



09:00-09:30 ABone Update -- Bob Braden

09:30-09:50 Active Network Management -- Livio Ricciulli

09:50-10:00 QCMD -- Carl Gunter

10:00-10:30 Break

10:30-10:45 Intro to CAIRN/Supernet/... -- Terry Gibbons

10:45-11:00 ABone in CAIRN -- Steve Berson

11:00-11:20 ABone Open Issues -- Peter Steenkiste & Bob Braden

11:20-12:00 Future Directions -- Panel & Group Discussion

**Bob Braden, Livio Ricciulli, Sandy Murphy, Peter Steenkiste,
and Carl Gunter**



ABone Update November 1999

**Bob Braden, USC/ISI
Livio Ricciulli, SRI
Steve Berson, USC/ISI**



OUTLINE

- o **The ABone plan**
- o **EE Management: Anetd**
- o **ABone Trust Model & Node Security**
- o **ABone Status**
- o **What a Core Node Administrator should do**
- o **What an EE Developer should do**



ABone -- Active Nets Testbed

- o **Large, quasi-stable virtual network infrastructure** to support testing and deployment of a growing set of AN software components:
 - Execution Environments (**EEs**)
 - Active Applications (**AAs**)
 - **Node OSs**
- o **Large: Must be scalable to O(100-1000) nodes**
- o **Security planned in.**

ABone ...

- o Assembled from existing nodes & links**
 - CAIRN nodes & links are part of the ABone**
 - Other nodes are provided by research sites**
 - Other links are Internet overlays.**
 - o Good news: Internet connectivity is available.**
 - o Bad news: Performance will be highly variable.**

- o Diverse platforms (NodeOS's)**

**Initially, ABone nodes will run Unix (FreeBSD, Linux, or Solaris).
NodeOSs designed for AN will be added soon.**



Shareable & Available

- o Any network testbed: conflict between using it and breaking it.
- o Active Nets is fundamentally about programmability with isolation for multiple simultaneous clients.

Therefore, normally operate ABone as always-available shared experimental resource.

But will also support exclusive-use periods for parts of infrastructure (DARTnet/CAIRN model).

Favor fixed real/virtual topologies for ABone

(Should ABone also support self-organizing virtual topologies?)



A Network of People

- o Encourage and enable collaboration**
- o Create active network building blocks**

Interchangeable parts => standard interfaces

-- AA/EE interface: EE-specific, but maybe recommended functionality

-- EE/NodeOS interface

- o Want EEs portable across NodeOS's.**
- o Two standard interfaces: Posix and EE/NodeOS**
- o Conflict with need to expose platform-specific features to AAs (?)**



Roles

We distinguish four ABone-related roles:

- o **Node Administrators**

- Configure node, maintain system, reboot upon request.

- o **NodeOS developers**

- o **EE developers**

- o **AA developers**

- Membership large, will change rapidly; *untrusted*

(Many researchers play multiple roles.)

There is also an **ABOCC** (ABone Coordination Center)

Core Nodes

Two-level scheme: “Core” nodes and “Edge” nodes.

Core [public] nodes:

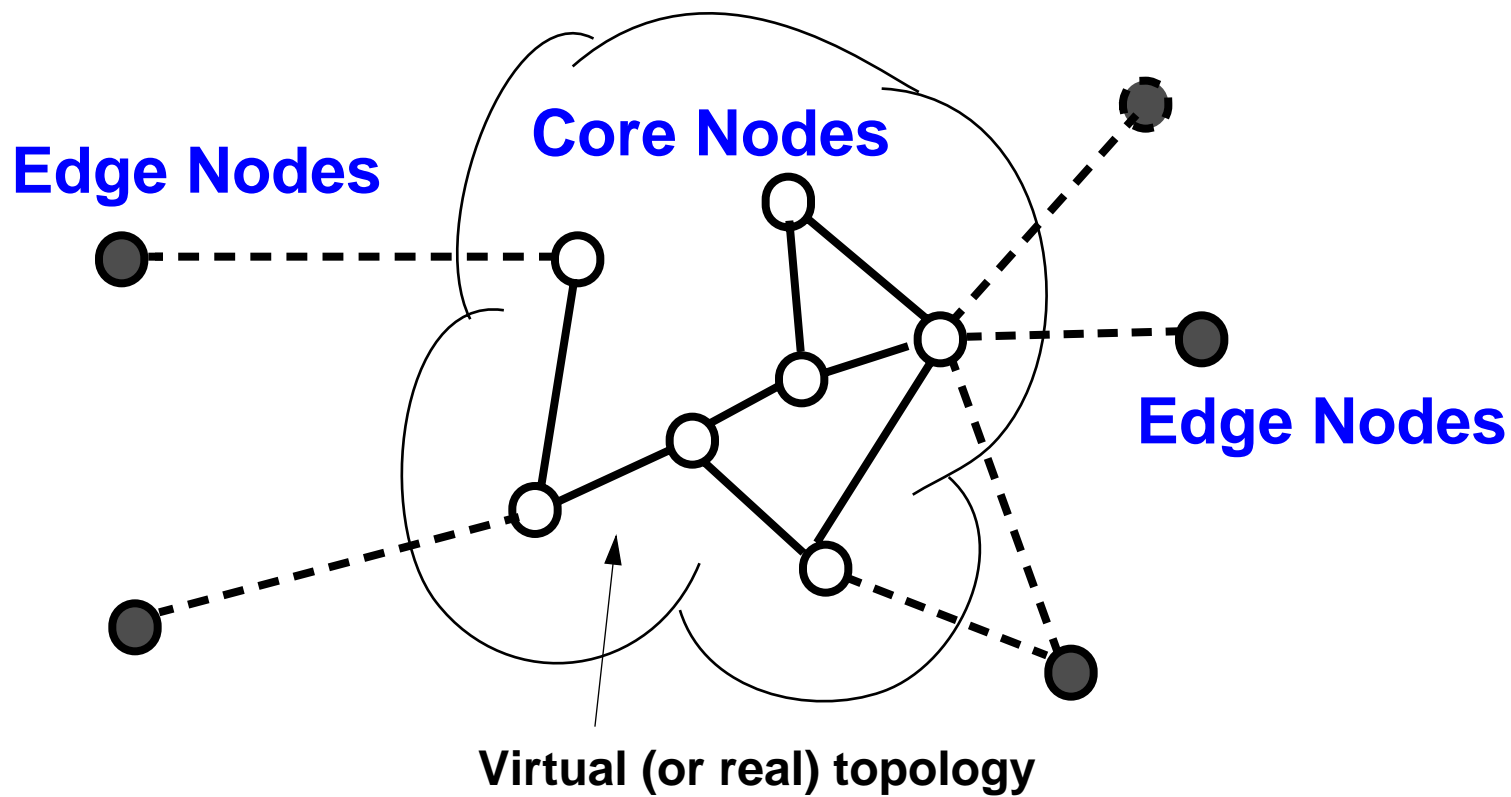
- o Public resource for AA and EE developers.
- o “Always” available -- ABOCC will monitor.
- o Support a set of “permanent” EEs.
 - An always-available resource for AA developers.
 - Fixed topolog(ies)
- o EE developers can also run private EEs on core nodes.



Edge Nodes

Edge [private] nodes

- o Private, “owned” by developer.
- o No availability requirement.
- o Dynamic addition to, removal from topology.
- o AA or EE developer starts local EE copy for a test.
- o AA developers may need private sharing agreements with other sites to access additional edge nodes.



EE Management: Anetd

o Nodes will be locally administered but remotely managed => **intense node security concerns!**

EE developers will use SRI's **Anetd** (Active Net Daemon) to remotely install and maintain EEs on core nodes.

ANetd has two functions:

1. **Remote management** of EEs.
2. ***Basic NodeOS for Unix** -- demuxing incoming active packets to EEs using ANEP header

Forks an EE as a sub-process, passes packets to it via stdin

*Note: (2) may be subsumed by NodeOS later.



Node Security, Trust, and EEs

- ** EEs must be trusted to be civil (perhaps buggy, but not malevolent).**
- ** An EE runs in user mode under Unix, but still, ... there are always system vulnerabilities.**
- ** (Wise) Node Administrators will not tolerate unknown remote users loading and executing arbitrary code on their systems!**
- ** Therefore, the ABone must have adequate node security.**



ABone Node Security Model

Anetd controls **WHO** can load & execute an EE on node

- o Public Key cryptography
- o **ACL** for each user ID on each node

and **WHENCE** such code can be loaded.

- o **TCL**: Trusted Code-server List

Remote management will use Anetd; only node administrators will have passwords for ABone accounts or root on their own systems.

ABOCC maintains master ACLs and TCLs, uses **QCMD** to securely push them into all (core) ABone nodes.



Trust Model for ABone nodes

EE developers must generally be trusted.

A cautious Node Administrator must trust ABOCC to supply 'safe' ACL and TCL.

E.G. initial TCLs might allow only ABOCC and DARPA-funded sites as code servers for loading EEs.

NOTE: Ideally: **NO restriction on AAs. An EE is expected to provide a safe execution environment for arbitrary, and perhaps evil, AAs.**



User Accounts on Core Nodes

~abocc

User ID owns files that contain:

Anetd code, QCMD code

ACLs and TCLs

Restart scripts, ...

The ABOCC uses an Anetd under this account to configure, control, and monitor the node.

~anee1, ... ~anee6

[~anee6 actually called ~anpub]

These are execution accounts for EEs under corresponding Anetd daemons



Why 6 EE accounts?

- o Best EE isolation if one account per EE, but this would be administrative hassle

- o Compromise:

 - A small **fixed set of accounts**

 - Allow multiple EEs to share same account

Think of these accounts as security compartments -- e.g.,
anee1 (most trusted) -> anpub (least trusted)

An EE CAN harm another EE in the same account, but only if it tries...

- o Each EE has (writable) **file subspace**:

~anee_k/**<princlD>**/

<princlD> is principal ID of person who installs EE.



AA Security

What can an AA do? That's up to the EE, of course.

For Java-based EEs:

- 1. Anetd installs its own Security Manager, to protect itself and the node from EE and AAs.**
- 2. An EE will be able to register its own Security Sub-Manager, which will be upcalled from the Anetd Security Manager.**

EE Sub-Manager can only REDUCE privileges.

Non-Java: ? (To be worked out)



Network Security

How about network security?

- o Could leave it up to EEs**
- o Seems better to put in place some basic security mechanism. (e.g., ISI proposal, discussed later).**



The ABOCC

o **ABOCC: A**Bone Coordination Center

Tries to organize things a little.

- Register nodes, EE developers, ...
- Maintain node access control lists using QCMD from UPenn
- Assign Numbers (e.g., TypeIDs, UDP ports)
- Monitor status of nodes and permanent EEs.
- Diagnose troubles, notify EE developers and/or node admins.
- Organize regular teleconferences.
- ...



Current ABone Status

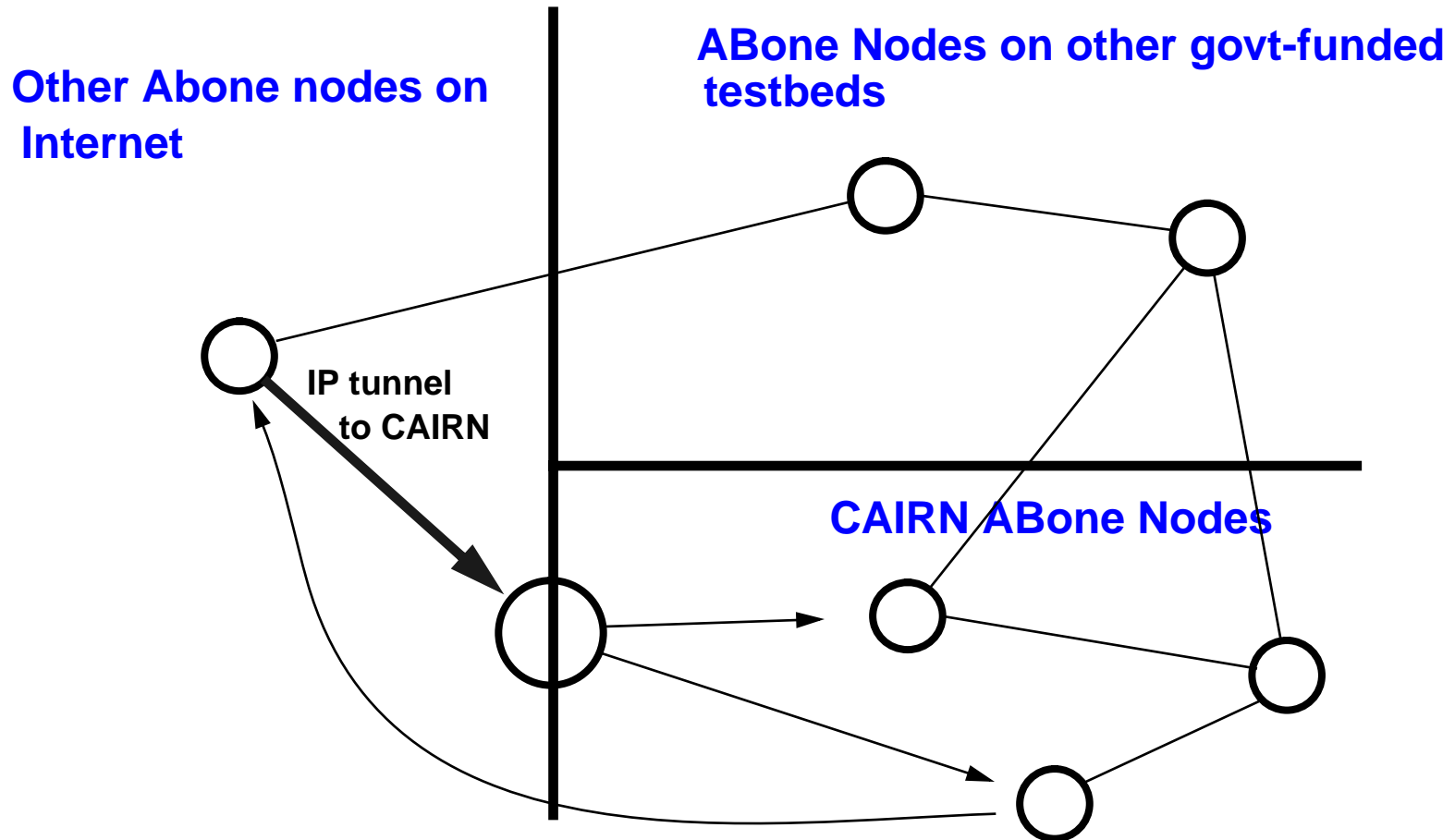
- o **Unix node configuration debugged in CAIRN**
 - 16 nodes running Anetd under multiple accounts
- o **QCMD installed and running**
- o **One permanent EE installed on all nodes (ASP EE)**
- o **Daily node monitoring, with email to Administrators of down nodes**
- o **Rebooting node causes Anetd and permanent EEs to auto-start.**



To be Done Real Soon...

- o Ask node administrators to install new configuration**
- o Install ANTS as permanent EE**
- o Improve monitoring to include permanent EEs**
- o Start looking at performance**

ABone Nodes: Routing





A Core Node Administrator Needs to...

- o **Register node with ABOCC using Web tools**
(currently: www.csl.sri.com/ancors/abone/)

Specify contact person(s)

- o **Use new ABOCC rules to [re-]configure node for isolation and security, and [re-]install Anetd and QCMD.**
<http://www.isi.edu/abone/configure.html>)



An EE Developer needs to...

- o Modify the EE to operate under anetd**
 - Packet input via stdin**
 - (Java) Supply EE security manager**
 - Provide some method for setting virtual topology**
- o Register as an ABone user.**
- o Get typeid assignment from ANANA [@ABOCC].**
- o Register code repository with ABOCC**
- o Tell ABOCC how to monitor EE.**
- o Set up Web page telling AA developers how to use EE.**



References

- o **Mailing lists:**

 - abone@csl.sri.com (THIS WILL MOVE TO UKANS)**

 - abocc@isi.edu**

- o **URLs**

 - <http://www.isi.edu/abone>**

 - <http://www.csl.sri.com/ancors/abone/>**



Open ABone Issues

Bob Braden, ISI



Large Open Issues

- o **Is the AA/EE/NodeOS model really applicable to the research efforts in the DARPA active nets community?**
- o **Can we start building on each other's work?**
 - Are there real common interfaces, and how will pieces fit together?**
- o **How & when should the ABone support kernel-invasive AN code?**
- o **Virtual or real protocol stacks, or both?**



Virtual or Real Protocol Stack?

- o **EE can model ANEP header as virtual link layer within virtual topology.**
 - Permanent EEs can be configured with “interesting” topologies.
 - AA can then construct entire new protocol stack above LL.
- o **Or, EE can model ANEP header as layer 4.5 (or 3.5) in real Internet protocol stack**



Small Open Issues

- o Establishing **virtual topologies**
 - Per-EE vs. node-wide topology
 - Standard forms and scripts
 - Support dynamic joining of edge nodes?
- o Common **traffic control** mechanism
- o Access to kernel routing table
- o Tools/facilities for AA **debugging**
- o Standard **network management** interfaces and tools
- o **Packet Capture**



Packet Capture

- o Anetd receives active pkts with UDP encapsulation
- o Could also receive them:

- with IP encaps (proto ID 107)

- Maybe ultimately the Right Answer, but requires Anetd to run as root...prefer not.

- from packet filter intercept

- Will look at packet firewall/redirection capabilities in Linux, FreeBSD. Should match NodeOS interface Filter (“channel”) may imply Typeid, no ANEP header needed.