

6TALK :

IPv6 Transition Toolbox



Joo-Chul Lee, ETRI

March. 13. 2002, Madrid Global IPv6 Summit

Contents Outline

- What is 6TALK ?
- Why 6TALK ?
- 6TALK Scenarios & Solution
- 6TALK Implementations
 - Function Specification
 - Hardware
 - User Interfaces
- 6TALK Test Environments
- Summary & Future Work



➔ What is 6TALK?

Why 6TALK ?

6TALK Scenarios & Solution

6TALK Implementations

6TALK Test Environments

Summary & Future Work

What is 6TALK ?

● 6TALK ?

- IPv6 TrAnsLator of Krv6
- *"Please talk with IPv6 ~"*

● 6TALK history

- Launched by ETRI, (2000-2002)
- Main solution and Strategy of Krv6 project (www.krv6.net)
- IPv4 Web server - www.6talk.net
- IPv6 Web server - www.lab.6talk.net

● 6TALK goal

- Smart Transition Toolbox Supporting IPv4/IPv6 Inter-working
 - Enables an IPv6 island to connect the IPv4 Internet and other IPv6 networks seamlessly in initial IPv6 transition period



What is 6TALK?

→ Why 6TALK?

6TALK Scenarios & Solution

6TALK Implementations

6TALK Test Environments

Summary & Future Work

Why 6TALK ?

- Initial IPv6 Transition Period
- Emerging IPv6 islands
 - Incremental deployment on existing network
 - Corporate networks
 - Large scale deployment of new infrastructure
 - Mobile 3G networks
- Facts of Traffic in IPv6 islands
 - IPv6 is NOT backwards compatible with IPv4
 - Most popular servers are running IPv4 today
 - Don't want isolation
- Requirement of IPv6 Traffic
 - Seamless Routing and Forwarding
 - IPv4 Translation
 - IPv6 Tunneling



What is 6TALK?

Why 6TALK?

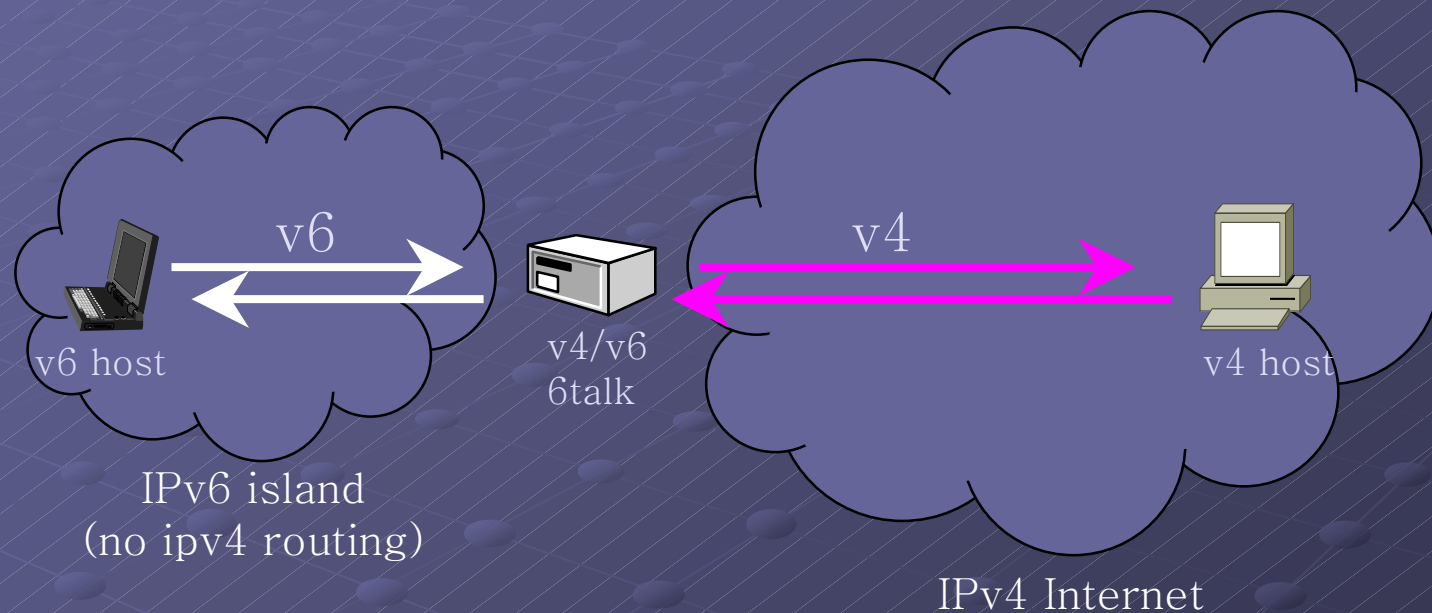
➔ 6TALK Scenarios & Solution

6TALK Implementations

6TALK Test Environments

Summary & Future Work

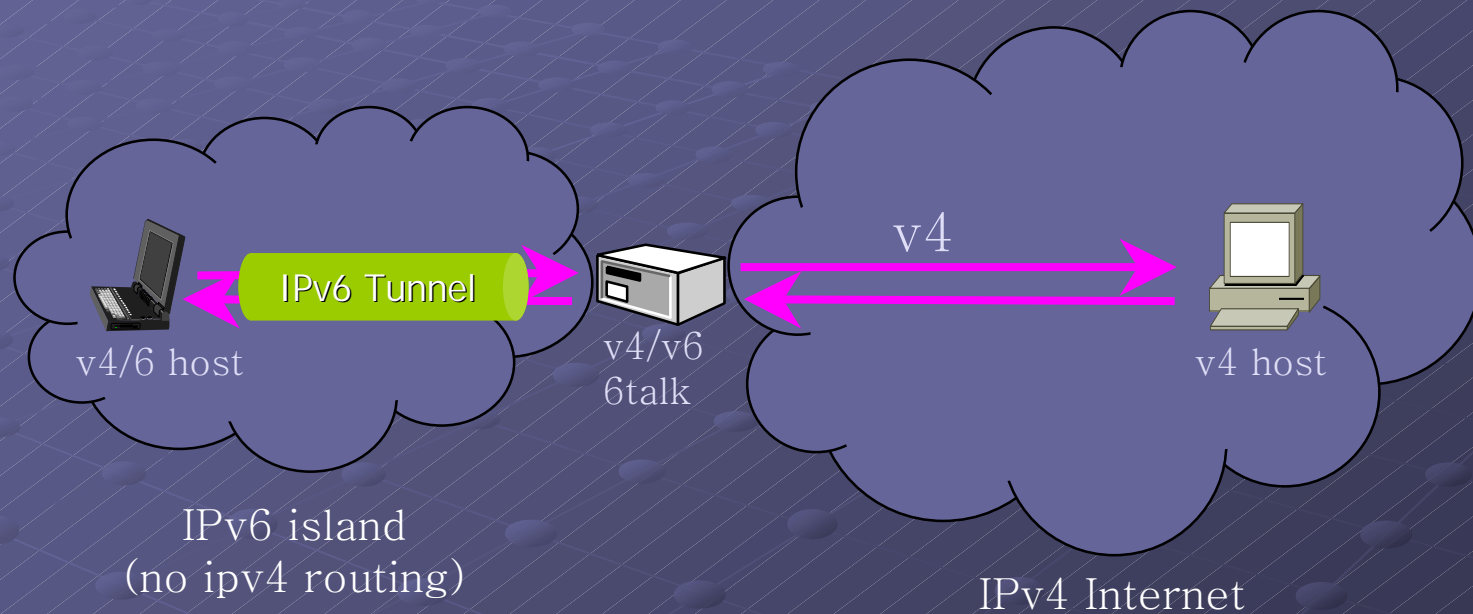
6TALK Scenarios(1/3)



- 6TALK box
NATPT - enable
(DSTM - disable)
- Decision – 6TALK box
All of IPv6 packets which have NAT-PT dummy prefix are translated into IPv4 packets, and vice versa
- Pros & Cons
Easy to deploy (no change of IPv6 hosts)
Does not support end-to-end connectivity and have scalability problem (# of flow)



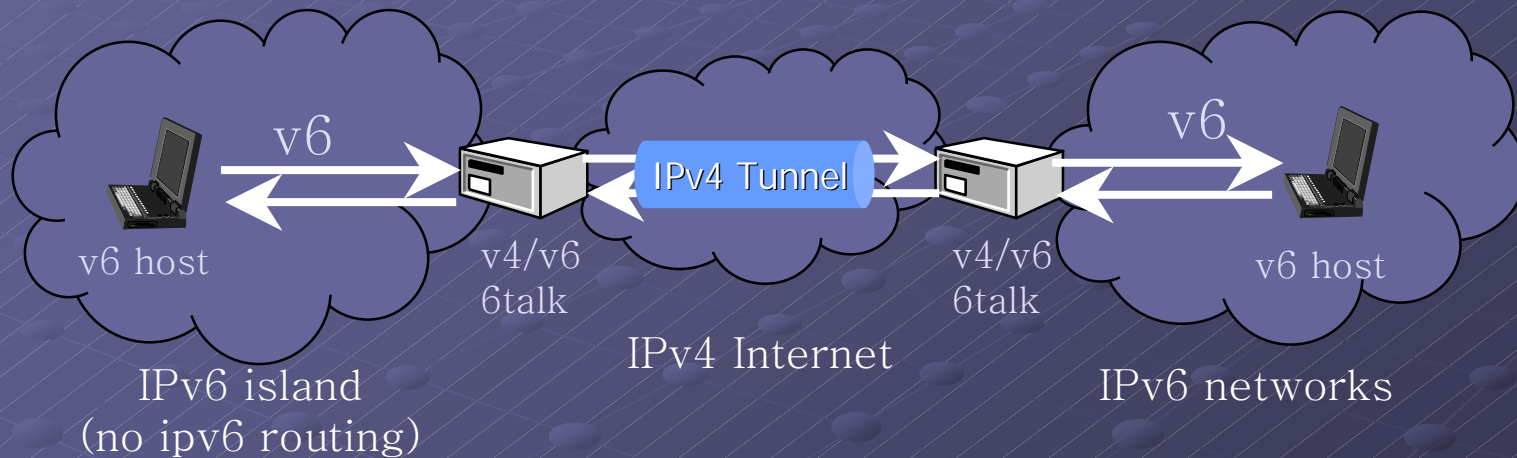
6TALK Scenarios(2/3)



- 6TALK box
 - (NATPT - disable)
 - DSTM - enable
- Decision - IPv6 host
 - If **DNS query result is an IPv4 address(A)** then select **DSTM mechanism**
- Pros & Cons
 - Support End-to-end connectivity
 - Need extra components (DHCPv6, DSTM client daemon,...)



6TALK Scenarios(3/3)



- 6TALK box
 - 6to4 or configured tunnel configuration
- Decision – IPv6 host
 - If **DNS query result is a 6to4 address (AAAA)**, then select **6to4 mechanism**
 - If **DNS query result is other IPv6 address**, then select **normal IPv6 routing**
- Pros & Cons
 - VPNv6 support



6TALK Solutions

- 6TALK

- Smart Toolbox for smooth migration towards IPv6
- NAT-PT/SIIT, DSTM, 6to4, Configured Tunnel, etc.

- 6TALK - IPv4/IPv6 Inter-working Solutions

- Scenario 1 & Scenario 3, or
- Scenario 2 & Scenario 3

- How can network administrators choose proper mechanisms ?

- # of IPv6 users (scalability), network transition period, IPv6 services type, etc.
- *DNS query information is important !*
 - Enhanced DNS ALGs



What is 6TALK?

Why 6TALK?

6TALK Scenarios & Solution

➔ 6TALK Implementations

6TALK Test Environments

Summary & Future Work

6TALK Specifications

- Transition mechanisms

- NAT-PT

- Implement NAT-PT(RFC2766) and SIIT(RFC2765)
 - It also supports NAPT-PT
 - It is based on Linux Netfilter framework

- DSTM & DSTM extensions

- Will implement DSTM(draft-ietf-ngtrans-dstm-05) and DSTM extension(draft-ietf-ngtrans-dstm-ext-00)
 - Plan to implement in 2002

- IPv6 over IPv4 Tunneling (including 6to4)

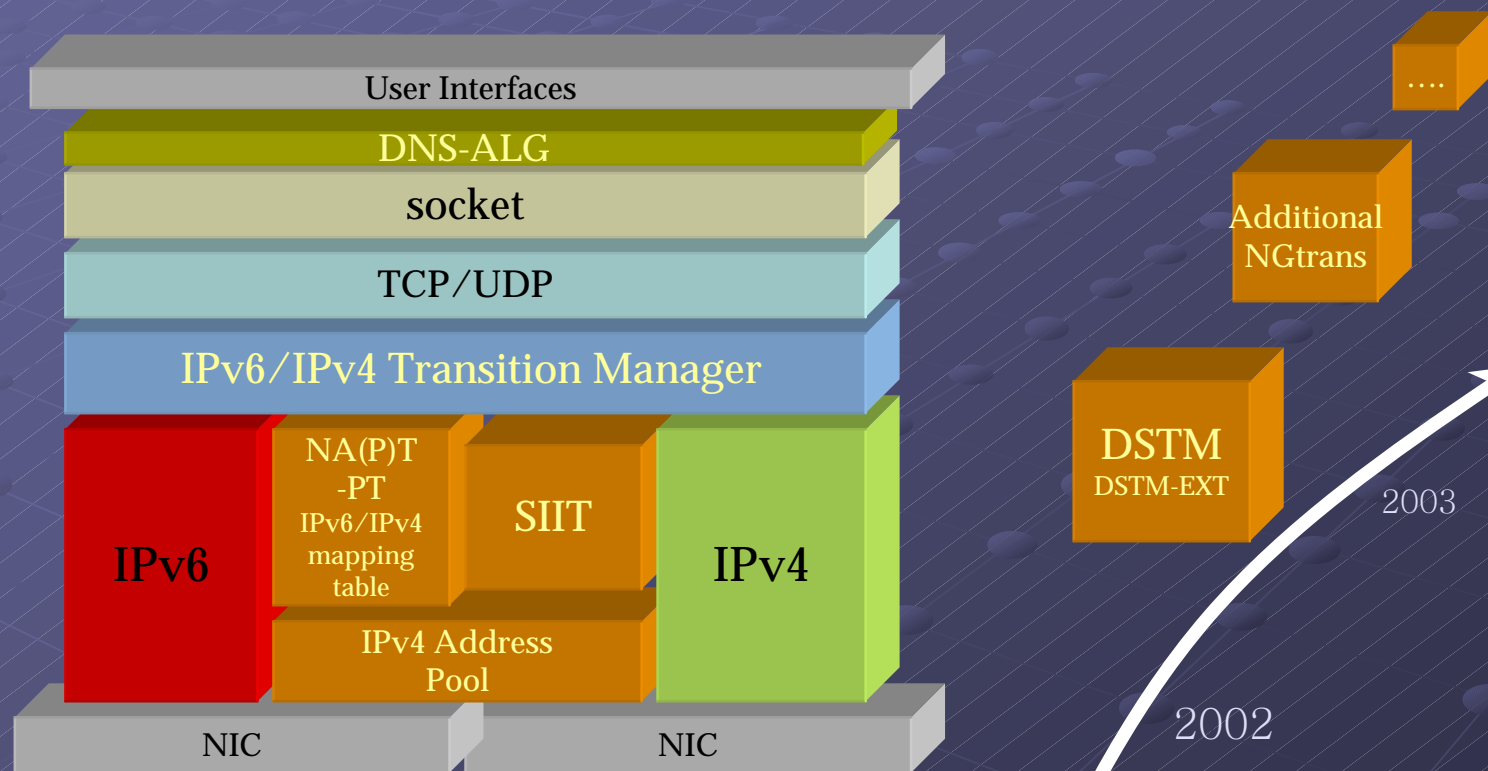
- Already implemented in Linux

- Operation environment

- Linux Kernel 2.4.8 and Embedded Linux using MPC8260 H/W



6TALK Function Modules



2001

2002

2003

6TALK Hardware

Item	Spec	Etc.
CPU	MPC8260Z U200A	
RAM	SDRAM 64M	
Flash Memory	1M	For PPC Boot
Flash Memory	16M	For Embedded Linux
Serial Port	2ports	
10M Ethernet	4ports	
Fast Ethernet	2ports	



6TALK User Interfaces

```

명령 프롬프트 - telnet 129.254.165.164

Welcome to the 6TALK system !!


      66
      66
      66
      66666   TTTTTT   AA   LL   KK   K
      66   66   TT   A   A   LL   KK   K
      66   66   TT   AAAAAA LL   KKKK
      6    6    TT   A   A   LL   KK   K
      6666   TT   A   A   LLLLLL KK   K

COMMAND: ip          natpt
         read        erase
         show

         ls          help          clear
         ping        telnet
         end (ctrl^D)

Loading configuraion data was done.
6TALK>
    
```

Console mode (text)


[IP](#)
[NATPT](#)
[Flash Write](#)
[Flash Erase](#)
[Show Flash](#)
[Show config](#)
[Change ID/PW](#)
[Logout](#)

Interface
Route
Default Route
Tunnel
Down

Show Interface
Show Route
Show Tunnel
Show Down

IP Layer Commands	
interface	Set an IPv4 or IPv6 address to an interface. Remove IPv4 or IPv6 address from the interface.
route	Add a new routing entry to IPv4 routing table or IPv6 routing table Delete the routing entry from IPv4 or IPv6 routing table.
default route	Add a new default routing entry to IPv4 routing table or IPv6 default routing table Delete the default routing entry from IPv4 or IPv6 routing table.
tunnel	Set a configured tunnel Delete the tunnel configuration.
down	Disable an interface. Enable an interface.
show interface	Show the information about interfaces.
show route	Show the information of IPv4 or IPv6 routing table.
show tunnel	Show the tunnel configuration.

Copyright 2001 by ETRI 6TALK

www.6talk.net

support@6talk.net

Web CGI mode



6TALK Implementation(1)

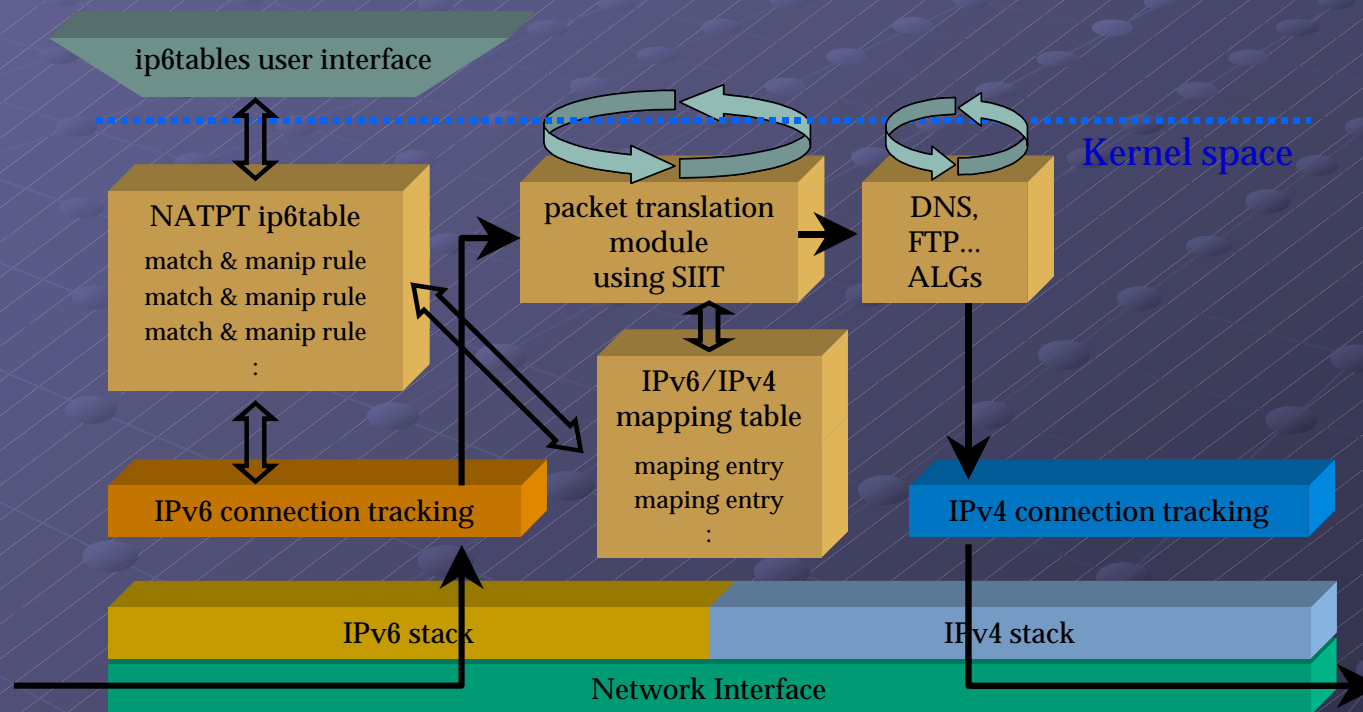
● NAT-PT Implementation

- Linux 2.4.x Netfilter framework Base
- Kernel netfilter framework
 - ip(6)table structure
 - List of packet matching & manipulation rule for each netfilter application (NAT, NAT-PT, packet mangle, packet filter, etc.)
 - connection tracking structure
 - Maintain status of each connection.
- User interface
 - ip(6)tables commands



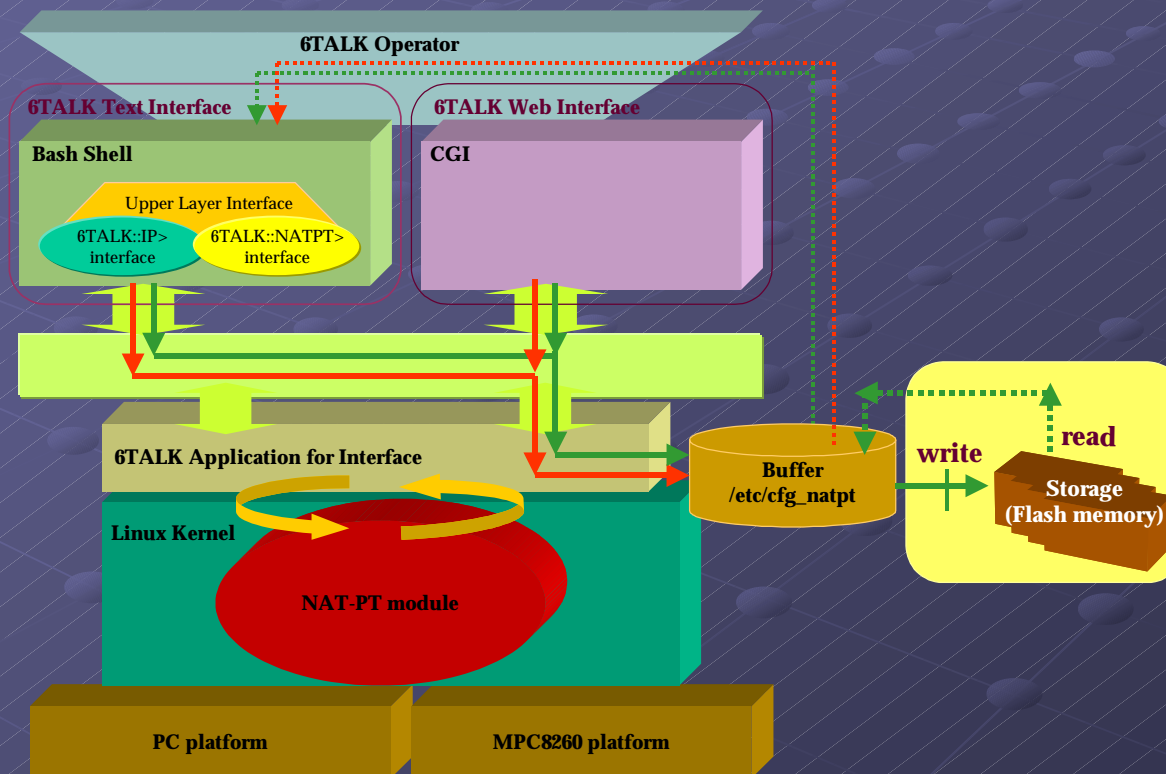
6TALK Implementation(2)

- Structure of NAT-PT in Netfilter



6TALK Implementation(3)

- Structure of NAT-PT user interface



6TALK Implementation Issues (1/2)

● Fragmentation issues

- MTU of IPv4 network and IPv6 network might be different. So we can't translate each fragmented packet directly
→ defragment first and fragment again

● ALG (Application Level Gateway)

- If upper layer protocol's payload includes ip address, we should translate that.
- 6TALK supports DNS-ALG



6TALK Implementation Issues (2/2)

- ICMP error message handling

- Also translate ip header embedded in icmp error message

- User interface extension

- Add shared library for 6TALK



What is 6TALK?

Why 6TALK?

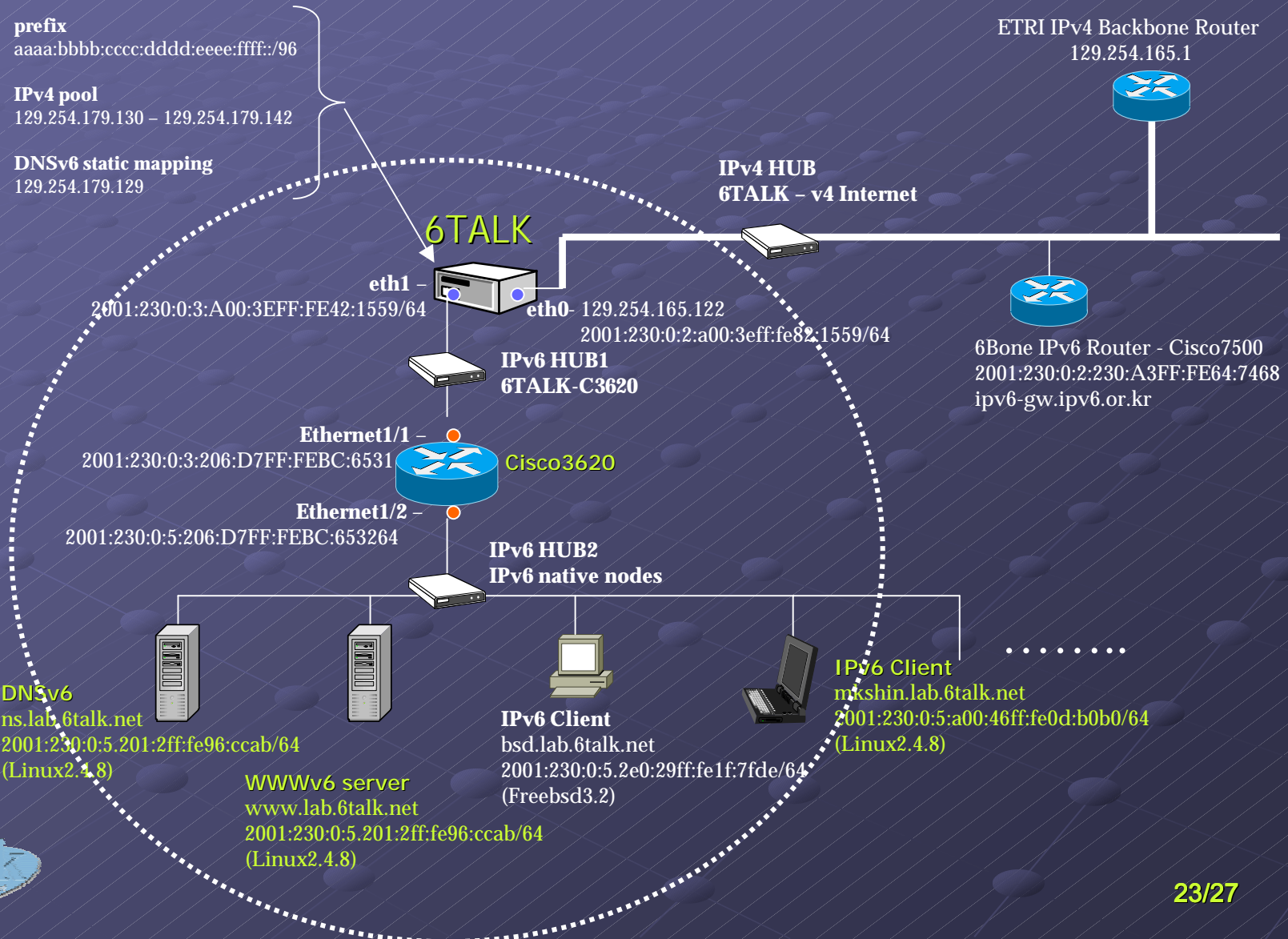
6TALK Scenarios & Solution

6TALK Implementations

➔ 6TALK Test Environments

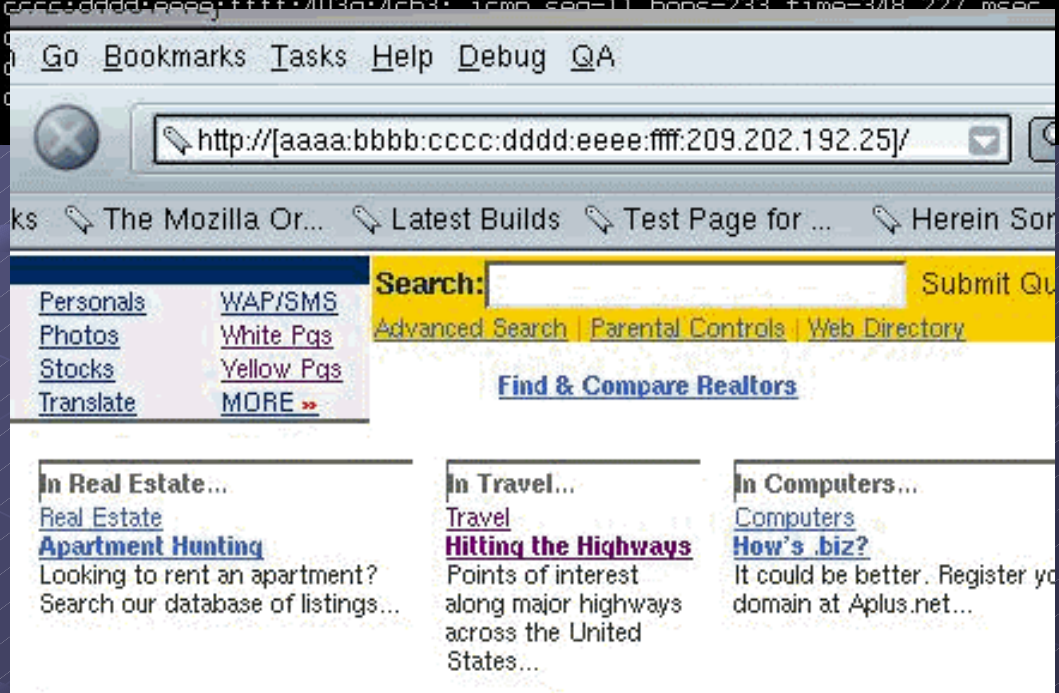
Summary & Future Work

6TALK Test Environments



Test

```
[root@Runic ~]# ping6 aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3 -s 10000
PING aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3(aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3) from 2001:230:
0:4:a00:46ff:fe06:7744 : 10000 data bytes
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=2 hops=233 time=297.156 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=3 hops=233 time=1.313 sec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=5 hops=233 time=1.294 sec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=7 hops=233 time=324.423 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=8 hops=233 time=321.038 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=9 hops=233 time=297.658 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=10 hops=233 time=301.293 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=11 hops=233 time=348.227 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=12 hops=233 time=348.227 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=13 hops=233 time=348.227 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=14 hops=233 time=348.227 msec
10008 bytes from aaaa:bbbb:cccc:dddd:eeee:ffff:403a:4cb3: icmp_seq=15 hops=233 time=348.227 msec
```



What is 6TALK?

Why 6TALK?

6TALK Scenarios & Solution

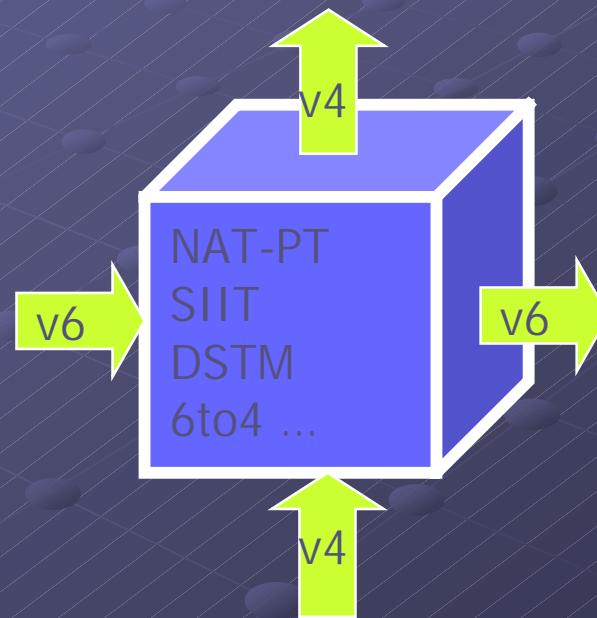
6TALK Implementations

6TALK Test Environments

➔ Summary & Future Work

Summary

- 6TALK : Smart Transition Toolbox supporting Inter-working for IPv6-IPv4 and/or IPv6-IPv4-IPv6 seamlessly



Summary & Future Work

- General Platform for IPv6 Transition Mechanism Implementation
 - Open architecture
 - Linux base
 - Sets of IPv6 transition mechanisms
 - IPv4/IPv6 Translation : NAT-PT / SIIT (Now)
 - Basic Tunneling : 6to4 and configured tunnel (Now)
 - DSTM and DSTM extensions (This year)
 - Combinations of transition tools (2003)
 - DSTM+SIIT, 6to4+ISATAP, etc.
- Optimization for performance
- Conformance test (i.e. by using TAHI)



