

# The X-Bone & its Virtual Internet Architecture 10 Years Later

Lars Eggert

Dagstuhl Seminar on Network Virtualization for the Future Internet

Schloss Dagstuhl, Germany

September 18-19, 2008



Nokia Research Center

September 18, 2008

Lars Eggert | Nokia © 2008

**NOKIA**

1

# Talk Outline

history

Virtual Internets

- why

- what

- architecture highlights

related projects at ISI (time permitting...)

- X-Bone, DynaBone, TetherNet



# History

**X-Bone** was a series of research projects at USC/ISI

X-Bone, DynaBone, TetherNet, X-Tend, NetFS, GeoNet, ...

1997-2005+

initial funding from DARPA, follow-on funding from the NSF

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

key results

**an architecture** (the “Virtual Internet” architecture)

**a deployment/management system** (the “X-Bone”)

follow-on work using virtual nets:

DynaBone

spread-spectrum virtual networks

TetherNet

rent real Internet behind firewall + NAT

GeoNet

geographically-routed virtual networks



# Prior & Related Work

## new services & protocols

Cronus, M/6/Q/A-Bone

## multi/other layers

Cronus, Supranet, MorphNet, VANs

## partial solutions

VPN, VNS, RON, Detour, PPVPN, SOS

## virtualization, revisitation, recursion

X-Bone, Spawning, Netlab/Emulab

## OS virtualization

VMware, jails, vserver, XEN, PlanetLab



# Virtual Internet – Why

“network equivalent of virtual memory”

protection

- separate topology, optionally secured
- test + deploy new protocol/service

sharing

- increase utility of infrastructure

abstraction

- adapt topology to application

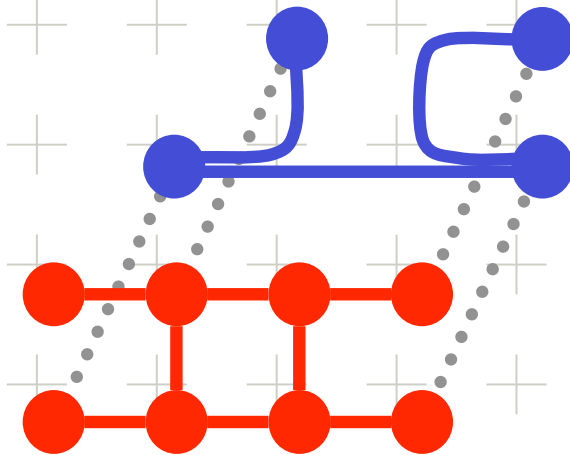


# Virtual Internet – What

network = hosts + routers + links

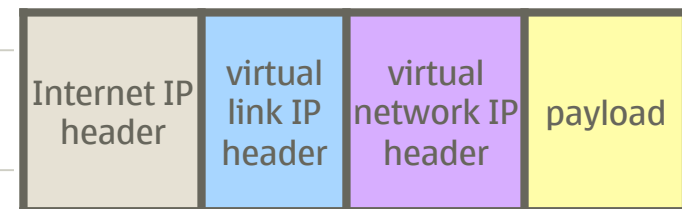
virtual network =

- + virtual host → packet src/sink
- + virtual router → packet gateway
- + virtual link → tunnel X over Y



virtual **Internet** – “network of networks”

- use Internet as physical media
- create virtual **link** & **network** layers
- strong L2 vs. weak L3 host model



a virtual Internet should look **exactly** like the real thing

“if an app can know it runs in a VI, we did it wrong”

# VI Architecture Feature – Recursion

virtual Internets **on top of** virtual Internets

our **litmus test**:

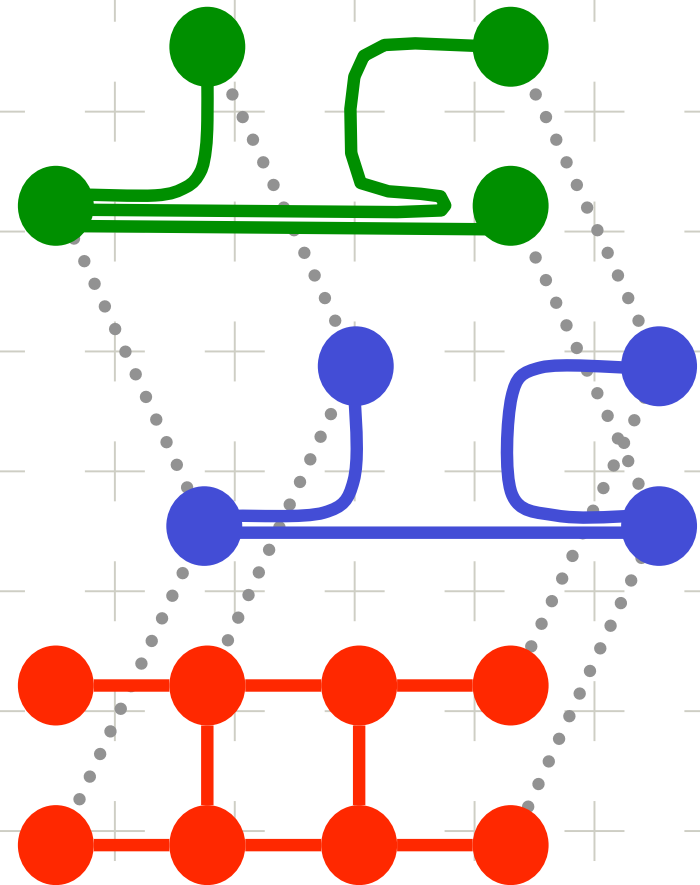
system should be able to do recursive VI-in-VI without hacks

recursion has real uses cases

e.g., allows transparent reconfiguration  
change outer VI w/o affecting inner  
fault tolerance, basis for DynaBone

also allows VI “embedding”

“router is a network inside”



# VI Architecture Feature – Concurrency

one node participates in multiple virtual Internets at the same time

basis for isolation & abstraction

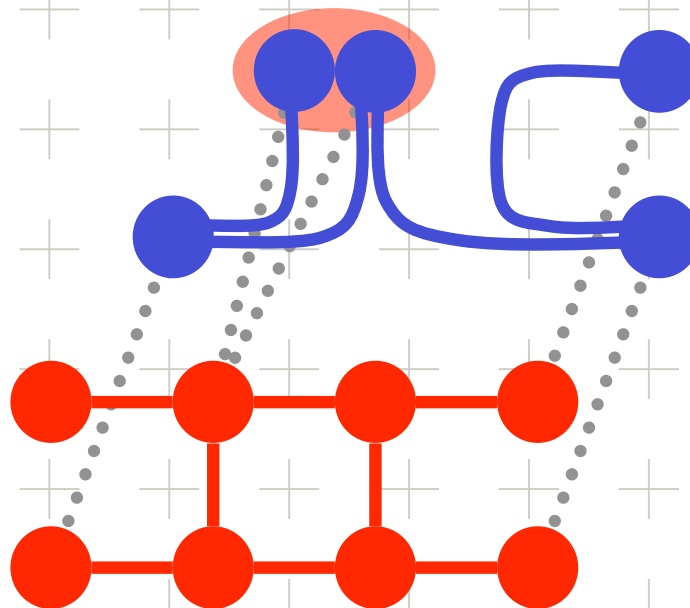
bind different apps/VMs to different VIs on the same physical node





# VI Architecture Feature – Revisitation

one node participates in the same virtual Internet but multiple times  
allows creation of VIs larger than physical resources  
fully decouples virtual from physical topologies



# VI Architecture Feature – Hop-by-Hop Security

security in the Virtual Internet architecture is a virtual link property

decoupled from topology

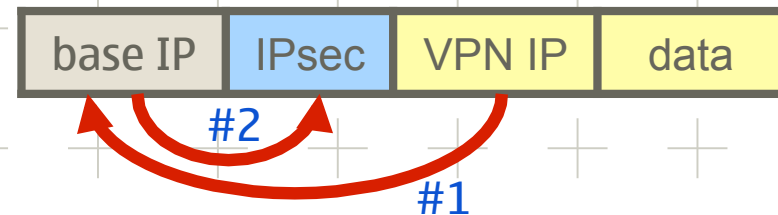
transparently coexists with end-to-end security inside the VI

transparently coexists with security underneath a VI

IPsec tunnel mode



IPIP tunnel + IPsec transport mode



IPIP tunnels + IPsec transport mode

modular tunnel mode equivalent

huge IETF debate around 2000 (draft-touch-ipsec-vpn-05.txt)



# The X-Bone System

deployment + management system for virtual Internets

programs → standardized API

humans → web interface

high-level virtual network description language

express virtual topology + services

XML

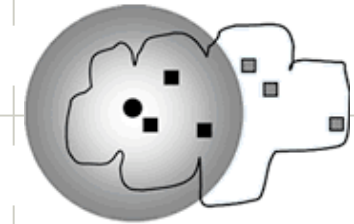
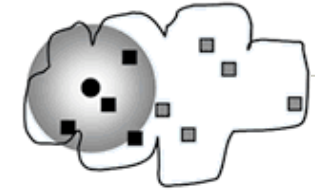
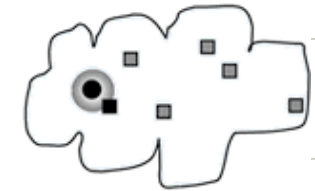
collaborating, distributed management daemons

multicast expanding-ring discovery

distributed resource reservation

instantiate + manage virtual network

non-goals: topology optimization, non-IP VIs, ...



# X-Bone Screenshots

**X-Bone Overlay Creation**

You are logged in with these credentials (taken from your X.509 certificate):

**User** Yu-Shun Wang <yushunwa@isi.edu>  
**Location** Marina del Rey, CA, US  
**Organization** USC Information Sciences Institute, Div 7

This page allows you to create a new overlay. Please fill out **all remaining red fields**.

**Overlay-Wide Properties**

**Name**  Name of the new overlay. Suffix ".xbone.net" will be added automatically. If "use DNS" is checked below, the overlay name will also become part of the DNS names of your overlay nodes.

**DNS**  use DNS If you check "use DNS", the overlay manager will assign DNS names in the OM's domain to the nodes of the new overlay. If unchecked, no DNS entries are created, and you will need to use IP addresses directly to reach overlay nodes.

**Search Radius**  Multicast search radius limiting the region in which the overlay manager will look for X-Bone hosts willing to participate in setting up the new overlay.

**Topology**  These topologies are available for new overlays:  
  
 Linear Ring Star

**Dynamic Routing**  use Dynamic Routing This option will determine whether to use Static Routing or Dynamic Routing within the overlay. **Only dynamic routing with RIP running GateD are supported.**

**Application Deployment**  Deploy Application Automatically deploy and start an application after the overlay has been set up. You need to specify the **complete URL of the deployment script**, eg. `http://file://`, or (anonymous) `ftp://`.

**Host Properties**

**Number of Hosts**  Number of hosts in the overlay. (Hosts are overlay nodes that do not route packets.)

**Host Operating System**  FreeBSD  Linux  Solaris  NetBSD Operating system requirements for the hosts. Only hosts of the checked operating systems will be picked for the new overlay.

**Router Properties**

**Number of Routers**  Number of routers in the overlay. (Routers are overlay nodes that route packets.)

**Router Operating System**  FreeBSD  Linux  Solaris  NetBSD Operating system requirements for the routers. Only routers of the checked operating systems will be picked for the new overlay.

**Link Properties**

**Authentication**  IPsec authentication algorithm used to authenticate all overlay traffic.

**Encryption**  IPsec encryption algorithm used to encrypt all overlay traffic.

100 ms Per-link transmission delay in milliseconds.

10 Kbps Per-link bandwidth limit.

100 Bytes Per-hop queue length limit.

0 % Per-hop loss probability.

Dummynet (FreeBSD only)

**X-Bone Overlay Status**

You are logged in with these credentials (taken from your X.509 certificate):

**User** Yu-Shun Wang <yushunwa@isi.edu>  
**Location** Marina del Rey, CA, US  
**Organization** USC Information Sciences Institute, Div 7

**Overlay Parameters**

**Name** line-test.xbone.net

**Topology** linear

**Overlay Properties** Authentication Encryption Dynamic Routing Dummynet  
 none 3des No No

**Creator** Yu-Shun Wang <yushunwa@isi.edu>

Role	Resource	Daemon	Local	Tunnel	End	Status
Router	chn.isi.edu		172.26.1.2	172.26.1.1	up	
	128.9.160.76		172.26.1.6	172.26.1.5	up	
	FreeBSD/KAME		172.26.1.13	172.26.1.14	up	
Router	hbo.isi.edu		172.26.1.14	172.26.1.13	up	
	128.9.160.75		172.26.1.17	172.26.1.18	up	
Host	mtv.isi.edu		128.9.160.79	172.26.1.9	172.26.1.10	up
			FreeBSD/KAME			
Router	sd.isi.edu		172.26.1.10	172.26.1.9	up	
	128.9.160.93		172.26.1.18	172.26.1.17	up	
Host	tlc.isi.edu		128.9.160.31	172.26.1.1	172.26.1.2	up
			FreeBSD/KAME			
Host	tnn.isi.edu		128.9.168.57	172.26.1.5	172.26.1.6	up
			FreeBSD/KAME			

Back to the [main X-Bone page](#).

# X-Bone Status

current release: 3.2

**mature**: 10 years of open source availability

platforms: FreeBSD, Linux

unofficial: NetBSD, Cisco

widely used (by 2003):

UCL, UPenn, Aerospace, DOD Canada, Sinica Taiwan + more

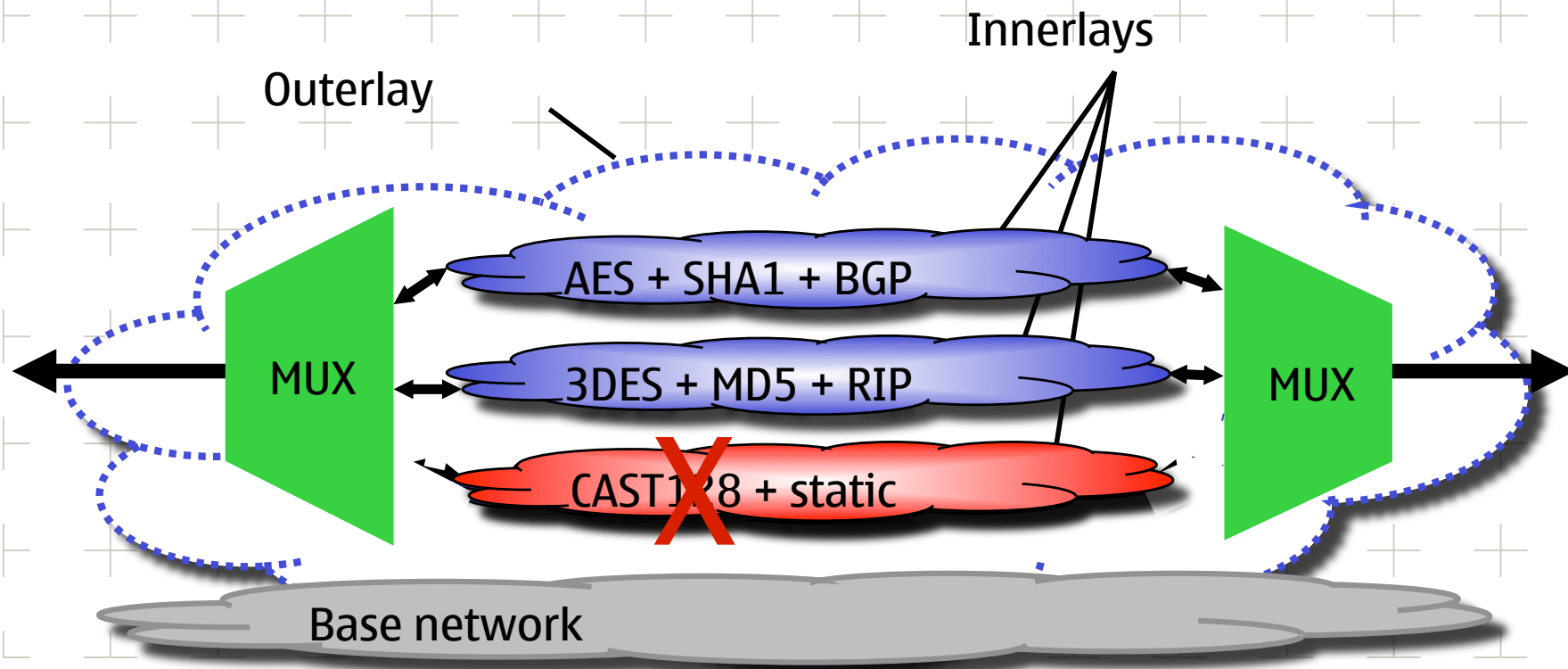


# Related Work at USC/ISI



# DynaBone

parallel inner virtual networks = algorithmic & protocol diversity  
spread-spectrum multiplexer, wrapped inside outer virtual network



# TetherNet

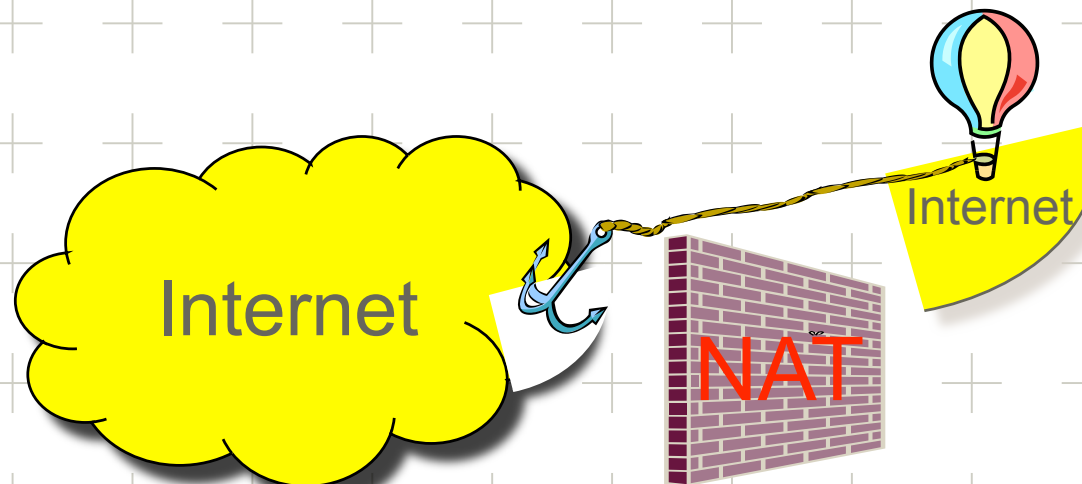
issue: firewalls, NATs, clueless ISPs

broken end-to-end connectivity

solution: relocate real Internet subnet

real = routable IP + DNS + no fw + ...

tunnel subnet from anchor router to tether router at remote site





# TetherNet Features

true Internet behind NATs  
and firewalls

IPv4 + IPv6

multicast

fwd/rev DNS

traffic shaping

802.11b AP

secure: IPsec for traffic, X.509 for user auth

web interface configuration

U.S. patent filed, talks with licensees



# TetherNet Screenshots

TetherNet Rental - Mozilla [Build ID: 2002091014]

File Edit View Go Bookmarks Tools Window Help Debug QA

Back Forward Reload Stop <https://router.local.lan/cgi-bin/start-tethern>

## TetherNet Rental

Required rental parameters:

Rental Site	Marina del Rey, USA 198.32.16.91	Pick a preconfigured TetherNet rental site close to you, or specify the IPv4 address of a custom one.
Subnet Size	9 hosts	Effective usable subnet size of the new TetherNet. Choose a large enough size for the planned number of client end hosts.
Access Code		Some TetherNet rental sites require access privileges. If you have been provided with an access code for a rental site enter it here, otherwise leave empty.

Start TetherNet Service

Optional rental features:

Relay Type	<input type="radio"/> TCP Local Port: auto <input checked="" type="radio"/> UDP Remote Port: auto <input type="radio"/> IPv4	Use specified relay method for TetherNet. Local and remote ports are only meaningful for TCP and UDP relays, and may be set to auto if no specific port setting is required to pass middleboxes.
Relay Encryption	<input type="checkbox"/> encrypt with aes	Optionally, the traffic between the TetherNet box and the rental site can be encrypted - <u>this does not provide end-to-end security.</u>

Optional advanced networking features:

IPv6	<input checked="" type="checkbox"/> enable	Enable IPv6 routing on the TetherNet, including autoconfiguration. IPv6-aware end hosts receive IPv6 addresses automatically through router solicitation.
Multicast	<input checked="" type="checkbox"/> enable IPv4 <input checked="" type="checkbox"/> enable IPv6	Configure IPv4 and/or IPv6 multicast connectivity for the TetherNet.
DHCP Server	<input checked="" type="checkbox"/> enable Range:	Start a DHCP server on the LAN interface, enabling end hosts to dynamically request IPv4 addresses. The Range field specifies how many IP addresses at the bottom of the allocated subnet block are handed out via DHCP (the rest are available for static assignment.)

Start TetherNet Service

TetherNet © 2001-2002 X-Bone/DynaBone at USC/ISI. [Version Info] [Contact] [Main Page]

Done

Rental Server Response - Mozilla [Build ID: 2002091014]

File Edit View Go Bookmarks Tools Window Help Debug QA

Back Forward Reload Stop <https://router.local.lan/cgi-bin/start-tethern>

## Rental Server Response

Rental information:

Rental Server	anchor.postel.org <larse@isi.edu>
Organization	USC/ISI, TetherNet
Location	Marina del Rey, CA, US
Local Time	Tue Sep 17 15:13:00 2002

Rented network block:

IP Block	206.117.27.16
Size	16

TetherNet properties:

Rental Site	198.32.16.91
LAN Size	206.117.27.16/28, 9 hosts, IP addresses 206.117.27.22 - 206.117.27.30
DNS Suffix	tethered.net
Tunnel Type	UDP (local port 35770, remote port 34213)
DHCP Service	on, handing out the range from 206.117.27.22 to 206.117.27.22
Tunnel Encryption	rijndael-cbc
IPv6	on, allocated prefix is 3fe:825:117:27:16::/64
IPv4 Multicast	on
IPv6 Multicast	on

Check status:

Check Status It may take several seconds to bring up the rental.

TetherNet © 2001-2002 X-Bone/DynaBone at USC/ISI. [Version Info] [Contact] [Main Page]

Done

# Other Projects

## X-Tend

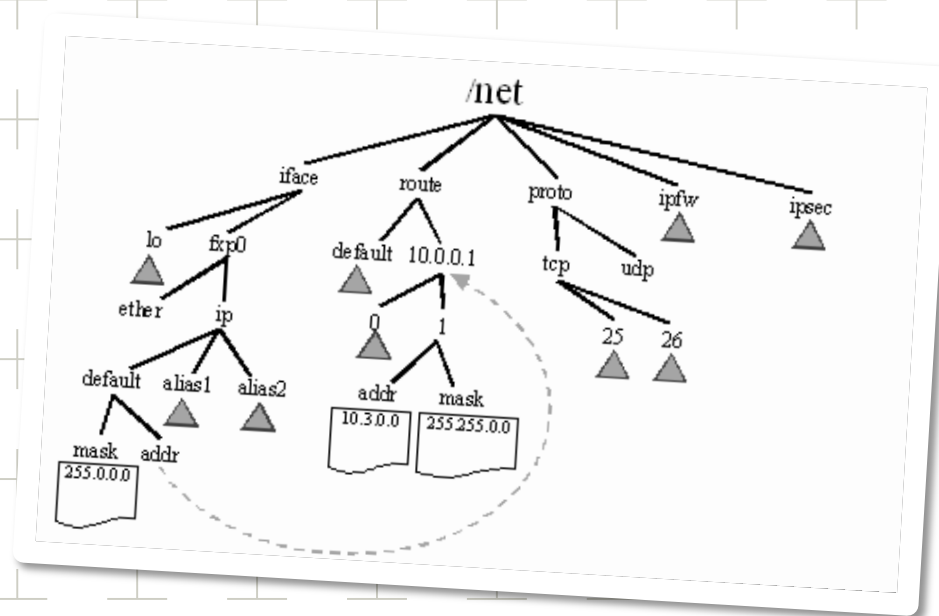
maintain + extend X-Bone as tool for research + education

## GeoNet

geographically-addressed overlays

## NetFS

access control for the network stack via a pseudo file system



**THANK YOU!**



**Nokia Research Center**

September 18, 2008

Lars Eggert | Nokia © 2008

**NOKIA**

20