



JANET

IPv6 Hands-on Workshop

Module 4: **IPv6 Transition and Deployment**

UKERNA, Lancaster University
and University of Southampton, 2006



Where are we?

- After yesterday's theory and lab sessions you should understand:
 - IPv6 addressing
 - Basic IPv6 protocol operation
 - Configuring a router to turn on IPv6 on the wire
 - IPv6 operation on a LAN
 - Basic IPv6 services: DNS, mail, web
- Today, you will undertake a basic IPv6 deployment
 - Implies connectivity upstream (JANET)
 - May require use of some IPv6 transition technologies

Today's goals

- IPv6 transition
 - Methods to allow IPv4 and IPv6 systems to interwork and integrate
 - Dual-stack networking within a site
 - Configure tunnelling of IPv6 in IPv4 for external connectivity
 - Gain some idea of which tools are suitable for you
- IPv6 routing
 - Internal (e.g. OSPFv3, IS-IS)
 - External (e.g. BGP4+)
- IPv6 multicast
 - A little taster if time permits 😊

IPv6 Transition and Integration

- IPv6 services being introduced at the edges (“islands”)
 - Exchange points becoming dual-stacked
 - Demand on providers should see them offer native with time
- Might have IPv6-only systems
 - IPv4 systems may need to access IPv6 services
 - IPv6 systems may need to access IPv4 services
- Ideally offer both protocols ubiquitously
 - But can only do that for so long – IPv4 address space is limited
- Need to offer IPv6 connectivity through IPv4 networks
 - Use IPv4 to carry IPv6 (encapsulation, tunnelling)
- During all this, the users shouldn't care or notice

Deploying IPv6

- Each site or network will need to form its own plan for IPv6 deployment
 - Currently this would be alongside existing IPv4
- Need to consider various factors
 - Technical – do we need upgrades? Applications?
 - Policy – how do we handle IPv6 traffic?
 - Education – are our people trained to operate IPv6?
- Then schedule the process
- You should have a copy of the IPv6 Tech Guide:
 - <http://www.ja.net/services/publications/technical-guides/ipv6-tech-guide-for-web.pdf>
 - Includes advice on deployment

Phase 1: advanced planning

- Phase 1 includes:
 - Adding IPv6 capability requirements to future tenders
 - Obtaining IPv6 address space from JANET Customer Services
 - Typically a /48 size prefix
 - IPv6 training (and here you are!)
 - Encourage in-house experiments by systems staff
 - e.g. using the JANET IPv6 Tunnel Broker
 - Review IPv6 security issues
 - IPv6 is often enabled by default

Phase 2: Testbed/Trials

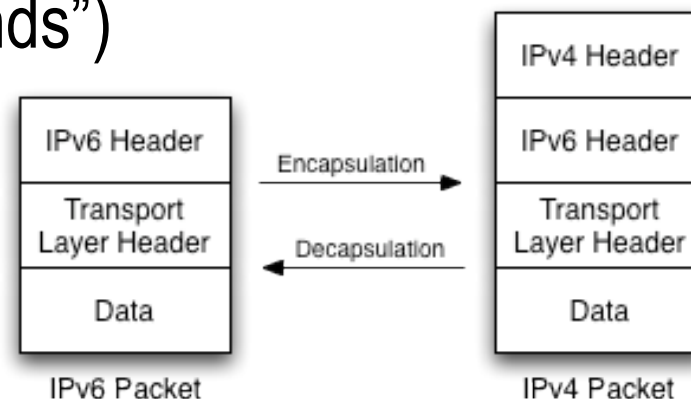
- Phase 2 includes:
 - Deploy IPv6 capable router, with cautious ACLs applied
 - Establish connectivity (probably a tunnel) to JANET
 - Probably to the Experimental Service, for which you can use your own address space
 - Set up an internal link with host(s), on a /64
 - Can be isolated from regular IPv4 network (e.g. a dual-stack DMZ)
 - Enable IPv6 on the host systems, add DNS entries if appropriate
- And in parallel
 - Survey systems and applications for IPv6 capabilities
 - Formulate an IPv6 site addressing plan
 - Document IPv6 policies (e.g. address assignment methods)

Phase 3: Production deployment

- Prudent to enable IPv6 on the wire, then services
- Phase 3 includes:
 - Plan initial deployment areas, e.g. your regular DMZ, WLAN, perhaps parts of your Computer Science department
 - Enable external IPv6 connectivity and ACLs/filters
 - Enable IPv6 routing ‘on the wire’ on selected internal links
 - Deploy IPv6 support in management/monitoring tools
- Then enable the services and advertise via DNS:
 - Enable IPv6 in selected services (e.g. web, SMTP)
 - Add IPv6 addresses to DNS, enable IPv6 DNS transport
- Remember IPv6 security:
 - e.g. include IPv6 transport in all penetration tests

Various transition approaches

- 1: Dual Stack
 - Servers/devices speaking both protocols
- 2: Tunnels (“connecting the IPv6 islands”)
 - IPv6 encapsulated over IPv4 links
 - IPv6 packet is payload of IPv4 packet
 - Requires “open” holes in firewalls
 - Packets whose Protocol field is ‘41’
- 3: Translation methods (“IPv4-only to IPv6-only”)
 - Rewriting IP header information
 - Rewriting TCP headers
 - Application layer gateways (ALGs)



1: Dual-stack

- Support both protocols on selected links (and nodes)
- Requires support in:
 - Host platforms
 - Router platforms
 - Applications and services
 - e.g. web, DNS, SMTP
- Adds considerations for
 - Security in all components
 - New policies dependent on IPv6-specific features
- Can run global IPv6 alongside NAT-ed IPv4

Dual-stack issues

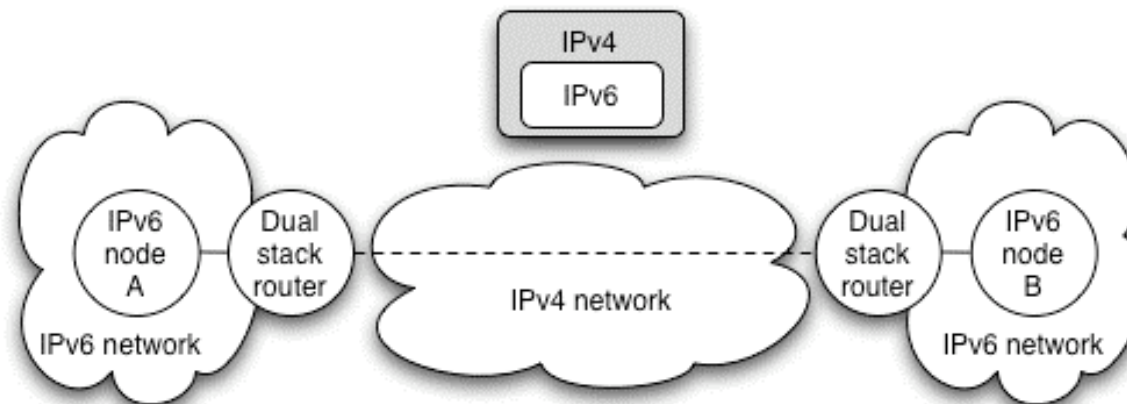
- Application must choose which IP protocol to use
 - DNS returns IPv4 (A record) and IPv6 addresses (AAAA record)
 - e.g. MSIE prefers IPv6
 - Don't advertise AAAA record for a host unless you have good IPv6 connectivity (for all services on host)
- Enabling IPv6 should not adversely impact IPv4 performance
 - Consider whether IPv6 tunnels use router CPU for example
 - So far is the case at Southampton
- Security should be no worse
 - Hosts listen on both protocols; secure both

Aside: IPv4 mapped addresses

- An IPv6 address used to represent an IPv4 address
- A socket API may receive an IPv4 connection as an IPv6 address, known as an IPv4-mapped address
 - Format is `::ffff:<ipv4-address>`
 - e.g. `::ffff:152.78.64.1`
- NB: This is one socket for both address families
- Should not be seen 'on the wire', i.e. not as source or destination address
- May appear in log files, depending on how the application handles a connection
- Typically seen in dual-stack deployments

2: Tunnelling

- IPv6 packets encapsulated in IPv4 packets
 - IPv6 packet is payload of IPv4 packet
- Usually used between edge routers to connect IPv6 'islands'
 - Edge router talks IPv6 to internal systems
 - Encapsulates IPv6 in IPv4 towards remote tunnel endpoint
- Initially IPv6 in IPv4, (much) later IPv4 in IPv6
 - May rely on Protocol 41 being allowed through firewalls



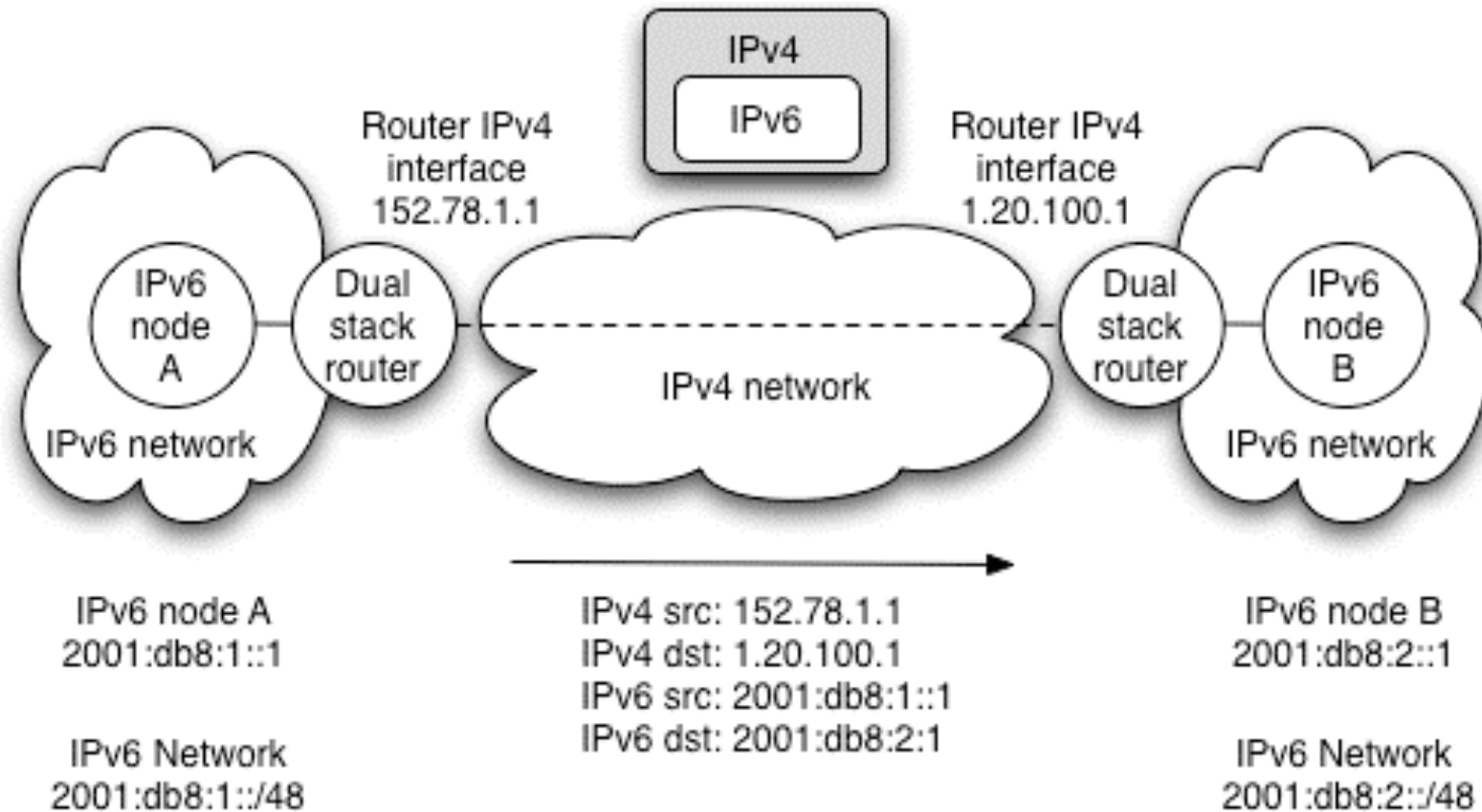
Manual or automatic?

- Can create tunnels manually or automatically
- Manual tunnels
 - Requires manual configuration, at both ends
 - Usually just one command/config line in the router at each end
 - Agreement on addresses to use for interfaces
 - Good from a management perspective: you know who your tunnels are created with
- Automatic tunnelling
 - Tunnels created on demand without manual intervention
 - Includes 6to4 (RFC3056)
 - Quite popular in SOHO deployments
 - Also: ISATAP and Teredo (out of scope of this workshop)

Manually configured tunnels

- Very easy to setup and configure
- Good management potential
 - ISP configures all tunnels, so is in control of its deployment
 - This is the current approach used by many NRENs (including UKERNA) to connect academic sites/users over IPv6 where native IPv6 connectivity is not available
 - In UKERNA case, sites apply for address space and tunnel connectivity usually in the same step (until RNOs support IPv6 natively)
- Usually used router-to-router or host-to-router
 - Desirable to allow end user to register (and subsequently authenticate) to request a tunnel
 - The IPv6 Tunnel Broker (RFC3053) offers such a system, usually for host-to-router connectivity, but sometimes for router-to-router
 - Removes some administration burden from the provider
 - See the JANET IPv6 Tunnel Broker: www.broker.ipv6.ac.uk

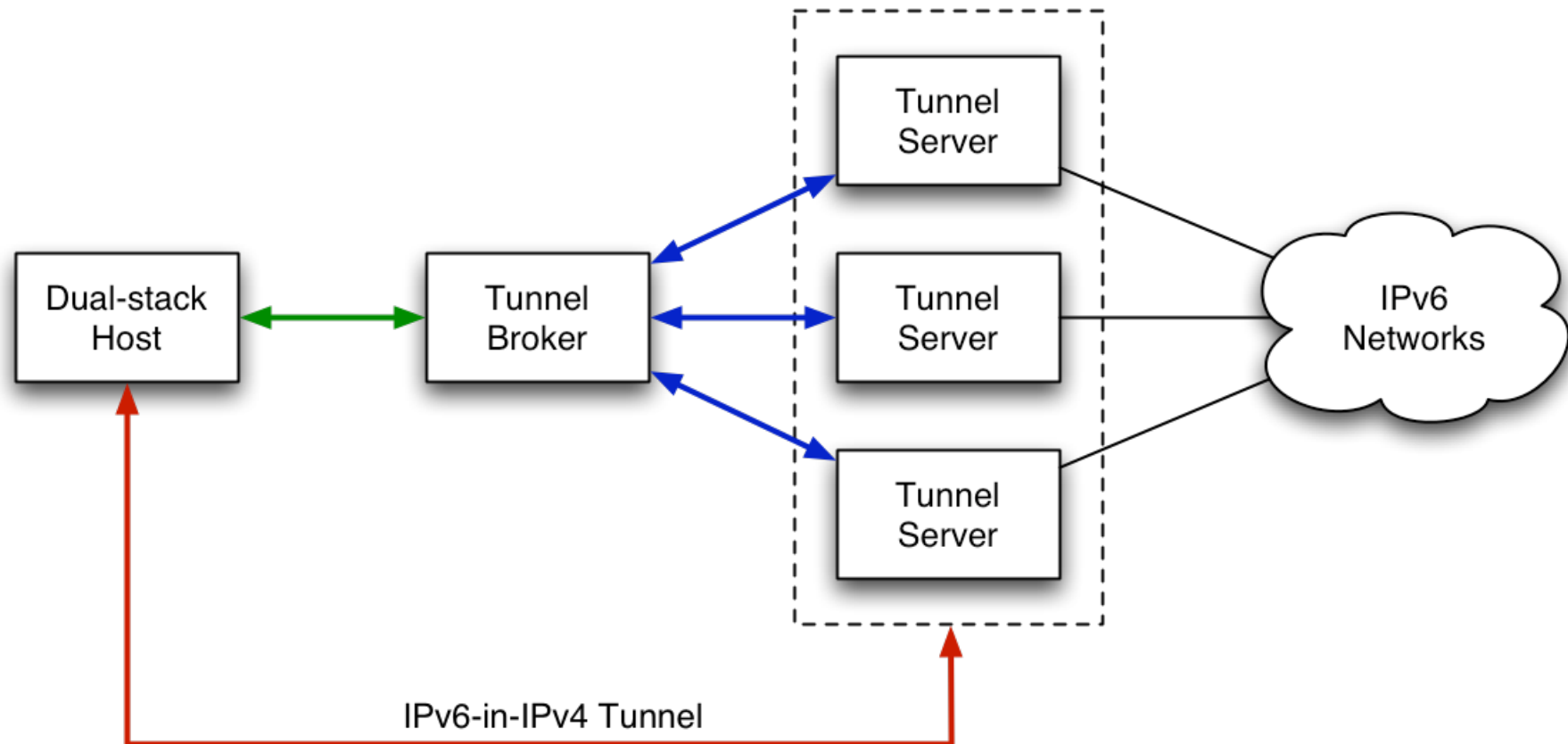
Manual tunnel addressing view



Tunnel broker

- Offer IPv6 connectivity to end user (usually a single host or subnet) over an IPv4 network
- User accesses tunnel broker web site
 - User may enter some authentication
- Broker configures tunnel end-point on tunnel server
 - User downloads and runs code to configure their local tunnel end point and allocated prefix/addresses
- Web server and tunnel server may be same system
 - e.g. as per the Hexago broker being piloted by UKERNA
- Can set up tunnels for hosts or networks
 - Tend to use broker for users, manual tunnels for campus sites
 - e.g. route /48 over tunnel to remote network, /64 to single subnet or /128 to single interface

Tunnel broker



1. User connects to Tunnel Broker web interface requesting tunnel
2. TB returns script to create tunnel to the Tunnel Server, and informs TS of new client
3. Client executes script, and gains access to IPv6 networks via the TS

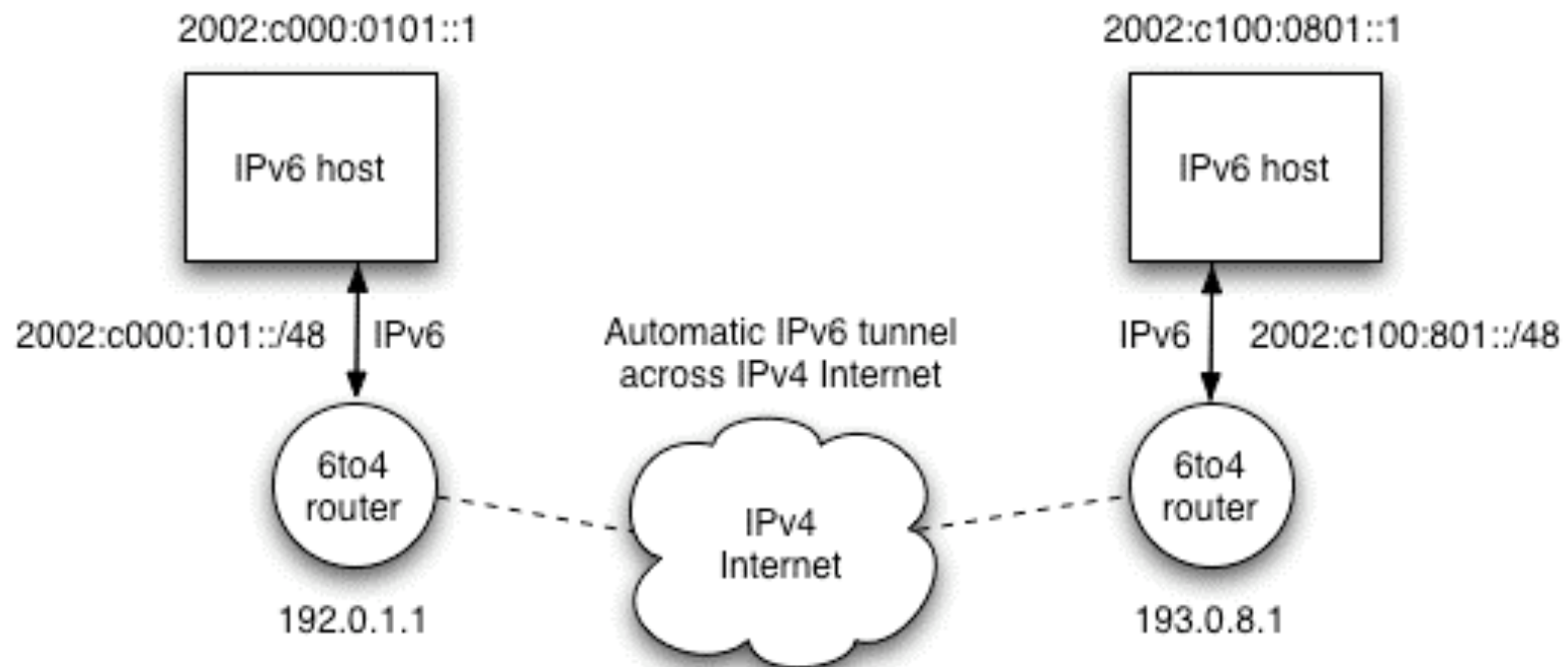
Tunnel broker issues

- Some considerations, for example..
 - Reduces demand on site administrators to set up a full IPv6 service
 - They may then not be aware that users want IPv6
 - Some potential problems where the IPv4 address used is dynamic (e.g. next host getting that IP may have unclosed tunnel running to it).
 - Needs extensions to work for a client that is located behind a NAT, e.g. UDP encapsulation
 - Hexago system has a client for NAT traversal
 - Protocol 41 must be allowed (not firewalled)

Automatic tunnels: 6to4

- An automatic router-to-router (usually) tunnelling method
- Thus in its basic configuration, 6to4 is used to connect two IPv6 islands across an IPv4 network
- 6to4 uses a special 'trick' for the 2002::/16 IPv6 prefix that is reserved for its use
 - Next 32 bits of the prefix are the 32 bits of the IPv4 address of the 6to4 router
 - For example, a 6to4 router on 192.0.1.1 would use an IPv6 prefix of 2002:c000:0101::/48 for its site network
- When a 6to4 router sees a packet with destination prefix 2002::/16, it knows to tunnel the packet in IPv4 towards the IPv4 address indicated in the next 32 bits
 - Hence the automatic tunnelling

6to4 basic overview



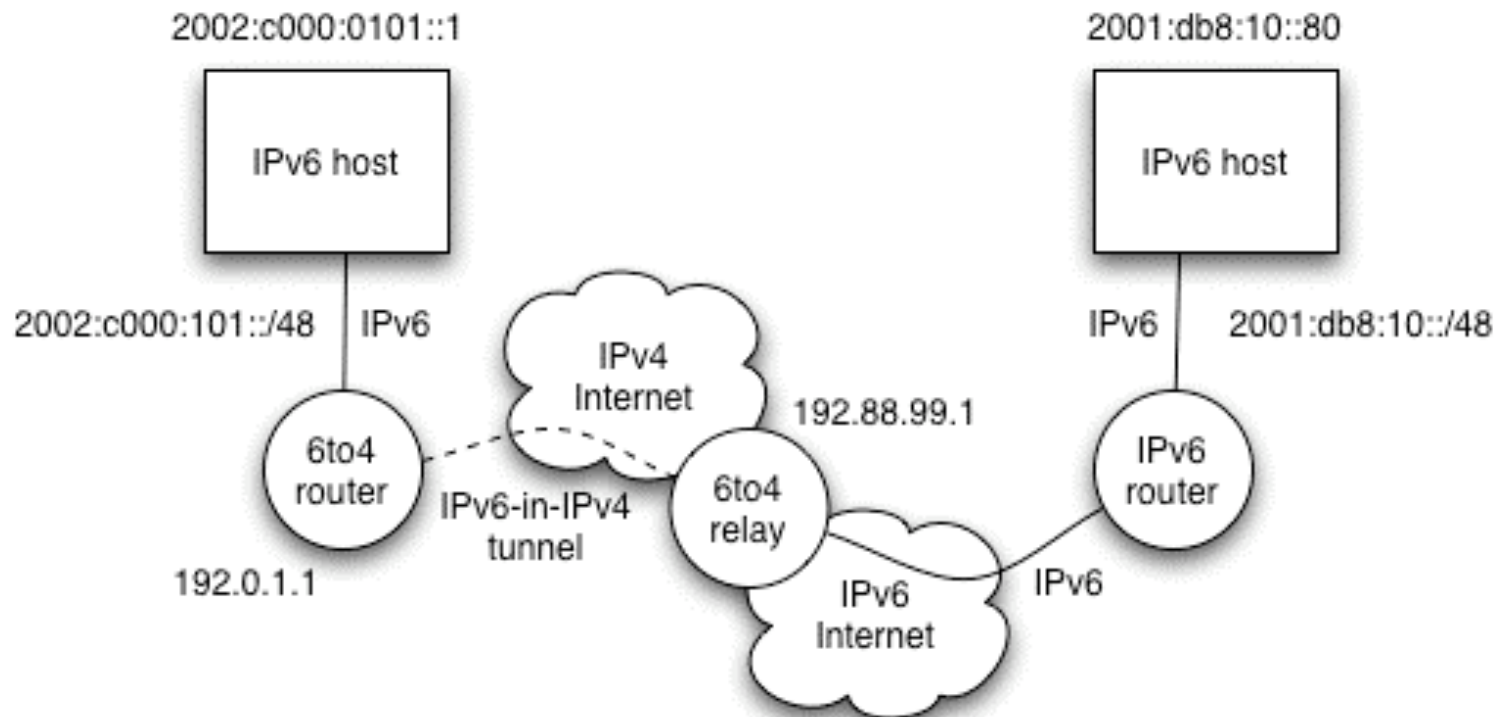
6to4 features

- On the plus side:
 - Simple to deploy and use
 - Fully automatic; no administrator effort per tunnel
 - Tunnelled packets automatically route efficiently to the destination network (following the best IPv4 path over the IPv4 Internet)
- But there's an important capability missing:
 - How does a node on a 6to4 site communicate with an IPv6 node on a regular, 'real' IPv6 site?
 - Without requiring all IPv6 sites to support 6to4
- Also, 6to4 relays can be abused (DoS attacks)
 - See RFC3964 for appropriate checks to deploy

6to4 relay

- A 6to4 relay has a 6to4 interface and a 'real' IPv6 interface
- Two cases to consider:
 - IPv6 packets sent from a 6to4 site to a destination address outside 2002::/16 are tunnelled using 6to4 to the relay, are decapsulated, and then forwarded on the relay's 'real' IPv6 interface to the destination site
 - The 6to4 relay is advertised on a well-known IPv4 anycast address 192.88.99.1.
 - IPv6 packets sent from a 'real' IPv6 site towards an address using the 2002::/16 prefix (a 6to4 site) are routed to the 6to4 relay and then tunnelled using 6to4 to the destination 6to4 site
 - The relay advertises 2002::/16 to connected IPv6 neighbours

6to4 with relay



Translation

- Translation is necessary for IPv6-only and IPv4-only hosts to communicate, which should be done near the edges
- Can be done at different communication layers:
 - NAT-PT – not recommended
 - Packet level, much the same as regular NAT
 - TCP-relay
 - Session level, e.g. KAME BSD Faith daemon, SOCKS64
 - Application level gateways – best solution
 - 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

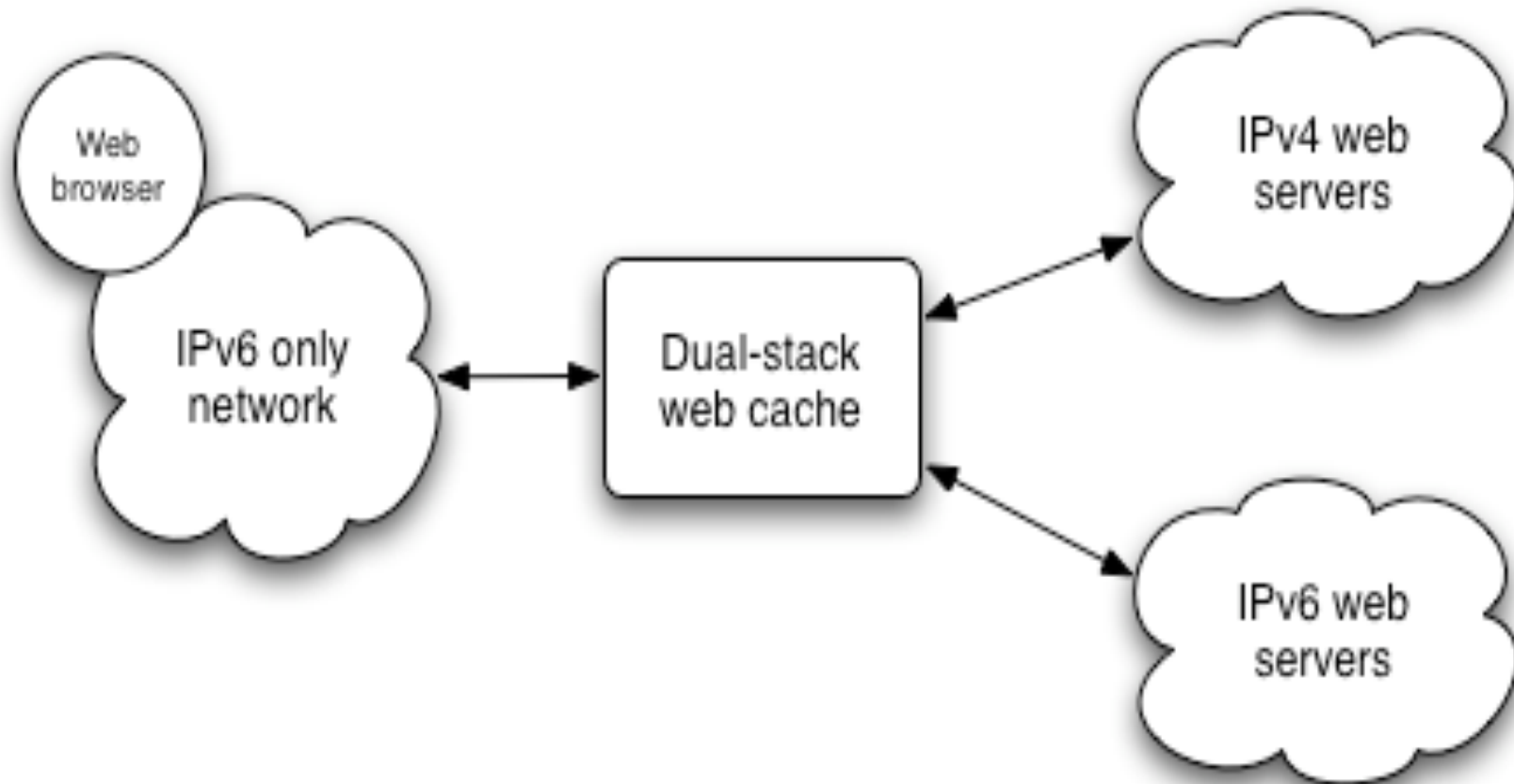
NAT-PT (RFC2766)

- Like IPv4 NAT, but also with protocol translation
 - Maintains pool of IPv4 addresses to map to
 - Performs IP header translation (SIIT, RFC2765)
 - But not all headers are translatable, e.g. IPv6 flow label
 - Has same problems as NAT if an upper layer protocol embeds IP addresses in it's PDUs, e.g. FTP
 - Likewise includes a DNS ALG (application layer gateway) for DNS interworking
 - (IPv4 has A records, IPv6 has AAAA records)
 - Being deprecated within the IETF to Experimental status
 - *draft-ietf-v6ops-natpt-to-exprmntl-03*

Application layer gateways

- ALG is an IP device running dual-stack
 - Can access both IPv4 and IPv6 services natively
- Uses “natural” proxy-style services
 - Web cache (e.g. squid)
 - SMTP gateway (e.g. sendmail as a relay)
 - Usenet news server
 - IRC server
 - H.323 proxy
- But doesn't handle all services, particularly those that have pure end-to-end requirements

ALG topology



ALG pros and cons

- Pros
 - Simple to deploy
 - ALGs already commonly in use, e.g.
 - Web cache to reduce bandwidth usage
 - SMTP relay to channel mail through one server
 - Avoids complexity of NAT-PT or TRT (transport layer relay)
- Cons
 - Requires client configuration to use ALG
 - Only works for specific ALG-supported applications

But what's the best method?

- We have a “toolbox” of IPv6 transition methods
- Some suited to certain scenarios
- IPv4 hosts will be around for a long time, with transition ongoing for many years (10-20+ years)
- Usage depends on scenario
 - A university may run dual-stack internally, and use a manual tunnel to the JANET Experimental Service IPv6 router externally
 - A home user with IPv6 enabled on their laptop may use a tunnel broker to gain IPv6 connectivity to their home
 - Alternatively, a SOHO environment may be suited to 6to4
 - Especially where a static IPv4 address is available
- There is no single ‘best’ solution

Finally: perspectives

- Potentially deployed by a (campus) site:
 - Manual tunnels
 - Dual-stack networking
 - ALGs
 - 6to4 router (for small, typically SOHO, sites)
 - NAT-PT (for IPv6-only subnets without ALG capability)
- Potentially offered/supported by an ISP/RNO:
 - Tunnel broker server (UKERNA is offering a pilot broker, see www.broker.ipv6.ac.uk)
 - Manual tunnels
 - 6to4 relay (JANET has a 6to4 relay deployed)