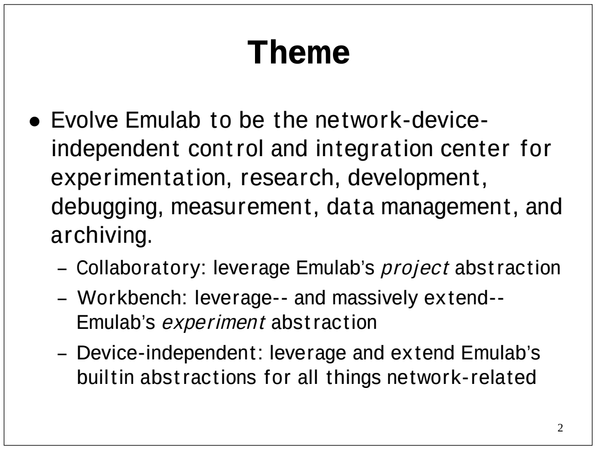
Jay Lepreau

University of Utah

www.emulab.net

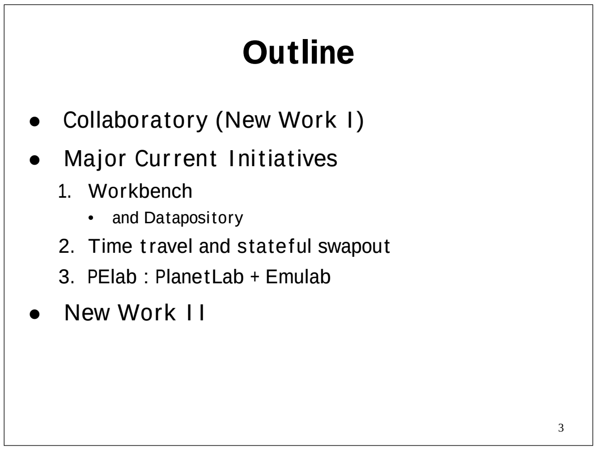
DETER Community Meeting

January 31, 2006



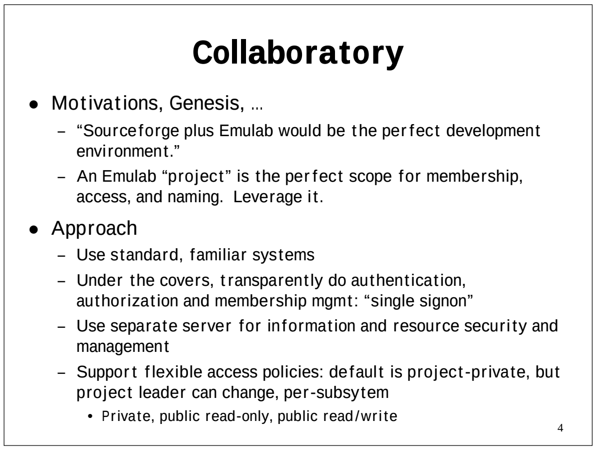
Theme

- Evolve Emulab to be the network-device-independent control and integration center for experimentation, research, development, debugging, measurement, data management, and archiving.
  - Collaboratory: leverage Emulab's *project* abstraction
  - Workbench: leverage-- and massively extend-- Emulab's *experiment* abstraction
  - Device-independent: leverage and extend Emulab's builtin abstractions for all things network-related



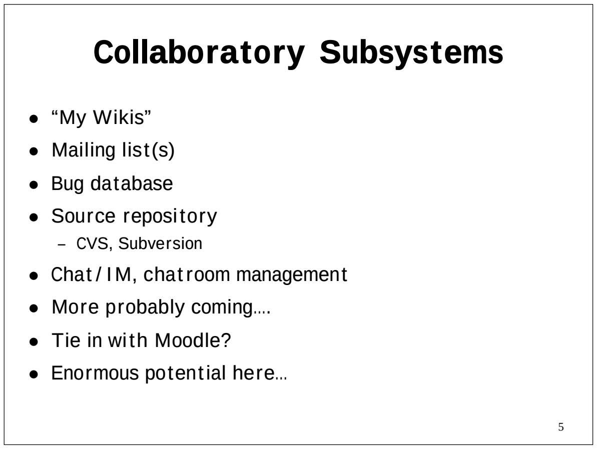
Outline

- Collaboratory (New Work I)
- Major Current Initiatives
  1. Workbench
     - and Datapository
  2. Time travel and stateful swapout
  3. PElab : PlanetLab + Emulab
- New Work II



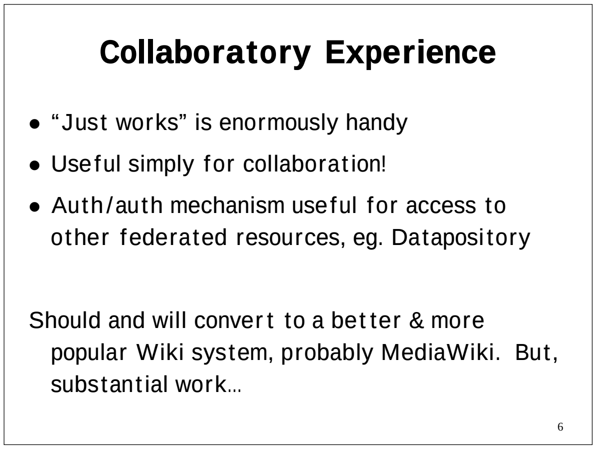
Collaboratory

- Motivations, Genesis, …
  - "Sourceforge plus Emulab would be the perfect development environment."
  - An Emulab "project" is the perfect scope for membership, access, and naming. Leverage it.
- Approach
  - Use standard, familiar systems
  - Under the covers, transparently do authentication, authorization and membership mgmt: "single signon"
  - Use separate server for information and resource security and management
  - Support flexible access policies: default is project-private, but project leader can change, per-subsytem
    - Private, public read-only, public read/write



Collaboratory Subsystems

- "My Wikis"
- Mailing list(s)
- Bug database
- Source repository
  - CVS, Subversion
- Chat/IM, chatroom management
- More probably coming….
- Tie in with Moodle?
- Enormous potential here…



Collaboratory Experience

- "Just works" is enormously handy
- Useful simply for collaboration!
- Auth/auth mechanism useful for access to other federated resources, eg. Datapository

Should and will convert to a better & more popular Wiki system, probably MediaWiki. But, substantial work…

1. Experimentation Workbench

Convergence of opportunity and demand

- Four types:
  - Workflow management (processes), including
    - Measurement and feedback steps
    - *mandatory pipelines.* Eg,
      - Enforce trace data anonymization based on user privileges
      - Just-in-time decryption of malware
  - Experiment management
  - Data management
  - Analyses

"Scientific Workflow" … with many differences

A Different Domain,
A Different Approach

- *Our* domain, our expertise.
  - "A systems viewpoint"
- Existing "experiment" model: pervasive
- Implicit vs. explicit specification
- History-based views
- Incremental adoption
- Pragmatic approach

Micro demo

Short paper in submission:

www.cs.utah.edu/papers/workflow-ftn2006-01-base.html

Related: "Datapository" for network-oriented measurement data

- Collaborative CMU (Dave Andersen) and Georgia Tech (Nick Feamster) effort to create an (Inter)net measurement "data repository"
- Currently running at datapository.net
- Federated with Emulab
- Temporarily using 16 TB file server at Utah
- Proposal under review

Short paper in submission:
www.pdl.cmu.edu/PDL-FTP/stray/CMU-PDL-06-102_abs.html

2. "Time Travel" and Stateful Swapout

- Time-travel of distributed systems for debugging
  - Generalize disk image format and handling (done)
  - Periodic disk checkpointing (prototyped, MS thesis)
  - Full state-save on swapout (prototyped)
  - Xen-based virtual machines (in progress)
  - Challenge: network state (packets in flight)
    - Ignore
    - Consistent checkpointing
    - Pragmatic middle ground: quiesce senders, flush buffers
- Stateful swapout/swapin [easier]
  - Allows transparent pre-emption experiment
- Related to workbench: history, tree traversal
  - Can share some mechanisms, some UI

3. "Pelab"

- Motivation:
  - PlanetLab (sort of) sees the "real Internet"
    - But its hosts are hugely overloaded, unpredictable
    - Internet and host variabiity ==> Takes many many runs to get statistical significance, and …
    - ==> Hard to debug
  - Emulab provides predictable, dedicated host resources and a controlled, repeatable environment in every way
    - But its network model is completely fake

Approach

- Goal: get the best of both worlds
  - Actually, better than the best of each world today
- Extreme formulation:
  Application runs on Emulab with its NICs on PlanetLab hosts

Possible Approaches

- Internet- and Model-oriented
  1. Measure the Internet over a long time
  2. Develop a model
  3. Make a super-Dummynet

  Drawbacks:
  - 1 and 2 are very hard.
  - "Rare events" are difficult to model and measure
- Delta to above:
  - Send real time Internet conditions into Emulab
- "Modeling and Emulating the Internet"

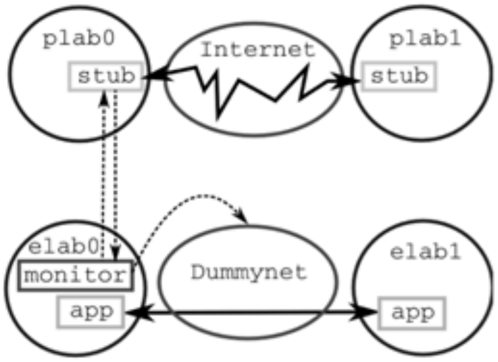
Possible Approach #2: View the Internet "through the PlanetLab lens"

- PlanetLab- and Model-oriented
  - Measure PlanetLab paths over a long time
    - Much more tractable than the whole Internet
  - Develop a model
  - Develop a super-Dummynet
- Additions to above:
  - Mirror real-time PlanetLab conditions onto Emulab
  - Use "stub" on Plab, peered with each Emulab node, sending that node's traffic into Plab. Needed if app's traffic evokes a reactive response from the Internet
- "Projecting PlanetLab into Emulab": Net -> Net'
- Drawbacks: Still hard in many ways, other…

Approach #3: Use the application traffic itself as the measurement traffic

- PlanetLab- and application- and realtime- oriented
  - Chosen Plab nodes peered with Emulab nodes
  - App starts up on Emulab
  - App-traffic gen and measurement stubs start up on Plab (TCP tracing)
  - Send real time network conditions to Emulab
  - Develop a super-Dummynet (done; useful separately)
  - Develop and continuously run adaptive Plab path-condition monitor
    - Pour results into Datapository
      Use for initial conditions or when app goes idle on certain pairs
- App -> App'

Pelab design



New Work (II)

1. Exploring a New “Assign”

Exploring the “Comet” domain-specific language for combinatorial optimization using constraint-based local search

- From Brown Univ (van Hentenryck, Michel); there’s a book.
- Goals:
  - Easier to understand and extend, especially by non-experts
  - More flexible
  - Should enable easy comparison of completely different optimization techniques (simulated annealing, other). Probably primarily of research interest.
- Have basic prototype implemented
- My instinct says it will be time-consuming and hard to match assign’s current level of performance and robustness

2. Security-related Improvements

- Secure “Experiment tear down” improved
  - Cleaned up, fixed some vulnerabilities
  - Added the MFS bootblock zapper program
  - enabled it for all firewalls
- Identified some holes in the control-net firewall rules and will be re-doing
- Switched to ssh2 keys
- Zeroing disks: support added to DB, not hooked in to UI
- Writing up a tech report on Emulab’s security-related design and implementation

3. Automatic Online Validation

Emulab is:

- an ongoing research and dev project, it’s big, and it’s complex
  - Bugs are likely
  - Bugs arise from subtle interactions: we’ve found that separate regression tests are insufficient
- … a public scientific facility: Stakes are high
- Approach:
  - Validate network config of *every* experiment
  - Make it so quick that this approach is acceptable

Online Validation (cont.)

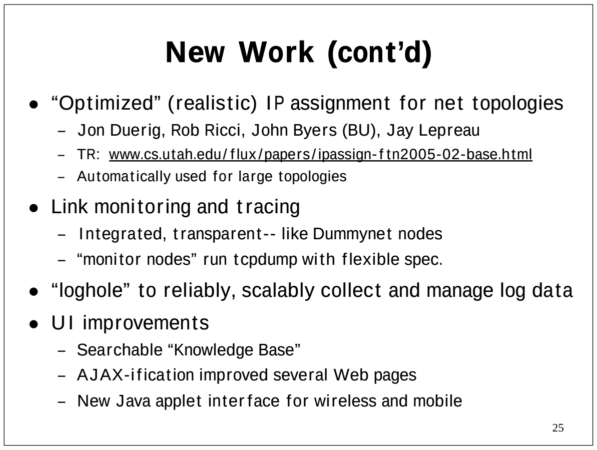
- Uses an entirely separate code path from Emulab configure path
  - No DB, no XML, no perl scripts, no nothing…
- A new state in experiment life cycle:
  - Invoked transparently as part of expt swapin, after all nodes up, but before “time 0”.

Automatic Online Validation (cont.)

- A validation program, *linktest*, runs after each swapin, modify, or upon user request
  - Validates the network configuration using end-to-end tests
- Linktest validates the following:
  - Duplex, simplex, and LAN links
  - Symmetric and asymmetric traffic shaping
    - Latency, loss, bandwidth
  - Static routing
  - Running invisibly in beta test, ~2 months

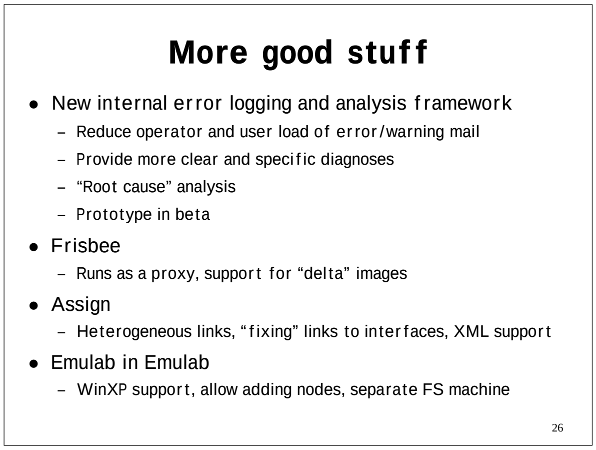
4. Major Cluster Expansion

- 160 high-end nodes, 3.0 GHz, 2GB, 6 Gbit NICs, 2 x 146G disks
- 2 new switches; 1 is very high bandwidth
- 360 total; back of envelope potential: 10,000 – 20,000 vnodes
- But: had significant bringup and scaling challenges
- Enormous boss/ops stability problems when they were moved to the new hardware. OS tweaks/fixes required.



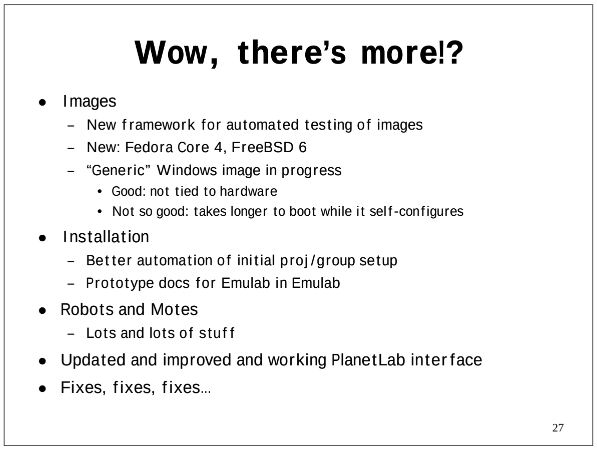
New Work (cont'd)

- "Optimized" (realistic) IP assignment for net topologies
  - Jon Duerig, Rob Ricci, John Byers (BU), Jay Lepreau
  - TR: www.cs.utah.edu/flux/papers/ipassign-ftn2005-02-base.html
  - Automatically used for large topologies
- Link monitoring and tracing
  - Integrated, transparent-- like Dummynet nodes
  - "monitor nodes" run tcpdump with flexible spec.
- "loghole" to reliably, scalably collect and manage log data
- UI improvements
  - Searchable "Knowledge Base"
  - AJAX-ification improved several Web pages
  - New Java applet interface for wireless and mobile



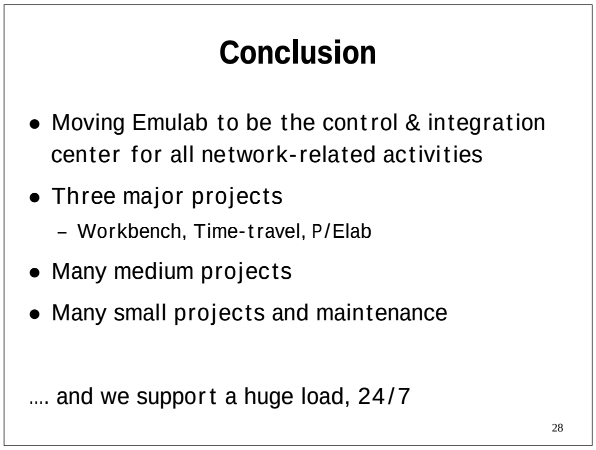
More good stuff

- New internal error logging and analysis framework
  - Reduce operator and user load of error/warning mail
  - Provide more clear and specific diagnoses
  - "Root cause" analysis
  - Prototype in beta
- Frisbee
  - Runs as a proxy, support for "delta" images
- Assign
  - Heterogeneous links, "fixing" links to interfaces, XML support
- Emulab in Emulab
  - WinXP support, allow adding nodes, separate FS machine



Wow, there's more!?

- Images
  - New framework for automated testing of images
  - New: Fedora Core 4, FreeBSD 6
  - "Generic" Windows image in progress
    - Good: not tied to hardware
    - Not so good: takes longer to boot while it self-configures
- Installation
  - Better automation of initial proj/group setup
  - Prototype docs for Emulab in Emulab
- Robots and Motes
  - Lots and lots of stuff
- Updated and improved and working PlanetLab interface
- Fixes, fixes, fixes…



Conclusion

- Moving Emulab to be the control & integration center for all network-related activities
- Three major projects
  - Workbench, Time-travel, P/Elab
- Many medium projects
- Many small projects and maintenance

…. and we support a huge load, 24/7