AFRL-IF-RS-TR-2003-182
Final Technical Report
August 2003

X-BONE

USC – Information Sciences Institute

Sponsored by Defense Advanced Research Projects Agency
DARPA Order No. G197

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the U.S. Government.

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2003-182 has been reviewed and is approved for publication.

APPROVED: /s/

PRISCILLA A. CASSIDY
Project Engineer

FOR THE DIRECTOR:

/s/

WARREN H. DEBANY JR., Technical Advisor
Information Grid Division
Information Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
Aug 2003

3. REPORT TYPE AND DATES COVERED
Final May 98 – Sep 01

4. TITLE AND SUBTITLE

X-BONE

5. FUNDING NUMBERS
C - F30602-98-1-0200
PE - 62301E
PR - G197
TA - 00
WU - 01

6. AUTHOR(S)

Joseph D. Touch

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

USC – Information Sciences Institute
4676 Admiralty Way
Marina del Rey, CA 90292

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Defense Advanced Research Projects Agency
3701 North Fairfax Drive
Arlington, VA 22203-1714

AFRL/IFGA
525 Brooks Road
Rome, NY 13441-4505

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

AFRL-IF-RS-TR-2003-182

11. SUPPLEMENTARY NOTES
DARPA Program Manager: Douglas Maughan, ATO, (703) 696-2373
AFRL Project Engineer: Priscilla A. Cassidy, IFGA, (315) 330-1887

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT ***(Maximum 200 Words)***
The X-Bone is a system for the rapid, automated deployment and management of overlay networks. It consists of a web interface, a coordinating back-end Overlay Manager, and per-node Resource Daemons that configure interfaces, add routes, set security keys, and arbitrate access. The X-Bone deploys IP overlay networks – virtual topologies of encapsulation tunnels – used for network experiments or for the incremental deployment of new network services. The project developed and implemented this distributed system for deploying and managing overlays as well as the architectural extension of the Internet on which it is based. The system was made available as a FreeBSD port and Linux RPM. The project discovered issues with current operating systems and protocols, which were patched and/or communicated to the relevant standards groups. The project indicated opportunities to integrate the Internet architecture extensions in broader ways, to more completely support virtual networks and concurrent distributed overlays; among these are ways to use multi-layer overlays for fault tolerance, and challenges to the interfaces used to configure and access network configuration parameters. The project demonstrated the utility of automated overlay deployment and the impact of general extensions to virtualize the Internet architecture.

14. SUBJECT TERMS
VPN, Virtual network, overlay, network management, network configuration, Internet architecture

15. NUMBER OF PAGES
38

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

Table of Contents

List of Figures

List of Tables

Acknowledgments

The members of the X-Bone project acknowledge the contributions of a number of collaborators who have been particularly helpful in developing the general concepts, debugging, and suggesting directions for research. These include the following individuals, with their particular contributions as indicated:

- Alexander S. Aakesson, USC/ISI - tests of the TetherNet system.
- Jun-ichiro "itojun" Hagino, KAME Project - helped debug numerous FreeBSD and KAME IPsec stack issues and helped highlight issues with non-alias multilayer IP-in-IP tunneling.
- Stephen Kent, BBN - feedback on the implications of using IPsec transport vs. tunnel mode.
- Peter Kirstein, Piers O'Hanlon, Panos Gevros, UCL - used the X-Bone to deploy web proxy cache overlays, debugging application deployment, also testing an early IPv6 port.
- Ted Faber, USC/ISI - feedback on the overall architecture.
- John Heidemann, USC/ISI - feedback on the overall architecture.
- Robert Watson, NAI Labs - helped with the development of the alias-based multilayer IP-in- IP tunnel, as well as general architectural issues

1. Progress and Status

The following is a brief introduction to the architecture of the X-Bone; a more complete discussion can be found in [43], and other related papers [45][46][48][49][51][52][53][55]. The X-Bone system provides automated deployment and management of overlay networks. An overlay network is a virtual topology mapped onto an underlying physical topology (Figure 1).

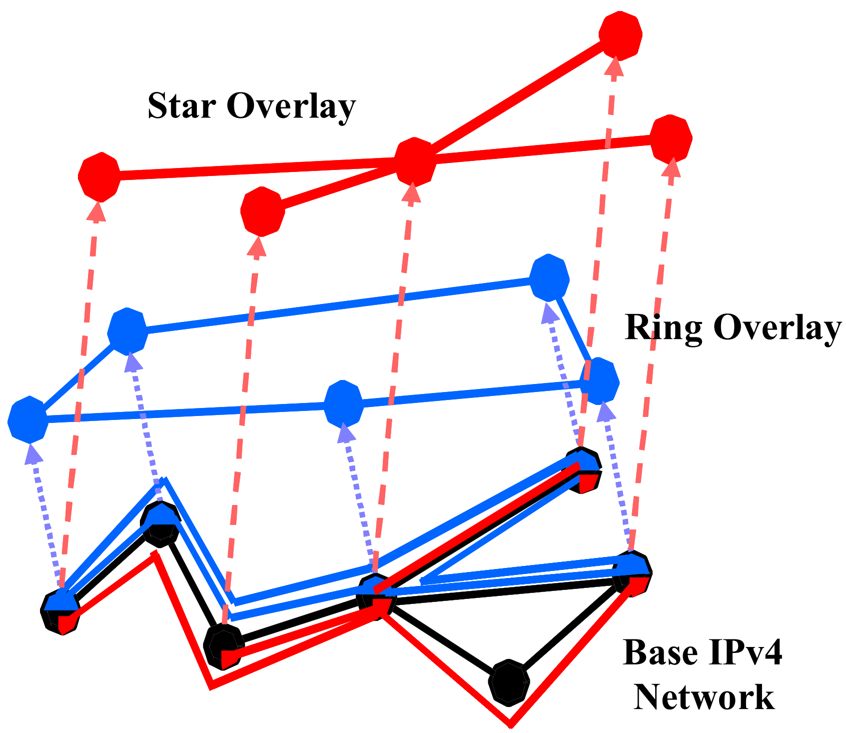

FIGURE 1. Overlay networks

Overlay networks are virtual topologies that use a combination of shared and dedicated resources, to provide a simple network view that conceals unnecessary details about the underlying topology. Overlay networks are composed of routing software installed at selected sites, connected by encapsulation tunnels [30][31][38] or direct links (Figure 1). Recent examples of overlays include the M-Bone for multicast IP [12] and the for IPv6 [1]. A single physical network can support multiple overlays, which can share both link and node resources.

Overlay networks can be useful for interconnecting groups of networked systems, as in Figure 2. They also provide virtual infrastructure on which to incrementally deploy new services or protocols [13][52][55]. Finally, they provide a convenient way to decouple topology from connectivity, to develop new protocols or configure demonstrations or experiments.

The X-Bone is designed to reduce the need for manual participation and intervention in creating, deploying, and managing overlay networks. Table 1 compares the tasks involved in overlay use before the X-Bone to those tasks using the X-Bone

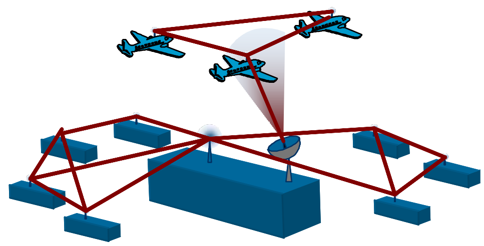

FIGURE 2. Application of overlay networks

Task	*Before X-Bone*	*With X-Bone*
Select properties	manual ad-hoc	manual or via program pick from menus
Select components	manual out-of-band e-mail, phone host-specific state commands	automated OM finds RDs via multicast
Design	manual	automated OM computes topology
Install	manual out-of-band e-mail, phone, telnet, or SNMP	automated OM configures RD via TCP/SSL
Monitor	Various in-band tools infer from visible state	X-Bone tools explicitly monitor state
Dismantle	telnet, SNMP, or e-mail to off-line recorded state	automated via OM, on command via RD, timer-based

TABLE 1 The Impact of X-Bone on Overlay Deployment Effort

The X-Bone is further designed to address two major issues with overlay use: user views and recursion. In the X-Bone, a single host or router may participate in multiple overlays, where each process (e.g., user program or routing instance) can attach to a particular overlay. Figure 3 shows this capability, depicting the simplified output of a real network mapping utility. Each window is a attached to a different overlay, thus the map output is different in each window.

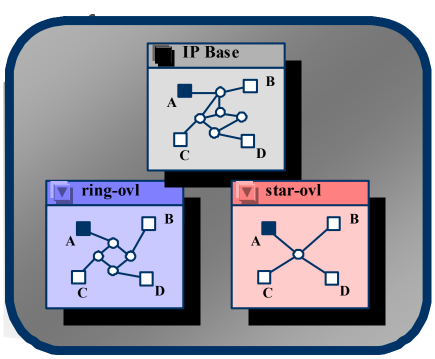

FIGURE 3. Per-process views of the attached overlay network

X-Bone overlays are capable of being deployed recursively, where one overlay is run on another overlay. The X-Bone is based on an architecture where subordinate overlays are represented as routers in the primary (superior) overlay, as shown in Figure 4. This allows layers of overlays, which are useful for experiments run on consortia of nodes, or to support fault tolerance [10]. The X-Bone also supports revisitation, where a single node may participate multiple times in a single overlay, allowing emulation of, e.g., a 100-node ring by revisiting each of 5 nodes 20 times. These capabilities - recursion and revisitation - have been developed as part of the X-Bone network architecture and tested in lab configurations, but have not been added to the automated management system.

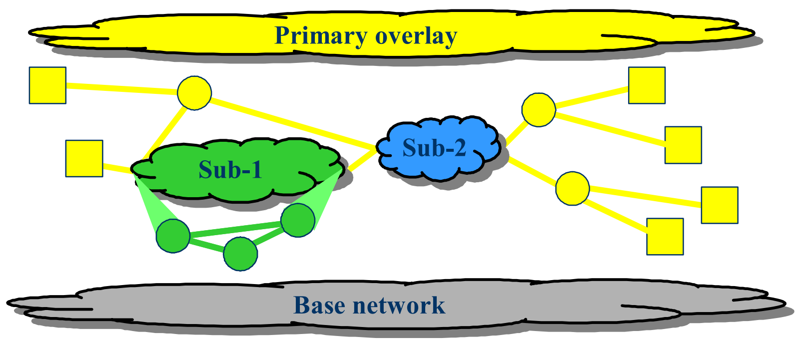

FIGURE 4. Layered overlays

The X-Bone focuses on a limited set of goals, and avoids issues that were deemed to unnecessarily complicate a solution. The X-Bone ***is*** the following, using (as much as possible) existing protocols, operating systems, and applications:

- an automated system for overlay deployment
- a way to interconnect a closed set of participating hosts and routers
- new ways to use overlays (recursion, revisitation, service deployment, fault tolerance)

The X-Bone considers capability to be more important than optimization, and leaves a number of detailed issues to plug-replaceable components. The mapping of the overlay onto base network links is not considered, notably because such optimization is not meaningful. The key purpose of encapsulation is to decouple the content from its path; it would require layer violation to detect whether encapsulated packets traversed the same path. The X-Bone also does not focus on non-IP protocols; because IP is *the* ubiquitous interoperation layer; it assumes only IP capabilities and provides an IP overlay. In many ways the X-Bone parallels the philosophy behind virtual memory - to provide a single, common abstract interface whose ubiquity and automated management are more important than its per-application optimization.

The X-Bone is related to VPNs, as well as to a number of past and current overlay management systems; these are addressed in related papers [43][46][48]. The X-Bone differs from these other systems in three primary ways:

1. ***Integrated end-to-end overlays:*** X-Bone considers overlays as more than an interim solution, more than a place where new protocols are tested before being integrated into the base network. The X-Bone is based on a more general, more permanent extension of the Internet architecture to support network virtualization.

2. ***Recursive Internet architecture:*** X-Bone extends the Internet beyond simple virtualization, including recursion (overlays on overlays), as well as revisitation (using a single node multiple times in a single overlay).

3. ***Deploy an alpha-grade tool:*** The X-Bone software is designed for use by other researchers to deploy testbeds, or for students to share class resources. The code is designed to be robust and to support user extension, to be safe, secure, and coordinated.

1.a. Methods and Procedures

The following is a description of the software architecture of the X-Bone, as well as the user's views of system use and its capabilities.

1.a.1. System Architecture

The X-Bone is implemented as a distributed system, composed of a Resource Daemon on each router or host, a set of Overlay Managers in the network, and a web graphical user interface (GUI) (Figure 5). Each overlay is associated with and managed by a single Overlay Manager (OM), and each resource is configured by a single Resource Daemon (RD). Users can use any web browser to access the Overlay Mangers, to deploy, manage, or dismantle overlays.

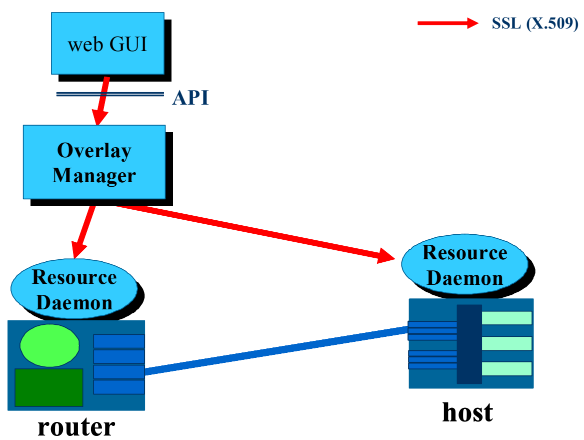

FIGURE 5. X-Bone distributed system architecture

The details of the architecture are discussed in [43], and outlined here. First, a user uses a web browser to issue a request to deploy an overlay to an OM. This OM and the set of machines to be connected are all on the Internet. Figure 6 shows such a group, before the OM begins, where *sin*, *eql*, *cos*, *div* and *sec* are all on a single LAN (shaded), and the other nodes are elsewhere on the Internet. Hosts are shown as rectangles, and routers as ovals.

Figure 7 shows the stages of overlay deployment. First (left), the OM issues UDP multicast invitations, which are authenticated using S/MIME [34][39]. These invitations are issued with particular IP time-to-live (TTL) counts, to reach over the multicast tree and discover sufficient nodes. Each RD receives the request and determines whether its node has sufficient resources and is willing to participate, based on advertised resource counts and user IDs in the invitation, as well as access control lists (ACLs) and usage counts in the RD.

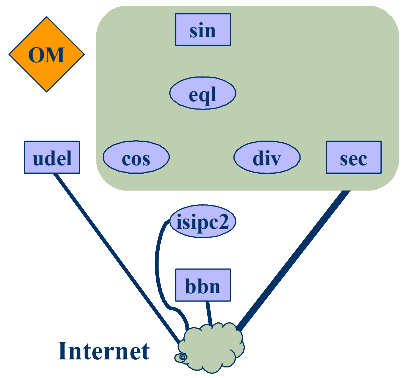

FIGURE 6. OM and nodes on the Internet

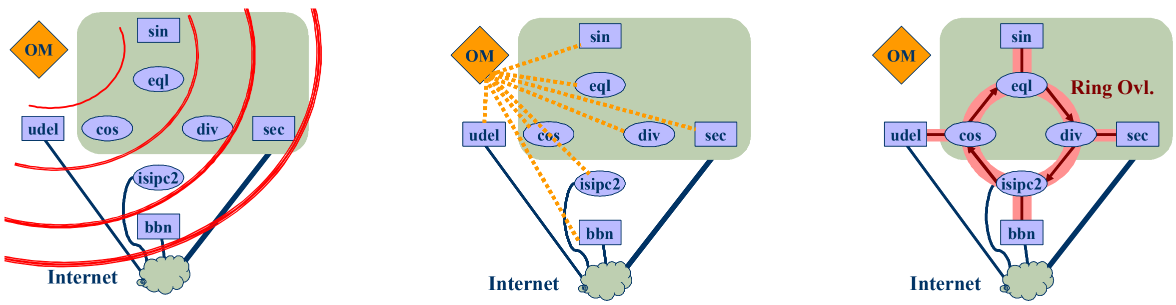

FIGURE 7. Stages of deployment: discovery/invite, configure, and result (left to right).

The OM gathers the replies from the RDs and responds to the GUI with a summary; this summary is either sufficient to deploy the requested overlay or the request is refused. In the former case, each RD temporarily reserves the requested resources on behalf of the requested overlay; these held resources are released if not used within a given time. The summary reply may be more than sufficient for the requested overlay, e.g., providing a list of 10 routers when only 6 are requested, in which case the GUI may refine the reply when it requests the overlay be deployed. The GUI then forwards this (possibly refined) request back to the OM, which proceeds to deploy the overlay. This two-phase invite/reserve and configure protocol is shown in Figure 8.

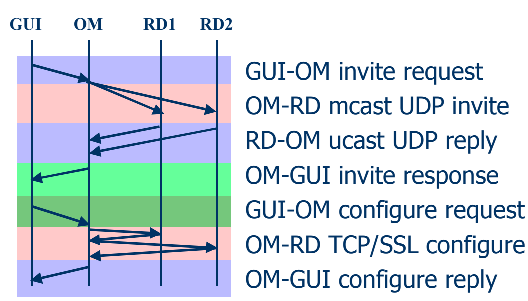

FIGURE 8. API two-phase configuration

The OM then opens TCP connections encrypted via SSL/X.509 (SSL uses X.509) to each RD (Figure 7, middle) [8][18]. The OM issues commands to the RDs to configure tunnels, add IPsec keys, and add routes [26]. The OM communicates to all components in parallel to reduce deployment delay, and implements a basic recovery and rollback procedure to clean up overlays when components fail to configure properly. The result is an overlay, in this case a ring, connecting a distributed set of components in a logical topology (Figure 7, right).

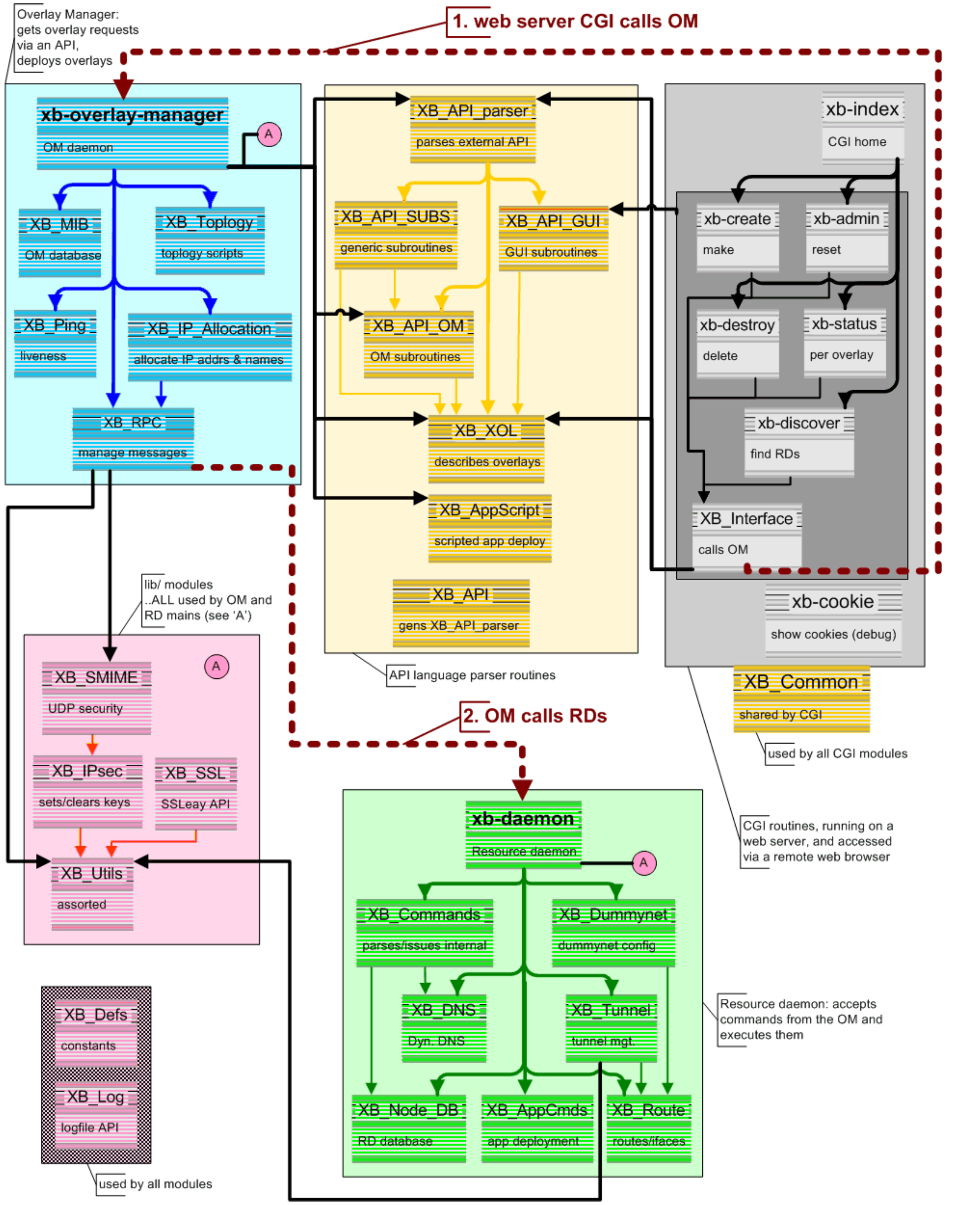

FIGURE 9. X-Bone software architecture flow diagram

Once the components are configured, the OM TCP/SSL connections are closed. After the system is configured, further configuration management, e.g., dismantling the overlay, is performed by the OM using new TCP/SSL connections. After the overlay is up, the OM sends UDP heartbeat messages to the RDs of each overlay, so that when an RD fails to hear the heartbeats it can perform a graceful cleanup of configurations associated with that overlay. Each RD saves a copy of its state to disk, as does the OM, so that the configuration recovers after reboot.

The components of the X-Bone software and their communication relationship is shown in Figure 9. It is largely modular where components are separated by function. There are two command interfaces that operate on reserved TCP port numbers: the X-Bone API (port 2165) and the X-Bone Control (port 265) [22]. The X-Bone API is used by the web interface and stand-alone programs to request overlay deployment, check on the status of overlays, and dismantle overlays. The X-Bone Control protocol is used by the OM for request/response control of the RDs.

The X-Bone system is implemented in Perl as persistent daemons; both the OM and RD run continually and listen for incoming commands. The OM listens on X-Bone-API (2165), and the RD listens on X-Bone-CTL (265); the different ports are used because of the different nature of the two protocols, and because the control protocol necessarily requires root-level access (thus it uses a privileged port number, below 1024).

1.a.2. User Capability

The most important feature of the X-Bone is the model of overlay networking it presents to the user. As shown in Figure 3, each process can attach to a single overlay, without the need for application, operating system, or protocol modifications. This model is achieved by the integration of DNS names with deployed overlay endpoints, as well as the use of non-overlapping IP addresses between sets of hosts and routers that share overlays.

The X-Bone presents a similarly simple model of deploying overlays. Deployment commands issued through the GUI are “do-what-I-mean” API commands, as “deploy a ring of 4 routers with 8 hosts”. Such topology-driven overlay requests are useful for deploying testbed topologies. These API commands are issued by a web server to the OM, where users control the web server via a web browser interface, as shown in Figure 10. This figure shows how overlays are named and defined (simple topologies are supported via the GUI); other specifications (OS type, IPsec algorithm, DummyNet parameters, etc.) are omitted at the bottom for brevity. [36]

After an overlay is successfully deployed, the web interface displays a status of the deployed overlay. Given an overlay named “blue”, hosts apple, pear, and lemon in the base network would be named apple.blue.xbone.net, pear.blue.xbone.net, and lemon.blue.xbone.net in the overlay. By using per-process DNS suffixes (e.g., LOCALDOMAIN in some Unixes), the hosts can be referred to by their prefix (e.g., *ping apple*) in different windows.

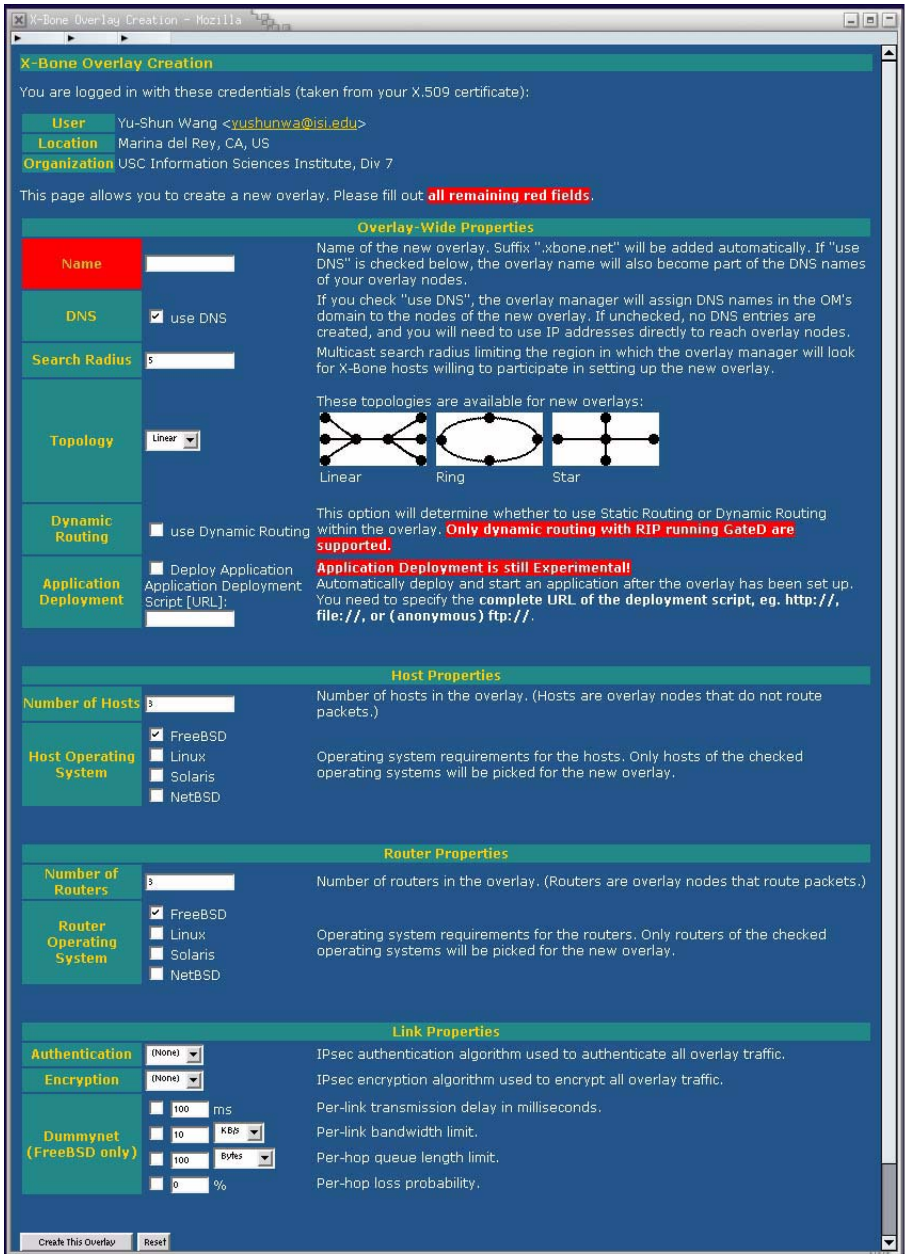

FIGURE 10. Web GUI requesting a ring overlay

1.a.3. Methodology and Principles

The primary goal of the X-Bone is to enable the deployment of "Bones," e.g., 6-Bone (IPv6), M-Bone (multicast), and A-Bone (Active Nets) [1][2][12]. This presumes that the initial configuration of an X-Bone node should be less effort than the configuration of a node for an overlay or non-overlay testbed. The further goal of the X-bone is that the deployment of an overlay in the X-Bone should require near-zero additional effort, and zero additional user training to deploy.

The development of the X-Bone system was incremental, focusing on basic capabilities first and making as many simplifying assumptions as necessary. The system was demonstrated very early, several months into the project, with regular demos thereafter. This forced project personnel to ensure the system was simple to install, simple to use, and reliable. Demonstrations included only features that were tested repeatedly weeks before.

The system was implemented in Perl as scripts, and made ample use of web interfaces throughout. The protocol itself was left in text form. In both cases, these decisions facilitated implementation and debugging. Because the system is intended for deploying networks, deployment speed is not a primary issue. In the current design, overlays with a few dozen nodes can be deployed in seconds, which was deemed sufficient even for a distributed tool.

The code was implemented using a variety of basic software engineering principles, including templates, documented interfaces and design rules, and regular code reviews. This proved critical to the success of the project, enabling a large number of short-term contributors of core code. Similarly, ample logging, error checking, and rollback and recovery assists debugging. The architecture was designed as modular and functional (Figure 9), using existing software where possible (web servers, SSL for TCP security, etc.)[18]. Where capabilities were not available, they were emulated with scripted equivalents, e.g., dynamic DNS was emulated by adding DNS entries to the nameserver config files and signalling it to reload [11].

The multicast-based invitation architecture is designed to distribute control. Nodes need not register with a central authority or even make their presence known until they want to participate in an overlay, and only to the OM issuing the invitation. This helps protect the privacy of the component nodes. Multicast simplifies the invitation architecture, but does compromise the list of possible RDs because they must join the multicast tree. An alternate mechanism could employ a publish/subscribe architecture for distributing invitations.

The centralized OM (as currently implemented) does limit scalability of a single overlay, but the recursive architecture (not yet implemented) supports divide-and-conquer scalability, resulting in a more scalable overall architecture utilizing a simpler building block. The system relies on a single OM per overlay, and the architecture supports backup OMs, though these were not implemented.

1.b. Results and Discussion

The X-Bone project resulted in a tool for deploying and managing Internet overlays, and a more general architectural extension of the Internet to support virtualization. During the course of the project, a number of issues were raised, including bugs discovered, architectural flaws in protocols and operating systems noted, and opportunities for further research observed. The system is being used by a number of researchers worldwide, and has been widely distributed.

1.b.1. Tool

The X-Bone project developed a distributed system for deploying Internet overlays, including:

web server scripts to provide a graphical user interface via a web browser, including scripts to interface to an OM

Overlay Manager (OM) as a persistent daemon written in Perl, listening on port 2165 for API commands

Resource Daemon (RD) as a persistent daemon written in Perl, listening on port 265, running with 'root' privileges, which listens for OM invitations and translates OM requests into local commands. This includes a variant of the RD which emulates an interface to dynamic DNS [11].

The system has been distributed in FreeBSD in the /usr/ports directory (v4.3 or later), and is available from the project web site as a Linux RPM [14][56]. The system has been ported to a number of earlier releases, and the final release supports FreeBSD 4.4 or later and Linux RedHat 7.0 or later (2.4 kernel) [35]. Performance measures indicated a modest performance impact of 15% bandwidth overall, 10's of microseconds per overlay hop.

The system supports the following features:

IP overlay deployment and management, including multiple concurrent overlays (recursion and revisitation are supported by the architecture, but are not implemented in the distributed tool).

Secure and safe overlay deployment, using SSL encrypted connections for configuration [18], S/MIME authenticated and replay-resistant invitations [34], per-user and per-resource access control lists for control, heartbeats for fail-stop operation, hard state (save to disk) with recovery on reboot, and idempotent nested transactions with rollback and recovery for fail-safe configuration.

IPsec authenticated/encrypted overlay links, compatible with dynamic overlay routing. Dynamic overlay routing is not yet supported, due to limitations of GateD and MRTd, and the deprecation of these open-source routing daemons [15][19][26][28].

Advanced configuration features, including automatic DNS names for overlay components, application deployment [51][55], and DummyNet configuration of overlay links [36]. This includes enhancements to deployment performance, including preemptive search (early termination) and parallel configuration, and static overlay routes, including support to manage static routes on nodes using dynamic routing on the base network via GateD or MRTd.

Documented, organized code, using protocol and software version numbers, taint-mode Perl with exception catching, unified subroutine template with standard returns, documented procedures (in, out, side-effects, notes), nonblocking I/O with time-outs to catch blackholes, and a documented API.

A number of other software components or systems were tested to determine the feasibility of porting or inclusion, including SNMP (to replace our custom OM-RD X-Bone-CTL protocol), DHCP, secure dynamic DNS, automatic IPsec keying (IKE), and IPv6 [11][26][32]. Tests were performed to determine the feasibility of porting the X-Bone system to WindowsNT, Windows 2000, and Cisco routers via a buddy host. In most cases, the protocol or system investigated was more complex than needed (IKE, SNMP) or lacked features required (SNMP, DHCP, WindowsNT, Windows 2000). IPv6 and Cisco tests were successful, but not implemented in the distributed code [32]. Other protocols were not sufficiently mature at the time of testing (dynamic DNS [11], Ascend's Tunnel Management Protocol [17]). In some cases, ports and tests resulted in patches to fix implementations, notably of IPsec in FreeBSD/CAIRN, or patches to allow layered encapsulation for Linux, KAME patches for FreeBSD 3.x, and FreeBSD 4.x [7][14][24].

There were a number of releases of the software system, at first internally related to demos, and later to the public. A number of demos were given, notably at each NGI, Active Nets, and Fault Tolerant Nets PI meeting attended, as well as to Rome Labs, the Aerospace Corporation, Centergate, and the Internet Collaboration Board [9][21]. The versions, dates of release, and features are summarized as follows:

v0.0 - Oct. 1998.

First demo - Oct. 1998 Nets PI meeting

FreeBSD 2.2.x/CAIRN

OM as a web script, run on demand, star topology, cleartext key authenticate (global key)

Multicast discovery/invitation, ACLs

v0.1 - Mar. 1999.

User interface revised as web browser to web server back-end

OM as daemon, user-OM encrypted via TCP/SSL, SSL/X.509 encrypt (1 global key)

v0.2 - June 1999.

OM-RD via TCP/SSL, per-overlay SSL keys, RD resource counts, dynamic DNS & IP

v1.0 - Aug. 1999.

First external release

FreeBSD 2.2.8/CAIRN, Linux

Static overlay routes via Gated, overlay links IPsec encrypted/authenticated

Ring & line topology, per-RD SSL/X.509 keys

v1.1 - Nov. 1999.

Demo'd at Educause

Added logging, extensive error recovery and rollback, installation scripts

v1.2 - May 17, 2000.

First wide-scale announced public release; picked up on SlashDot.

FreeBSD port and Linux RPM

Static route updates via MRTd socket, streamlined installation, ping (liveness test)

Web log analysis: software retrieved by 370 users at over 150 different sites, home pages accessed over 11,000 times by over 3,100 sites.

v1.3 - Oct. 2000.

FreeBSD 4.x

DNS as option, invitation replay protection, 2-layer tunnels using aliases

Heartbeats for fail-stop operation, arbitrary IP block allocation (previously /24)

v1.3.1 - Nov. 2000.

Dumbbell (line w/>2 hosts), per-OM messages (invites, destroy-all), dynamic IP allocate

v1.4 - May 2001.

Parallel deployment (10-node in 5 sec., down from 60), S/MIME authenticated invite
Dummynet, increased GUI monitoring, better autoconfigured install
X-Bone port in FreeBSD /usr/ports as of Jan. 20, 2001.

v2.0 - Nov. 2001.

Application deployment, preemptive discovery, API as a true parser
FreeBSD 4.4 (dynamic gifs)
Linux 2.4 (RedHat 7.1/7.2) support incl. bug fix in net/ipv4/ipip.c (patch included)

Other tools and packages were developed to facilitate deployment, simplify implementation, or assist with debugging. A user-space IP-in-IP tunnel *ip-tun* was developed for FreeBSD 3.x [23]; it was obviated by kernel support for tunneling in FreeBSD 4.x [14]. A packet trace analysis tool *IPdump* was developed to decode packets with multiple encapsulations [56].

1.b.2. Architecture

The X-Bone is based on an extension of the Internet architecture to support persistent virtualization, a Virtual Internet [48][53]. The model allows hosts and routers to be members of multiple overlays concurrently. It also allows overlays to be layered, where an overlay at a lower layer overlay is represented as a router in the upper layer overlay (Figure 4). The model supports existing protocols and applications, and can be used in existing operating systems with only minimal support.

The Virtual Internet architecture is motivated by the lack of architectural extensions of the Internet to support VPNs [48][53]. Although there are some IETF working groups starting to address the issue (e.g., Provider-Provisioned VPNs [PPVPN], Mobile-IP, etc.), most are vendor specific or focused on a narrow subset of an implementation, rather than the general architectural issues [20].

The primary contribution of the Virtual Internet architecture is the need for two-layer encapsulation. Internet networking is presumed to rely on a single IP header, but this discounts the extent to which link layer protocols pervade network layer issues, e.g., address resolution, dynamic routing, etc. [25]. The X-Bone architecture developed a unique two-layer encapsulation, where the application packet addressed with the overlay endpoints is wrapped in one IP header indicating the overlay link addresses, and another layer indicating the base network addresses (Figure 11). This architecture allows packets to have the same endpoint headers at every hop in the virtual network, and allows the virtual network to support recursion and revisitation [43][48][53]. The method has been used for subsequent architectures supporting encapsulation other than IP, e.g., in Nortel's PPVPN proposal and Telcordia's recent overlay architecture [27][40].

FIGURE 11. Two-layer encapsulation

Another key contribution of the Virtual Internet is the integration of IPsec with dynamic routing. IPsec includes a tunnel mode, in which both encryption and encapsulation are performed in a single step (Figure 12) [26]. This causes a problem for dynamic routing over hop-by-hop IPsec tunnels, in which a packet arriving at A might need to be keyed with K1 and encapsulated as addressed to B or keyed with K2 and encapsulated as addressed to C, depending on the forwarding table [27][45]. The problem is that IPsec key databases are not integrated with forwarding tables, so that the decision of which key and encapsulation address cannot be made by IPsec.

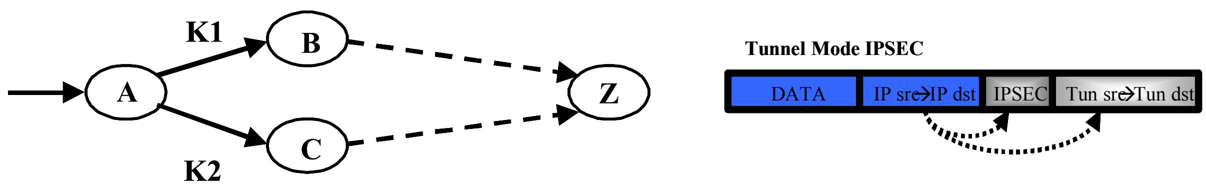

FIGURE 12. IPsec tunnel mode defeats per-hop keys

The solution involves performing forwarding prior to IPsec [45]. In this case, packets are forwarded to internal virtual interfaces V1 or V2, where those interfaces perform encapsulation to B or C correspondingly (Figure 13). IPsec then processes the packets, having sufficient information about the destination (B or C) to determine which key (K1 or K2) to use. This solution has the benefit of simplifying IPsec, as it replaces IPsec tunnel mode with conventional IP in IP tunneling and IPsec transport mode [30]. Alternate proposals in the IETF IPsec WG include integrating routing into IPsec, which is as problematic and complex as adding tunneling, and may not ultimately be as flexible [20].

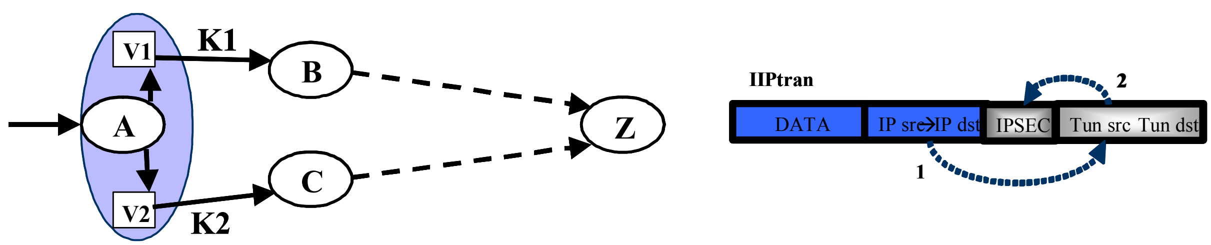

FIGURE 13. IP encapsulation followed by IPsec transport mode enables per-hop keys

There are several other aspects to the Virtual Internet architecture which are summarized in [48] and [53].

1.b.3. Issues and Impact

The development of the X-Bone tool and Virtual Internet architecture raised a number of issues, including bugs in existing kernels and protocols, the need for the development of an API for overlay management, and challenges in supporting concurrent overlays via existing operating system interfaces. It also raised the opportunity to develop hierarchical overlays, which can be used for fault tolerance, and to develop solutions for the challenges presented.

Negatives.

The X-Bone tool exposed bugs in the FreeBSD and Linux operating systems, notably encapsulation limits. Protocol implementers assumed that an IP packet might be encapsulated once, but that attempts to encapsulate further implied a forwarding loop, largely assuming there would be no reason for multiple encapsulations. The X-Bone is not the only reason for needing such; running multicast over a VPN, or mobile IP over multicast might yield similar needs for multiple layers of encapsulation. There remains a desire to avoid resource hogging that might result from an accidental forwarding loop, so the solution is to allow a finite number of successive encapsulations rather than only one. The X-Bone project developed patches to both FreeBSD and Linux enabling this capability.

The X-Bone tool also exposed bugs in implementations of IPsec, notably padding bugs in the CAIRN IPsec stack as well as inconsistencies in the KAME command line interface [7][24]. The project developed patches for the former, and indicated the latter to the KAME group. Further bugs in other programs, notably in DHCP clients and ICMP redirects were also noted and indicated to their respective developers.

During the development of the X-Bone tool, a number of protocols were planned for potential inclusion. Some protocols, as noted earlier (DHCP, SNMP, IKE) were insufficient for inclusion, while others (TMP, RSVP over tunnels) did not materialize as anticipated [17][50][57]. Some operating systems, such as Windows NT and Windows 2000 and the Active Networks OS "Janos" were found to have insufficient networking capability, lacking the ability to tunnel or to completely configure tunnels. IPsec on FreeBSD was found sufficient, as was the NIST IPsec patches for Linux (which are no longer maintained) [29]; the FreeS/WAN Linux IPsec patches (which are more common and supported) were found to be insufficient, due to the inability to associate transport mode IPsec associations with tunneled interfaces.

A number of software components on which the X-Bone tool relies were in substantial flux during the project, notably KAME (esp. its command line interface), the FreeBSD and Linux kernels, and Perl modules for SSL and X.509 [8][18]. Maintaining compatibility with a variety of deployed software proved particularly challenging, requiring tests in the code to determine configuration and use appropriately.

Positives.

A major accomplishment of the project was to demonstrate the capability of Internet-based overlay networks. The X-Bone uses no new protocols, requires no application recompilation or relinking, and requires no operating system extensions. The operating system and protocol modifications required merely fix bugs or relax safety checks. The X-Bone tool has deployed as many as 800 concurrent overlays including nodes distributed world-wide.

X-Bone Internet overlays have been shown to uniquely support recursion, revisitation, and dynamic routing inside separate concurrent overlays. The Virtual Internet architecture on which the X-Bone is based has been shown to be a consistent extension of the Internet architecture, as defined in the host and router requirements specifications [5][6]. This includes support for dynamic routing inside overlays, even where IPsec is used on overlay links.

Part of developing the X-Bone tool was to develop an API (GUI-OM) and a control protocol (OM-RD). The control protocol was deemed to be a simple variant of SNMP with support for multicast, though SNMP was not used due to its complexity. The API was developed and documented as a generic interface for overlay deployment, and is currently being extended to support revisitation and recursion. A component of the API supports application deployment and is developed and documented separately [51][55].

The X-Bone tool's automated overlay deployment has been used to support several other research projects. Application deployment inside the X-Bone has been used to dynamically deploy UCL's RadioActive web servers and an Anetd-based A-Bone [2][4]. This sort of dynamic overlay deployment and application control is considered a way to deploy and manage Active Networks, as well as an non-Active Nets alternative with similar capability [3].

NATs proved a significant challenge to developing and deploying the X-Bone tool [16]. The components of the tool are services that operate on reserved ports, a capability that NATs defeat [44]. This made distributed demonstrations difficult, especially due to the variety of networking environments at demo locations. The result was to apply a variant of the X-Bone technology to lease IP subnets and tunnel them back through NATs to the regular Internet. This technique, known as "TetherNet", enabled not only X-Bone demos, but was also supported dozens of other Active Nets and Fault Tolerant Nets DARPA PI meeting demos, and is being developed into a turnkey device for future use [42].

Impact.

The overall impact of the X-Bone project has been addressed throughout this document, and is summarized here. There are a number of groups who are evaluating the X-Bone tool for use in their projects, including:

- Akesson/ISI - protocol experiments in CAIRN [7]
- Alaettinoglu/USC - used for graduate networking lab class [54]
- Braden/ISI - deployment of the A-Bone [2]
- Canadian Research Center - evaluated for testbed deployment
- Fisher/UCB - high-speed California testbed experiments
- Gotfriend/Raytheon - evaluated for testbed deployment
- Handley/XORP - design of reentrant routing daemons
- Hollander/Aerospace Corp - use for testbed deployment
- Kirstein/UCL - used to configure testbeds for RadioActive (Active Nets) demos
- International Collaboration Board (ICB) - evaluated for testbeds [21]
- LUT, a Finnish university, for use by a Finnish 3G wireless testbed consortium
- Savage/UCSD - protocol experiments
- SRI/ISI - mods to reentrant A-Bone anetd [4]
- Villanueva/U. Catalonia - application deployment issues
- PlanetLab - install X-Bone to deploy overlays [33]

In addition, the architectures and protocols developed by the X-Bone project have impacted a number of standards groups and software organizations. In particular, SRI's implementation of anetd was augmented to support the X-Bone/Virtual Internet style of 'network reentrancy', where the interface subset associated with an overlay is explicitly specified, and all other parameters (usernames, log files, configuration files) are passed on the command line [4][43][48]. The IETF's PPVPN and IPsec working groups are incorporating aspects of the Virtual Internet's two-layer tunneling and considering its use of IPsec transport with IP-in-IP encapsulation as a viable alternative to IPsec tunnel mode [20][26][45][45]. The KAME group is developing IKE extensions (key exchange) for these IPsec transport/IP encapsulation tunnels [24][45]. The FreeS/WAN group is also considering modifications to their system to support keys on virtual interfaces, due to complications noted when the X-Bone was tested on their system in Linux. Finally, the X-Bone overlay deployment system is currently distributed with FreeBSD in the /usr/ports directory, facilitating simple installation [14]

The TetherNet software has been used to support dozens of demos at a number of DARPA Active Nets, Dynamic Coalitions, and Fault Tolerant Nets PI meetings. In these preliminary tests, it has reduced the complexity of configuring demos to local site characteristics, and increased the number of demos that were successful. It is currently being developed as a turnkey system for future meetings, conferences, and demos,[42].

The X-Bone group initiated a summer research program where graduate students participate in real project operation in exchange for educational credit, entitled "Summer Graduate Research Experience Program (SGREP) [37][47]. USC graduate students participated in this program through the X-Bone project, and contributed to tests of WindowsNT, Windows2000, DHCP, SNMP, dynamic DNS, and RSVP, and developed code that was incorporated into the released system to support Linux. The program was also useful for evaluating students for potential addition to the project, some of whom joined as supported GRAs for 1-2 semesters.

1.c. Conclusions

The X-Bone is a system for dynamic deployment and management of Internet overlays, having proven that existing Internet protocols and operating systems are capable of supporting concurrent and recursive overlays. The X-Bone project's Virtual Internet architecture is providing a platform to examine the virtualization of the Internet, with specific solutions for increasing the flexibility and utility of IPsec and the wide-spread use of tunneled overlays.

The Virtual Internet architecture has shown to be sufficiently powerful and flexible, with substantial power to describe revisitation, recursion, and hierarchical deployment. The X-Bone tool is a proof-of-concept that automated deployment using 'do what I mean' (DWIM) interfaces can break the barrier of entry to complex network configuration systems. The X-Bone system has shown that overlay systems have the power to deploy distributed applications similar to that of Active Networks in a simpler, more direct architectural extension of the Internet.

1.c.1. Recommendations

The success of the X-Bone suggests a number of future directions. The Virtual Internet architecture has only been lightly developed; there are further uses of virtualization as a first-class component of the Internet, including the deployment of new services, such as geographic delivery or to provide network-level support for peer-to-peer networks [13]. Layered overlays are useful for providing dynamic routing where not supported in the underlying network, useful to deploy dynamic routing across non-cooperating autonomous systems (AS's), or to hide fault-tolerant heterogeneity; this latter concept is being developed as the DynaBone [10].

The X-Bone tool itself is proof that network management would benefit from hierarchical, distributed automation. Current network configuration is automatic only at the leaves, where DHCP is used, but only at the expense of moving that work up to the DHCP server. The X-Bone removes and automates the work of network management coordination, as has been shown in the use of similar automation to deploy tethered Internets (TetherNet) [42]. There is additional opportunity to extend this sort of automation in a truly hierarchical fashion to other systems, including base network configuration, DNS configuration, and parameter tuning; this general concept is expressed as "Turnkey networks", and is under development.

The use of the X-Bone tool is only now beginning, with substantial opportunities, such as PlanetLab and classroom use, on the near horizon [33][54]. There is also an effort underway to develop the X-Bone tool for further classroom and research use, including support for additional features (IPv6, Cisco routers via buddy hosts), as well as incorporation into the tool of features tested in the lab (revisitation, recursion).

Finally, there is follow-up on the impact of the Virtual Internet architecture on operating system and application design, called "NetFS". The goal is to support per-overlay network management configuration permission without releasing full 'root' privileges. Additional work compartmentalizes the view of the network to that of a single overlay, on hosts or routers that are members of multiple overlays.

2. Topic Areas

2.a. Evaluation of progress

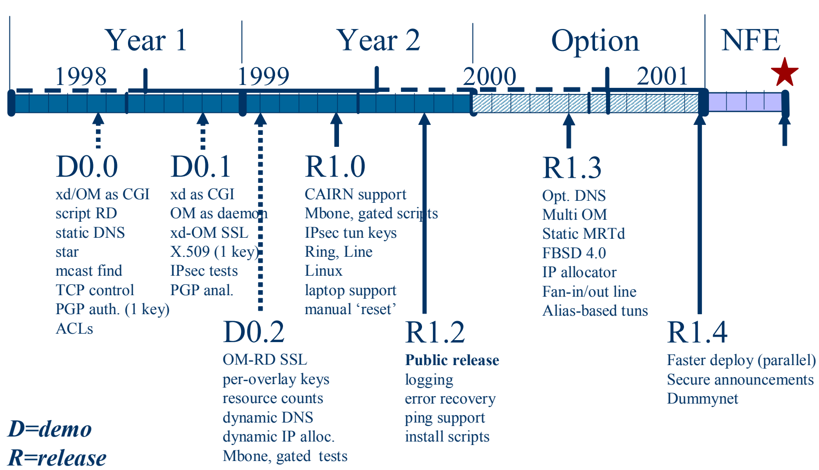

FIGURE 14. Timeline of the X-Bone project, including planned releases.

The X-Bone project achieved most of its deliverables on-time (Figure 14). Some of the project objectives were preempted by the lack of anticipated development, e.g., protocols for in-band configuration of tunnels, interface to RSVP, or advanced routing table management. In addition to its primary goal of a demonstration system for deploying overlay networks, the X-Bone provided the basis for a more general virtual extension of the Internet architecture.

There were a large number of demonstrations of the X-Bone technology. The first occurred at the 1998 DARPA NGI PI meeting, several months ahead of schedule, only a few months into the project. Demos were given at every DARPA NGI PI meeting, as well as at DARPA Active Nets PI meetings, the latter in anticipation of potential use by the Active Nets community for testbed deployment [3]. The X-Bone team developed a system to deploy an A-Bone, and assisted several other projects in the use of this technology to deploy their demos [2]. Finally, the X-Bone architecture (layered tunnels) was applied to develop TetherNet to support network connectivity at DARPA meetings, replacing NAT'd or partial Internet configurations with a more complete, standard Internet environment for demos [16][42][44]. The prototype TetherNet was tested at a few meetings, and was shown to reduce demo debugging and configuration time and to reduce the need for network administrator support.

During the first year, configuration demos included control of host, tunnel, and host-based router components, DNS entries for overlay IP addresses, a registry for shared state and resource location, and a management tool to access components. Resource management included count-based limits and access control lists (ACLs), and the system included the use of a graphical user interface. This phase used SSL/X.509-encrypted connections for configuration management, and supported a star-based virtual topology [8][18].

During the second year, a new technique for virtual overlays was developed which supported revisitation and recursion, and was consistent with IPsec and dynamic routing. The software was released to the public, including support for FreeBSD (generic), FreeBSD (CAIRN patches), and Linux [7]. Monitoring was added, including logging, error recovery and graceful shutdown. The system deployed IPsec-keyed tunnels, and also supported ring and line virtual topologies. The software was packaged with installation scripts for rapid deployment into the FreeBSD (/usr/ports) and Linux (RPM) communities. Tunnel management and tunneled RSVP protocols were not forthcoming as expected, and kernel support for overlapping routing tables was replaced with the use of non-overlapping IP addresses, to avoid the need for custom OS patches [17][41][57]. DHCP lacked sufficient support for multicast and asynchronous interaction.

During the third option year, support for a more advanced and consistent tunnel architecture was developed, using IP aliases. The deployment architecture was accelerated using parallelism, and security was added to the invitation phase. Dynamic IP address allocation was added, as was support for fan-in/fan-out (dog-bone) topologies. The system was augmented to support multiple Overlay Managers (controllers), authenticated invitations, and support additional routing daemons (MRTd) [28]. Loadable forwarding engines were not deemed feasible, but were substituted with application deployment [51][55]. Topology-based optimizations were deemed inconsistent with the use of tunnels, so the GUI was augmented with DummyNet options to increase latency, loss, or limit bandwidth to emulate more constrained networks [36].

During the NFE, the software was prepared for its final public release.

2.b. Publications

Finn, G., Touch, J., “Network Construction and Routing in Geographic Overlays,” ISI Technical Report ISI-TR-2002-564, July 2002. **http://www/touch/pubs/isi-tr-2002-564/**

Hughes, A., Touch, J., “Cross-Domain Cache Cooperation for Small Clients,” Proc. Netstore ‘99, Seattle, WA, Oct. 14-15, 1999.

Karn, P. (ed), “Advice for Internet Subnetwork Designers,” (work in progress), July 2002, draft-ietf-pilc-link-design-12.txt (Touch, J., contributing author)

Touch, J., “Dynamic Internet Overlay Deployment and Management Using the X-Bone,” Computer Networks, July 2001, pp. 117-135. A previous version appeared at Proc. ICNP 2000, pp. 59-68. **http://www.isi.edu/touch/pubs/comnet2001/**

Touch, J., “Those Pesky NATs,” IEEE Internet Computing, July/August 2002, pp. 96. **http://dsonline.computer.org/0207/departments/w4icon.htm**

Touch, J., Eggert, L., “Use of IPsec Transport Mode for Dynamic Routing,” (work in progress), June 2002. **http://www.isi.edu/touch/pubs/draft-touch-ipsec-vpn-04.txt**

Touch, J., Hotz, S., “The X-Bone,” Third Global Internet Mini-Conference at Globecom ’98 Sydney, Australia Nov. 8-12, 1998. **http://www.isi.edu/touch/pubs/gi98/**

Touch, J., Hughes, A., Wang, Y., Eggert, L., “The ISI Summer Graduate Research Experience Program (SGREP),” ISI Technical Report ISI-TR-2002-563, August 2002. (In participants proceedings, 2002 Sigcomm Education Workshop) **http://www/touch/pubs/isi-tr-2002-563.pdf**

Touch, J., Wang, Y., Eggert, L., “Virtual Internets,” ISI Technical Report ISI-TR-2002-558, July 2002. **http://www/touch/pubs/isi-tr-2002-558.pdf**

Touch, J., Wang, Y., Eggert, L., “Virtual Internets for Lab and Class Experiments,” ISI Technical Report ISI-TR-2002-562, August 2002. (In participants proceedings, 2002 Sigcomm Education Workshop). **http://www/touch/pubs/isi-tr-2002-562.pdf**

Villanueva, O.A., Touch, J., “Web Deployment and Management Using the X-Bone,” Spanish Symposium on Distributed Computing, SEID2000, Sept. 25-27, 2000, Ourense, Spain. http://www.ei.uvigo.es/seid2000/welcome_Ingles

Wang, Y., Touch, J., “Application Deployment in Virtual Networks Using the X-Bone,” DANCE: DARPA Active Networks Conference and Exposition, May 2002, pp. 484-493. **http://www.isi.edu/touch/pubs/dance2002/dance2002.pdf**

Wang, Y., Touch, J., Finn, G., Eggert, L., “An API for Describing and Deploying Virtual Networks,” Nov. 2002 (tech. report in progress).

Wang, Y., Touch, J., “Architecture for Layer 3 Virtual Networks,” (work in progress), June 2002, draft-touch-l3vn-arch-00.txt.

2.c. Personnel

Joseph D. Touch, Project Leader (95%-90% throughout; 50% Quarters 9 and 10)

B.S. 1985 University of Scranton (Biophysics, Computer Science)

M.S. 1987 Cornell University (Computer Science)

Ph.D. 1992 University of Pennsylvania (Computer Science)
thesis entitled "Mirage: A Model for Latency in Communication"

Anindo Banerjea, Research Scientist (50% Quarters 1 to 6)

B.S. 1989 Indian Institute of Technology, Delhi (Computer Science)

Ph.D. 1994 University of California, Berkeley (Computer Science)
thesis entitled "Fault Management for Realtime Networks"

Gregory G. Finn, Research Scientist (100% Quarters 2 through 13)

B.S. 1973 Brandeis University (Physics)

M.S. 1978 University of Southern California (Computer Science)

Jarda Flidr, Research Scientist (50% Quarters 4 to 6)

B.S. 1988 Charles University, Prague (Applied Physics)

M.S. 1990 Charles University, Prague (Applied Physics)

Ph.D. 1998 Cornell University (Nuclear Science)
thesis entitled "Understanding and Controlling the Evolution of Surface Morphology"

Stephen Hotz, Research Scientist (50% Quarters 1 to 4)

B.S. 1985 Bowling Green State University (Computer Science)

M.S. 1995 University of Southern California (Computer Science)

Bill Manning, Research Scientist (10% Quarters 5 to 7)

Graduate students:

Wei-Chun Chou (Quarters 5 to 8)

Alper Demir (Quarters 5 to 8)

Osama Dosary (Quarters 9 to 11)

Lars Eggert (Quarters 2 to 9, 11 to 13)

Savas Guven (Quarters 3 to 4)

Amy S. Hughes (Quarters 3 to 9, 11 to 13)

Shitanshu Shah (Quarters 3 to 4)

Ankur Sheth (Quarters 7 to 12)

Stephen Suryaputra (Quarters 2 to 3)

Yu-Shun Wang (Quarters 4 to 9, on loan to A-Bone 9 to 11, back 11 to 13)

Visiting scholar:

Oscar A. Villanueva (Quarters 4 to 5) Univ. Catalonia, Spain

Summer students (no cost to the project):

Wei-Chun Chou, Deepesh Chouhan, Alper Demer, Tze-Wei Sou, Yu-Shun Wang (1999)

Osama Dosary, Ankur Sheth, Dong-Jin Son, Noparut Vanitchanant (2000)

Tzu-Hao Liu, Eunil Seo, Amol Sonpatki (2001)

2.d. Papers Presented at Meetings, Conferences, and Seminars

Joe Touch attended the Grids '98 Workshop in Chicago, IL, July 26-28, 1998.

Joe Touch attended the IETF-42, Chicago, IL, August 23-28, 1998. There he met with various potential users of the X-Bone technology, and participated in working groups on overlay issues.

Anindo Banerjea attended High Performance Networking (HPN'98) in Vienna, Austria in September 21-25, 1998.

Steve Hotz attended Sigcomm'98, Vancouver CA, Oct. 1-4, to meet with Tom Anderson (UWash) and Hui Zhang (CMU), and to solicit additional feedback from the research community.

Joe Touch gave an invited presentation on the X-Bone to the Internet Research Group at Sprint's Advanced Technology Labs, Burlingame, CA, October 5, 1998. There he met with Bryan Lyles and Christophe Diot.

Anindo Banerjea attended the Open Signalling (OPENSIG'98) workshop in Toronto, Ontario October 5-6, 1998 to present "The Xbone: Building Overlay Networks" as part of a panel on "Enabling Virtual Networks."

Joe Touch gave a presentation on advanced network technologies, including the X-Bone, at Loyola Marymount University, Westchester, CA, October 22, 1998.

Joe Touch met with DARPA/ITO PM Jean Scholtz in Fairfax, VA, October 23, 1998, to discuss the application of X-Bone to other ITO programs.

Steve Hotz gave a presentation on the status of the X-Bone at the DARPA NGI PI meeting in Herndon, VA, October 26-29, 1998. Steve also gave the first functional demonstration of X-Bone technology, running remotely in a USC/ISI lab.

Joe Touch gave a presentation entitled "The X-Bone" at the Third Global Internet Mini-Conference, held in conjunction with Globecom '98, in Sydney, Australia, November 9-10, 1998.

Joe Touch attended the DARPA Active Networks PI meeting, at the invitation of Hilarie Orman, DARPA/ITO, to participate in discussions on the use of the X-Bone to coordinate the deployment of the A-Bone, the Active Networks overlay. The meeting was held in NY, NY, November 19-20, 1998.

Joe Touch attended the IETF-43, Orlando, FL, Dec. 6-11, 1998. There he met with various potential users of the X-Bone technology, and participated in working groups on overlay issues.

Joe Touch attended the DisCom2 ASCI workshop in La Jolla, CA, Mar. 11-12, 1999.

Joe Touch attended the IETF-44, Minneapolis, MN, Mar. 16-19, 1999. There he met with potential X-Bone users (e.g., other CAIRN sites), and participated in the WREC and PILC working groups.

Joe Touch co-chaired the IEEE Gigabit Networks Workshop, visited IBM T.J Watson Research Center, and attended IEEE Infocom '99, NY, NY, Mar. 21-25, 1999. At IBM he attended local "Intelligent Networks" meeting. At Infocom he attended TPC and Executive Committee meetings for Infocom 2000.

Joe Touch presented a talk at the Usenix Embedded Systems Workshop, Cambridge, MA, Mar. 29-31. There he met with Kevin Mills, NIST, regarding tech. transfer for X-Bone.

Anindo Banerjea attended the INFOCOM'99 conference in New York, New York, March 21-25, 1999, as well as the associated workshops, GigaBit Networking (GBN'99) March 21, and Open Architectures and Network Programming (Openarch'99) March 26-27.

Steve Hotz and Anindo Banerjea attended an Active Nets Middleware Team Review in Los Angeles, CA, March 30, 1999.

Joe Touch attended the DARPA ITO Active Nets PI meeting in La Jolla, CA, April 15-16, 1999.

Joe Touch attended the IETF in Oslo, Norway, July 12-16, 1999. There he participated in the PILC, NAT (RSIP), and transport working groups, as well as a variety of meetings on overlay networking.

Joe Touch visited Cisco in San Jose Aug. 16, 1999, to discuss overlay network issues with Chase Bailey (Cisco) and John Wakerley (Cisco/Stanford).

The X-Bone group hosted a visit by Jerry Leppert, Bob Kamisky of Rome Labs on Aug.18, 1999, to discuss the status of the X-Bone project, to provide a demonstration of its use, and to discuss its potential applications in teleconferencing.

The X-Bone group hosted presentations by the SGREP summer students, presented to ISI on Aug. 19, 1999. SGREP students each presented their work to the division and invited guests.

Joe Touch co-chaired the IFIP Protocols for High-Speed Networks Workshop '99 (PfHSN) in Salem, MA Aug. 25-27, 1999. While there, he also attended ACM Sigcomm in Cambridge, MA Aug. 30-Sep. 3, and held meetings with Mari Maeda of DARPA (our PM), Scott Bradner of Harvard regarding overlay network security issues and impact on router architectures, and David Wetherall of U. Wash. regarding graph embedding optimizations and how they apply to overlay deployment.

Joe Touch attended the DARPA Active Nets demos in Wash. DC Sept. 23-25, 1999. As part of that trip, he also gave an invited talk at the Univ. of Penn. and met with Roch Guerin Sept 28-29, 1999.

Lars Eggert attended the Gigabit Kits Workshop in St. Louis, MO, July 12-13, 1999. His travel and expenses were supported by the Workshop.

Amy S. Hughes presented a paper on “Cross-Domain Cache Cooperation for Small Clients” at Netstore ‘99 in Seattle, WA, Oct 14-15, 1999.

Joe Touch attended the Infocom 2000 TPC and Executive Committee meetings in NY, NY, Oct. 16-17, 1999.

Joe Touch presented an invited talk at Local Computer Networks in Lowell, MA, Oct. 19, 1999, and presented an invited talk at U. Mass. in Amherst, MA Oct 20, 1999. During this visit he met with Jim Kurose (U. Mass.) Kevin Mills (NIST, at LCN), and Scott Bradner (Harvard).

Anindo Banerjea and Joe Touch presented a demonstration of the X-Bone system at Educause, in Long Beach, CA, Oct. 26-28, 1999. This demonstration was run tethered, remotely using hosts in the lab only. Several potential users, notably Joe Henderson of Dartmouth, expressed interest in secure overlay deployment for work in conjunction with the CDC.

Joe Touch attended the 46th IETF in Wash,. DC, Dec. 7-11, 1999. There he met with Bob Aiken and Bruce Davie (Cisco), and participated in various working groups.

Joe Touch attended the DARPA Active Networks PI meeting and demos in Albuquerque, NM, Nov 17-19, 1999. There he held a working meeting with other overlay networkers, including Peter Steenkiste (DARWIN, CMU), Danillo Florissi and Yechiam Yemini (Columbia), and Bob Braden and Ted Faber of USC/ISI.

The X-Bone group hosted a visit by Doug Maughan, of DARPA/ITO, Dec. 6, 1999, to discuss the state of the project, present a demo, and examine X-Bone’s use in supporting fault tolerance.

Joe Touch gave an invited presentation on the X-Bone at CMU in Pittsburgh, PA, Dec. 14, 1999. There he met with Peter Steenkiste and Hui Zhang.

Greg Finn and Joe Touch attended the DARPA NGI PI meeting in Wash., DC., Dec 15-17, 1999. There they gave a demonstration of the X-Bone system. This demonstration utilized the X-Bone’s security capabilities, with the control system in DC and the overlay in a lab in Los Angeles. Joe Touch also met with Bill Decker of the NSF, to discuss the use of the X-Bone to support shared infrastructure.

Joe Touch and Greg Finn attended the 47th IETF in Adelaide, Australia, March 26-31, 2000. There he presented the X-Bone’s proposal for use of transport mode IPsec to support overlay networks to the IPsec working group, and met with Steve Kent of BBN. They also met with Peter Kirstein, of Univ. College London (UCL), to investigate use by the International Collaboration Board.

The X-Bone group presented a demo of the project to Frank Fernandez, Director of DARPA, Shankar Sastry, Director of ITO, and DARPA PM Sri Kumar on May 26, 2000.

The X-Bone group presented a demo of the project to Jerry Leppert, Rome Labs, on April 19, 2000.

The X-Bone group presented a demo of the project to Cliff Hollander and Jack Carrol, Aerospace Corp., on April 27, 2000.

Joe Touch gave an invited presentation on the X-Bone, and coordinated an installation at the University College of London, May 12, 2000. There he met with Peter Kirstein to discuss collaborations using the X-Bone, notably for dynamic virtual routing in UCL's research backbones.

Joe Touch gave an invited presentation on the X-Bone at the 47th meeting of the International Collaboration Board at FGAN near Dusseldorf, Germany, on May 17-19, 2000.

Joe Touch attended the DARPA Active Nets PI meeting in Portland, Oregon, May 24-25, 2000.

Joe Touch gave an presentation at the DARPA IA&S and FTN joint PI meeting in Honolulu, HI, July 17-21, 2000.

Joe Touch attended the IETF in Pittsburgh, PA, July 31-Aug. 4, 2000. There he had meetings with Robert Watson of NSI and FreeBSD, regarding tunnel support in FreeBSD. He also met with Phil Karn of Qualcomm regarding distributed denial of service, and Ian Heavens regarding TCP RST handling. He also participated in the BEEP (application layer multiplexing) and VPN meetings.

Joe Touch met with Doug Maughan, DARPA PM in Washington DC on Aug. 8, 2000. There they discussed the current and future plans for the X-Bone project, including the status of current funding and plans for the Option year.

Joe Touch attended ACM Sigcomm in Stockholm, Sweden on Aug. 28-Sept. 1, 2000, as Publicity Chair, and was invited to participate in Sigcomm 2001 as Finance Chair and as a member of the Technical Program Committee.

Joe Touch participated in a panel on "End2End Argument vs. Programming the Internet: Are the Two Complementary?" at Opensig 2000 in Napa, CA, Oct. 11-12, 2000.

Joe Touch participated in the TPC meeting for Infocomm 2001 in NYC, NY, Oct. 14, 2000.

Joe Touch presented a paper on the X-Bone at ICNP in Osaka, Japan, Nov. 13-17, 2000.

Joe Touch attended a workshop on end-to-end issues in Stanford, CA, Dec. 1, 2000.

Joe Touch gave a presentation and demo of the X-Bone technology, in conjunction with UCL, at the DARPA Active Networks PI meeting in Atlanta, GA, Dec. 4-8, 2000.

Lars Eggert and Joe Touch attended meetings of the Transport, IPsec, IPsec remote access, transport, NBVPN, and BXXP working groups at the 49th IETF in San Diego, CA, Dec. 11-15, 2000.

The X-Bone group hosted a visit by DARPA PM Doug Maughan at USC/ISI Oct. 25, 2000. Research issues and current project status were discussed, and a demo presented.

Joe Touch presented a talk on the X-Bone and a demo at the DARPA Fault Tolerant Networks (FTN) PI meeting in St. Petersburg, FL, Jan. 16-19, 2001.

Joe Touch attended the 50th IETF in Minneapolis, MN, Mar. 19-23, 2001. There he participated in IPsec, PILC, Transport, Middlebox, IPv6 multihoming, and NBVPN working group meetings.

Yu-Shun Wang and Joe Touch attended the "ABone Status & Anetd V.2 Meeting" at ISI on Feb. 21, 2001. Yu-Shun gave a presentation on "Deploying ABone using XBone".

Joe Touch met with Roberta Gotfried et al. at Raytheon in El Segundo, CA, on Mar. 26, 2001. They discussed using the X-Bone in Raytheon's wireless embedded system, which would require porting the X-Bone to a real-time system such as VxWorks. (this event was omitted last quarter)

Joe Touch attended the Sigcomm 2001 TPC meeting in Philadelphia, PA on April 12, 2001.

Joe Touch attended the Gigabit Networks Workshop (GBN), Openarch, and Infocom 2001 in Anchorage, AK, April 22-29, 2001. Joe chaired meetings of the IEEE TCGN (gigabit networks) and ITC (Internet), and gave an invited presentation on overlays and peer networking at Openarch.

Joe Touch, Lars Eggert, and Yu-Shun Wang gave a demo of the X-Bone at a USC Faculty Retreat in Santa Monica, CA, April 30 2001. The need to support NAT'd Internet connections was raised.

Joe Touch gave a presentation on the Postel Center at the CENIC meeting in San Diego, CA, May 10, 2001. This travel was supported by other funds.

Gregory Finn and Yu-Shun Wang attended the Active Nets PI meeting in Jackson Hole, WY, June 4-6. Greg and Yu-Shun gave a talk on "Deploying ABone Using XBone." They met with Peter Kirstein (UCL) regarding the need for IPv6 and incremental update in the X-Bone.

Joe Touch and Yu-Shun Wang gave a demo of the X-Bone system at DISCEX II in Anaheim, CA, June 13, 2001. They discussed deployment with Cmdr. Ellen Jewett, and developed the NAT fix.

Joe Touch discussed a collaboration on a traffic control module for X-Bone overlays with Mathieu Lemay, of the Communications Research Center (CRC) in Ottawa, Canada, June 2001.

Amy Hughes gave a presentation at the Web Caching Workshop in Boston, MA, June 19-24, 2001. There she met with Oscar Villanueva regarding application deployment on overlay networks.

Joe Touch participated in a review of the Interplanetary Internet in Newark, DE, on July 12, 2001. The IPN was observed to be a variant of a two-layer overlay, spatial on top of local, connected with TCP-based tunnels.

Joe Touch and Yu-Shun Wang attended the DARPA FTN PI meeting in Colorado Springs, CO, July 30 - Aug. 2, 2001. There they gave a talk on the X-Bone, presented a demo, and deployed the TetherNet for use by both the FTN and DC (the earlier week) PI meetings. There they met with Peter Reiher (UCLA) regarding collaboration.

Joe Touch and Lars Eggert attended the IETF in London, U.K., Aug. 6-10, 2001. There they met with Peter Kirstein's group from UCL regarding support for IPv6, incremental modification, and support for netlist-specified overlays. They met with Oscar Villanueva, a former X-Bone project visiting student, who is completing his Ph.D. at the University of Catalonia, Spain, on overlay application deployment. They met with Jun-ichiro "itojun" Hagino, who coordinates modifications to FreeBSD's 'KAME' IPsec stack, regarding issues in extending IPsec dynamic key exchange (IKE) to support the X-Bone's style of IPsec transport-mode encrypted IP encapsulated tunnels. They also participated in the IPsec, PPVPN, Multi6, and Transport working groups regarding the use of the X-Bone's overlay network architecture.

Joe Touch attended Opticomm 2001 in Denver, CO, Aug. 20-22, 2001.

Joe Touch, Lars Eggert, Yu-Shun Wang, and Amy Hughes attended Sigcomm 2001 in San Diego, CA, Aug. 27-31, 2001. There they met with a number of faculty, notably from the UC system (UCSD, UCB, UCLA, UCSC, UCI) regarding the use of the X-Bone for networking experiments on testbeds as well as for teaching networking lab classes.

Joe Touch reviewed the status of the X-Bone with Doug Maughan, the DARPA PM of the FTN program, in Fairfax, VA on July 13, 2001. There he indicated that the X-Bone had been selected for "red team" security analysis by Sandia Labs, and that the X-Bone had also been selected for integration in the NCIC. The "TetherNet" NAT-traversal solution was also discussed for use at the upcoming DC and FTN PI meetings.

Peter Reiher, UCLA, visited on Sept. 20, 2001, regarding potential uses of the X-Bone at UCLA for teaching networking labs and in projects he leads.

2.e. Consultative and Advisory Functions

Joe Touch met with DARPA/ITO PM Jean Scholtz in Fairfax, VA, October 23, 1998, to discuss the application of X-Bone to other ITO programs.

Joe Touch attended the DARPA Active Networks PI meeting, at the invitation of Hilarie Orman, DARPA/ITO, to participate in discussions on the use of the X-Bone to coordinate the deployment of the A-Bone, the Active Networks overlay. The meeting was held in NY, NY, November 19-20, 1998.

Joe Touch presented a talk at the Usenix Embedded Systems Workshop, Cambridge, MA, Mar. 29-31. There he met with Kevin Mills, NIST, regarding tech. transfer for X-Bone.

Joe Touch attended the DARPA ITO Active Nets PI meeting in La Jolla, CA, April 15-16, 1999.

The X-Bone group hosted a visit by Jerry Leppert, Bob Kamisky of Rome Labs on Aug.18, 1999, to discuss the status of the X-Bone project, to provide a demonstration of its use, and to discuss its potential applications in teleconferencing.

Joe Touch attended the DARPA Active Nets demos in Wash. DC Sept. 23-25, 1999. As part of that trip, he also gave an invited talk at the Univ. of Penn. and met with Roch Guerin Sept 28-29, 1999.

Joe Touch presented an invited talk at Local Computer Networks in Lowell, MA, Oct. 19, 1999, and presented an invited talk at U. Mass. in Amherst, MA Oct 20, 1999. During this visit he met with Jim Kurose (U. Mass.) Kevin Mills (NIST, at LCN), and Scott Bradner (Harvard).

Joe Touch attended the DARPA Active Networks PI meeting and demos in Albuquerque, NM, Nov 17-19, 1999. There he held a working meeting with other overlay networkers, including Peter Steenkiste (DARWIN, CMU), Danillo Florissi and Yechiam Yemini (Columbia), and Bob Braden and Ted Faber of USC/ISI.

Joe Touch met with Bill Decker of the NSF, to discuss the use of the X-Bone to support shared infrastructure, Dec. 18, 1999.

Joe Touch and Greg Finn met with Peter Kirstein, of Univ. College London (UCL), to investigate use by the International Collaboration Board March 26-31 at the 47th IETF in Adelaide, Australia.

The X-Bone group presented a demo of the project to Frank Fernandez, Director of DARPA, Shankar Sastry, Director of ITO, and DARPA PM Sri Kumar on May 26, 2000.

The X-Bone group presented a demo of the project to Jerry Leppert, Rome Labs, on April 19, 2000.

The X-Bone group presented a demo of the project to Cliff Hollander and Jack Carrol, Aerospace Corp., on April 27, 2000.

Joe Touch gave an invited presentation on the X-Bone at the 47th meeting of the International Collaboration Board at FGAN near Dusseldorf, Germany, on May 17-19, 2000.

Joe Touch attended the DARPA Active Nets PI meeting in Portland, Oregon, May 24-25, 2000.

Joe Touch gave a presentation and demo of the X-Bone technology, in conjunction with UCL, at the DARPA Active Networks PI meeting in Atlanta, GA, Dec. 4-8, 2000.

Joe Touch met with Roberta Gotfried et al. at Raytheon in El Segundo, CA, on Mar. 26, 2001. They discussed using the X-Bone in Raytheon's wireless embedded system, which would require porting the X-Bone to a real-time system such as VxWorks.

Gregory Finn and Yu-Shun Wang attended the Active Nets PI meeting in Jackson Hole, WY, June 4-6. Greg and Yu-Shun gave a talk on "Deploying ABone Using XBone." They met with Peter Kirstein (UCL) regarding the need for IPv6 and incremental update in the X-Bone.

Joe Touch and Yu-Shun Wang gave a demo of the X-Bone system at DISCEX II in Anaheim, CA, June 13, 2001. They discussed deployment with Cmdr. Ellen Jewett, and developed the NAT fix.

Joe Touch discussed a collaboration on a traffic control module for X-Bone overlays with Mathieu Lemay, of the Communications Research Center (CRC) in Ottawa, Canada, during this quarter.

Joe Touch participated in a review of the Interplanetary Internet in Newark, DE, on July 12, 2001. The IPN was observed to be a variant of a two-layer overlay, spatial on top of local, connected with TCP-based tunnels.

2.f. New Discoveries

During Quarter 5 (1 July 1999 - 30 Sept. 1999) a two-level IP encapsulation system was developed, including the use of transport-mode IPsec to provide and secure overlay links. This system supports the use of unmodified dynamic routing protocols in an overlay, independent of the use of IPsec on its links.

During Quarter 9 (1 July 2000 - 30 Sept. 2000) a new technique for two-layer tunneling using aliases for the first layer was developed. This new technique avoids the need for corrections to most kernel-based tunneling mechanisms to support two-layer tunnels, including FreeBSD and Linux.

3. References

[1] 6-Bone web pages - **http://www.6bone.net/**

[2] Abone (ISI) web pages, **http://www.isi.edu/abone/**

[3] Active Networks web pages, **http://www.ittc.ukans.edu/Projects/ActiveNets/**

[4] Anetd (SRI) web pages, **http://www.csl.sri.com/activate/anetd**/

[5] Baker, F., (ed), “Requirements for IP Version 4 Routers,” RFC-1812, June 1995.

[6] Braden, R., ed. “Requirements for Internet Hosts -- Application and Support,” Internet RFC 1123, IETF, Oct. 1989.

[7] CAIRN web pages, **http://www.cairn.net**/

[8] CCITT. Recommendation X.509: “The Directory - Authentication Framework”. 1988.

[9] Centergate web pages, **http://www.centergate.com/**

[10] DynaBone web pages, **http://www.isi.edu/dynabone/**

[11] Eastlake, D., “Secure Domain Name System Dynamic Update,” RFC-2137, April 1997.

[12] Eriksson, H., “MBone: The Multicast Backbone,” *Communications of the ACM*, Aug. 1994, Vol.37, pp.54-60.

[13] Finn, G., Touch, J., “Network Construction and Routing in Geographic Overlays,” ISI Technical Report ISI-TR-2002-564, July 2002. **http://www/touch/pubs/isi-tr-2002-564**/

[14] FreeBSD project web pages. **http://www.freebsd.org/**

[15] GateD web pages, **www.gated.org**

[16] Hain, T., “Architectural Implications of NAT,” RFC-2993, Nov. 2000.

[17] Hamzeh, K., “Ascend Tunnel Management Protocol - ATMP,” RFC-2107, Feb. 1997.

[18] Hickman, Kipp, “The SSL Protocol”, Netscape Communications Corp., Feb. 9, 1995.

[19] Hunt, C., "Appendix C: A gated reference," in *TCP/IP Network Administration*, O'Reilly & Associates, Inc., CA, May 1994.

[20] IETF web pages, **http://www.ietf.org**

[21] International Collaboration Board, **http://www.cs.ucl.ac.uk/research/icb/**

[22] Internet Assigned Numbers Authority, **http://www.iana.org/**

[23] ip-tun man pages, **http://www.isi.edu/xbone/software/ip-tun**

[24] KAME group web pages, **http://www.kame.net/**

[25] Karn, P. (ed), "Advice for Internet Subnetwork Designers," (work in progress), July 2002, draft-ietf-pilc-link-design-12.txt

[26] Kent, S., Atkinson, R., "Security Architecture for the Internet Protocol," RFC-2401, Nov. 1998.

[27] Knight, P., "Protocol Actions for Virtual Router VPNs," (work in progress), Oct. 2002, draft-knight-ppvpn-vr-protocol-00.txt

[28] mrtd web pages, **http://www.merit.net/net-research/mrt/html/**

[29] NIST Cerberus IPsec patches for Linux, **http://www.antd.nist.gov/itg/cerberus/**

[30] Perkins, C., "IP Encapsulation within IP," RFC-2003, Oct. 1996.

[31] Perkins, C., "Minimal Encapsulation within IP," RFC-2004, Oct. 1996.

[32] Perl 'socket6', **http://www.imasy.or.jp/~ume/ipv6/**

[33] PlanetLab web pages, **http://www.planet-lab.org/**

[34] Ramsdell, B., (ed), "S/MIME Version 3 Message Specification," RFC-2633, June 1999.

[35] Redhat Linux web pages, **http://www.redhat.com**

[36] Rizzo, L., "Dummynet: A Simple Approach to the Evaluation of Network Protocols," ACM Computer Communications Review (CCR), v27 n1, Jan. 1997, pp. 31-41.

[37] SGREP web pages, **http://www.isi.edu/touch/sgrep**

[38] Simpson, W., "IP in IP Tunneling," Internet RFC 1853, Daydreamer, Oct. 1995.

[39] S/MIME web pages, **http://www.openssl.org/docs/apps/smime.html**

[55] Villanueva, O.A., Touch, J., "Web Service Deployment and Management Using the X-Bone," Spanish Symposium on Distributed Computing, SEID2000, Sept. 25-27, 2000, Ourense, Spain. **http://www.ei.uvigo.es/seid2000/welcome_Ingles**

[56] X-Bone web pages, **http://www.isi.edu/xbone**

[57] Zhang, L., Deering, S., Estrin, D., Shenker, S., and Zappala, D., "RSVP: A New Resource ReSerVation Protocol," *IEEE Network*, Sept. 1993.

Appendix A: X-Bone Release Annoucement

```
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
x                                                                   x
x           XX    XX      X-BONE Overlay System                     x
x             X  X                                                  x
x              XX              Software Release                     x
x             X  X                                                  x
x           XX    XX           March 2000                           x
x                                                                   x
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

```
The X-Bone system for automated deployment of VPN / overlay networks
is now publicly available. This is the first major public release, v1.2.

X-Bone dynamically deploys and manages Internet overlays to reduce
configuration effort and increase network component sharing. X-Bone
discovers, configures, and monitors network resources to create
overlays over existing IP networks.

The X-Bone is implemented in Perl, and open source is provided.

The X-Bone can be used for:

      - deploying VPNs

      - sharing lab or wide-area networks
         for multiple, concurrent projects
         for testing protocols and apps on new topologies

X-Bone uses two-layer IP in IP tunneled overlays and supports existing
applications and unmodified routing, multicast, and DNS services.
X-Bone also support IPSec within overlays. Applications can use the
X-Bone without modification or recompilation.

The X-Bone is available for the following operating systems:

      - FreeBSD
              CAIRN 2.5, 3.*, 3.* + KAME IPsec patches

      - Linux RedHat
              6.0, 6.0 + NIST Cerberus IPsec patches, 6.1

The FreeBSD port and Linux RPM have been submitted to the FreeBSD
ports and RedHat Linux RPMs sites; further information and details and
the port and RPM files are currently available at:

      http://www.isi.edu/xbone/

- Joe Touch
  Project Leader, X-Bone group, USC/ISI
  touch@isi.edu
  http://www.isi.edu/touch
  +1 (310) 448-9151

-------------------------------------------------------------------
 Effort sponsored by the Defense Advanced Research Projects Agency
 (DARPA)and Air Force Research Laboratory, Air Force Materiel
 Command, USAF, under agreement number F30602-98-1-0200.

 Copyright (c) 1998-2000 by the University of Southern California.
 All rights reserved.
-------------------------------------------------------------------
```