Introduction to IPv6

Stig Venaas, UNINETT

2001-10-25

Overview

- What is wrong with IP?
- IPv6 basics
- More IPv6
- Implementations
- Getting started
- IPv6 connectivity and transition techniques
- UNINETT IPv6 network and activities
- UNINETT future and experiences

What is wrong with IP?

- IP (IPv4) has been a great success, why change it?
- A victim of its own success
  - Running out of IP addresses
  - Routing tables are too large

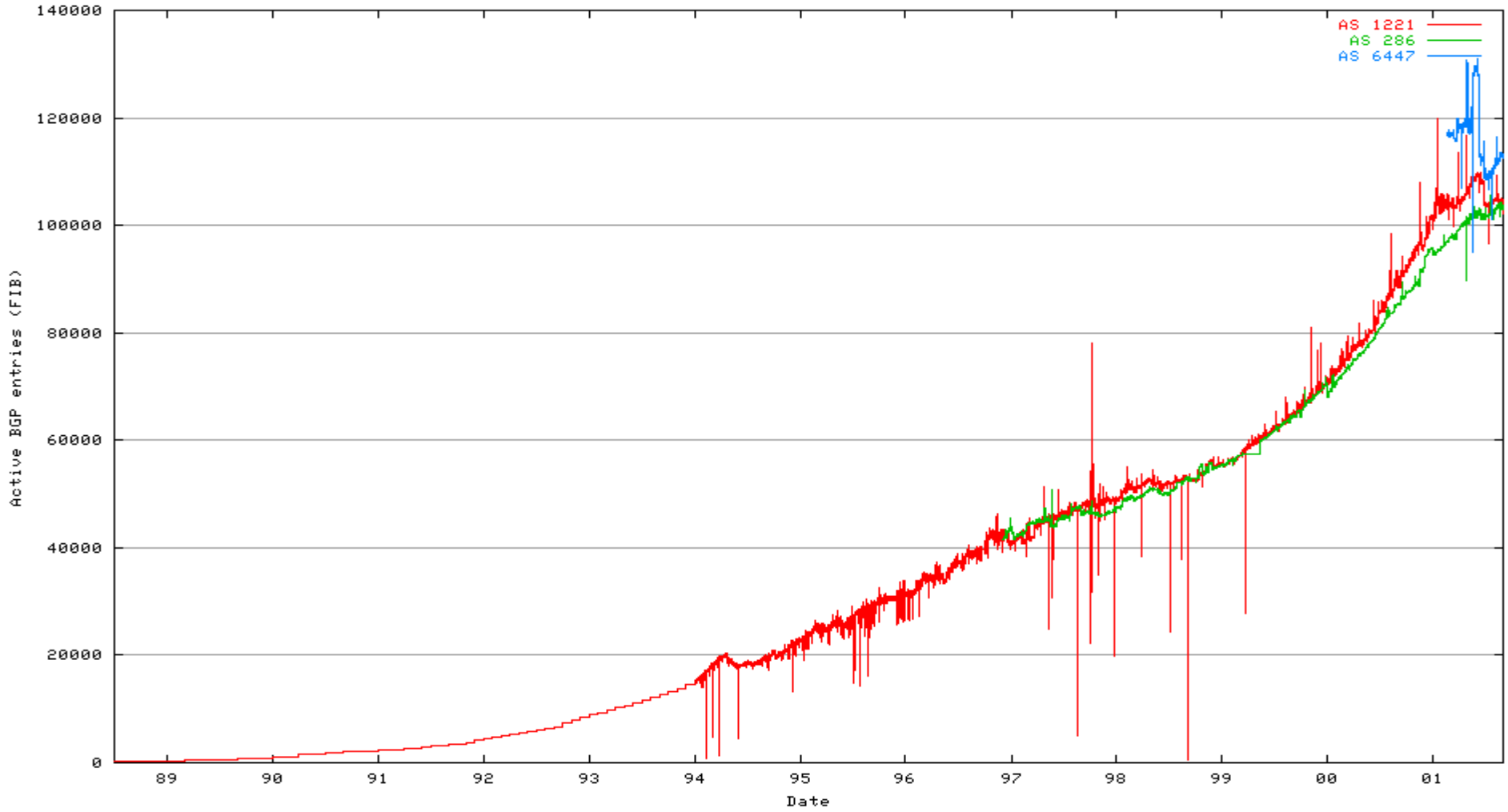
Lack of IP addresses

- Internet design is based on each host having a globally unique identifier (address)
- Increasing number of hosts
- Techniques like NAT can help, but there are serious problems
- IPv4 addresses can be distributed in better ways, but difficult in practice and only short term solution
- Still IPv4 addresses left, but already hard/expensive to get enough.

Increasing number of hosts

- Always on in the home
  - xDSL, cable, satellite, 802.11, ...
- Mobile phones (GPRS, UMTS)
  - Each phone at least one address when connected (always?)
- Always on everywhere
  - GPRS, UMTS, 802.11, Bluetooth? Connectivity from cars, planes..
- Internet gaining popularity in new countries
  - Lack of addresses in Japan
  - China will run into problems.

Size of routing tables(1)



Size of routing tables(2)

- Rapid growth, want less detailed routes in global routing tables
- Main problem is slow BGP convergence
- Better aggregation with IPv6
  - Customers get addresses from providers
  - Customers change addresses when switching providers
- Still not clear how to do multihoming
  - Multihoming is getting more and more popular
  - Often leads to more routes in global tables
  - The IETF is trying to find a solution for IPv6, might apply to IPv4.

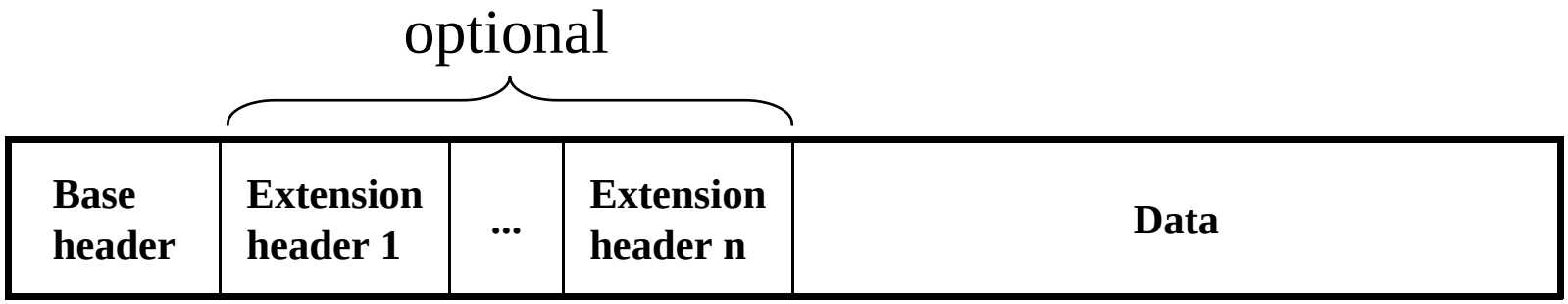
IPv6 basics

- Addresses
- Packet headers
- MTU and fragmentation
- Address architecture
- Neighbor discovery
- Auto configuration

IPv6 addresses

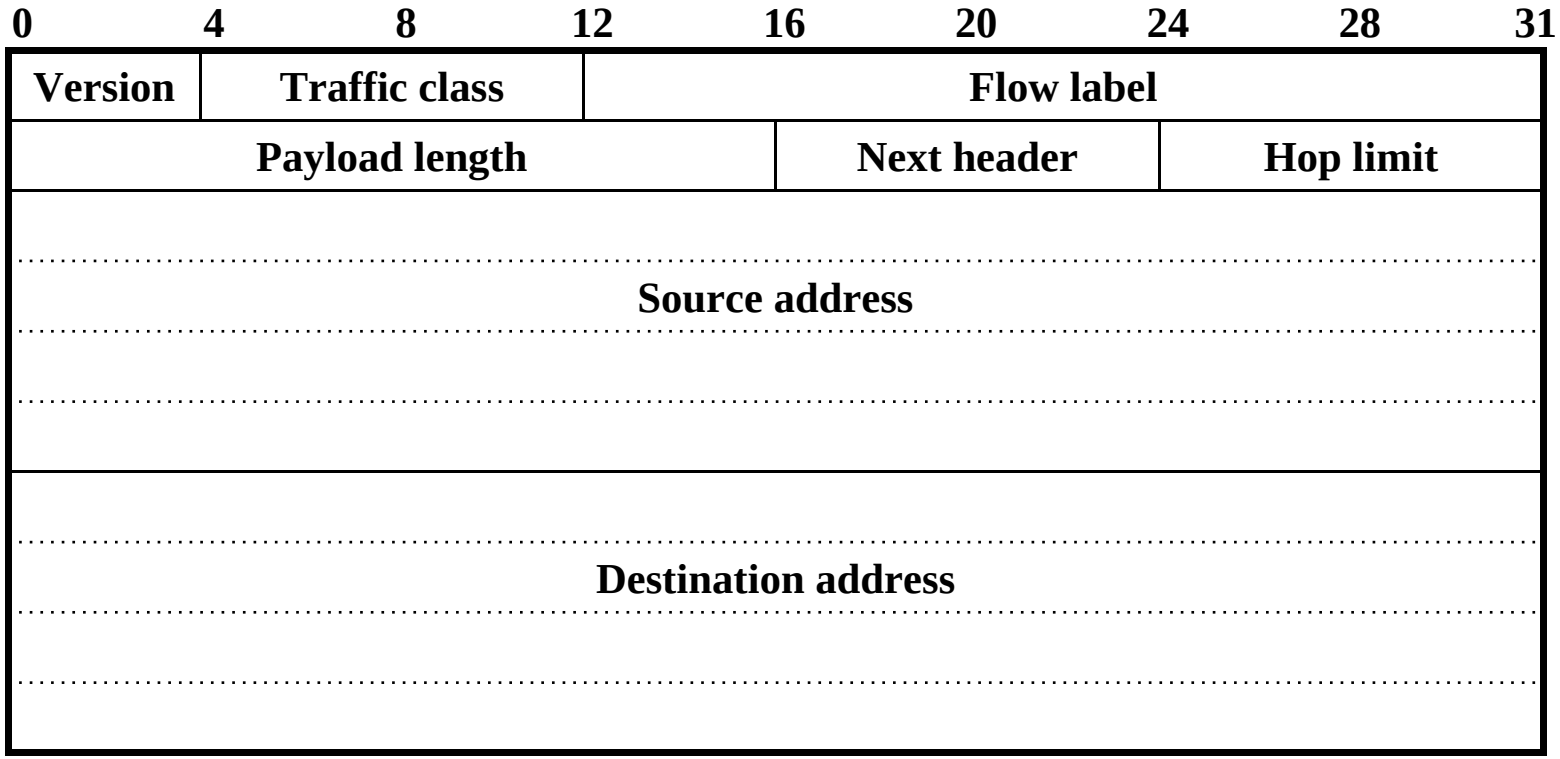
- 128 bits, enormous number
- How to write them:
  - 8 blocks of 16 bits, each written in hex separated by :
    - 3ffe:2a00:100:7020:0:0:dead:beef
    - 2001:700:700:1:0:0:0:2
  - 0-compression, consecutive 0 blocks written as ::, can be used only once (else ambiguous)
    - 2001:700:700:1::2
  - Dotted deimal suffix, useful during IPv4/IPv6 transition.
    - ::ffff:129.241.210.18

IPv6 packet headers



- Each header contains a next field describing the next header. next=tcp, next=udp etc describes the payload
- IPv4 has only one header that might contain options, protocol field describes payload.

IPv6 base header



MTU and fragmentation

- IPv6 requires MTU to be at least 1280 bytes
- Only sender fragments (like IPv4 with DF) and usually only receiver reassembles
- If a packet is to be forwarded onto a link with MTU less than packet size, ICMP Packet Too Big is sent to the source address
- Recommended that nodes support Path MTU discovery, but a minimal implementation might restrict itself to sending packets of size <=1280
- Fragmentation should be avoided, better that the application reduces its packet size

Address architecture

- Loopback address ::1 (127.0.0.1 in IPv4)
- Link-local unicast addresses
- Site-local unicast addresses
- Global unicast addresses
- No broadcast addresses

Multicast addresses

- 4 bits flags (well known/transient), 4 bits scope, 32 bits group ID
  - Node local ff01, link local ff02, site local ff05, organization local ff08
- Some special multicast addresses:
  - All nodes addresses ff01::1, ff02::1
  - All routers addresses ff01::2, ff02::2, ff05::2
  - Solicited-node address ff02::1:ffXX:XXXX
    - For each unicast address a node responds to the solicited-node address where the X’es are the last 24 bits of the unicast address

Neighbor discovery

- Replaces IPv4 ARP, no broadcast
- Uses multicast and ICMP
- Address resolution
  - Sends solicitation message to the solicited-node multicast address of the target address (sender includes its link layer address). Target responds with an advertisement message containing its link layer address
- Individual hosts will receive much less address resolution packets than in IPv4
- Using ICMP is more media-independent than ARP and allows for IP security mechanisms

Auto configuration(1)

- Stateless autoconfiguration of addresses, default router, MTU etc. Not DHCPv6 (separate protocol)
- Uses multicast and ICMP
- Router advertisement messages
  - Sent by routers at regular intervals or when prompted
  - Info about prefixes, hop limit, MTU, life time etc.
  - Sent to all-nodes multicast address or a specific host
- Router solicitation messages
  - Prompts for a router advertisement, advertisement is sent to the source address

Auto configuration (2)

- Interface identifier
  - 64 bit identifier, required to be unique on the link
  - One way is IEEE EUI-64, for ethernet EUI-64 gives a way of creating 64 bits identifier from 48 bits MAC
- Link local address FE80::EUI-64
  - Check uniqueness by sending neighbor solicitation
- Global and site-local addresses
  - Append interface identifier to known prefixes
  - Prefixes usually known from router advertisements

More IPv6

- DNS
- Ipsec
- Mobile IPv6
- Porting applications

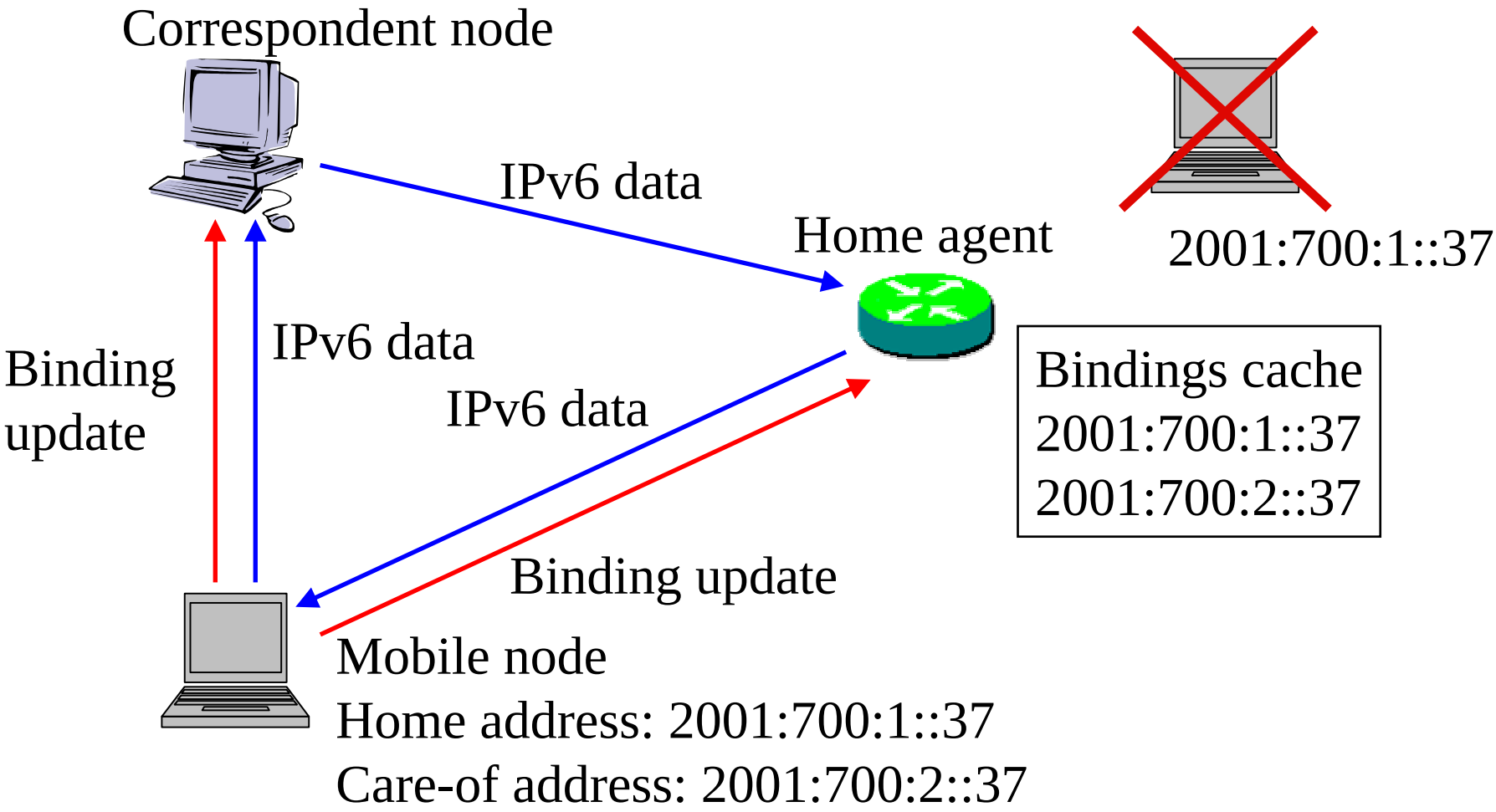
DNS

- Very few IPv6 name servers, no official root servers accessible with IPv6
- IPv6-only host will need help of a dual-stack name server for resolving (or some translation magic)
- Registering hosts with IPv6 addresses in the DNS
  - kattem AAAA 2001:700:1:0:290:27ff:fe55:fe7b
  - $ORIGIN 0.0.0.0.1.0.0.0.0.0.7.0.1.0.0.2.ip6.int
    b.7.e.f.5.5.e.f.f.f.7.2.0.9.2.0 PTR kattem.uninett.no.

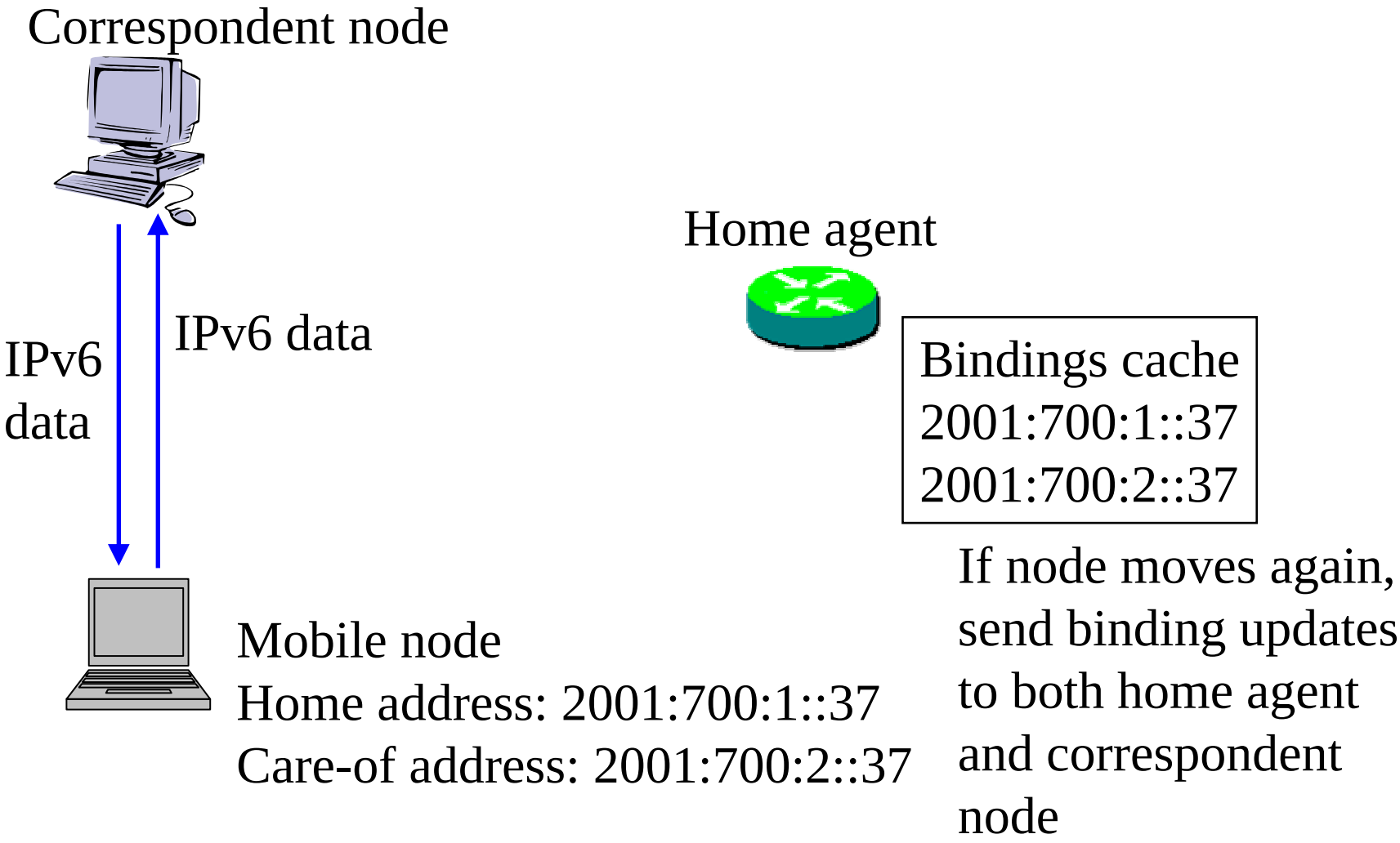
IPsec

- Integrity and confidentiality for IP packets
- Two extension headers included in full IPv6 implementations
- IP authentication header (AH)
  - Connectionless integrity, data origin authentication, optional anti-replay service. Integrity for both most of IP header and payload
- Encapsulating security payload (ESP)
  - Encryption of payload, may also provide the same as AH
- IPsec can be used in transport mode or tunnel mode. Security gateways use tunnel mode

Mobile IPv6(1)



Mobile IPv6(2)



Porting applications

- No radical changes in socket API
- Many platforms support PF_INET6 socket for both IPv6 and IPv4 (using IPv4-mapped IPv6 addr.), else application might need one for each
- sockaddr_in6, sockaddr_storage
- Replace gethostbyname() with getaddrinfo(), thread safe
- Replace inet_aton(), inet_addr() with inet_pton()
- Replace inet_ntoa() with inet_ntop(), thread safe

Implementations

- Linux, UC Berkeley 4.4 BSD, Net/Free/OpenBSD, AIX 4.3, Tru64 5.1, Solaris 8, HP-UX 11i IPv6, Windows NT/2000 (add-on), Windows XP, MacOS X
- Cisco IOS 12.2T (for most Cisco routers), Nortel BayRC 12.0 (hw announced), Ericsson/Telebit software, Hitachi and Juniper software (hw Q4/01)
- MIPv6 for Linux, some BSD, Windows
- Few proper IPsec implementations

Getting started

- http://www.ipv6.org/
  - Pointers to howtos, implementations, specifications
- http://www.ipv6forum.com/
  - IPv6 news, presentations, list of events, links, etc.

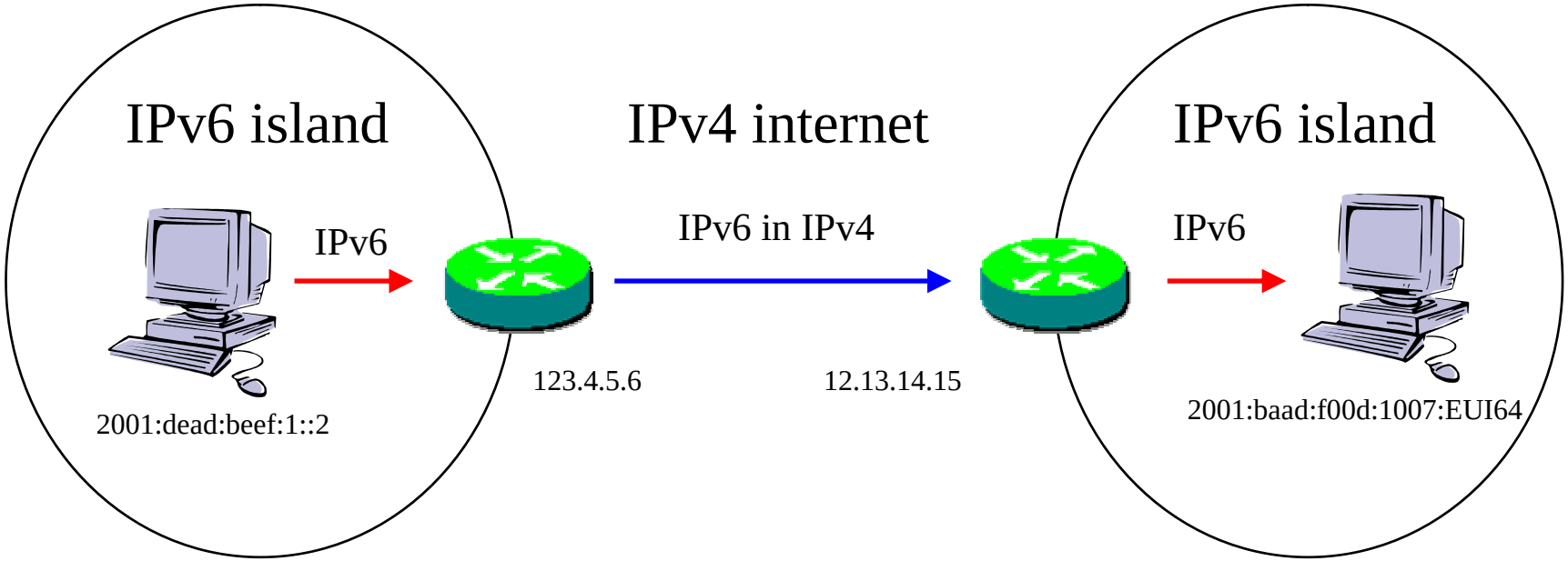
IPv6 connectivity

- Native IPv6 and IPv4; want to avoid separate physical links, easier if same provider
- Tunnel from any IPv6 provider, maybe use free tunnel broker for testing
- 6to4, need just one static IPv4 address, no IPv6 delegation. Need to find a friendly 6to4 relay to talk to non-6to4 users

Transition techniques

- Dual stack
  - Not really a technique, but a way for IPv4 and IPv6 to co-exist. Hosts and routers can support both, with native IPv4 and IPv6 on the same links
- Tunneling
  - Connecting sites
  - Intrasite
- Translation

Tunneling IPv6 in IPv4



src	2001:dead:beef:1::2
dst	2001:baad:f00d:1007:EUI64

IPv6 packet

src	123.4.5.6
dst	12.13.14.15
protocol	41

encapsulating IPv4 packet, IPv6 packet as payload

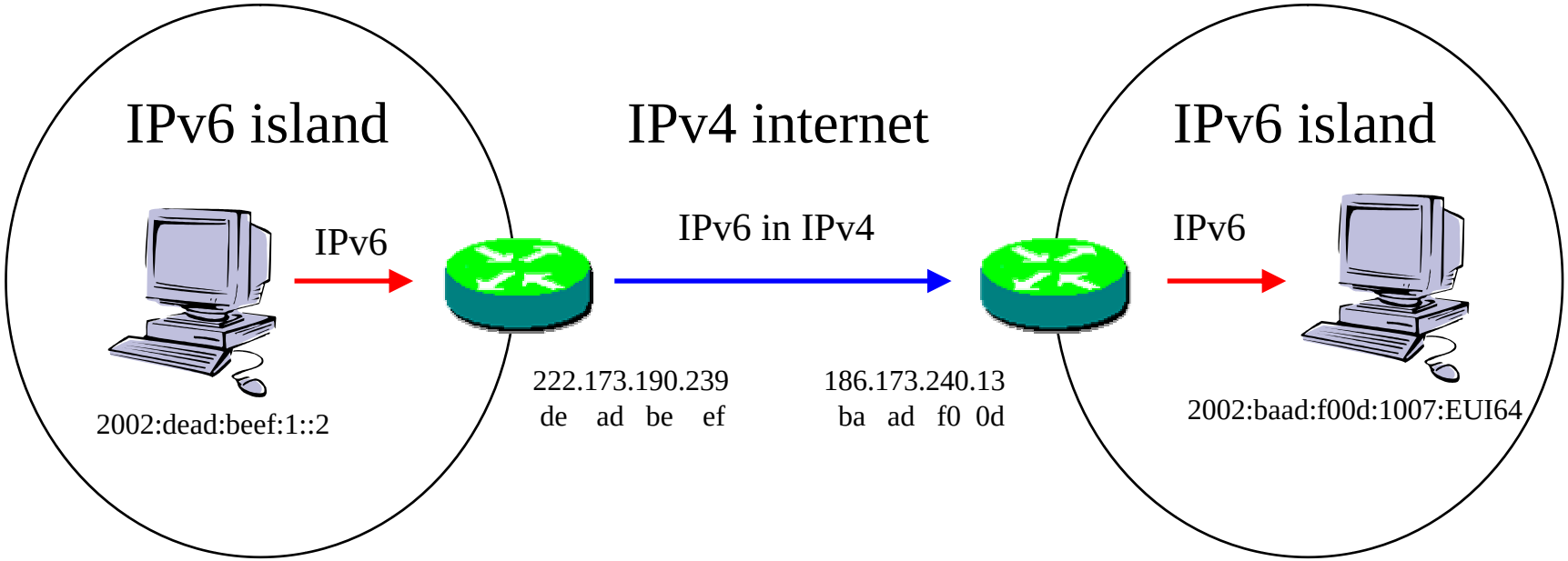
Tunneling

- Tunnel brokers, 6to4 etc for connecting sites
- ISATAP, 6over4 for intrasite connectivity
  - No need for IPv6 routers on all links, in theory all hosts can have IPv6 connectivity with just one IPv6 router at the site
- DSTM (dynamic IPv4 allocation, tunneling)
  - Can use IPv4 to talk to IPv4 hosts without the need for translation, can have more hosts than IPv4 addresses
  - Tunneling can be used to avoid internal IPv4 routing infrastructure

Tunnel brokers

- End user uses for instance web interface to register with name, e-mail address etc, and configuration data like IPv4 address.
- The provider allocates an IPv6 prefix and configures the provider side, automated
- End user configures hir side
- End user can change configuration via web later, or take tunnel up/down
- http://tb.ipv6.bt.com/v6broker/
  http://tunnel.be.wanadoo.com/

6to4



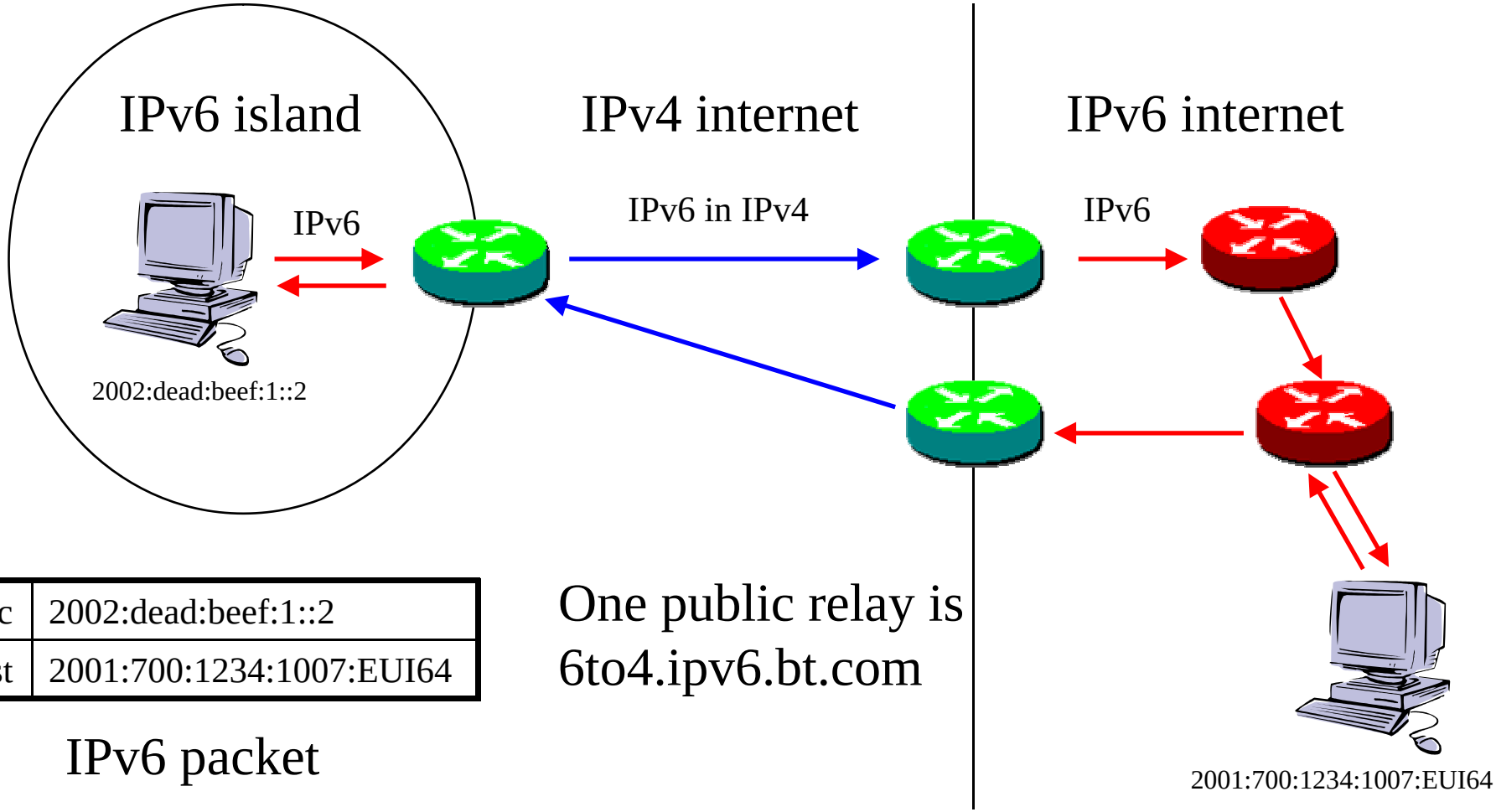
src	2002:dead:beef:1::2
dst	2002:baad:f00d:1007:EUI64

IPv6 packet

src	222.173.190.239
dst	186.173.240.13

encapsulating IPv4 packet

6to4 relay



src	2002:dead:beef:1::2
dst	2001:700:1234:1007:EUI64

IPv6 packet

One public relay is 6to4.ipv6.bt.com

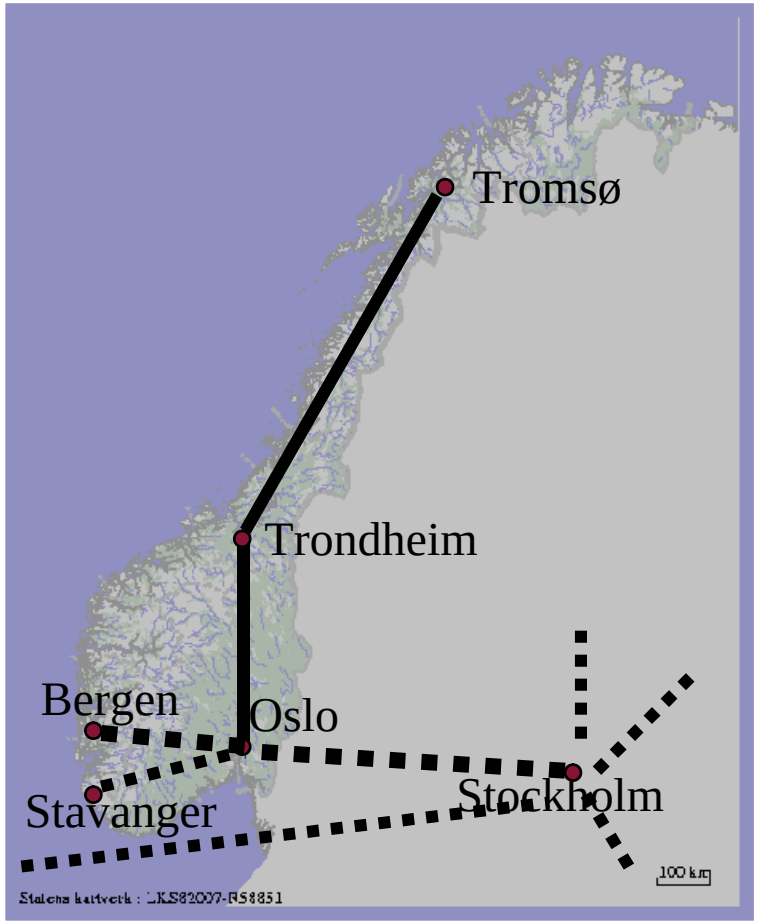
Translation

- Translation necessary for IPv6-only and IPv4-only hosts to communicate, should be done near edges, and we prefer dual stack if possible
  - NAT-PT
    - Packet level, much the same as normal NAT
  - TCP-relay
    - Session level, KAME Faith tested by UofTromsø
  - Application level gateways
    - Application level, basic proxies. Sometimes fit well with current architecture. For instance make an existing web cache dual stack, or make firewall with proxies dual stack

UNINETT IPv6 network

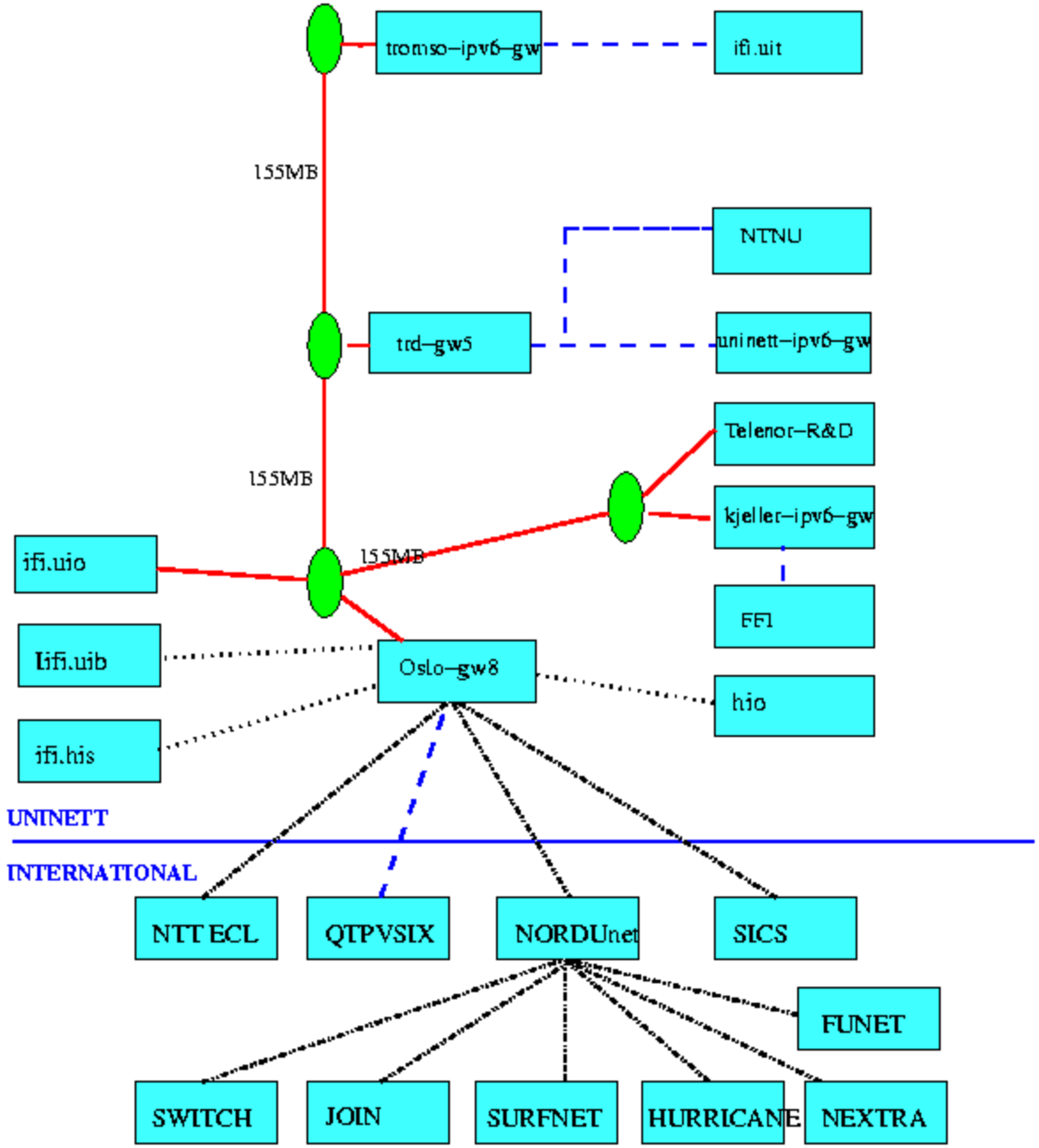
- Native IPv6 155Mbit/s Oslo-Trondheim-Tromsø, with native connections to several local sites in those cities; since 1998
- Tunnels from Oslo to Bergen, Stavanger and Stockholm (to 6bone)

External connectivity



- All NORDUnet members can set up native link or tunnel to a NORDUnet IPv6 router in Stockholm
- Use BGP4+ and mainly tunnels to peer with others and get transit access. Maybe also native through 6NET/GTPv6

6bone: 3ffe:2a00::/28 Sub TLA: 2001:700::/35



IPv6 at UNINETT office

- Native IPv6 and IPv4 in parallel in the UNINETT office, also wireless
- About half of the workstations are dualstack, mostly Linux and a few NetBSD
- Some employees have IPv6 at home, native wireless or tunnel

Application services

- DNS
  - BIND9 server reachable over IPv6, needed by IPv6-only clients
- HTTP/FTP-proxy
  - Squid proxy on dual-stack host
- www.uninett.no, ldap.uninett.no
  - Production services that are reachable over both IPv4 and IPv6

UNINETT future

- ISIS testing, maybe move from OSPF to ISIS for IPv4 also
- Core network with both native IPv4 and IPv6
- Port in-house management tools, also need some vendor support
- Multicast (one-way distribution and conferencing, also interest in reliable multicast)
- Mobile IPv6 (PDAs with wireless and streaming media)

Experiences(1)

- Network administrators with good IPv4 knowledge easily learns IPv6
- Hard to get enough experience from just a test network, and not all the necessary software and hardware exists, so we will deploy IPv6 gradually
- We want a network that is easy to manage, and maintain end-to-end and transparency, only possible with IPv6. The transition mechanisms create new complexity, but communication between IPv6 hosts will be transparent

Experiences(2)

- We have been and still are running experimental IPv6 code in many places. Some accidents and bugs, but works well most of the time
- Production quality IPv6 implementations from several vendors, more will follow