IPv6 Transition and Coexistence with IPv4

Agenda

- Approaches to deploying IPv6
  - Standalone (IPv6-only) or alongside IPv4
- Considerations for IPv4 and IPv6 coexistence
- Approaches to coexistence
  - 1: Tunnelling
  - 2: Translation
  - 3: Dual-stack
- Specific examples
  - 6to4
  - Tunnel broker
  - ISATAP

Deploy IPv6 standalone

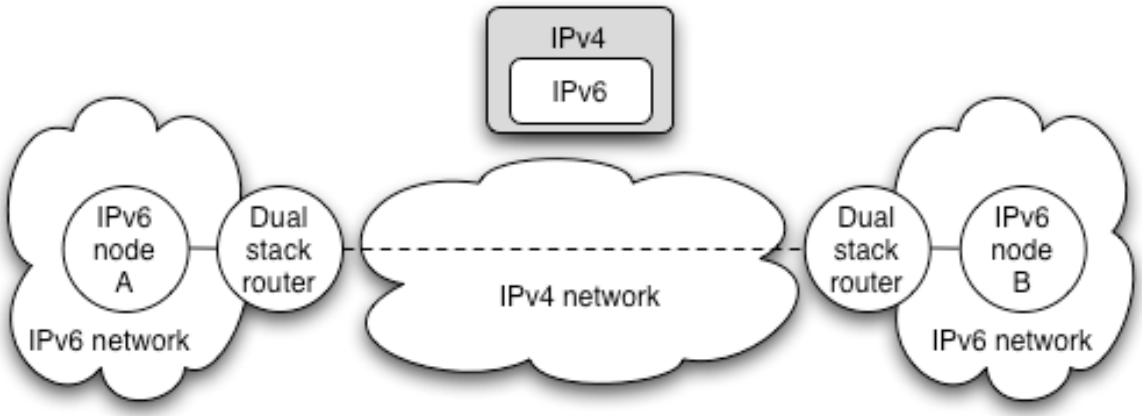
- One option is to deploy an IPv6-only network
- Introduces specific requirements:
  - All components must be IPv6-capable
  - Likely to need to talk to IPv4-only systems
    - So need some way to ‘translate’ between the protocols at some layer
  - Likely to want to communicate with remote IPv6 network ‘islands’ that may only be connected through existing IPv4 networks
    - Need a way to send IPv6 packets over/through an intermediate IPv4-only network

Deploy IPv6 alongside IPv4

- Existing network runs IPv4
- Introduce IPv6 to the same network
- Deploy IPv6 in parallel to IPv4
  - Known as ‘dual-stack’ operation
  - Hosts and routers are able to talk using either protocol
- Choice of protocol is application-specific
  - DNS returns IPv4 and IPv6 addresses for a given hostname
  - As an example, MS Internet Explorer by default prefers IPv6 connectivity, but can fall back to IPv4 (after a timeout)
  - Thus need to be confident IPv6 connectivity is good, else the application may perform worse than in an IPv4-only network

1: Tunnelling

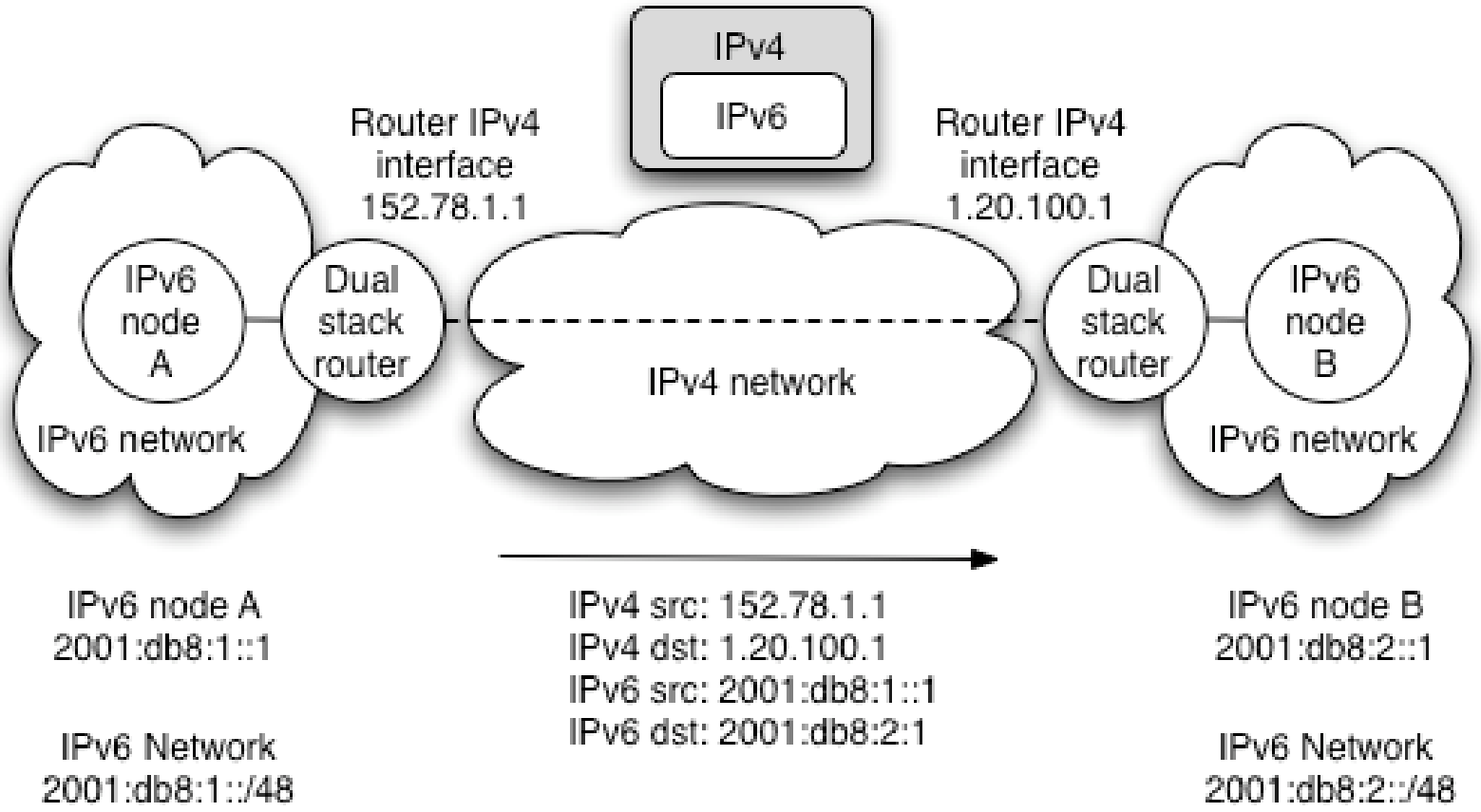
- IPv6 packets encapsulated in IPv4 packets
  - IPv6 packet is payload of IPv4 packet
- Usually used between edge routers to connect IPv6 'islands'
  - Edge router talks IPv6 to internal systems
  - Encapsulates IPv6 in IPv4 towards remote tunnel endpoint



Packet delivery over the tunnel

- IPv6 node A sends packet to IPv6 node B
  - Routed internally to edge router A
- Edge router A sees destination network B is reachable over tunnel interface
  - Encapsulates IPv6 packet in IPv4 packet(s)
  - Sends resulting IPv4 packet(s) to edge router B
  - Delivered over existing IPv4 Internet infrastructure
- Edge router B decapsulates IPv6 packet from payload of received IPv4 packet
  - Packet routed internally in network B to node B
  - Node B receives the IPv6 packet

Tunnel addressing view



Fragmentation

- IPv6 requires that packet fragmentation only occurs at end systems, not on intermediate routers
  - Use Path Maximum Transmission Unit (PMTU) Discovery to choose the MTU
  - Achieved using special ICMP messages
  - Minimum MTU is 1280 bytes in IPv6
- When tunnelling IPv6 in IPv4, the IPv4 packets may be fragmented
  - Depends on the IPv4 packet size
  - Additional IPv6 headers (e.g. Authentication Header) will affect this

Tunnel solution considerations

- These include:
  - Security
  - Manual or automatic setup
  - Ease of management
  - Handling dynamic IPv4 addresses
  - Support for hosts or sites to be connected
  - Scalability: 10, 100, or 10,000 served tunnels?
  - Support for NAT traversal
  - Tunnel service discovery
  - Support for special services (e.g. multicast)
  - Tunnel concentration/bandwidth usage issues
- We'll come back to these later…

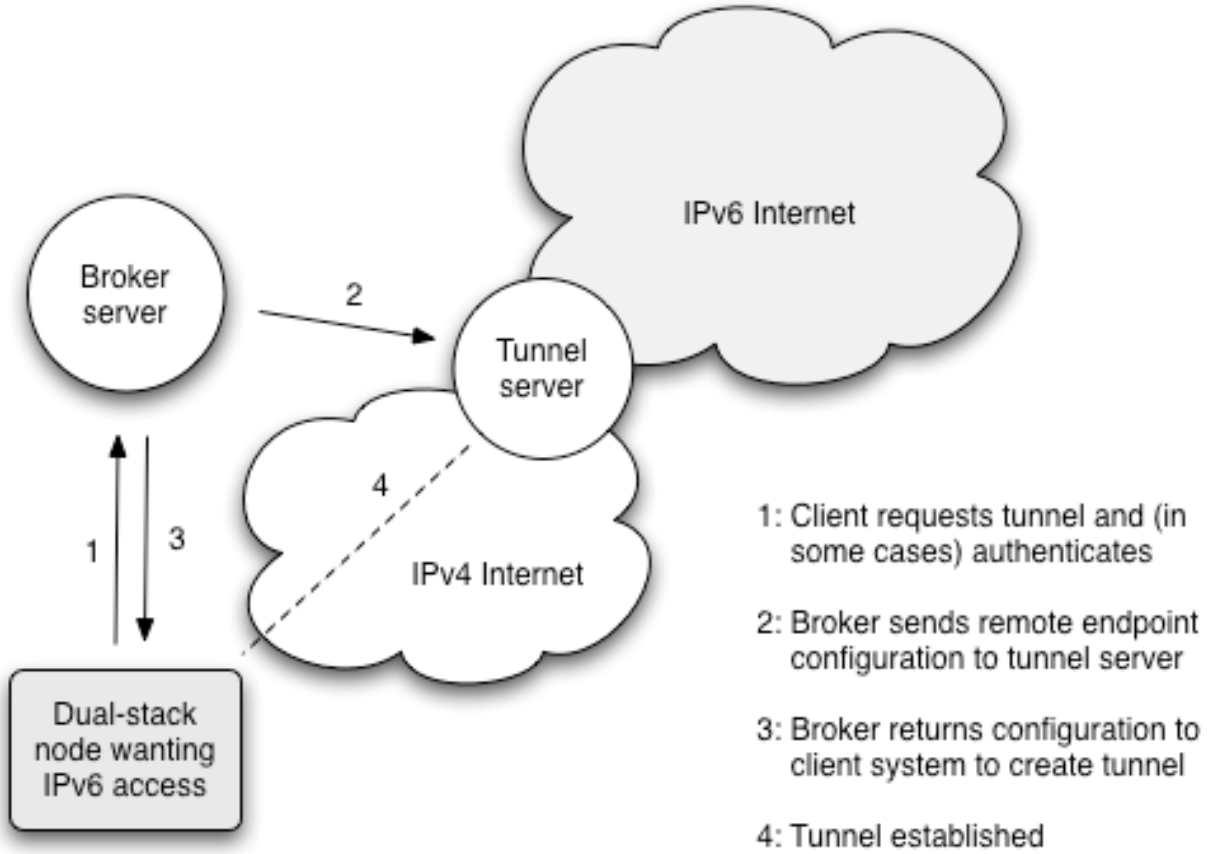
Manually configured tunnels

- 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 and Renater) to connect academic sites/users over IPv6 where native IPv6 connectivity is not available
- 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.

Tunnel broker

- Very popular in IPv6 user community
- Most well-known broker is www.freenet6.net
  - Hosted in Canada by Hexago
- General mode of operation is:
  - User/client registers with the broker system
  - A tunnel is requested from a certain IPv4 address
  - The broker sets up its end of the requested tunnel on its tunnel server
  - The broker communicates the tunnel settings to the user, for client-side configuration
- Can traverse a NAT, e.g. if UDP tunnelling used

Broker architecture



Broker issues

- Broker's key advantage is its manageability
  - ISP can track usage levels
- A few downsides:
  - If broker is topologically remote, round trip times for data may suffer
    - e.g. using freenet6 in Canada to reach UK sites
  - Not well-suited if IPv4 address is dynamic
    - Common problem in home DSL networks
  - If using a remote tunnel broker, your own ISP may not perceive a demand for IPv6

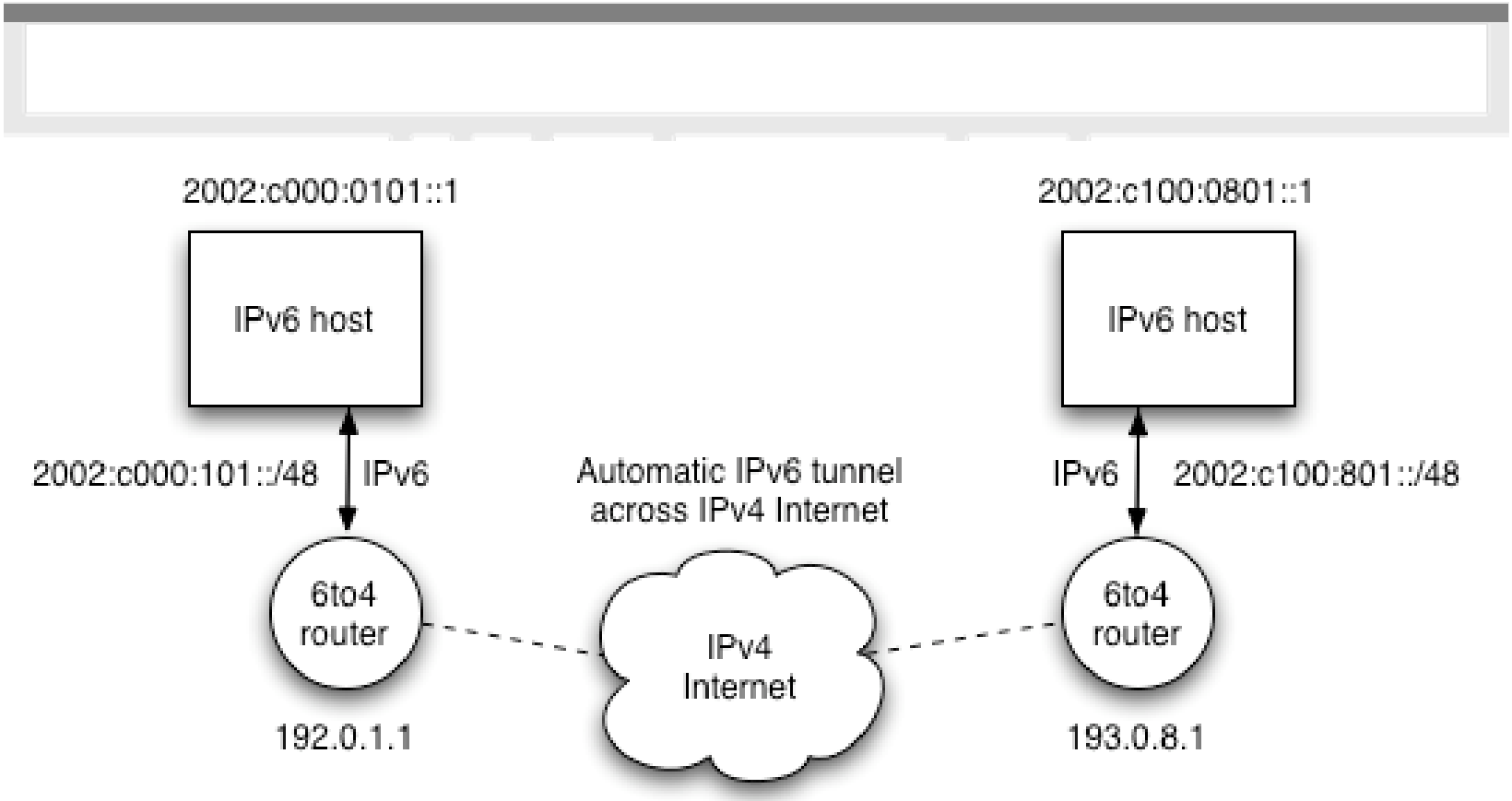
Automatic tunnelling

- Goal is to avoid requiring support staff effort to setup and maintain tunnels
- Set up required tunnels on demand
- Make deployment and usage simple(r) for the end user
- Most common automatic method is 6to4 (RFC3056)
  - Generally used router-to-router
  - Well supported in commercial routing platforms
- Other methods include ISATAP (RFC4214) and Teredo
  - We don't cover Teredo (RFC4380) here; it is a NAT-traversing IPv6 connectivity method used by Microsoft in XP/Vista.

6to4

- In its basic configuration, 6to4 is used to connect two IPv6 islands across an IPv4 network
- Uses special ‘trick’ for the 2002::/16 IPv6 prefix that is reserved for 6to4 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

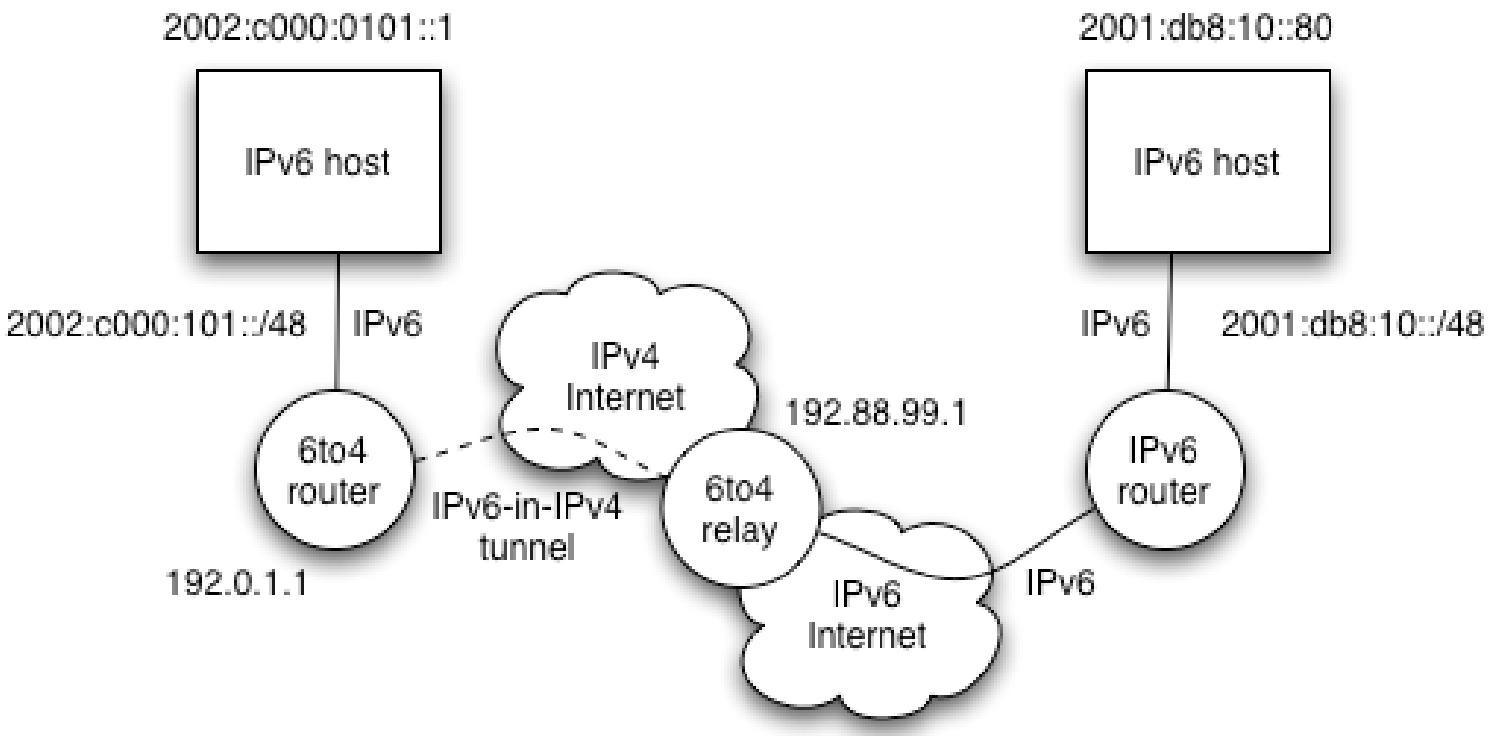
6to4 basic overview



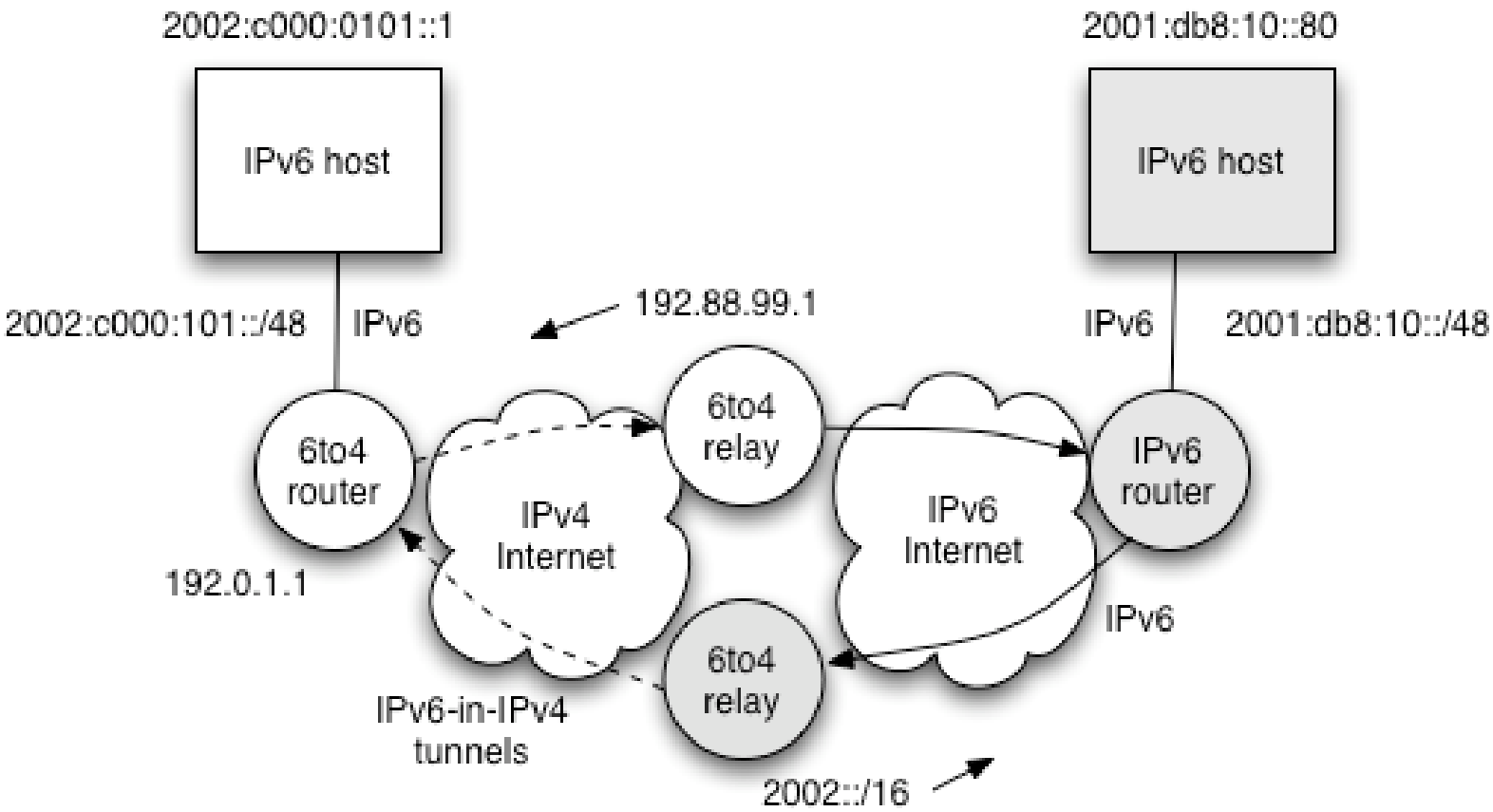
6to4 relay

- 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
  - 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

6to4 with relay



Asymmetric 6to4



6to4 and broker features

Feature	6to4	Tunnel broker
Security	Potential for abuse	Supports authentication
Setup	Automatic	Manual
Ease of management	Poor (automatic)	Good
Dynamic IPv4 addresses	Poor	Poor
Host or site tunnels	Primarily site	Primarily host
Scalability	Very good	Good
NAT traversal	Tricky	Yes, with TSP
Tunnel service discovery	Automatic	Manual configuration
Special service support	Variable	Variable
Bandwidth concentration	Only at 6to4 relay	At tunnel server

2: Translation

- When an IPv4-only system needs to communicate with an IPv6-only system some form of translation is required
- Can be done at various layers
- Network layer
  - Rewrite IP headers
- Transport layer
  - Use a TCP relay
- Application layer
  - Use an application layer gateway (ALG)
- Ideally avoid translation
  - Use IPv4 to speak to IPv4 systems and IPv6 for IPv6 systems

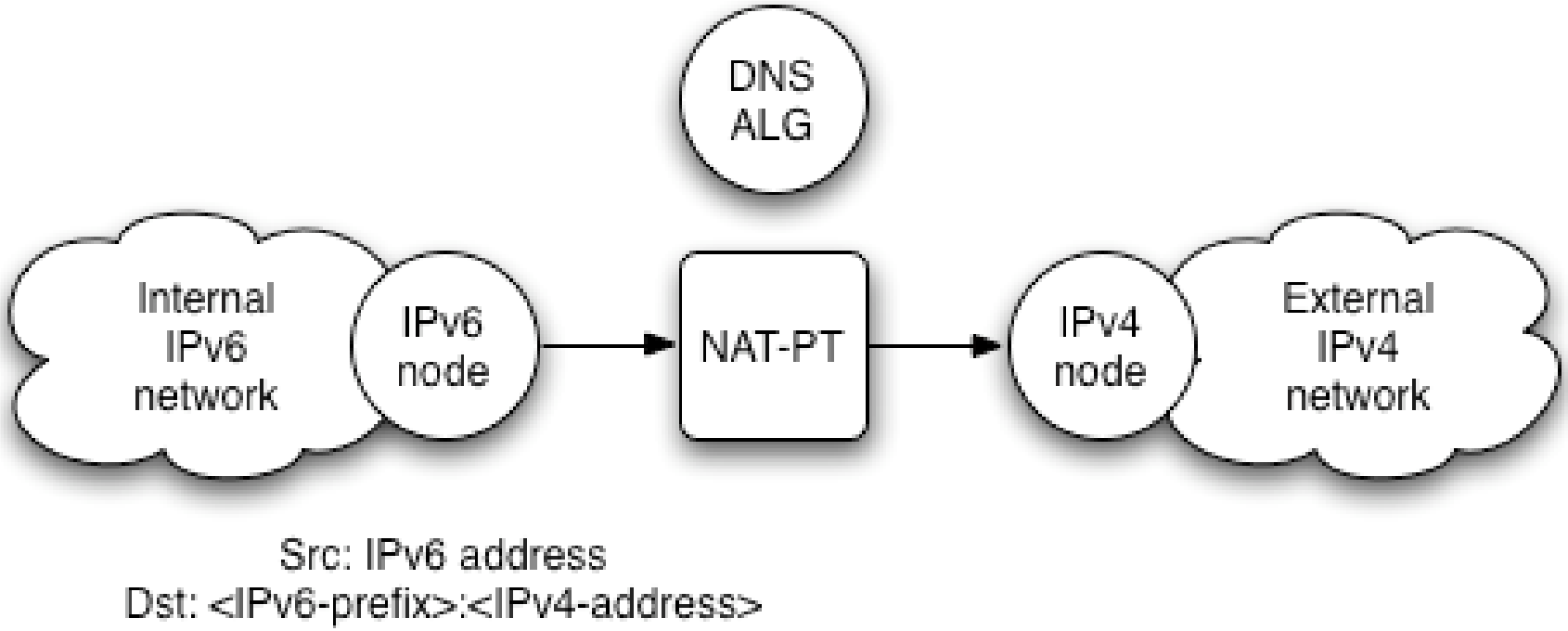
Translation scenarios

- Generally when deploying IPv6-only network elements and you need them to communicate with IPv4-only systems
  - Legacy applications that cannot be ported to support IPv6
    - Or perhaps source code not available
  - Legacy IPv4-only operating systems
    - For example Windows 98
  - Legacy IPv4-only hardware
    - Printers

Network layer: NAT-PT

- Network Address Translation - Protocol Translation
  - Defined in RFC2766
  - Like IPv4 NAT, but with protocol translation
- Uses Stateless IP/ICMP Translation (SIIT)
  - Defined in RFC2765
  - SIIT defines algorithms to translate between the IPv4 and IPv6 header fields, where possible to do so
- NAT-PT extends SIIT by using IPv4 address pools
  - IPv4-to-IPv6 and IPv6-to-IPv4 supported

NAT-PT topology



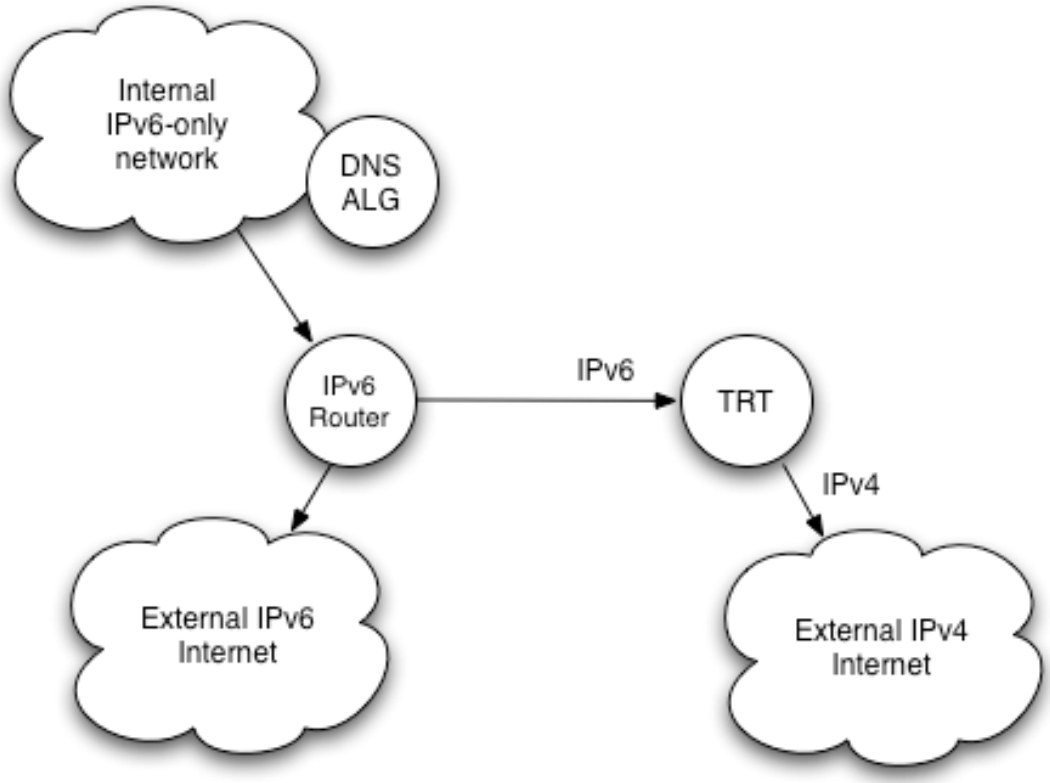
NAT-PT and DNS

- Internal network IPv6 only
- DNS ALG watches for IPv6 (AAAA) DNS queries outbound, and translates to IPv4 (A) queries
- When IPv4 DNS response comes back, DNS ALG maps the result to an IPv6 address
  - <IPv6-prefix>:<IPv4 address>
  - A special NAT-PT IPv6 prefix is taken from the IPv6 network's address space
- Querying host can now use an IPv6 destination that NAT-PT can map to the real IPv4 destination

Transport layer: TRT

- Transport Relay Translator (TRT)
  - Designed for use in IPv6-only networks wishing to connect to external IPv4-only systems
  - TRT has internal IPv6 and external IPv4 interfaces
- External IPv6 connections work as usual
- Trick is handling connections to IPv4 networks
  - Relies on use of a DNS proxy
  - Internal IPv6 host looks up IP address of destination
  - If an IPv6 address, traffic is sent out to IPv6 Internet
  - If an IPv4 address, traffic needs to route to the TRT

TRT topology



DNS proxy address mapping

- If internal IPv6 host is trying to reach an IPv4-only system, the DNS proxy (ALG) returns a special IPv6 destination
  - First 64 bits assigned to be unique locally
  - Next 32 bits all zero
  - Last 32 bits are the real IPv4 destination
    - <IPv6-prefix>:0:0:<IPv4 address>
- The <ipv6-prefix> is routed internally to the TRT
  - Which terminates the TCP/IPv6 connection
  - And opens a connection to the real IPv4 destination

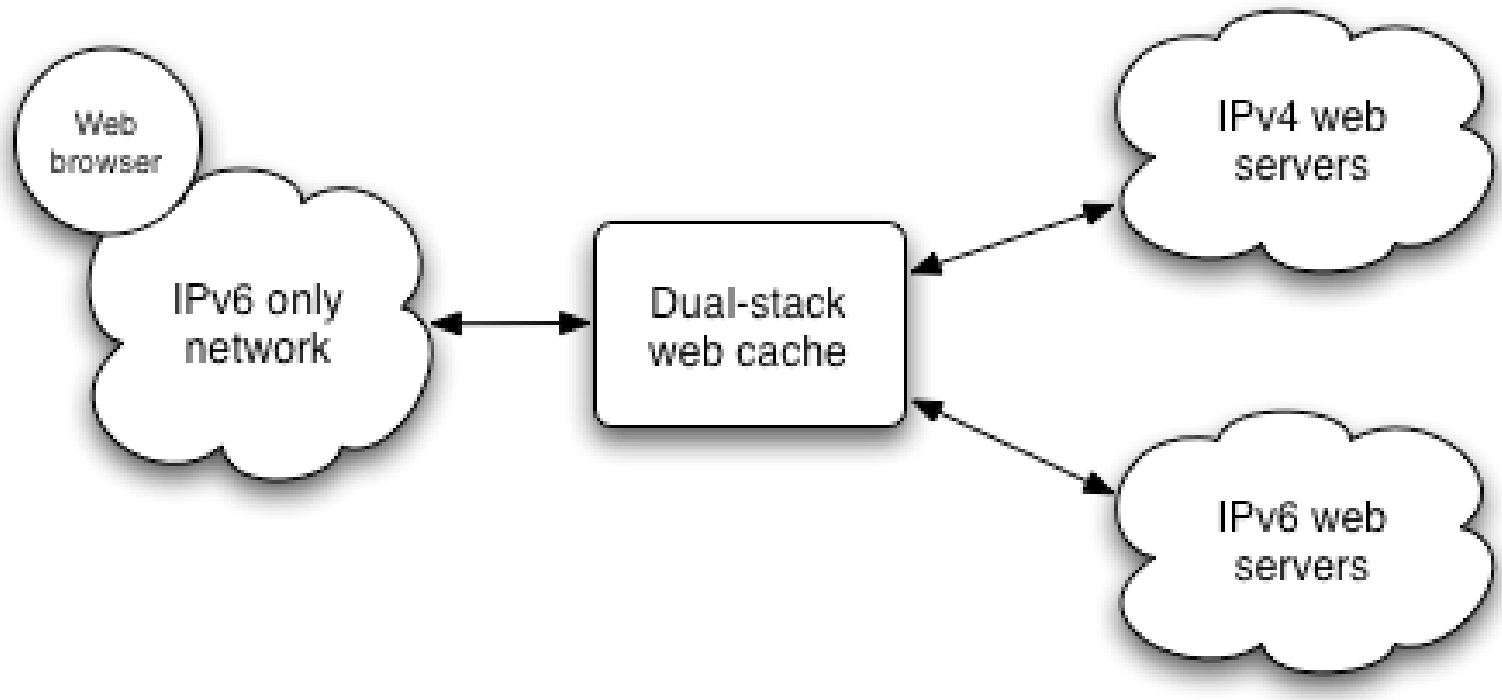
TRT pros and cons

- Pros
  - Transparent to hosts/applications
  - Scalable - can use multiple TRTs, with one internal /64 prefix used per TRT device
  - TRT can work with one global IPv4 address
- Cons
  - Like NAT, problems with embedded IP addresses in payload (e.g. FTP)
  - No simple way to allow connections initiated inbound from external IPv4 to internal IPv6 hosts

Application: ALGs

- NAT-PT and TRT are somewhat complex
- Luckily, application layer gateways (ALGs) offer a simpler alternative
- Many applications support ALGs already
  - Web cache
  - SMTP gateway
  - DNS resolver
  - SIP proxy
  - etc
- We can leverage this in a simple way

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
- Cons
  - Requires client configuration to use ALG
  - Only works for specific ALG-supported applications

3: Dual-stack

- Support both protocols on 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

Dual-stack issues

- Application must choose which IP protocol to use
  - Given DNS returns IPv4 and IPv6 addresses
  - 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
- Security should be no worse
  - Hosts listen on both protocols; secure both

Conclusions

- There is a large set of IPv6 transition tools available
  - No single ‘best’ solution
  - Transition plan is likely to be site-specific
- Current ‘best practice’ is dual-stack deployment
  - Natural path via procurement cycles
  - Allows experience in IPv6 operation to be gained early
- IPv6-only networks can be deployed
  - But very limited in number to date, and missing some apps
- Ultimate driver is IPv4 address space availability
  - But also need IPv4 addresses for a smooth transition