IP - IPv6

History of IPv4

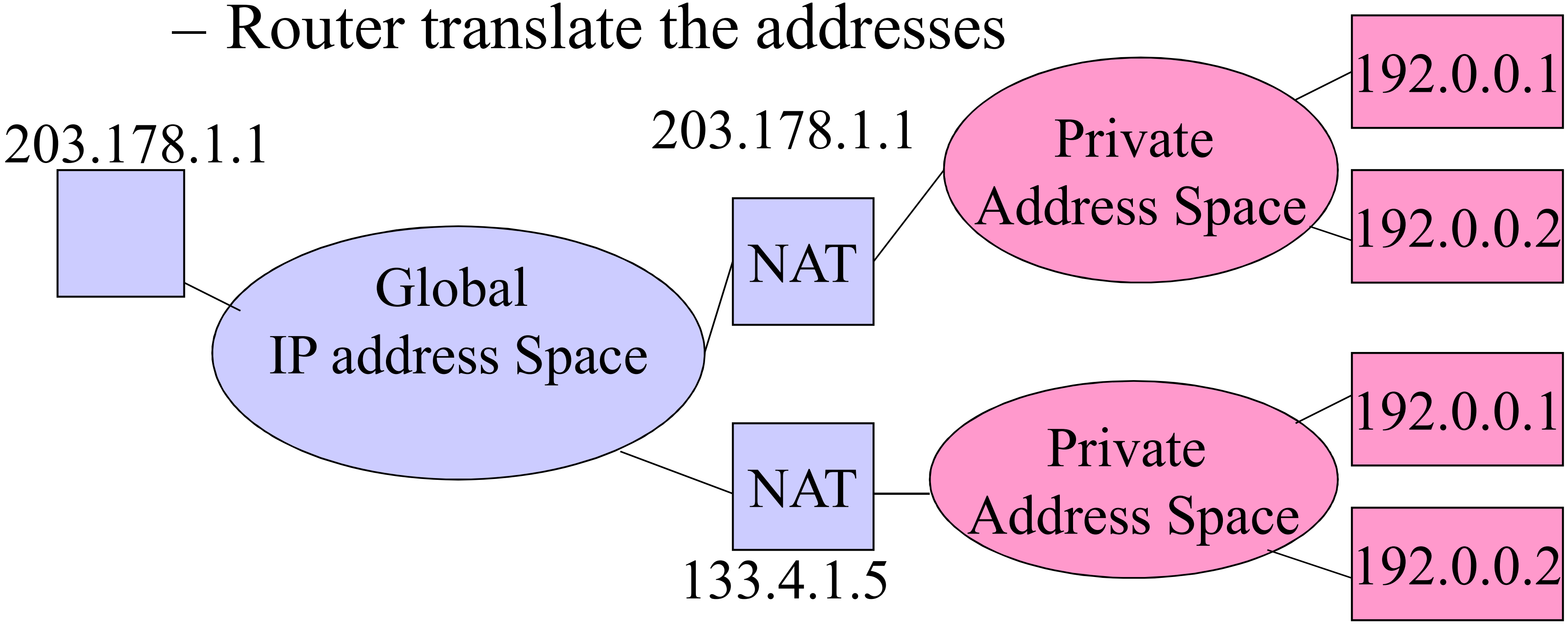
- allocate 1 class B per 1 organization(1980’s)
- Projected exhaustion of class B address(1990’s)
  - allocate multiple class C address per organizations
- Rapid increase of routing tables
  - reducing them by using CIDR(address+mask)
- Projected exhaustion of whole IPv4 addresses (1990’s)
  - 4 bytes = 4,300,000,000

Problems of IPv4

- Problems
  - rapid increase of Routing tables
    - 60,000 entries in the Internet now
  - exhausted IPv4 addresses
    - will exhaust by 2008±

Can NAT solve the problems ?

- NAT : Network Address Translation
  - Assign private addresses to the internal systems
  - Router translate the addresses



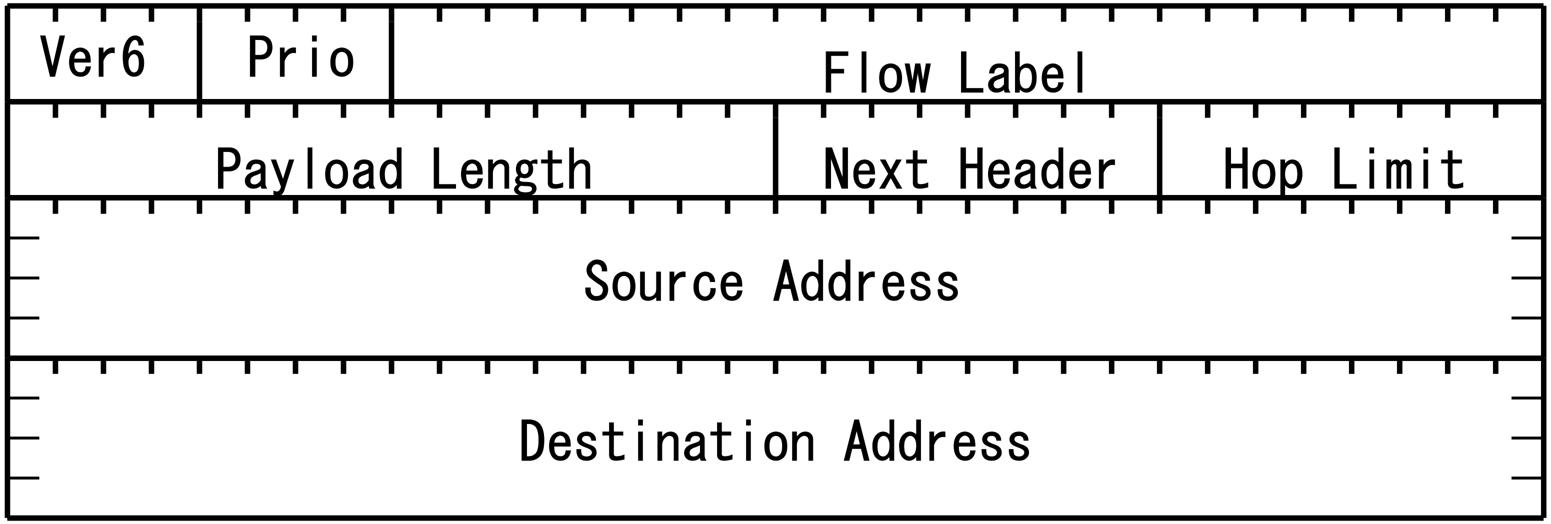
One solution – NAT

- NAT(Network Address Translator)
  - Popular on Dial-ups, SOHO and VPN networks
  - will save IPv4 address
  - lost of the end-to-end model
  - Asymmetric identifier/communication model

Why not NAT ?

- NAT breaks “end-to-end communication”
  - Routers monitors the communication
  - Routers changes the data
- NAT breaks “Bi-directional communication”
  - Hosts with global address can not initiate the communication to the hosts with private address.

IPv6 Header format



Ver6

Prio

Flow Label

Payload Length

Next Header

Hop Limit

Source Address

Destination Address

IPv4 Header Format

<table>
<tr><td colspan="4">Ver4</td><td colspan="4">*IHL*</td><td colspan="8">*Type of service*</td><td colspan="16">Total Length</td></tr>
<tr><td colspan="16">*Identification*</td><td colspan="3">*Flgs*</td><td colspan="13">*Fragmentation Offset*</td></tr>
<tr><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">*Header Checksum*</td></tr>
<tr><td colspan="32">Source Address</td></tr>
<tr><td colspan="32">Destination Address</td></tr>
<tr><td colspan="32">*Options*</td></tr>
</table>

Italics have been removed in IPv6

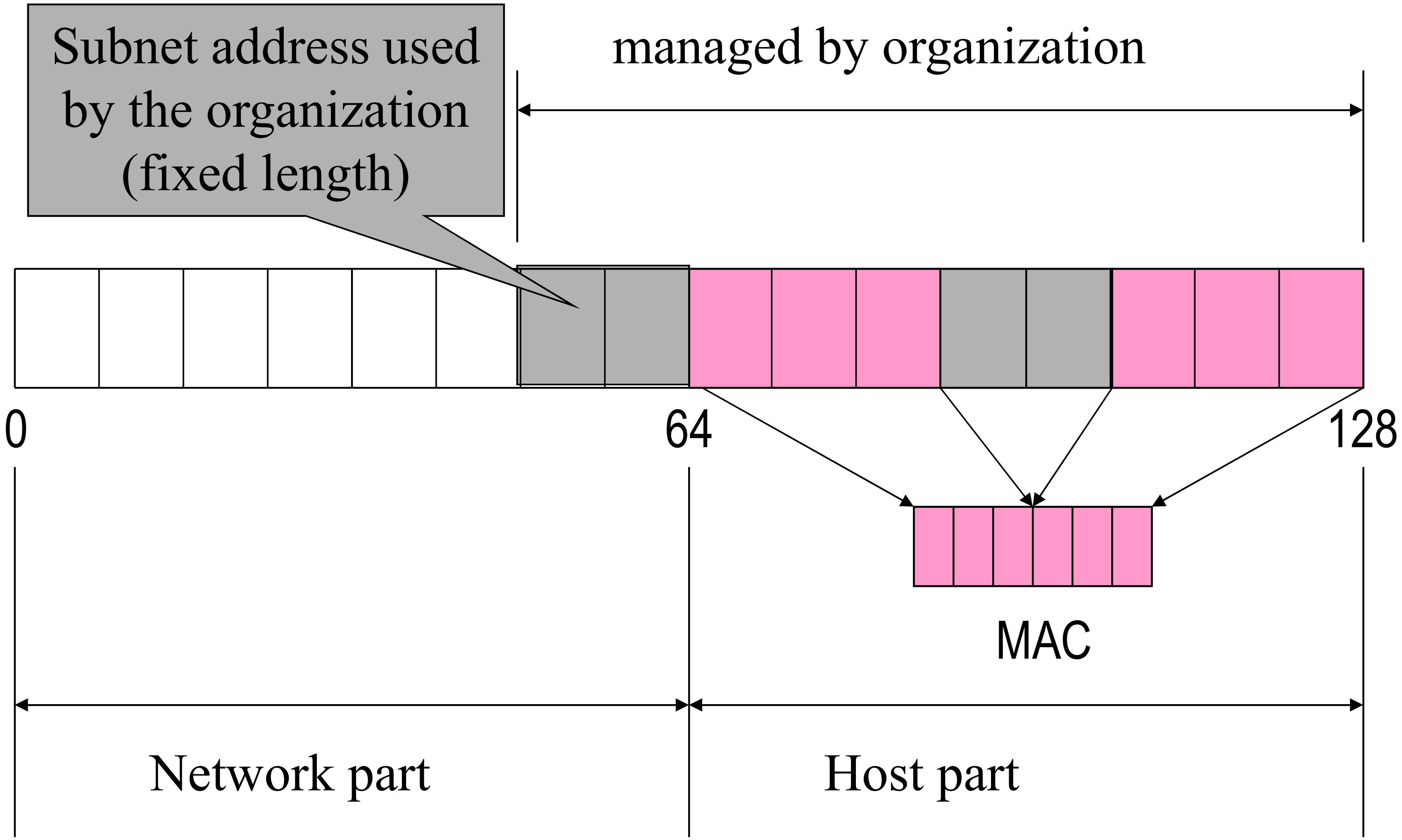
What’s good about IPv6

- Larger Address space
  - 128 bit: 3.4 × 10^38
- Re-design to solve the current problems such as;
  - Routing
  - Security
  - Auto-configuration
  - Plug & Play

Is IPv6 really good ?

- IPv6 can not easily solve (same as IPv4);
  - Security
  - Multicast
  - Mobile
  - QoS

IPv6 Address



IPv6 Address notation

- Basic rules
  - “:” in every 2 bytes
  - Hex digits
- shorthand
  - heading 0s in each block can be omitted
  - “0000” → “0”
  - “0:all zeros in between :0” can be “::”

IPv6 address notation – example

- 3ffe:0501:0008:0000:0260:97ff:fe40:efab
  - 3ffe:501:8:0:260:97ff:fe40:efab
  - 3ffe:501:8::260:97ff:fe40:feab

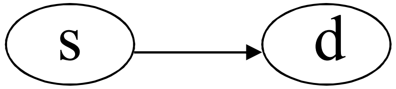
- ff02:0000:0000:0000:0000:0000:0000:0001
  - ff02:0:0:0:0:0:0:1
  - ff02::1

Types of addresses

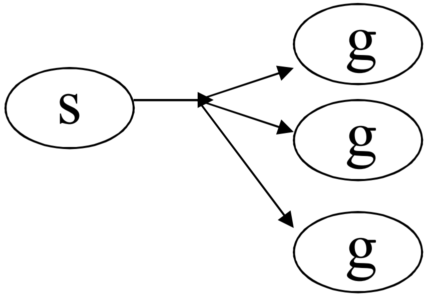
1st 4bits of the adddress	Use
0 (0000)	Special address
1 (0001)	Special address
2 (0010)	**Aggregatable global unicast address**
3 (0011)	**Aggregatable global unicast address**
4 (0100)	Unassigned
5 (0101)	Unassigned
6 (0110)	Unassigned
7 (0111)	Unassigned
8 (1000)	Unassigned
9 (1001)	Unassigned
a (1010)	Unassigned
b (1011)	Unassigned
c (1100)	Unassigned
d (1101)	Unassigned
e (1110)	**link-local, site-local, multicast**
f (1111)	**link-local, site-local,multicast**

IPv6 addresses:uni/multi/any-cast

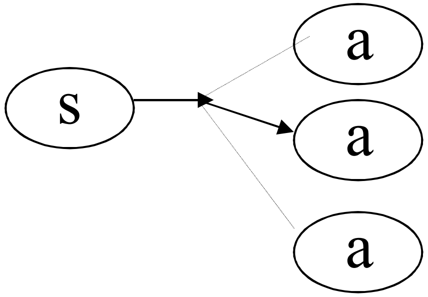
- **unicast**
  - communicate specified 1 computer
- **multicast**
  - communicate group of computers
- **anycast**
  - send group address that can receive multiple computers,but receive 1 computer



unicast



multicast

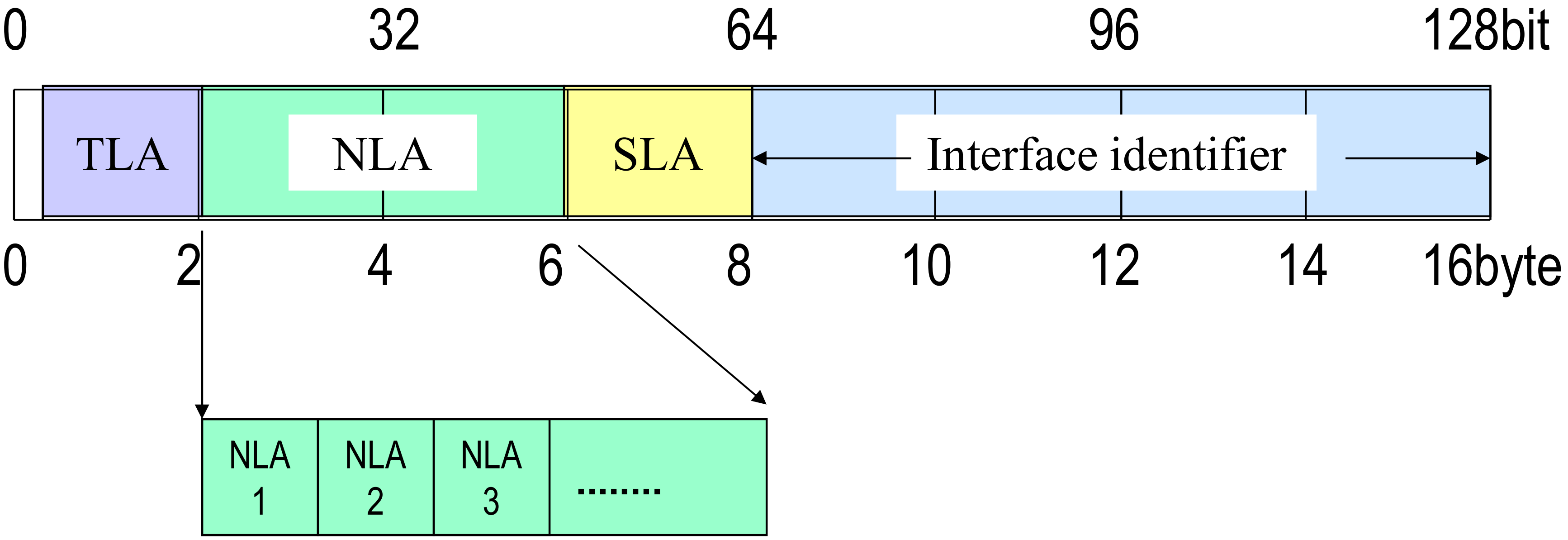


anycast

Special address

- ::
  - Used by plug & play
- ::1
  - loopback (same as 127.0.0.1 in many IPv4 implementations)
  - ping ::1 should answer myself.

Aggregatable global unicast address

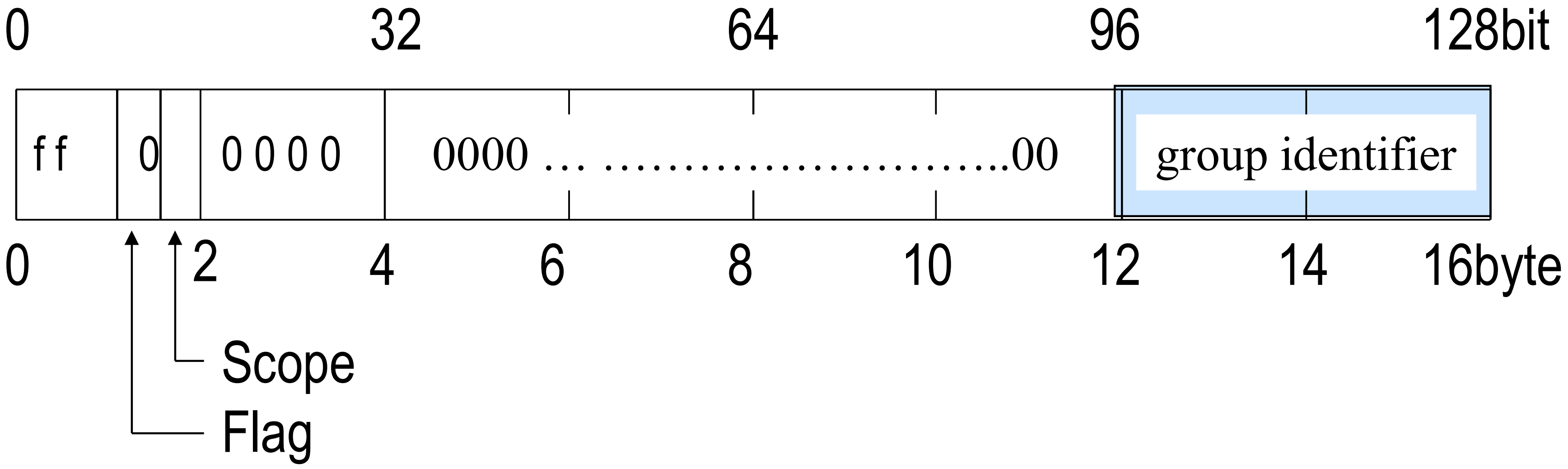


TLA – Top Level Aggregator … assigned for 8K major providers(13+3bits)

NLA – Next Level Aggregator … assigned for smaller providers

SLA – Site Level Aggregator … subnet numbers within organizations (16bits)

Multicast Address



- **Scope**

  1: node local scope

  2: link local scope

  5: site local scope

  8: organization local scope

  e: global scpe

- **Group ID**

  0000:0000 ～ feff:ffff

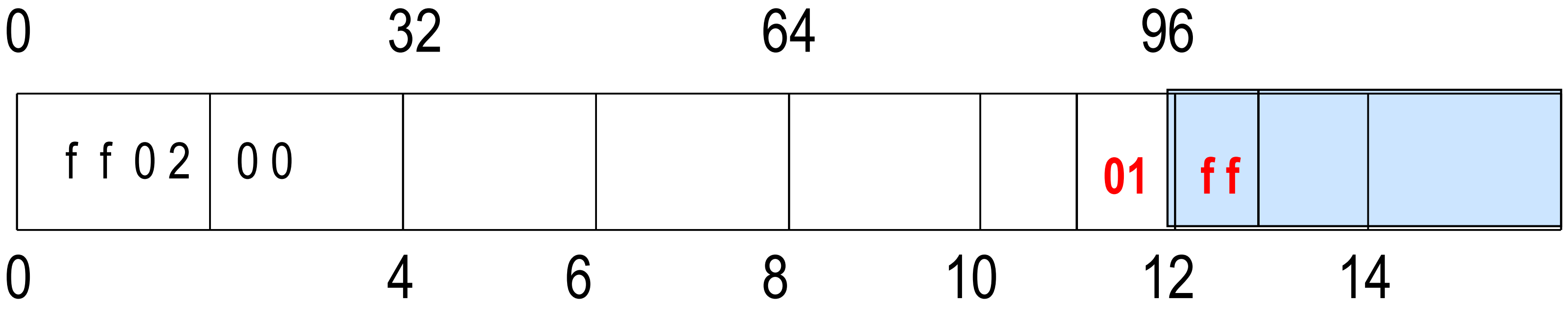
  0000:0001 … all nodes

  0000:0002 … all routers

Multicast Address - example

- ff01::2
  - node local & all routers
- ff02::1
  - link local & all nodes
- ff02::2
  - link local & all routers

Solicitation Multicast Address



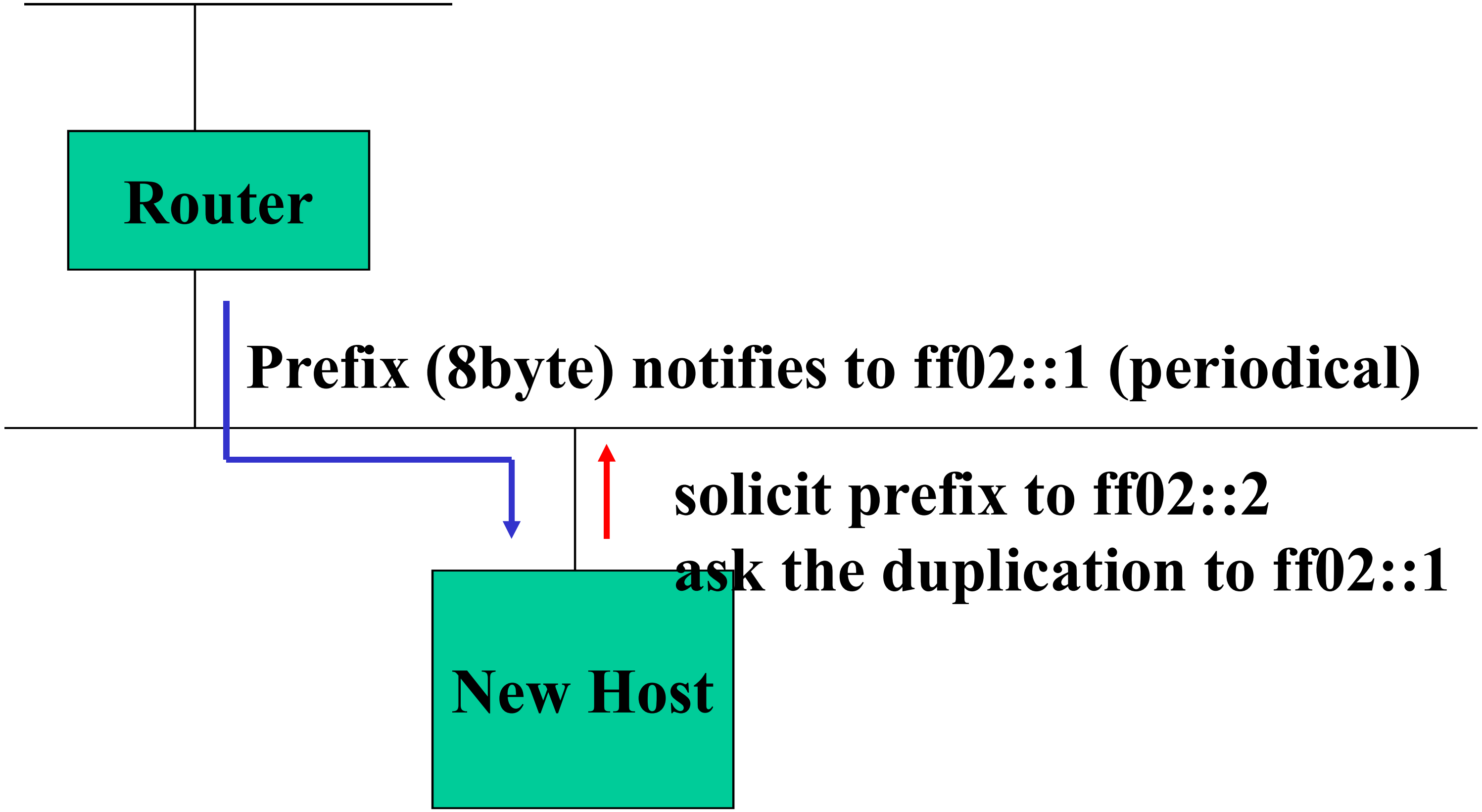
- **Used for address resolution (ARP)**
- **ff02::1:ff40:efab =**

  Solicitation multicast address for fe80::260:97ff:fe40:efab

Plug & Play

- generate a IPv6 address automatically from global network address and ether MAC address
- sense duplicated address
- detect default route to the appropriate router
- redirect to the router if host’s connection was lost

Plug & Play



Router

Prefix (8byte) notifies to ff02::1 (periodical)

solicit prefix to ff02::2

ask the duplication to ff02::1

New Host

Ethernet Address(6bytes) = 00:60:97:40:ef:ab

Interface-id(8bytes EUI) = 260:97ff:fe40:efab

IPv6 Address = Prefix : Interface-id

IPv6 ready application

- handle “:” in address correctly
- handle IPv4 and IPv6 addresses

```
% ftp ftp.kame.net
% ftp 3ffe:501:4819:2000:5254:ff:fedc:50d2
```

NOTE: Use of ‘:’ may cause confusions!

(http:://xxx, %scp xx:foo.txt)

DNS for IPv6

- ‘A’ record of DNS(IPv4)

www.kame.net A 203.178.141.212

- ‘Qaud A’ record of DNS(IPv6)

www.kame.net AAAA 3ffe:501:4819:2000:5254:ff:fedc:50d2

IPv4 programming example

```
int i, s;
struct hostent *hp;
struct servent *sp;
struct sockaddr_in sin;

s = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP);
hp = gethostbyname("www.kame.net");
sp = getservbyname("http", "tcp");
for (i = 0; hp->h_addr_list[i]; i++) {
  memset(&sin, 0, sizeof(sin));
  sin.sin_family = AF_INET;
  sin.sin_len = sizeof(sin);
  sin.sin_port = htons(sp->s_port);
  memcpy(&sin.sin_addr, hp->h_addr_list[i], hp->h_length);
  if (connect(s, &sin, sizeof(sin)) < 0)
    continue;
  break;
}
```

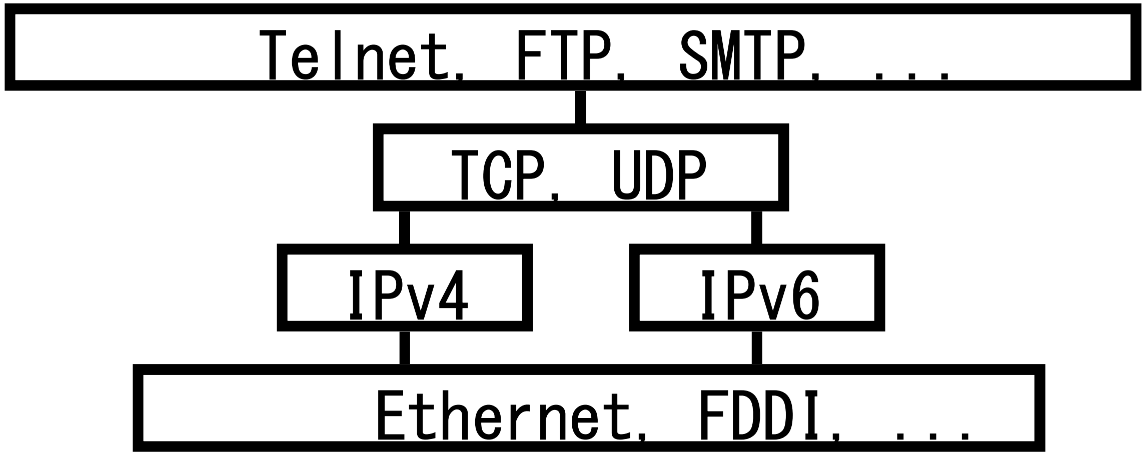
IPv6 programming example

```
int s;
struct addrinfo hints, *res, *res0;

memset(&hints, 0, sizeof(hints));
hints.ai_family = PF_UNSPEC;
hints.ai_socktype = SOCK_STREAM;
getaddrinfo("www.kame.net", "http", &hints, &res0);
for (res = res0; res; res = res->ai_next) {
   s = socket(res->ai_family, res->ai_socktype, res->ai_protocol);
   if (connect(s, res->ai_addr, res->ai_addrlen) < 0) {
      close(s);
      continue;
   }
   break;
}
freeaddrinfo(res0);
```

Technical term

- IPv6/IPv4 (dual) node
  - Node with both IPv4 and IPv6
  - Handle tunneling



- IPv6/IPv4 header translation router
  - IPv6/IPv4 nods which translates the header format

Transition Plan

- Current status
  - Only IPv4
- Phase I
  - IPv4/v6 Dual node
  - IPv4 address in IPv6 address format
  - IPv6 tunneling on the IPv4 Network
- Phase II
  - Combination of IPv6 infrastructure and IPv4 infrastructure
  - Translate between IPv4/v6 each other(optional)
  - More IPv6 nodes

Migration to IPv6

- dual stack
- tunnel
- translator

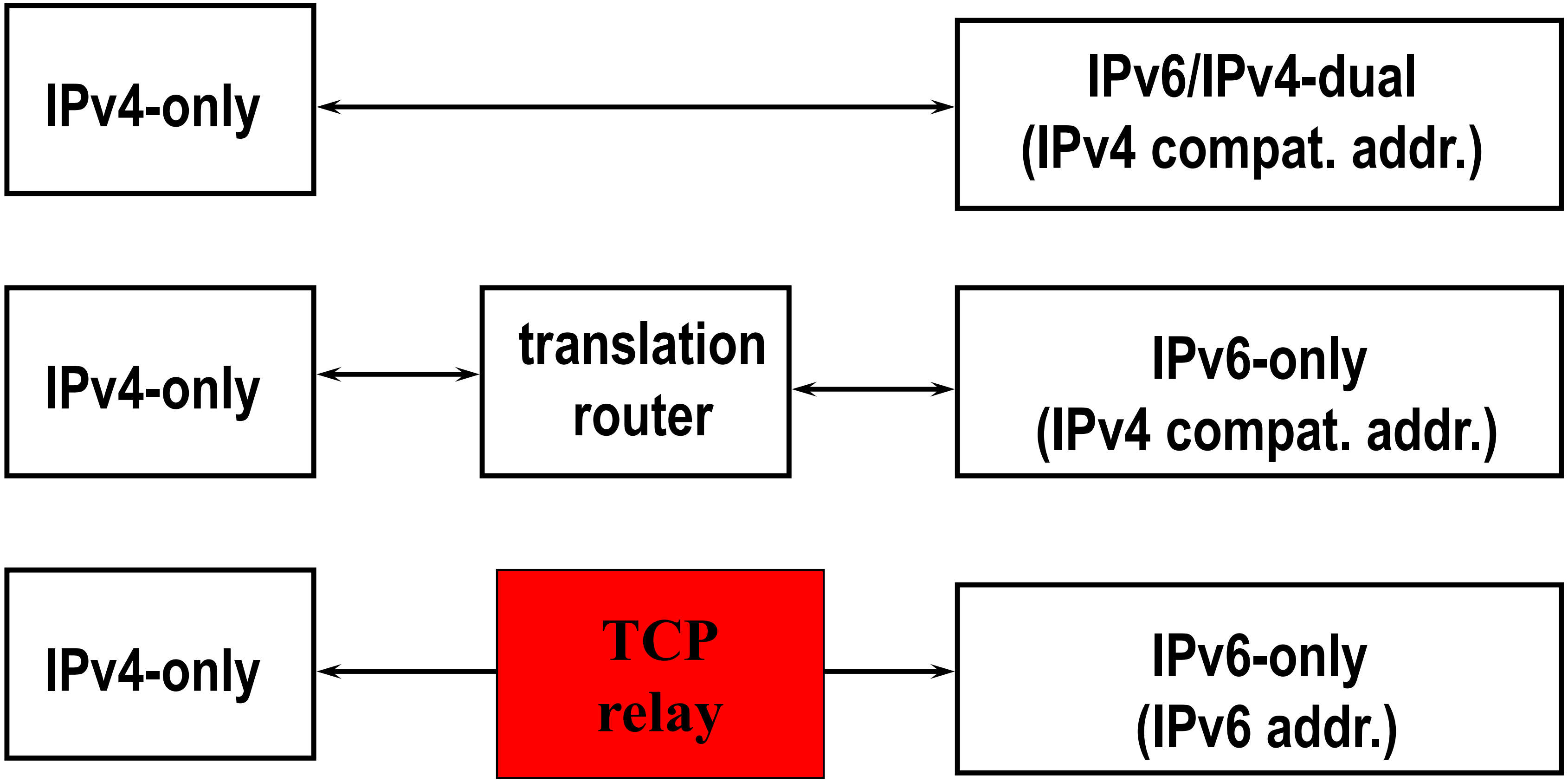
dual stack

- Dual stack host can speak both IPv4 and IPv6
  - communicate with IPv4 host by IPv4
  - communicate with IPv6 host by IPv6
- Dual stack host look up DNS entry by IPv4

translator

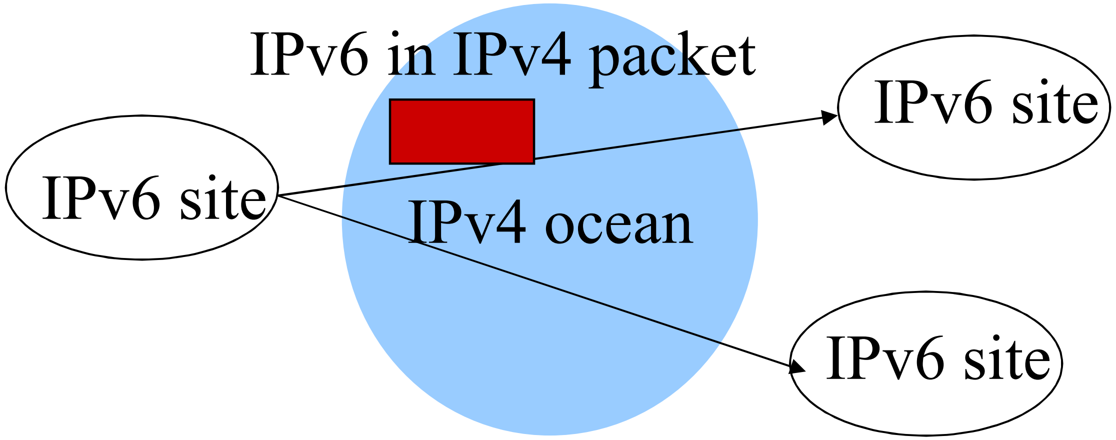
- Translate IPv6 only host to IPv4 only host(vice versa)
  - protocol translation
  - mapping address

Communication between IPv4 nodes and IPv6 nodes

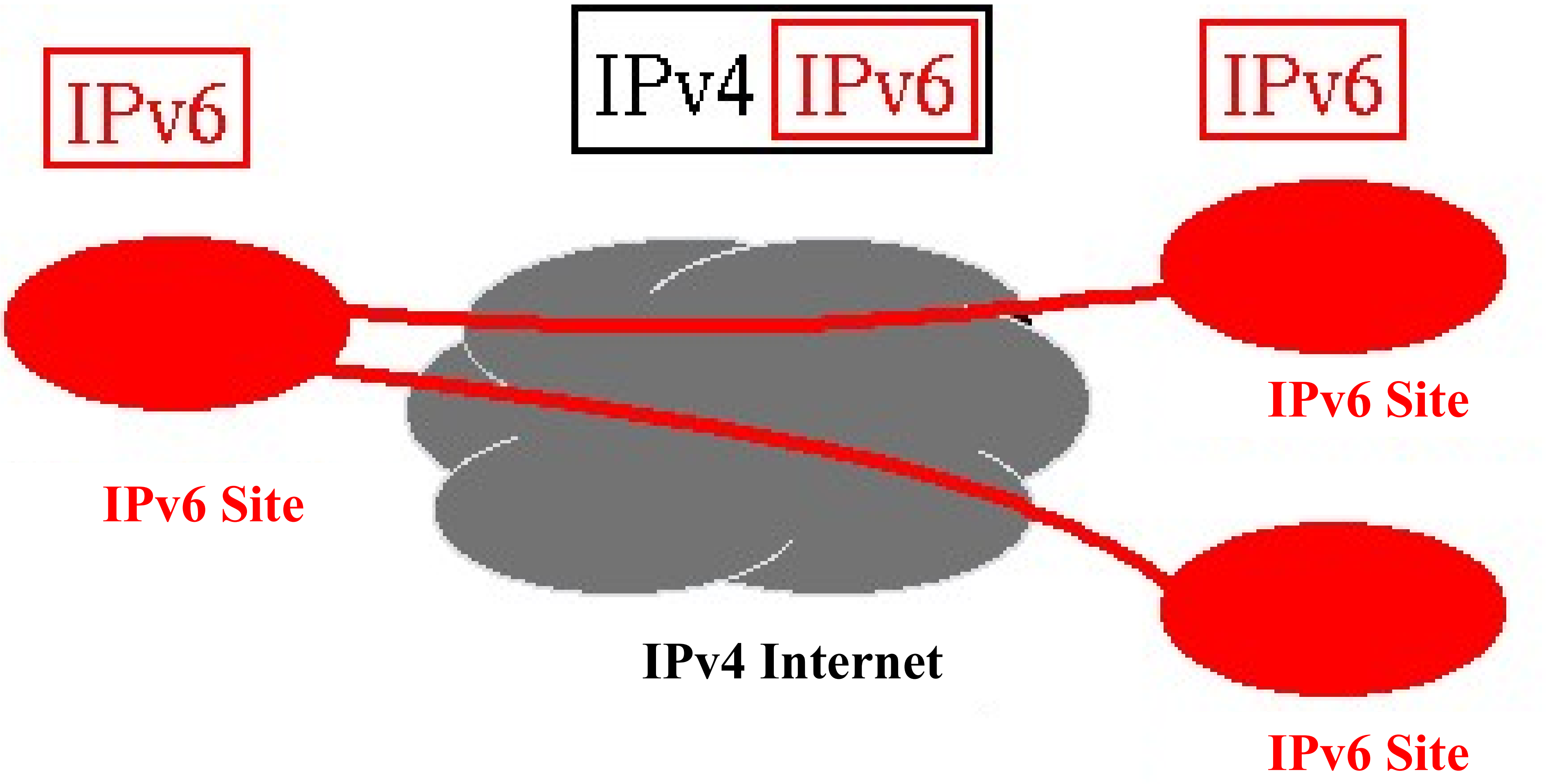


IPv6 in IPv4 tunnel

- IPv6 site is island surrounded IPv4 ocean
- connect IPv6 island each other
  - encapsulate IPv6 packet in IPv4 packet
  - threat as IPv4 as data link layer

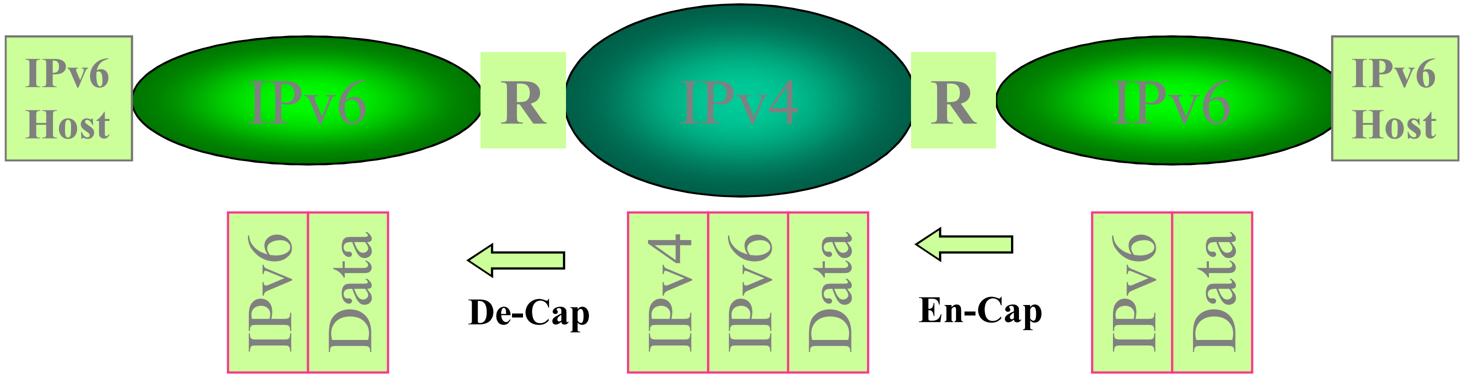


IPv6 in IPv4 Tunnel

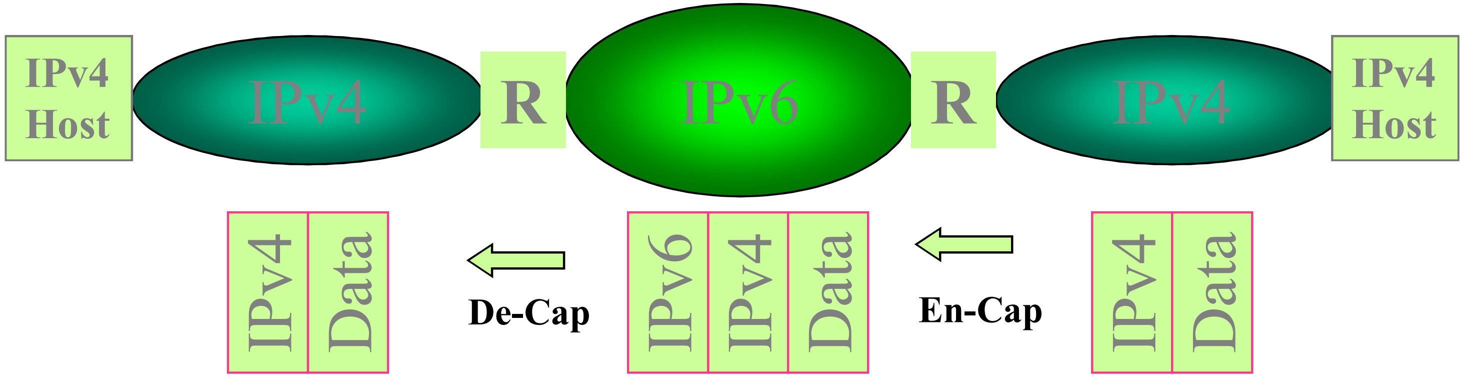


Tunneling

- IPv6 packets goes through IPv4 network



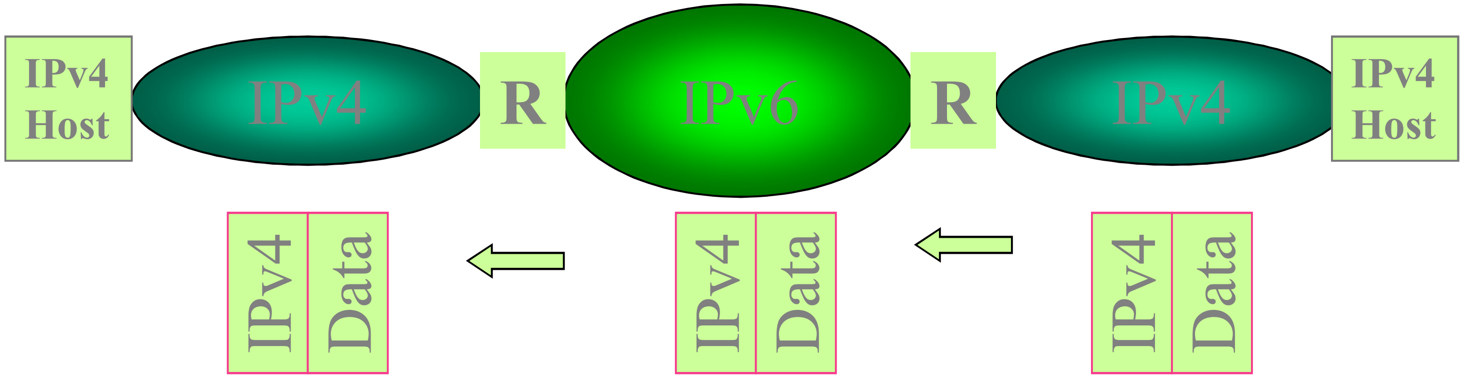
- IPv4 packets goes through IPv6 network



IPv4 communications through IPv6 Network

- IPv6 Backbone (IPv6/IPv4 Dual-Stack) convey IPv4 packets

Routing: **<only IPv4>** **<IPv4/IPv6>** **<only IPv4>**



IPv4 Host

IPv4

R

IPv6

R

IPv4

IPv4 Host

IPv4 | Data

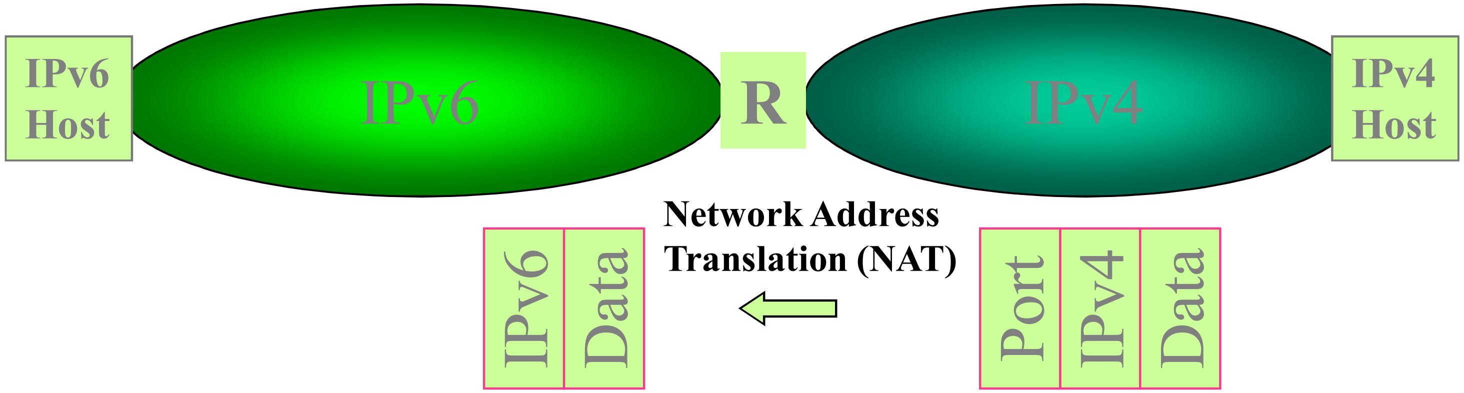
IPv4 | Data

IPv4 | Data

1. Native IPv4 Packet
2. embedded IPv4

Address Translation

- IPv6 packets goes through IPv4 network



- IPv6 hosts IP address resolve (DNS)

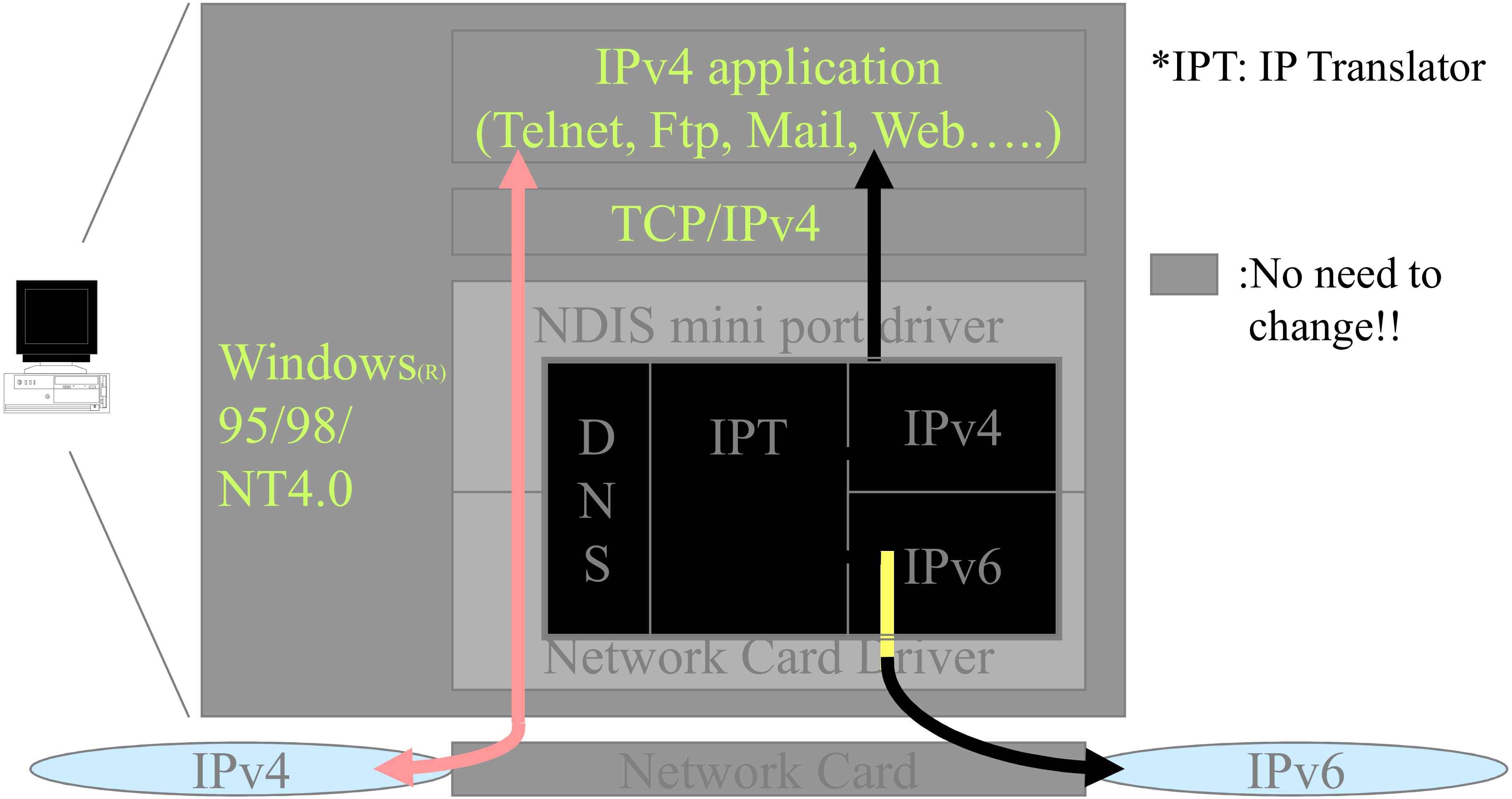
  (a) In IPv4 network

  v6-1.rdc.toshiba,co.jp => {IPv4, port}

  (b) In IPv6 network

  v6-1.rdc.toshiba.co.jp => IPv6

Hitachi V6 stack for Win98



*IPT: IP Translator

▨ :No need to change!!

KAME Project

KAME Project

- A single effort
  - 8 core members from 7 Japanese companies
  - Fujitsu, Hitachi, IIJ, NEC, Toshiba, YDC, Yokogawa
  - Two-years joint project
  - April 1998 - March 2000
  - The core members work for IPv6 three days a week
- KAME
  - A short word of KArigoME, where our office locates
  - Turtles

Purpose

- Reference code
  - IPv6, IPsec, and advanced networking
    - FreeBSD, NetBSD, and BSD/OS
  - Provided "AS IS" like BSD
    - Free and no warranty
  - Commercial use is OK
- Release schedule
  - SNAP - every Monday
  - STABLE - every other month
  - RELEASE - quality assurance by TAHI Project
    - http://www.tahi.org
    - Specification conformance
    - Interoperability

The current status

- The basic spec has been implemented
  - IPv6, DNS, BGP4+, RIPng, translator, laptop computer support
  - IPsec, IKE
  - IPComp, IPv4 NAT, ECN, ATM, ALTQ
  - Many applications
    - SMTP, POP, HTTP, FTP, TELNET, SSH, X11,...
- Used in the worldwide 6bone
  - More than 20 countries, 200 people
  - Both as routers and hosts
- Will be merged into NetBSD
  - The other BSD variants are waiting for "unified stack"
    - KAME, NRL, INRIA

Future plans

- Obtaining more experiments
  - IKE interoperability
  - PIM multicast routing
- Another new features
  - IPv6 router renumbering
  - New DNS (A6 record)
- Maintenance
  - Catching up to updates of BSD variants
- The KAME Y2K problem
  - Who will maintain KAME after April 2000?

History of the 6bone-JP

- June 9, 1996 creation of the 6bone-JP

  Tokyo NOC and Nara NOC are connected by a IPv6 dedicated line.

- July 16, 1996 joins 6bone

  Nara NOC and Cisco are linked by means of a tunnel.

- October 1997 **network address renumbering**

  The 6bone-JP changes over to the Aggregatable Global Unicast Address and is assigned a pTLA ID.

- January 1, 1998 creation of a registry system

  6bone-JP Registry System is created in order to begin the assignment of IPv6 addresses to the public at large.

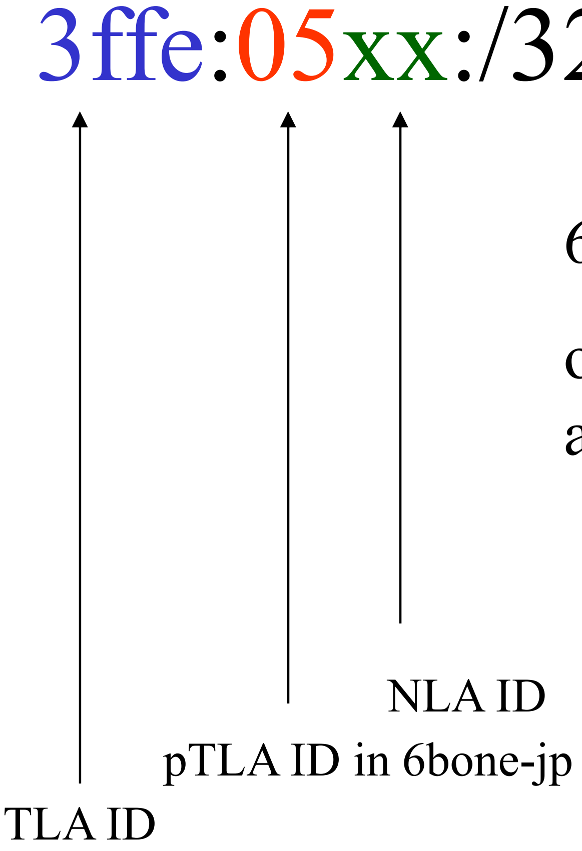
Current Status

- 6bone-JP is responsible for the assignment of IPv6 addresses and connectivity within the Japan, Asia region

- Present address assignment situation (as of January 1999)

– NLA	10 sites
– SLA	50 sites

Address assignment policy of the 6bone-JP(1)



3ffe:05xx:/32

NLA ID

pTLA ID in 6bone-jp

TLA ID

6bone-JP allocates NLA IDs for organizations throughout Japan and Asia

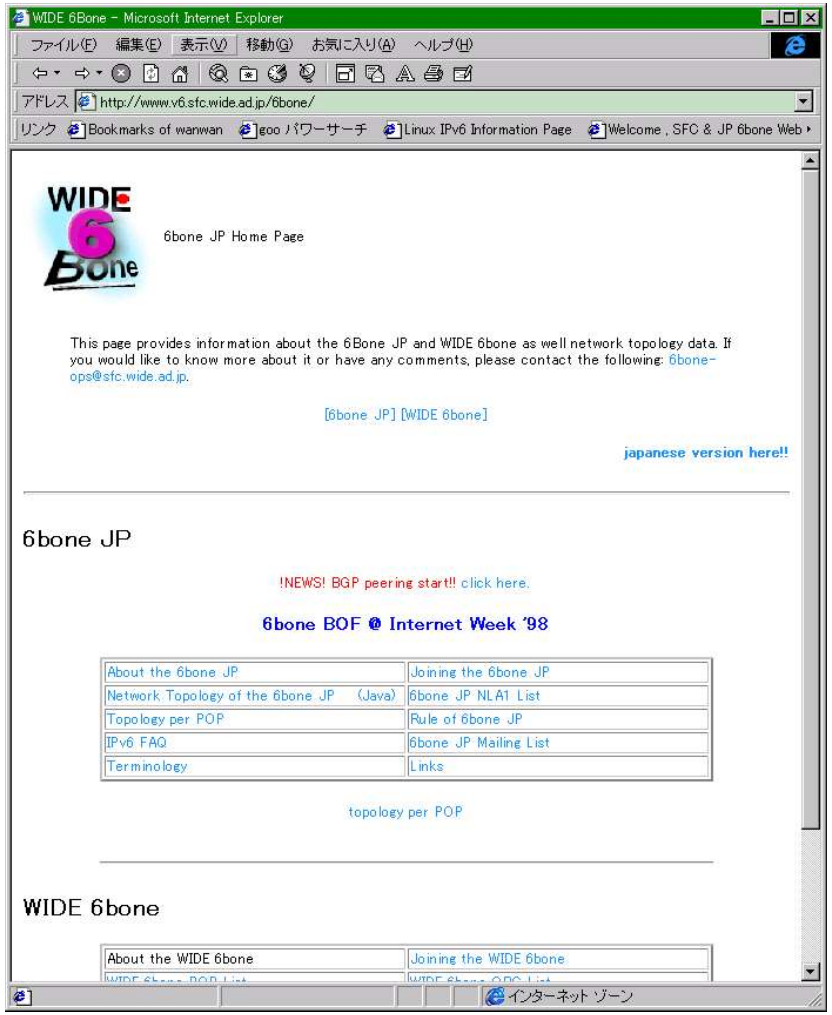
Address assignment policy of the 6bone-JP(2)

- Organizations with NLA ID's assign addresses in turn to other organizations according to their own address assignment policies.
- Organizations with NLA ID's accept address assignment requests through the Web.
    - WIDE Project http://www.v6.sfc.wide.ad.jp/6bone/
    - NTT Software Lab. http://www.nttv6.net/
    - IIJ Lab. http://playground.iijlab.net/6bone/6bone-policy.html
    - IMASY http://www.imasy.or.jp/~ichiro/v6/6bone.html

6bone-JP Registry System

- Web based IPv6 registry system from 1997
- easy to update and view
- uses PGP public-key for authentication of maintainer
- can apply for IPv6 address via the Web

http://v6.sfc.wide.ad.jp/6bone/


WIDE 6Bone - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
アドレス http://www.v6.sfc.wide.ad.jp/6bone/
リンク Bookmarks of wanwan goo パワーサーチ Linux IPv6 Information Page Welcome , SFC & JP 6bone Web
WIDE 6 Bone
インターネット ゾーン

6bone JP Home Page

This page provides information about the 6Bone JP and WIDE 6bone as well network topology data. If you would like to know more about it or have any comments, please contact the following: 6bone-ops@sfc.wide.ad.jp.

[6bone JP] [WIDE 6bone]

japanese version here!!

6bone JP

!NEWS! BGP peering start!! click here.

6bone BOF @ Internet Week '98

About the 6bone JP	Joining the 6bone JP
Network Topology of the 6bone JP (Java)	6bone JP NLA1 List
Topology per POP	Rule of 6bone JP
IPv6 FAQ	6bone JP Mailing List
Terminology	Links

topology per POP

WIDE 6bone

About the WIDE 6bone	Joining the WIDE 6bone

Management of the 6bone-JP

- IPv6 NOC

  There are 10 NOCs in existence which accept IPv6 connections, either through IPv6 dedicated lines or tunneling. The WIDE Project is mainly responsible for the maintenance of these NOCs. Some ISPs participate in this maintenance as well.

- A backbone created on ATM lines

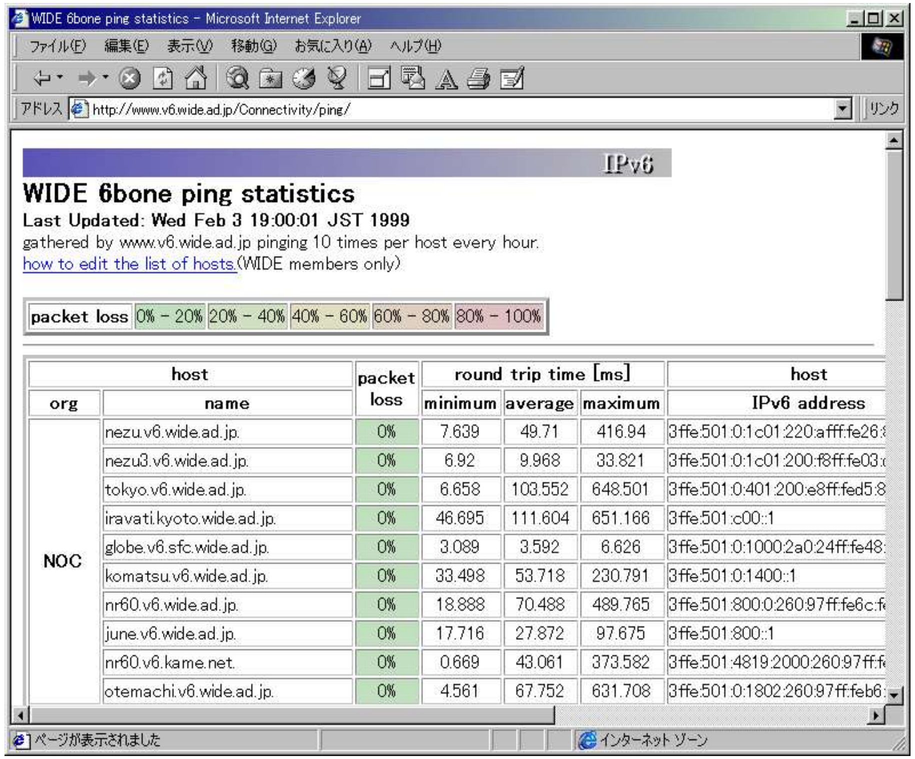
  IPv6 dedicated ATM lines are used abundantly for the creation and maintenance of the 6bone-JP backbone.

- Routing protocols

  Routing is done either by RIPng or BGP4+.

6bone-JP Statistics

- Ping statistics for IPv6 nodes which are connected to the 6bone-JP are now available


WIDE 6bone ping statistics - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
アドレス http://www.v6.wide.ad.jp/Connectivity/ping/
リンク
IPv6
ページが表示されました
インターネット ゾーン

WIDE 6bone ping statistics

Last Updated: Wed Feb 3 19:00:01 JST 1999

gathered by www.v6.wide.ad.jp pinging 10 times per host every hour.

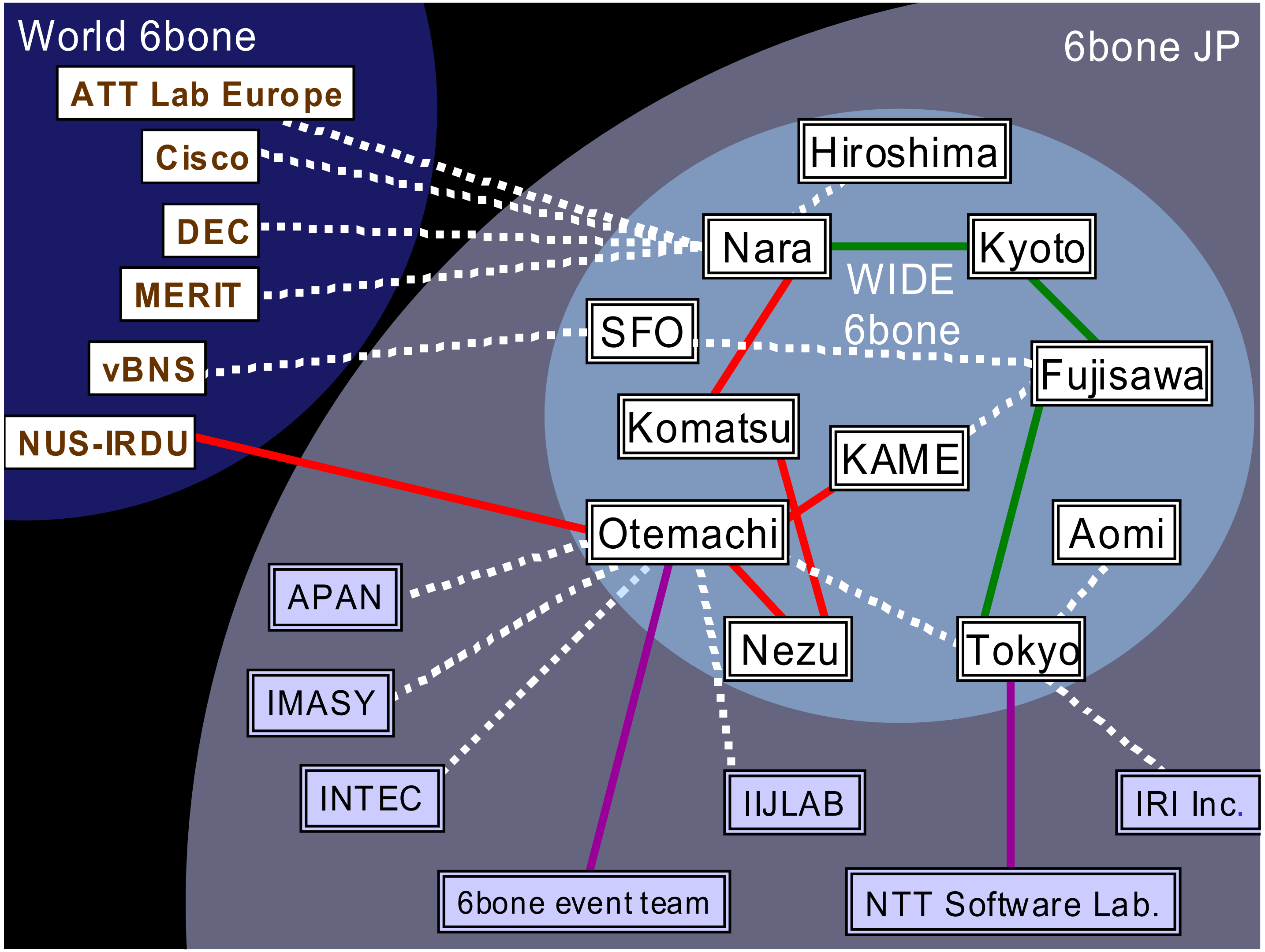
how to edit the list of hosts.(WIDE members only)

packet loss	0% - 20%	20% - 40%	40% - 60%	60% - 80%	80% - 100%

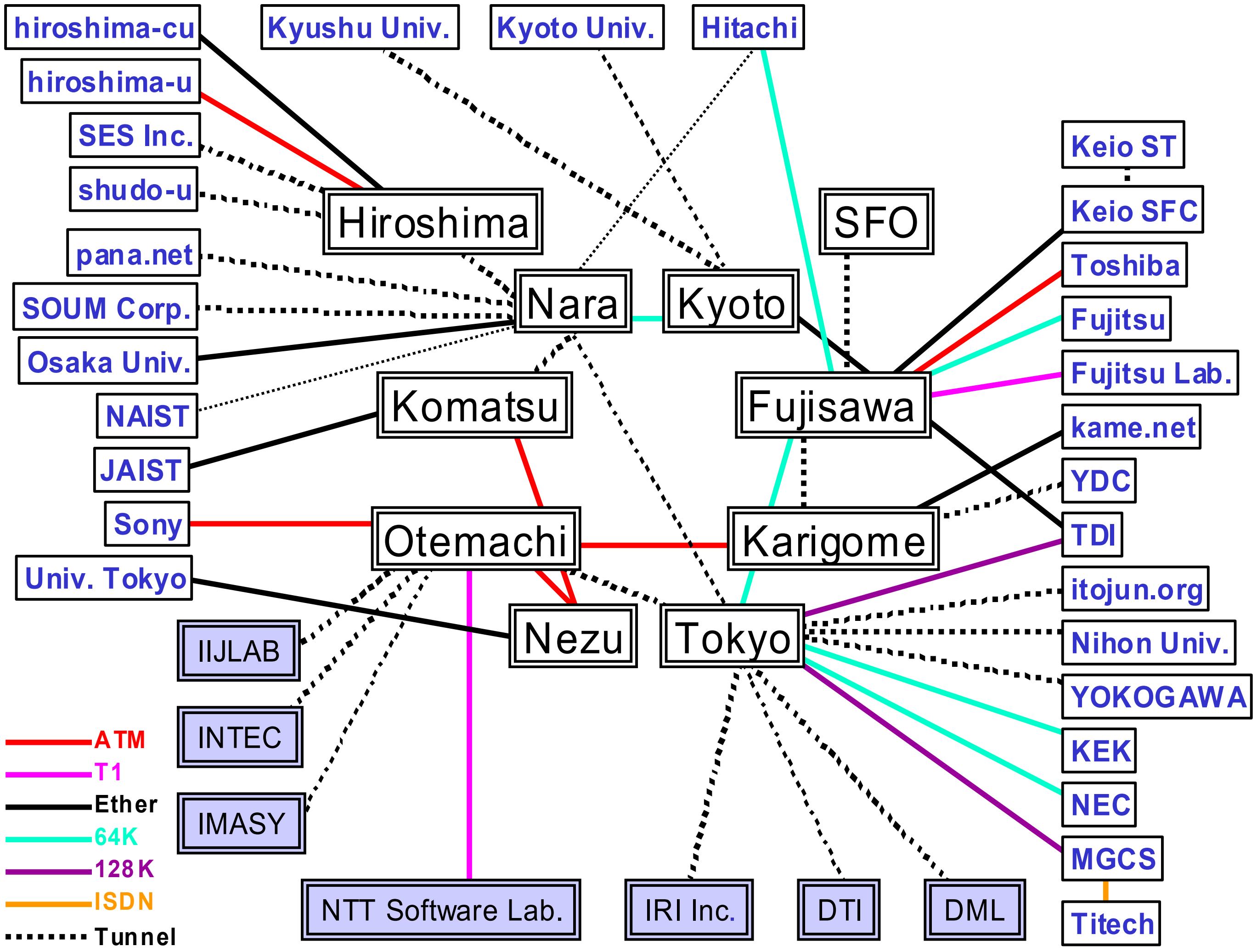
host		packet loss	round trip time [ms]			host
org	name		minimum	average	maximum	IPv6 address
NOC	nezu.v6.wide.ad.jp.	0%	7.639	49.71	416.94	3ffe:501:0:1c01:220:afff:fe26:
	nezu3.v6.wide.ad.jp.	0%	6.92	9.968	33.821	3ffe:501:0:1c01:200:f8ff:fe03:
	tokyo.v6.wide.ad.jp.	0%	6.658	103.552	648.501	3ffe:501:0:401:200:e8ff:fed5:8
	iravati.kyoto.wide.ad.jp.	0%	46.695	111.604	651.166	3ffe:501:c00::1
	globe.v6.sfc.wide.ad.jp.	0%	3.089	3.592	6.626	3ffe:501:0:1000:2a0:24ff:fe48:
	komatsu.v6.wide.ad.jp.	0%	33.498	53.718	230.791	3ffe:501:0:1400::1
	nr60.v6.wide.ad.jp.	0%	18.888	70.488	489.765	3ffe:501:800:0:260:97ff:fe6c:f
	june.v6.wide.ad.jp.	0%	17.716	27.872	97.675	3ffe:501:800::1
	nr60.v6.kame.net.	0%	0.669	43.061	373.582	3ffe:501:4819:2000:260:97ff:f
	otemachi.v6.wide.ad.jp.	0%	4.561	67.752	631.708	3ffe:501:0:1802:260:97ff:feb6:

http://www.v6.wide.ad.jp/Connectivity/ping/

World Topology of the 6bone-JP



Domestic Topology of the 6bone-JP



Brief history of v6 wg

- [1994/07] SIP was chosen as IPng
  - SIP → IPv6
- [1995/09] WIDE Project organized IPv6 working group
  - http://www.wide.ad.jp/
- [1995/12] The basic spec was published
  - RFC 1883
- [1995/12] The first interoperability test in WIDE Project
- [1996-1997] Many IPv6 products in WIDE Project
  - Parallel efforts appeared ineffective
  - Interoperability became less important

KAME Project

- A single effort
  - 8 core members from 7 Japanese companies
  - Fujitsu, Hitachi, IIJ, NEC, Toshiba, YDC, Yokogawa
- Two-years joint project
  - April 1998 - March 2000
  - The core members work for IPv6 three days a week
- KAME
  - A short word of KArigoME, where our office locates
  - Turtles

Purpose

- Reference code
  - IPv6, IPsec, and advanced networking
    - FreeBSD, NetBSD, OpenBSD, and BSD/OS
  - Provided "AS IS" like BSD
    - Free and no warranty
  - Commercial use is OK
- Release schedule
  - SNAP - every Monday
    - STABLE - every other month
    - RELEASE - quality assurance by TAHI Project
      - http://www.tahi.org
      - Specification conformance
      - Interoperability

The current status

- The basic spec has been implemented
  - IPv6, DNS, BGP4+, RIPng, translator, laptop computer support
    - IPsec, IKE
    - IPComp, IPv4 NAT, ECN, ATM, ALTQ
    - Many applications
      - SMTP, POP, HTTP, FTP, TELNET, SSH, X11,...
- Used in the worldwide 6bone
  - More than 20 countries, 200 people
  - Both as routers and hosts
- Reference code
  - Merged: NetBSD
  - Will be merged: BSD/OS, FreeBSD, OpenBSD

IPv6 on Production

- JB
  - WIDE backbone
    - IPv6, Diffserve, Multicast
    - # of IPv6 over IPv4 tunnel is being decreased
- Remote class
  - University of Wisconsin
    - Introduction to Computer Networks
    - Professor Lawrence Landweber
    - Digital Video via 6TAP
  - Credits
    - Keio University
    - Nara Institute of Science and Technology

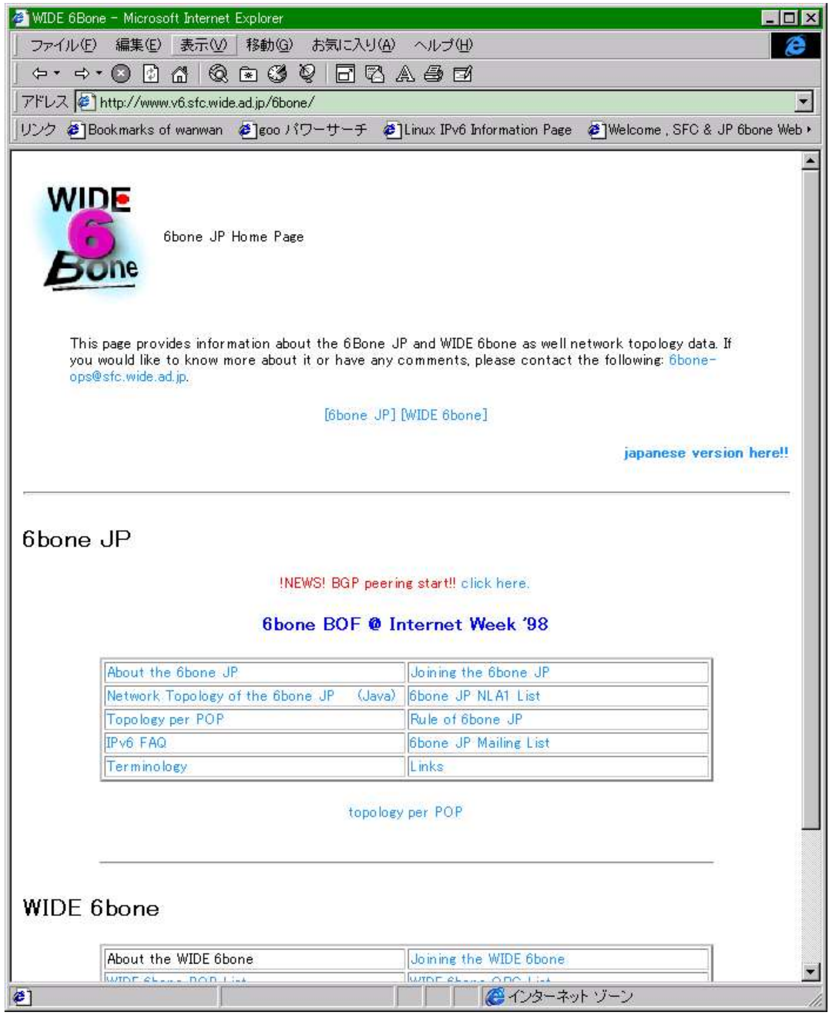
IPv6 address allocation

- ICANN
  - Working with IANA, RIR not to slow down the process
- APNIC
  - 2001:200::/35 WIDE Project
  - 2001:208::/35 National University of Singapore
  - 2001:210::/35 CONNECT AT
  - 2001:218::/35 OCN (NTT)
- JPNIC
  - Started working with APNIC

6bone-JP Registry System

- Web based IPv6 registry system from 1997
- easy to update and view
- uses PGP public-key for authentication of maintainer
- can apply for IPv6 address via the Web

http://v6.sfc.wide.ad.jp/6bone/


WIDE 6Bone - Microsoft Internet Explorer
ファイル(F) 編集(E) 表示(V) 移動(G) お気に入り(A) ヘルプ(H)
アドレス
http://www.v6.sfc.wide.ad.jp/6bone/
リンク
Bookmarks of wanwan
goo パワーサーチ
Linux IPv6 Information Page
Welcome , SFC & JP 6bone Web
WIDE 6 Bone
インターネット ゾーン

6bone JP Home Page

This page provides information about the 6Bone JP and WIDE 6bone as well network topology data. If you would like to know more about it or have any comments, please contact the following: 6bone-ops@sfc.wide.ad.jp.

[6bone JP] [WIDE 6bone]

japanese version here!!

6bone JP

!NEWS! BGP peering start!! click here.

6bone BOF @ Internet Week '98

About the 6bone JP	Joining the 6bone JP
Network Topology of the 6bone JP (Java)	6bone JP NLA1 List
Topology per POP	Rule of 6bone JP
IPv6 FAQ	6bone JP Mailing List
Terminology	Links

topology per POP

WIDE 6bone

About the WIDE 6bone	Joining the WIDE 6bone