2BA5: Introduction to IPv6

David Malone

<dwmalone@maths.tcd.ie>

<David.Malone@nuim.ie>

<dwmalone@FreeBSD.org>

10 January 2007

Introduction

- Why IPv6 was designed.
- IPv6 Addressing (format, types and policy).
- How IPv6 interacts with other layers.
- Transition mechanisms.
- The current state of IPv6.

IPv4: Late 80s/Early 90s

- Hosts went from 10,000 to 100,000 between 1987 to 1989.
- IP space was *classful* (126 class A of 16M hosts, 16K class B of 64K, 2M class C of 253).
- Concern about routing and addressing.
- By 1993, people reckoned there was ¡ 1yr worth of address space left.
- Lead to CIDR *Classless Interdomain Routing*.

CIDR

Class A, B and C had network/host boundary.
CIDR puts the boundary on any bit.

Net	First addr	netmask	prefix len
MIT	18.0.0.0	255.0.0.0	/8
TCD	134.226.0.0	255.255.0.0	/16
School	87.32.0.0	255.240.0.0	/12

Required updating routing protocols to store prefix length and some TCP/IP stacks, which assumed netmasks

NAT

Network address translation made it possible to use fewer addresses. Idea: rewrite addresses using rules. Allows use of private address space.

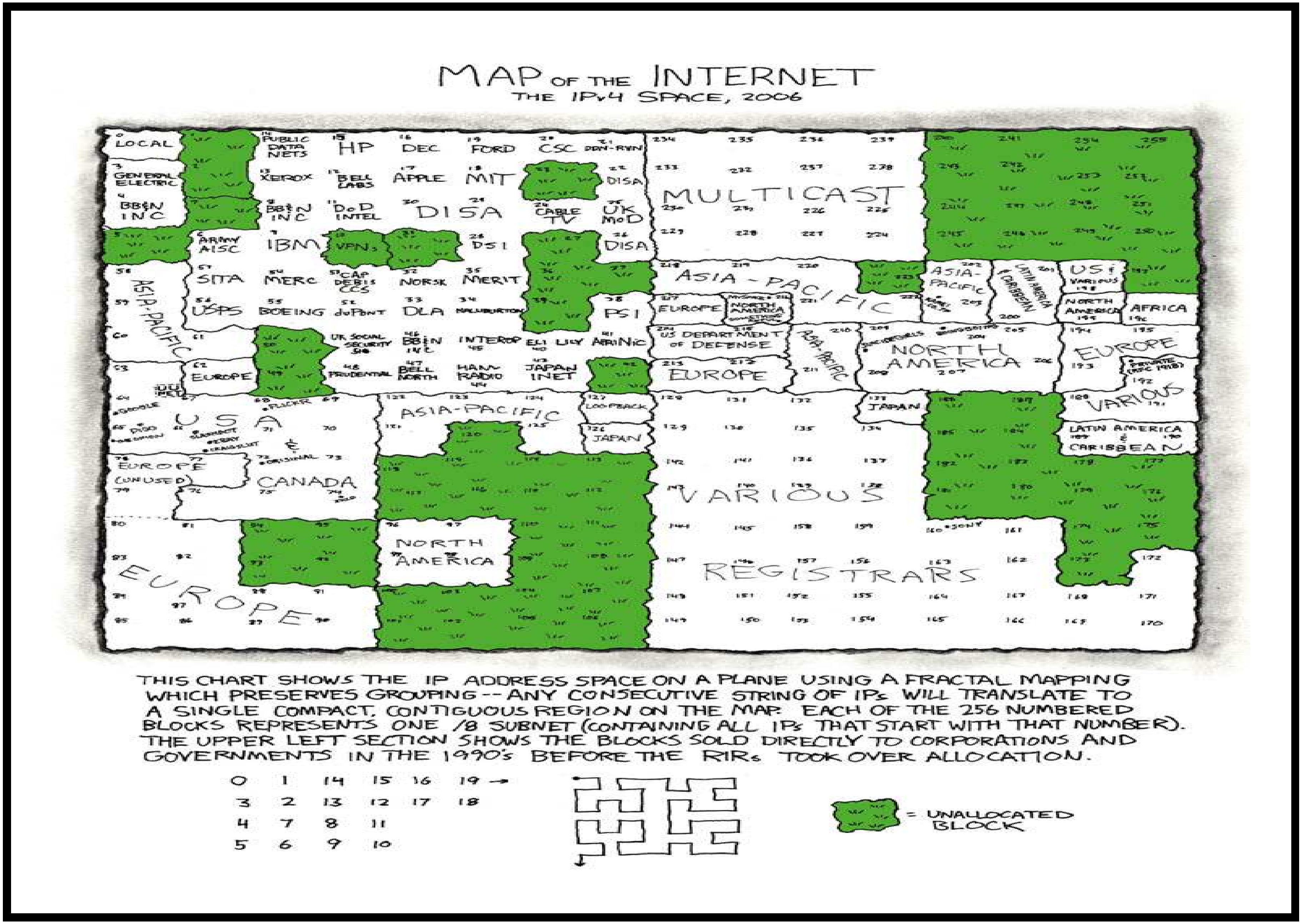
- Connection from private block is made.
- Allocate public address/ports, record in rules.
- Outgoing packets have private address/port replaced.
- Incoming packets have public address/port replaced.

IPv4 Today

CIDR and NAT have bought more time than expected. Has given IPv6 (too much?) time to grow.

Problems with IPv4 Internet.

- Security (spam, viruses, botnets, exploits, . . .)
- Routing (scalability, stability, multihoming)
- NAT (inhibitor, robustness, performance, cost)
- Politics/Market (scarce resources)
- . . .


MAP OF THE INTERNET
THE IPv4 SPACE, 2006
LOCAL
GENERAL ELECTRIC
BBN INC
PUBLIC DATA NETS
HP
DEC
FORD
CSC
DDN-RVN
XEROX
BELL LABS
APPLE
MIT
DISA
BBN INC
DoD INTEL
DISA
CABLE TV
UK MoD
ARMY AISC
IBM
VPNs
DSI
DISA
MULTICAST
SITA
MERC
CAP DEBIS CCS
NORSK
MERIT
USPS
BOEING
duPONT
DLA
HALLIBURTON
PSI
ASIA-PACIFIC
UK SOCIAL SECURITY
BBN INC
INTEROP
ELI LILY
AfriNIC
EUROPE
PRUDENTIAL
BELL NORTH
HAM RADIO
JAPAN INET
ASIA - PACIFIC
EUROPE
NORTH AMERICA
US DEPARTMENT OF DEFENSE
EUROPE
ASIA-PACIFIC
NORTH AMERICA
ASIA-PACIFIC
LATIN AMERICA & CARIBBEAN
US & VARIOUS
NORTH AMERICA
AFRICA
EUROPE
PRIVATE (RFC 1918)
USA
GOOGLE
FLICKR
ASIA-PACIFIC
LOOPBACK
JAPAN
JAPAN
VARIOUS
LATIN AMERICA & CARIBBEAN
EUROPE (UNUSED)
CANADA
VARIOUS
SONY
NORTH AMERICA
EUROPE
REGISTRARS
THIS CHART SHOWS THE IP ADDRESS SPACE ON A PLANE USING A FRACTAL MAPPING WHICH PRESERVES GROUPING -- ANY CONSECUTIVE STRING OF IPs WILL TRANSLATE TO A SINGLE COMPACT, CONTIGUOUS REGION ON THE MAP. EACH OF THE 256 NUMBERED BLOCKS REPRESENTS ONE /8 SUBNET (CONTAINING ALL IPs THAT START WITH THAT NUMBER). THE UPPER LEFT SECTION SHOWS THE BLOCKS SOLD DIRECTLY TO CORPORATIONS AND GOVERNMENTS IN THE 1990's BEFORE THE RIRs TOOK OVER ALLOCATION.
0 1 14 15 16 19 →
3 2 13 12 17 18
4 7 8 11
5 6 9 10
= UNALLOCATED BLOCK

Longer term solution: IPv6

- 1993 proposals sought for successor (IPng).
- Several proposals merged to get IPv6 idea by 1995.
- Standardisation continues to the day, like IPv4.
- Mainly replaces the layer 3 in the OSI model.

OSI Layers

Layer	Name	Description	Example
7	Application	Applications and associated protocols	HTTP
6	Presentation	Data syntax and semantics	XDR
5	Session	Session management for applications	
4	Transport	Packetisation, retransmission, . . .	TCP
3	Network	How subnets interoperate	IP
2	Data Link	Management of interface	Ethernet (upper level)
1	Physical	Physical operation of the medium	Ethernet over UTP

Keep TCP, UDP, HTTP, ... Just replace Layer 3.

Need to update the glue though.

Major changes

- Bigger addresses (128 bit from 32).
- Better extensibility (extension headers).
- Built in autoconfiguration (DHCP and PPP still possible).
- Mandatory IPsec.
- More integrated multicast.
- ARP replaced with Neighbour Discovery.

Addresses

- Compromise between variable and 64 bit.
- 128 bit addresses: 340282366920938463463374607431768211456,
- In practice you loose space to structure (see RFC 3194).
- 64 bits network, 64 bits host, CIDR for *aggregation*
- Written in 8 hex quads,
- Several shortcuts allowed.

Examples

- 2001:0770:0010:0300:0000:0000:86e2:510b
- 2001:770:10:300:0:0:86e2:510b
- 2001:770:10:300::86e2:510b
- 2001:770:10:300::134.226.81.11

Structured Addresses

- 2001::/16 = Chunk of production address space
- 2001:770::/32 = HEAnet (ISP prefix)
- 2001:770:10::/48 = TCD (organisation prefix)
- 2001:770:64:200::/56 = CS (dept prefix)
- 2001:770:64:200::/64 = CS (subnet)

Special Addresses

::	Unspec
::1	localhost
fe80::/10 block	link-local
fec0::/10 block	site-local (deprecated)
ff00::/8 block	multicast

Multiple addresses on each interface.

Destinations

Unicast Destined to a single machine (normal).

Broadcast Destined to all machines (ARP).

Multicast Destined to all in a particular group (IP TV, ND).

Anycast Destined to any one of a particular group (DNS Root, 6to4).

Autoconfiguration

Addresses formed from network prefix + host id.

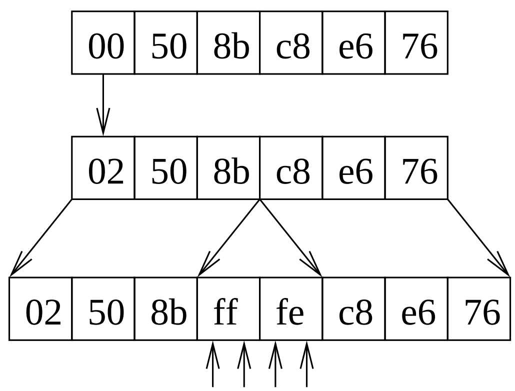
1. Find host-id and form link-local.
2. Check unique.
3. Multicast router solicitation to get prefix(es).
4. Form global address(es).
5. Duplicate address detection.
6. Manual, DHCPv6 and privacy addressing also possible.

Address Scope

IPv6 allows link-local on each interface. How does it know which?

Must specify address scope — for link-local typically use interface ID.

Example



Flip universal/local bit.

Insert fffe to make 64 bits.

```
% ifconfig -a
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
    inet6 ::1 prefixlen 128
    inet6 fe80::1%lo0 prefixlen 64 scopeid 0x3
    inet 127.0.0.1 netmask 0xff000000
xl0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
    ether 00:b0:d0:f4:c6:c5
    inet6 fe80::2b0:d0ff:fef4:c6c5%xl0 prefixlen 64 scopeid 0x1
    inet 147.252.43.5 netmask 0xffffff00 broadcast 147.252.43.255
    inet6 2001:770:68:1ff:2b0:d0ff:fef4:c6c5 prefixlen 64 autoconf
```

IPv4 Header

Version 4 bit	Head Len 4 bit	ToS 8 bit	Total Length 16 bit		
ID 16 bit			Flags 3 bit	Frag Offset 13 bit	
Time to Live 8 bit		Protocol 8 bit	Header Checksum 16 bit		
Source Address (32 bit)					
Destination Address (32 bit)					
Options — *variable*					

IPv6 header

Version 4 bit	Traffic Class 8 bit	Flow Label 20 bit
Payload Length 16 bit	Next Header 8 bit	Hop Limit 8 bit
Source Address 128 bit		
Destination Address 128 bit		

Header Differences

- Bigger addresses.
- Drop uncommonly used (frags, IP opts).
- Drop unneeded (header len, checksum).
- ToS $\rightarrow$ TC, TTL $\rightarrow$ HL, Protocol $\rightarrow$ NH.
- Alignment to 64 bits, min length from 20B to 40B.
- New flow label.

Extension Headers

Chaining headers together:

IPv6 header	Routing header	TCP header and data
Next Header = 43	Next Header = 6	

Terminal types (6 = TCP, 17 = UDP, 58 = ICMPv6, 59 = No Next Header).

Extension headers (0 = Hop-by-Hop Options, 43 = Routing, 44 = Frag, 60 = Destination Options). Also Authentication and Encapsulated Security Payload.

Header type	subtype	RFC
Hop-by-Hop options		
	Padding	2460
	Router alert	2711
	Jumbo Payload	2675
Destination options		
	Padding	2460
	Binding update	3775
	Binding acknowledgement	3775
	Binding request	3775
	Home Address	3775
Routing header		
	Type 0	2460
	Mobility	
Fragment header		2460
Authentication header		2402
ESP header		2406

Options: skip, drop, drop & send ICMPv6 parameter problem, ditto if not multicast.

ICMPv6

ICMP closely tied with IP $\Rightarrow$ new version.

- Ping, error reporting, redirects, PMTU.
- Router/Neighbour solicitation/advertisement.
- Neighbour discovery: dropping ARP.
- Provides Duplicate Address Detection, Neighbour Unreachability Detection.
- Also provides configuration/reconfiguration (routers, lifetimes, MTUs, routes).

ND details largely define interactions with layer 2.

Interactions with upper layers

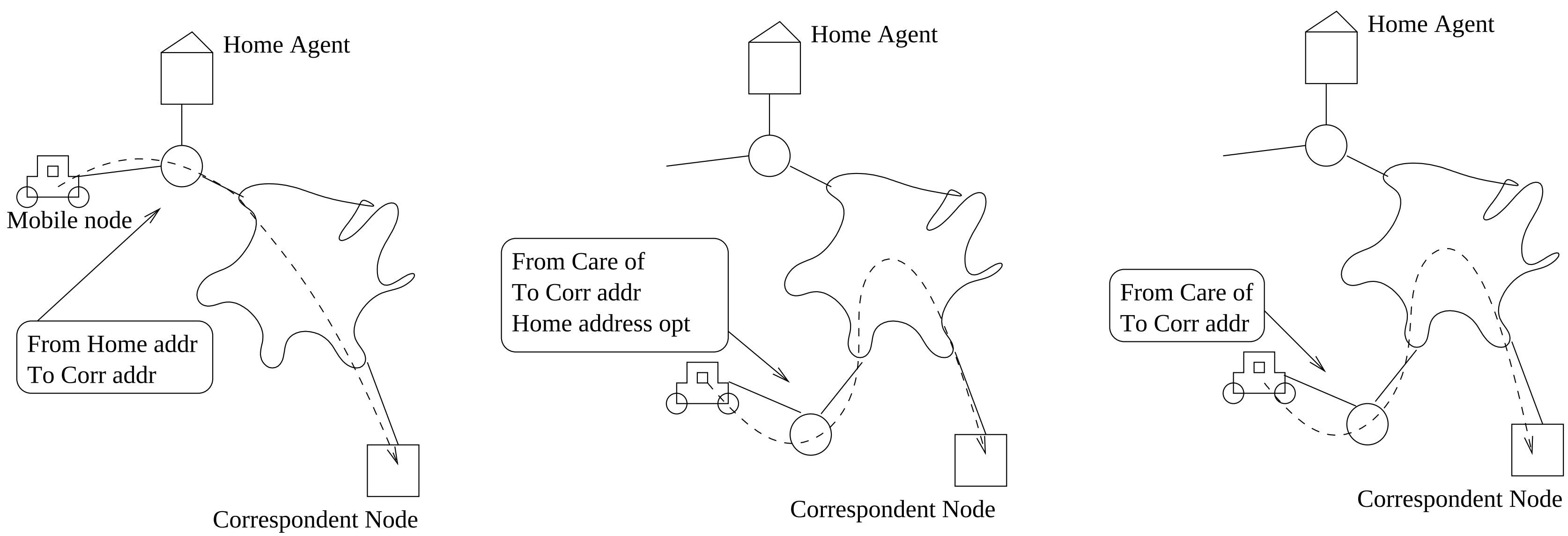
- TCP/UDP require a pseudo-header to checksum.
- Some special features for jumbograms.
- Interactions with protocols that embed IP addresses: DNS, FTP, SMTP, NTP.
- Some changes trivial.

```
Recieved: from 134.226.81.11 ...
Recieved: from [IPv6:2001:770:10:300::86e2:510b] .
```

Example: DNS

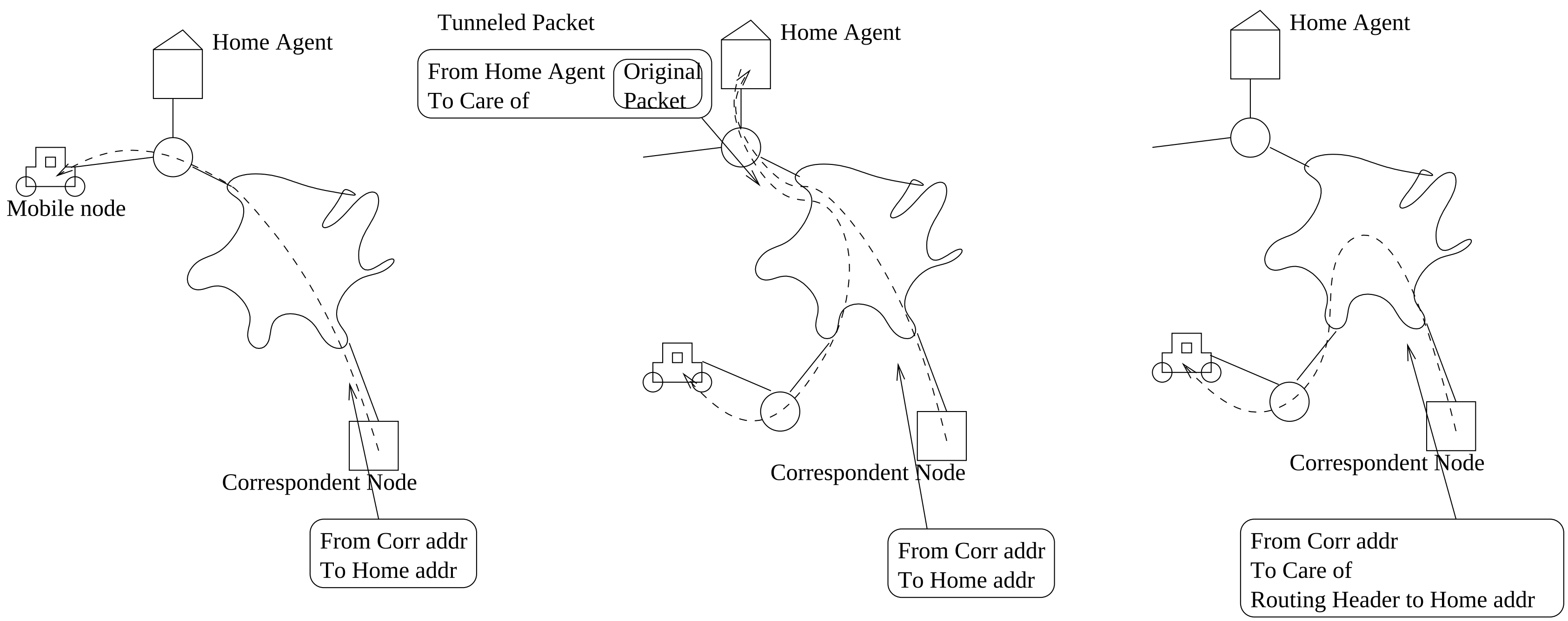
- DNS messages carried in TCP/UDP. Easy.
- But keys and records relate to addresses.
- A: salmon →134.226.81.11.
- PTR: 11.81.226.134.in-addr.arpa → salmon
- AAAA: salmon.ipv6 → 2001:770:10:300::...
- PTR: ...0.7.7.0.1.0.0.2.ip6.arpa → salmon.ip6
- (Admin: New reverse zone delegated from upstream.)

Example Extension: MIPv6



Path from mobile to correspondent node.

Mobile IPv6



Path from correspondent to mobile node.

Transition Mechanisms

- Large IPv4 network, want IPv6 network.
- IPv4 only hardware, software and people.
- How to get IPv6 working around this?
- A lot of effort on Transition Mechanisms for getting IPv6 working.
- Broad strategies: dual stack, tunnelling, translation and proxies.

Method	Summary	Type
Dual-stack	Run IPv4 and IPv6 on nodes.	DS
DSTM	Dual-stack, but dynamically allocate IPv4 addresses as needed.	DS
Configured tunnel	Virtual point-to-point IPv6 link between two IPv4 addresses.	Tun
Automatic tunnel	Automatic encapsulation of IPv6 packets using “compatible addresses”.	Tun
6to4	Automatic assignment of /48 network to each public IPv4 address.	Tun
Teredo	IPv6 in UDP through a NAT.	Tun
6over4	Using IPv4 as a link layer for IPv6, using IPv4 multicast.	Tun
ISATAP	Using IPv4 as a link layer for IPv6, using a known router.	Tun
SIIT	Rules for translating IPv6 packets straight into IPv4.	Tran
NAT-PT	Using SIIT to do NAT with IPv4 on one side and IPv6 on the other.	Tran
TRT	Translating IPv6 to IPv4 at the UDP/TCP layer.	Tran
BIS	Using SIIT to do make IPv4 applications speak IPv6.	Tran
BIA	Using a special library to make IPv4 applications speak IPv6.	Tran
Proxies	Using application level trickery to join IPv4 to IPv6 networks.	Prox

Dual Stack

- Have IPv4 and IPv6 on the machine.
- Do both IPv4/IPv6 depending on what packets arrive (eg. try autoconfig).
- If user enters IPv4 address, do IPv4, if they enter IPv6, do IPv6.
- If name is used, check for IPv4 and IPv6 addr.
- Usually prefer IPv6.

Usual case, except where software constrained by resources. Allows gradual deployment.

Tunnelling

- Embedded/chained headers is normal.
- Why not put an IPv6 packet in an IPv4 packet.
- Protocol 41 — IPv6 in IPv4.
- Configured tunnel: fixed IPv4 addresses between two points.
- Tunnel brokers can provide these addresses (e.g. SixXS).
- Could use other protocols (teredo = IPv6-in-UDP-in-IPv4++).

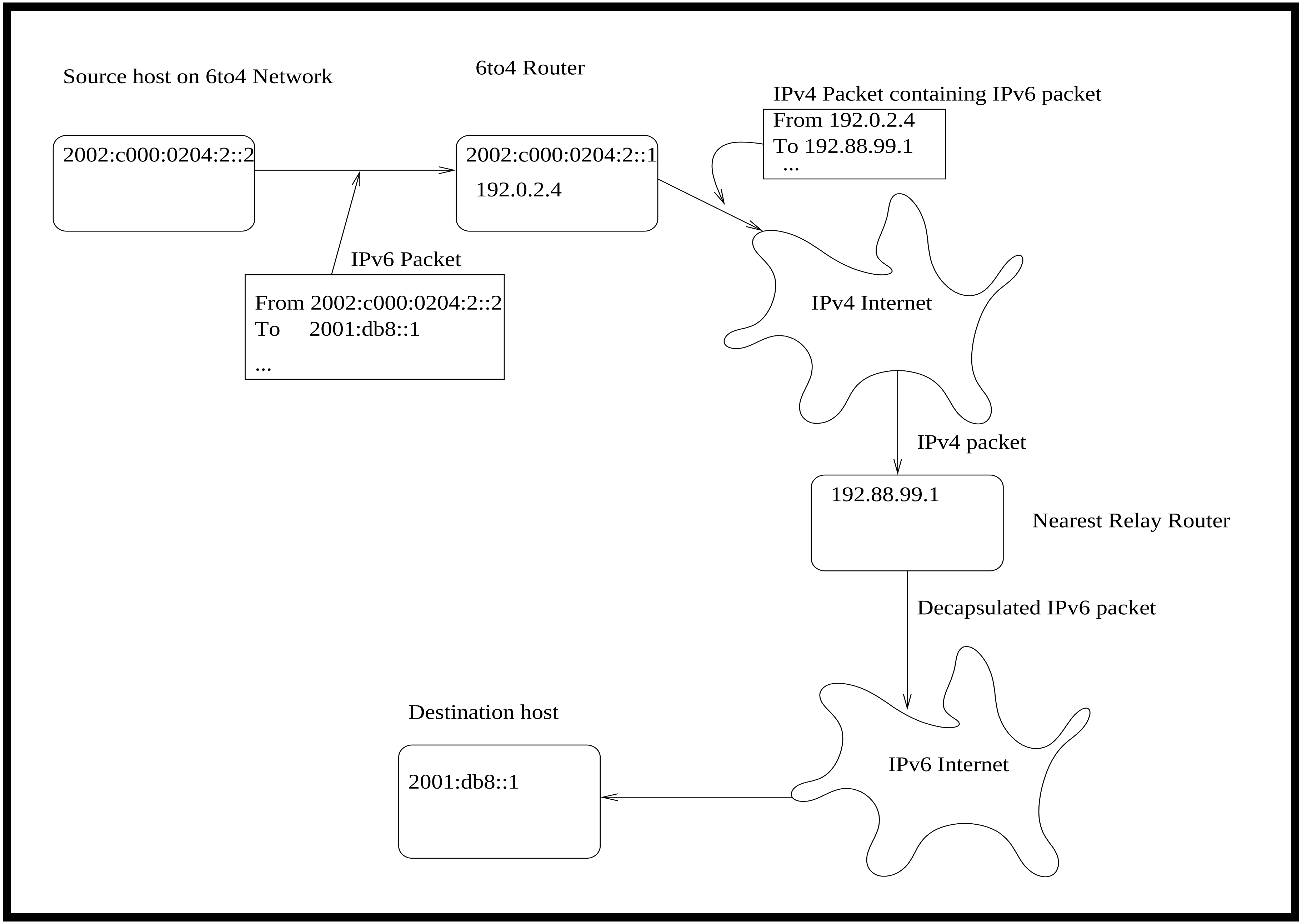
Tunnelling quite common.

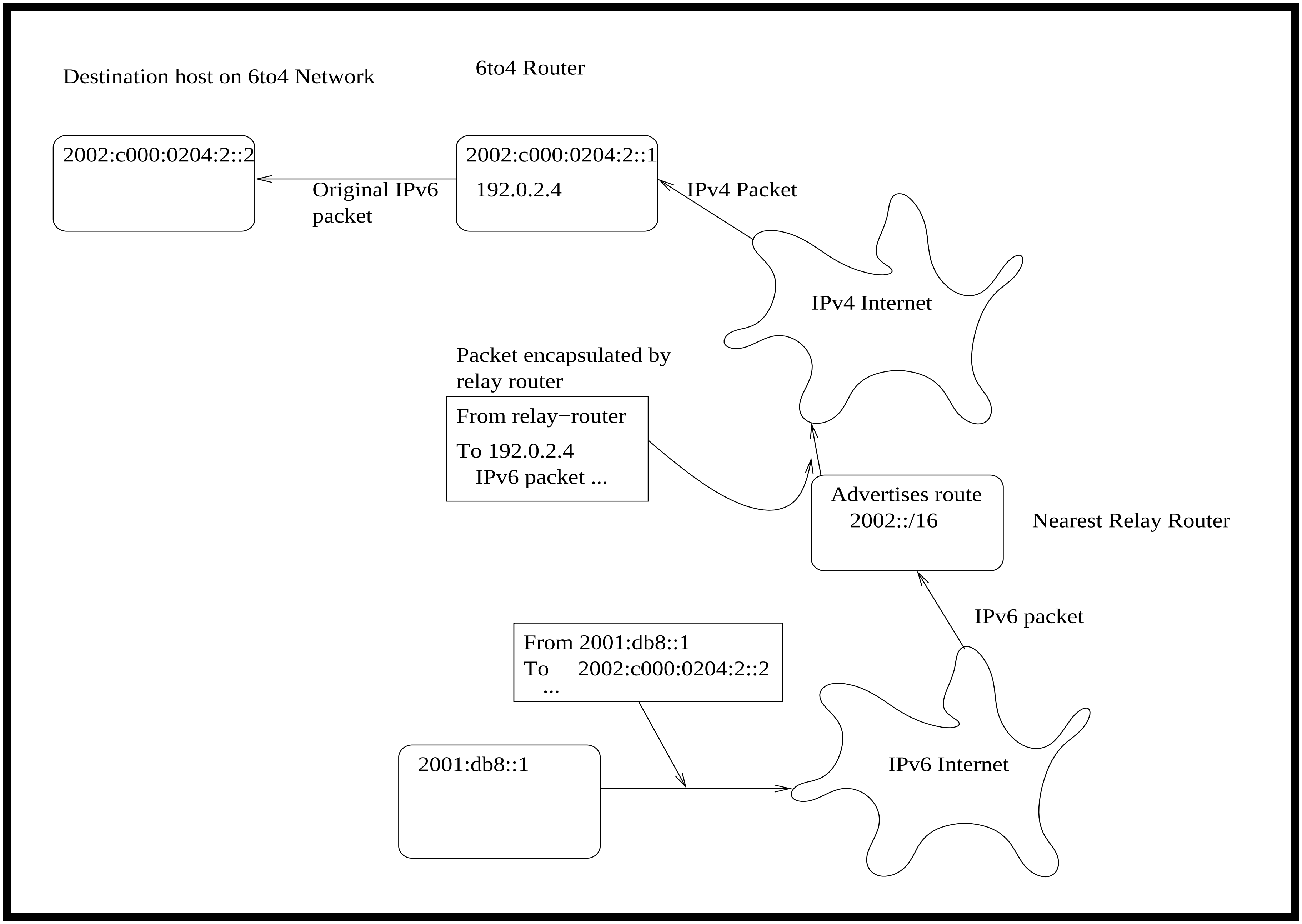
Could be used to get around IPv4 only router (at home, on campus, in ISP) or longer distance (between two research groups).

More exotic tunnelling is also useful.

6to4: IPv6 from any public IPv4

- 6to4 — very easy, easy config.
- www.xxx.yyy.zzz → 2002:WWXX:YYZZ::
- Uses anycast.
- Eircom and HEAnet offer relays.


Source host on 6to4 Network
6to4 Router
IPv4 Packet containing IPv6 packet
From 192.0.2.4
To 192.88.99.1
...
2002:c000:0204:2::2
2002:c000:0204:2::1
192.0.2.4
IPv6 Packet
From 2002:c000:0204:2::2
To 2001:db8::1
...
IPv4 Internet
IPv4 packet
192.88.99.1
Nearest Relay Router
Decapsulated IPv6 packet
Destination host
2001:db8::1
IPv6 Internet


Destination host on 6to4 Network
6to4 Router
2002:c000:0204:2::2
2002:c000:0204:2::1
192.0.2.4
Original IPv6
packet
IPv4 Packet
IPv4 Internet
Packet encapsulated by
relay router
From relay−router
To 192.0.2.4
IPv6 packet ...
Advertises route
2002::/16
Nearest Relay Router
IPv6 packet
From 2001:db8::1
To 2002:c000:0204:2::2
...
2001:db8::1
IPv6 Internet

Translation

- Do NAT like tricks in network, stack, libraries.
- Will suffer from NAT's problems.
- Defeats many advantages of IPv6.
- IPv6 not yet at the stage where this is really necessary?

Proxies

- Dual stack proxies can work at application layer.
- Eg. Dual stack web proxy: connect IPv4/IPv6 clients to IPv4/IPv6 servers.
- Very natural in many situations (Web, DNS, SMTP).
- Sometimes very easy to deploy.

Local history

2000/08/03 Maths upgraded to FreeBSD 4.1.

2000/08/09 I harass davew re IPv6 address.

2001/03/29 With mknell merge IPv6 networks.

2002/02/06 DNS/addresses allocated by HEAnet.

2002/06/04 AAAA added to production names.

2002/12/20 Natively over Gigabit Ethernet.

2003/07/21 AAAA glue in .ie zone.

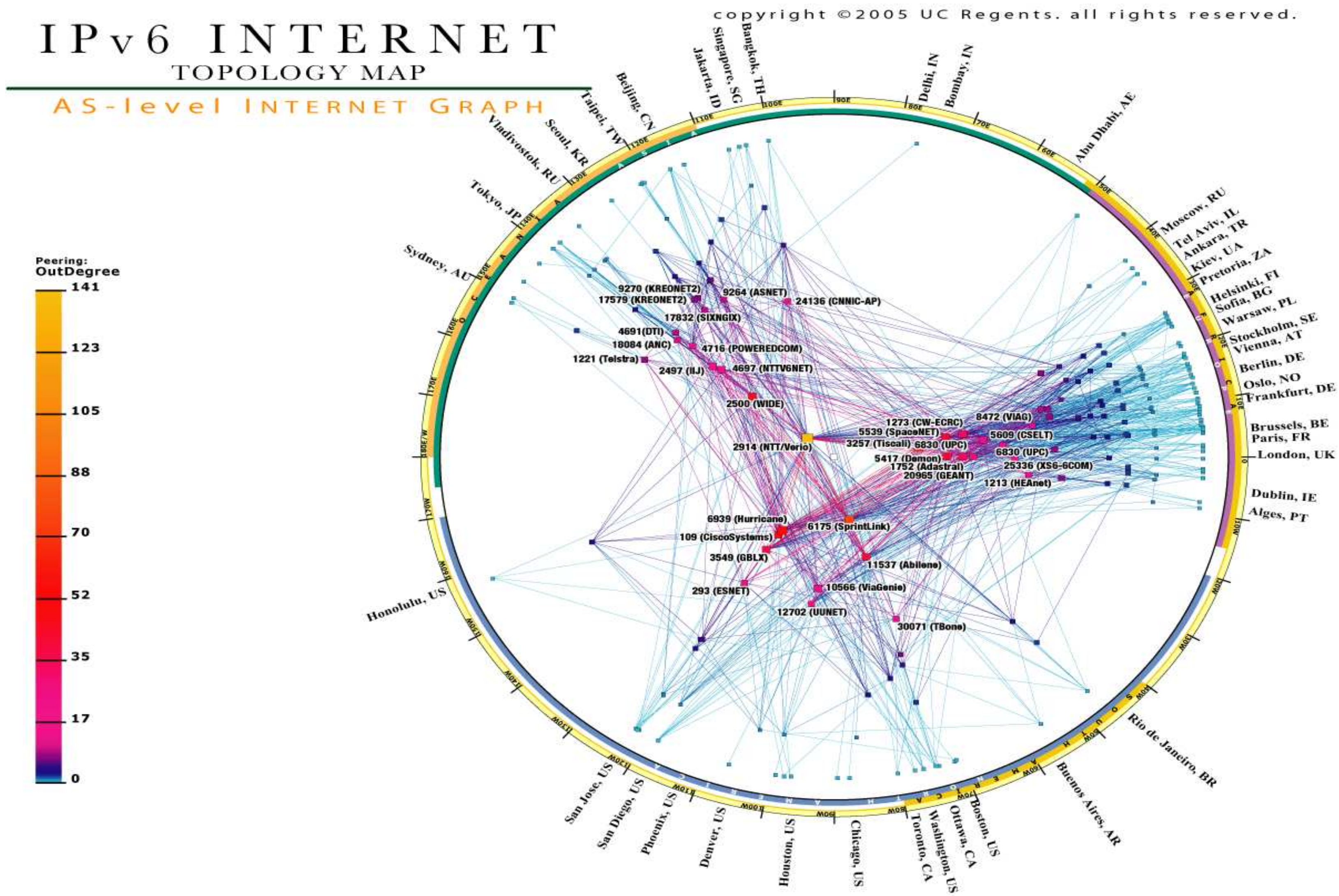
2004/09/06 IPv6 capable .ie server.

2005/03/14 Google get a block of IPv6 addresses.

Current IPv6 Issues

- Basics and software porting largely done.
- Deployment well underway, though not common.
- Stuck on businesses and some ISPs.
- Vista should have IPv6 on by default.
- Some remaining standardisation to be done (multihoming, anycast, DNS config).
- Address policy developing, routing technically as IPv4, but policies different.

- Some features growing towards usefulness (router advertisements, address selection).
- Personal favourite: Protocol resilience. If you screw up your IPv4/6 network, firewall, router, then you can still use IPv6/4 to get in and fix it.
- What impact will IPv6 have on worms? Bigger address space.
- Bigger addresses also mean more choice for attacks, like algorithmic complexity attacks.


IPv6 INTERNET
TOPOLOGY MAP
AS-level INTERNET GRAPH
copyright ©2005 UC Regents. all rights reserved.
Peering:
OutDegree
141
123
105
88
70
52
35
17
0
9270 (KREONET2)
17579 (KREONET2)
9264 (ASNET)
24136 (CNNIC-AP)
17832 (SIXNGIX)
4691(DTI)
18084 (ANC)
4716 (POWEREDCOM)
1221 (Telstra)
2497 (IIJ)
4697 (NTTV6NET)
2500 (WIDE)
2914 (NTT/Verio)
1273 (CW-ECRC)
5539 (SpaceNET)
3257 (Tiscali)
6830 (UPC)
8472 (VIAG)
5609 (CSELT)
5417 (Demon)
1752 (Adastral)
20965 (GEANT)
25336 (XS6-6COM)
1213 (HEAnet)
6939 (Hurricane)
109 (CiscoSystems)
6175 (SprintLink)
3549 (GBLX)
11537 (Abilene)
293 (ESNET)
10566 (ViaGenie)
12702 (UUNET)
30071 (TBone)
Beijing, CN
Taipei, TW
Seoul, KR
Vladivostok, RU
Tokyo, JP
Sydney, AU
Jakarta, ID
Singapore, SG
Bangkok, TH
Delhi, IN
Bombay, IN
Abu Dhabi, AE
Moscow, RU
Tel Aviv, IL
Ankara, TR
Kiev, UA
Pretoria, ZA
Helsinki, FI
Sofia, BG
Warsaw, PL
Stockholm, SE
Vienna, AT
Berlin, DE
Oslo, NO
Frankfurt, DE
Brussels, BE
Paris, FR
London, UK
Dublin, IE
Alges, PT
Rio de Janeiro, BR
Buenos Aires, AR
Boston, US
Ottawa, CA
Washington, US
Toronto, CA
Chicago, US
Houston, US
Denver, US
Phoenix, US
San Diego, US
San Jose, US
Honolulu, US