IPv6 Deployment Study

AfNOG 2011 AR-E Workshop

Notes

- This presentation is still under development
  - I started writing it in 2006 as ISPs started to deploy IPv6
  - Apologies for the holes and blanks
  - Content being gathered as experiences are being gained, related to me, etc
  - Feedback welcome…

  - Philip Smith

Agenda

- Network Audit
- Network Optimisation
- Procuring IPv6 Address Space
- IPv6 Address plan
- Deploying Addressing & IGP
- Deploying iBGP
- Seeking IPv6 Transit
- Forward and Reverse DNS
- Services & Customers

Network Audit

What can run IPv6 today, and what needs to be upgraded?

Audit

- First step in any deployment:
  - Audit existing network infrastructure
- Primarily routers across backbone
  - Perhaps also critical servers and services (but not essential as initial focus is on routing infrastructure)

Process

- Analyse each PoP
- Document
  - Router platform
  - RAM (installed and used)
  - FLASH memory
  - IOS release versions
  - RANCID (www.shrubbery.net/rancid/) makes this very easy
- Sanity check
  - Check existing connectivity
  - Remove unused configuration
  - Shutdown and clean up unused interfaces

Software Issues

- Software images:
  - Need “AdvancedIPServices” or “IP Plus” images to support IPv6
- 12.3 Cisco IOS has limitations on some platforms:
  - 2600 (non XM) and 3620 have no OSPFv3
  - 2500 needs 16M RAM and 16M FLASH but has no SSH/crypto support
- 12.4 Cisco IOS generally fine, but older platforms not supported

Next Steps

- Upgrade RAM and FLASH for platforms identified as being deficient
- Replace routers which can not run most recent Cisco IOS software (12.2S, 12.3, 12.4 & 15.0)
  - This will impact 2600 (non-XM), 3620, elderly 7200s (pre NPE200), &c
- Decide on a software strategy
  - Mix of 12.3 and 12.4
  - 12.4 everywhere (bigger impact as some platforms which support 12.3 aren't supported for 12.4 - e.g. 2500)

Software Strategy

- GSRs
  - 12.0S supports IPv6
  - Or use IOS-XR
- 6500 and 7600
  - 12.2SXI & 12.2SRE support IPv6 – no work should be required
  - But unless Sup720 3BXL or later is used, FIB sizes must be watched
- Remaining platforms recommendation:
  - Run 12.4 if supported
  - Otherwise run 12.3 (legacy platforms) and make a note to replace/upgrade in next refresh cycle

Result

- Once the previous steps are completed, entire network is running IPv6 capable software
- Deployment of IPv6 can now begin

Network Optimisation

Is the IPv4 network the best it can be?

Optimisation

- IPv4 networks have been deployed and operational for many years
  - Your network may fall into this category
- Optimisation means:
  - Does the iBGP design make sense?
  - Are the OSPF areas in the right places?
  - Does the ISIS backbone make sense?
  - Do all routing protocols have the latest best practices implemented?
  - Are the IGP metrics set so that primary and backup paths operate as expected?

Motivation for Optimisation

- IPv6 deployment will be dual stack
  - So sitting alongside existing IPv4 configurations
- Aim is to avoid replicating IPv4 "shortcuts" or "mistakes" when deploying IPv6
  - IPv6 configuration will **replicate** existing IPv4 configuration
- Improvements in routing protocol BCPs should be deployed and tested for IPv4
  - Take the opportunity to "modernise" the network

iBGP considerations

- Full mesh iBGP still?
  - Perhaps consider migration to route reflectors
- Route reflector configuration
  - Proper redundancy in place?
  - Overlapping clusters, one reflector per cluster
  - Direct path between client and reflector
- BGP best practices deployed
  - Peer-group strategy? (Will have to be replicated for IPv6)
  - Full routes in core iBGP?
  - Partial routes in edge/rr client iBGP
  - Community strategy for internal and external announcements?

OSPF considerations

- IOS 12.4 OSPFv2 supports same CLI as OSPFv3
  - **`network x.x.x.x 0.0.0.m area A`** command syntax is replaced by configuring OSPF on the actual interface
  - As for OSPFv3 (and ISIS)
  - Convert OSPFv2 to modern CLI – then easy to replicate configuration for OSPFv3
- Are the OSPF areas configured as intended?
  - Contiguous area 0, with redundant links?
- Are the interface metrics configured as intended?
  - Easy to miss bits of configuration
  - They will be replicated in IPv6 (unless the intention is to have different traffic flow patterns from IPv4)

ISIS considerations

- This is a good time to check NSAP numbering plan
- Need to deploy wide metrics
  - Multi-topology ISIS requires the use of wide metrics
  - (Narrow metrics don't scale for modern networks anyway!)
- Are the interface metrics configured as intended
  - Easy to miss bits of configuration
  - They will be replicated in IPv6 (unless the intention is to have different traffic flow patterns from IPv4)

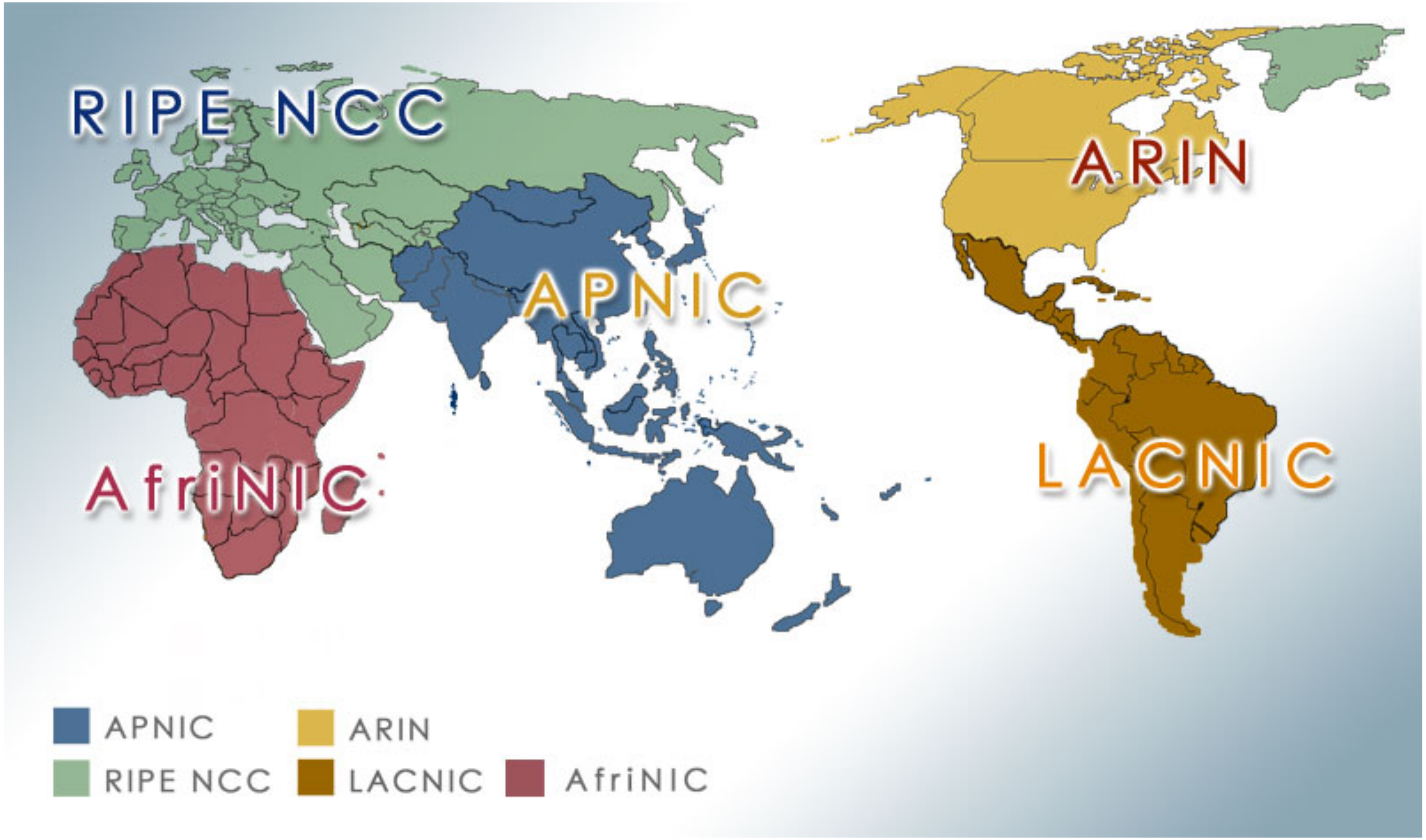
Procuring IPv6 address space

Now we need addresses…

Where to get IPv6 addresses

- The Regional Internet Registries:
  - Africa
    - AfriNIC – http://www.afrinic.net
  - Asia and the Pacific
    - APNIC – http://www.apnic.net
  - North America
    - ARIN – http://www.arin.net
  - Latin America and the Caribbean
    - LACNIC – http://www.lacnic.net
  - Europe and Middle East
    - RIPE NCC – http://www.ripe.net/info/ncc
- From your upstream ISP

Internet Registry Regions



Getting IPv6 address space (RIR)

- If existing Regional Internet Registry account holder with an IPv4 allocation:
  - Just ask for an IPv6 allocation and it will be given – it really is as simple as that!
- Become an account holder of your Regional Internet Registry and get your own IPv6 allocation
  - Requires a plan for a year ahead
  - IPv6 allocation policies are documented on each RIR website
  - The following slides describe considerations when constructing such a plan
- Note Well: There is plenty of IPv6 address space
  - The RIRs require high quality documentation

Getting IPv6 address space (non-RIR)

- From your upstream ISP
  - Get one /48 from your upstream ISP
  - More than one /48 if you have more than 65k subnets
- Use 6to4
  - Take a single public IPv4 /32 address
  - 2002:<ipv4 /32 address>::/48 becomes your IPv6 address block, giving 65k subnets
  - Requires a 6to4 gateway
- These two options are NOT viable for service providers though – a /32 from an RIR is the only way

Addressing Plans – ISP Infrastructure

- Address block for router loop-back interfaces
  - Number all loopbacks out of **one** /64
  - /128 per loopback
- Address block for infrastructure
  - /48 allows 65k subnets
  - /48 per region (for the largest international networks)
  - /48 for whole backbone (for small to large networks)
  - Summarise between sites if it makes sense

Addressing Plans – ISP Infrastructure

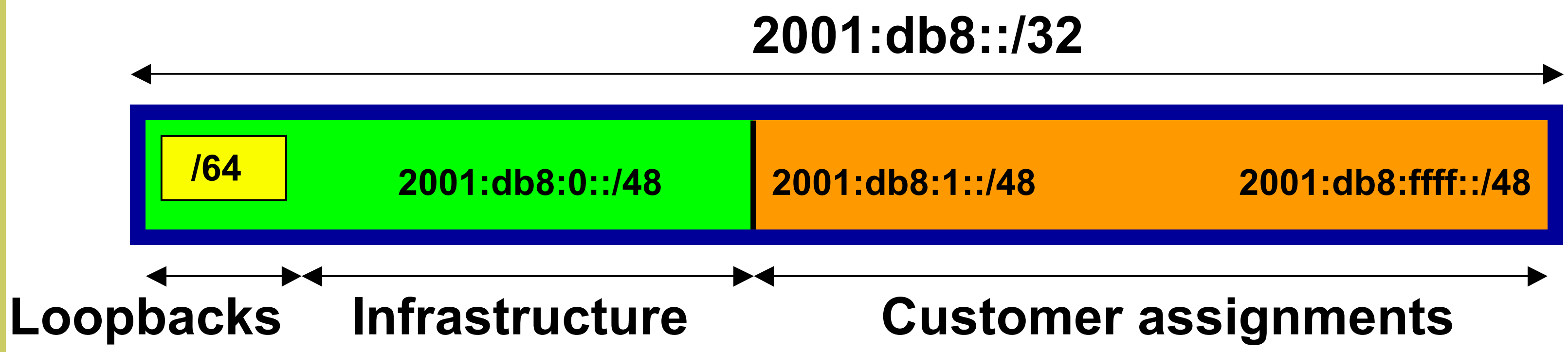
- What about LANs?
  - /64 per LAN
- What about Point-to-Point links?
  - Expectation is that /64 is used
  - /126s and /127s are being used
    - Mobile IPv6 Home Agent discovery won't work
  - /112s are being used
    - Leaves final 16 bits free for node IDs
  - Some ISPs are considering /80s or /96s
  - RFC3627 and datatracker.ietf.org/doc/draft-ietf-6man-prefixlen-p2p provide interesting perspectives

Addressing Plans – Customer

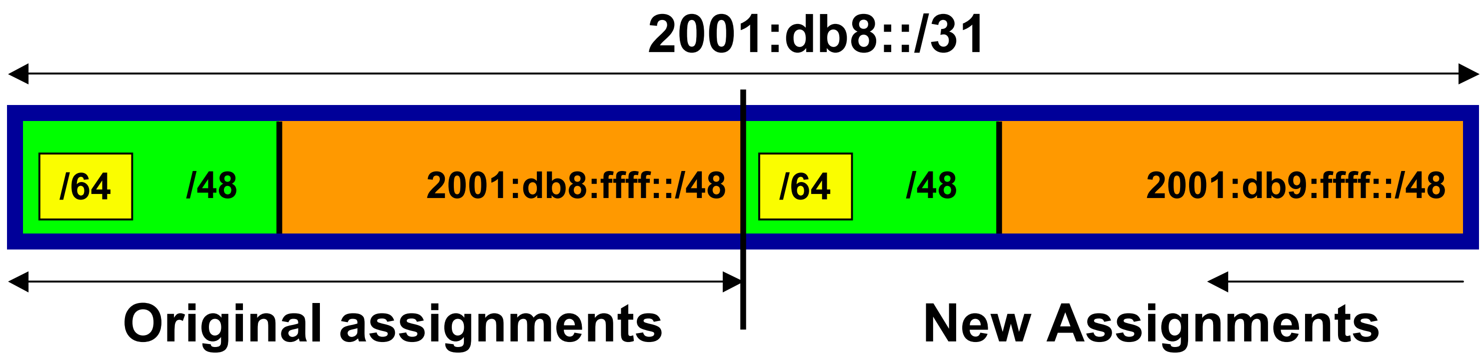
- Customers get **one** /48
  - Unless they have more than 65k subnets in which case they get a second /48 (and so on)
  - See later for further discussion about customer addressing
- Should not be reserved or assigned on a per PoP basis
  - ISP iBGP carries customer nets
  - Aggregation within the iBGP not required and usually not desirable
  - Aggregation in eBGP is very necessary

Addressing Plans – ISP Infrastructure

- Phase One



- Phase Two - second /32



Addressing Plans

- Registries will usually allocate the next block to be contiguous with the first allocation
  - Minimum allocation is /32
  - Very likely that subsequent allocation will make this up to a /31
  - So plan accordingly

Addressing Plans (contd)

- Document infrastructure allocation
  - Eases operation, debugging and management
  - Makes IPv6 DNS easier to operate
- Document customer allocation
  - Customers get /48 each (see later)
  - Prefix contained in iBGP
  - Eases operation, debugging and management
  - Submit network object to RIR Database

Addressing Tools

- Examples of IP address tools (which support IPv6 too):
  - IPAT nethead.de/index.php/ipat
  - ipv6gen techie.devnull.cz/ipv6/ipv6gen/
  - sipcalc www.routemeister.net/projects/sipcalc/
  - freeipdb home.globalcrossing.net/~freeipdb/

Constructing a Deployable Addressing Plan

We have got the address space, what next…

Deployable Address Plan

- Documentation
  - IPv4 addresses are probably short enough to memorise
  - IPv6 addresses are unlikely to be memorable at all
- Document the address plan
  - What is used for infrastructure
  - What goes to customers
  - Flat file, spreadsheet, database, etc
  - But documentation is vital
  - Especially when coming to populating the DNS later on

Deployable Address Plan

- Pick the first /48 for our ISP infrastructure
  - Reason: keeps the numbers short
  - Short numbers: less chance of transcription errors
  - Compare:

    2001:db8:ef01:d35c::1/128

      with

    2001:db8::1/128

    For Loopback interface addresses
- Out of this /48, pick the first /64 for loopbacks
  - Reason: keeps the numbers short

Deployable Address Plan

- For the infrastructure /48:
  - First /64 for loopbacks
  - Remaining 65535 /64s used for internal point to point links
- Second /48:
  - Use for point to point links to customers
  - Unless you use unnumbered interfaces
  - That gives 65536 /64s for 65536 customer links
- Remaining /48s are for delegation to customers

Example: Loopback addresses

- 2001:db8:0::/48 is used for infrastructure
- Out of this, 2001:db8:0:0::/64 is used for loopbacks
- ISP has 20 PoPs around the country
  - Scheme adopted is:
    - 2001:db8::X:Y/128
  - Where X is the PoP number (1 through FFFF)
  - Where Y is the router number (1 through FFFF)
  - Scheme is good for 65535 PoPs and 65535 routers per PoP, and keeps addresses small/short

Example

- Loopbacks in PoP 1:

Cr1	2001:db8::1:1/128
Cr2	2001:db8::1:2/128
Br1	2001:db8::1:3/128
Br2	2001:db8::1:4/128
Gw1	2001:db8::1:10/128
Gw2	2001:db8::1:11/128
Gw3	2001:db8::1:12/128
Gw3	2001:db8::1:13/128

…etc…

Loopbacks in PoP 10:

Cr1	2001:db8::a:1/128
Cr2	2001:db8::a:2/128
Br1	2001:db8::a:3/128
Br2	2001:db8::a:4/128
Gw1	2001:db8::a:10/128
Gw2	2001:db8::a:11/128
Gw3	2001:db8::a:12/128
Gw4	2001:db8::a:13/128

…etc…

Example: Backbone Point to Point links

- ISP has 20 PoPs around the country
  - Scheme adopted is:
    - 2001:db8:0:MNXY::Z/64
  - Where
    - MN is the PoP number (01 through FF)
    - XY is the LAN number (00 through 0F)
    - XY is the P2P link number (10 through FF)
    - Z is the interface address (1 or 2)
  - Scheme is good for 16 LANs and 240 backbone PtP links per PoP, and for 255 PoPs

Example

- PtP & LANs in PoP 1:

LAN1	2001:db8:0:100::/64
LAN2	2001:db8:0:101::/64
LAN3	2001:db8:0:102::/64
PtP1	2001:db8:0:110::/64
PtP2	2001:db8:0:111::/64
PtP3	2001:db8:0:112::/64
PtP4	2001:db8:0:113::/64
PtP5	2001:db8:0:114::/64
…etc…	

- PtP & LANs in PoP 14:

LAN1	2001:db8:0:e00::/64
LAN2	2001:db8:0:e01::/64
LAN3	2001:db8:0:e02::/64
LAN4	2001:db8:0:e03::/64
LAN5	2001:db8:0:e04::/64
PtP1	2001:db8:0:e10::/64
PtP2	2001:db8:0:e11::/64
PtP3	2001:db8:0:e12::/64
…etc…	

Links to Customers

- Some ISPs use “ip unnumbered” for IPv4 interface links
  - So replicate this in IPv6 by using “ipv6 unnumbered” to address the links
  - This will not required one /48 to be taken from the ISP’s /32 allocation
- Other ISPs use real routable addresses
  - So set aside the second /48 for this purpose
  - Gives 65536 possible customer links, assuming a /64 for each link

Example

- Customer PtP links
  - Customer1 2001:db8:1:0::/64
  - Customer2 2001:db8:1:1::/64
  - Customer3 2001:db8:1:2::/64
  - Customer4 2001:db8:1:3::/64
  - Customer5a 2001:db8:1:4::/64
  - Customer5b 2001:db8:1:5::/64
  - Customer6 2001:db8:1:6::/64
  - …etc…

Example: Customer Allocations

- Master allocation documentation would look like this:
  - 2001:db8:0::/48 Infrastructure
  - 2001:db8:1::/48 PtP links to customers
  - 2001:db8:2::/48 Customer1
  - 2001:db8:3::/48 Customer2
  - 2001:db8:4::/48 Customer3
  - 2001:db8:5::/48 Customer4

    …
  - 2001:db8:ffff::/48 Customer65534
- Infrastructure and Customer PtP links would be documented separately as earlier

Summary

- First /48 for infrastructure
  - Out of that, first /64 for Loopbacks
- PoP structure within IPv6 addressing is very possible
  - Greater flexibility than with IPv4
  - Possible to come up with a simple memorable scheme
- Documentation vitally important!

Deploying Addressing and IGP

Let's now touch the network…

Deploying addressing and IGP

- Strategy needed:
  - Start at core and work out?
  - Start at edges and work in?
  - Does it matter?
- Only strategy needed:
  - Don't miss out any PoPs
  - Connectivity is by IPv4, so sequence shouldn't matter
  - Starting at core means addressing of point to point links is done from core to edge (many ISPs use strategy of low number towards core, high number towards edge)
  - But it really doesn't matter where you start…

Deploying: Router1 in PoP1

- Start with addressing
  - Address all the PtP links on Router1

```
interface serial 0/0
  ipv6 address 2001:db8:0:110::1/64
interface hssi 1/0
  ipv6 address 2001:db8:0:111::1/64
```

  - Go to the other end of each PtP link and apply the corresponding addressing there also

```
interface serial 2/0/0
  ipv6 address 2001:db8:0:110::2/64
```

...and...

```
interface hssi 3/1
  ipv6 address 2001:db8:0:111::2/64
```

Deploying OSPF

- Configure OSPFv3 on the links that will run OSPF

```
ipv6 router ospf 100
  log adjacency-changes detailed
  passive-interface default
  no passive-interface serial 0/0
  no passive-interface hssi 1/0
interface serial 0/0
  ipv6 ospf 100 area 0
interface hssi 1/0
  ipv6 ospf 100 area 0
```

- No need to do the OSPF on the other end yet
  - Those routers will be done in due course, and saves time jumping back and forth

Deploying ISIS

- Configure ISIS on the links that will run ISIS

```
ip router isis as100
  <existing isis for ipv4 configuration>
  metric-style wide
interface serial 0/0
  ip router isis as100
  ipv6 router isis as100
interface hssi 1/0
  ip router isis as100
  ipv6 router isis as100
```

- Must do ISIS on the other end too
  - Otherwise ISIS adjacency will go down due to address family mismatch

Deploying the IGP

- Repeat this strategy for all remaining routers in the PoP
  - IPv6 addresses are active
  - OSPF/ISIS is ready to run

Deploying on PoP LANs

- LANs need special treatment
  - Even those that are only point to point links
- Issues:
  - ISPs don't want to have Router Advertisements active on network infrastructure LANs
  - Activating IPv6 on a LAN which isn't adequately protected may have security consequences
    - Servers may auto configure IPv6
    - No firewall filtering means no security $\Rightarrow$ compromise

Deploying on PoP LANs

- Example of Point to Point link (12.3 and 12.4):

```
interface GigabitEthernet0/0
  description Crossover Link to CR2
  ipv6 address 2001:db8:0:115::1/64
  ipv6 nd suppress-ra
  ipv6 ospf 100 area 0
```

- Example of local aggregation LAN (12.4T):

```
interface GigabitEthernet0/1
  description Gateway Aggregation LAN
  ipv6 address 2001:db8:0:100::1/64
  ipv6 nd ra suppress
  ipv6 ospf 100 area 0
```

Deploying on LANs

- Example of local services LAN (12.4):

```
interface GigabitEthernet0/1
  description Services LAN
  ipv6 address 2001:db8:0:101::1/64
  ipv6 nd suppress-ra
  ipv6 traffic-filter SERVER-IN in
  ipv6 traffic-filter SERVER-OUT out
```

  - Where the server-in and server-out filters are ipv6 access-lists configured to:
    - Allow minimal access to servers (only ssh for now),   or
    - To match their IPv4 equivalents

Deploying OSPF on LANs

- When implementing OSPF, use the same metrics and configuration as for the IPv4 version of the IGP
  - If OSPFv2 configuration set the two core routers to be Designated and Backup Designated routers, make it the same for IPv6:

```
interface FastEthernet 0/0
 ip ospf priority 10
 ipv6 ospf priority 10
```

  - Any other OSPFv2 metrics should be replicated for OSPFv3:

```
 ip ospf hello-interval 3
 ip ospf dead-interval 15
 ipv6 ospf hello-interval 3
 ipv6 ospf dead-interval 15
```

Deploying ISIS on LANs

- ISIS has concept of DIS only for a LAN
  - Existing IPv4 DIS will be used for IPv6 because topology is congruent

```
interface FastEthernet 0/0
 isis priority 96 level-2
```

  - No changes needed when adding IPv6

Checks

- Before launching into BGP configuration
  - Sanity check the OSPFv3 configuration
- Are all adjacencies active?
  - Each router should have the same number of OSPFv2 and OSPFv3 adjacencies
- Does each interface with an “ip ospf <pid>” configuration have a corresponding “ipv6 ospf <pid>” configuration?
- Have interfaces not being used for OSPFv3 been marked as passive
  - And do they match those marked as passive for OSPFv2?

Checks

- Does the number of entries in the OSPFv3 routing table match the number of entries in the OSPFv2 routing table
  - Compare the number of entries in “sh ip route ospf” and “sh ipv6 route ospf”
  - Examine differences and work out the reason why
- Do IPv4 and IPv6 traceroutes through the network
  - Are the paths the same?
  - Are the RTTs the same?
  - Discrepancies must be investigated and fixed

Deploying iBGP

Functioning IGP means all routers reachable…

Deploying iBGP

- Strategy is required here
  - Starting at edge makes little sense
  - Starting at core means route reflector mesh builds naturally
- Modify BGP defaults
- Prepare templates
  - Set up peer-groups in master configuration file
  - There should already be a master configuration for IPv4

Modify BGP defaults (1)

- Disable default assumption that all peers are IPv4 unicast peers

```
no bgp default ipv4-unicast
```

- Failure to do this doesn't break anything
  - But makes the IOS configuration and "sh bgp ipvX" output look messy
  - There will be lots of

```
no neighbour x:x:x::x activate
```

  - for IPv6 peers in the IPv4 address family, and lots of

```
no neighbour x.x.x.x activate
```

  - for IPv4 peers in the IPv6 address family

Modify BGP defaults (2)

- Switch BGP to using address families
  - Happens "auto-magically" once first address family configuration entered
  - But remember to apply
    - IPv4 configuration information to the IPv4 address family
    - IPv6 configuration information to the IPv6 address family

```
router bgp 100
 address-family ipv4
  <enter IPv4 configuration as before>
 address-family ipv6
  <enter all IPv6 configuration here>
```

Modify BGP defaults (3)

- Make BGP distances all the same:

```
distance bgp 200 200 200
```

  - This makes eBGP, iBGP and locally originated prefixes have all the same protocol distance
  - (This should already be configured for IPv4)
- Switch off synchronisation
  - Off by default, but no harm caused by including the command in templates

```
no synchronization
```

  - (There is no auto summarisation as there is for IPv4)

Creating IPv6 templates

- Typical iBGP peer-groups might be:
  - core-ibgp router participates in full mesh iBGP
  - rr-client neighbour is a client of this route reflector
  - rr neighbour is a route reflector
- These should be replicated for IPv6:
  - corev6-ibgp router participates in full mesh iBGP
  - rrv6-client neighbour is a client of this route reflector
  - rrv6 neighbour is a route reflector
  - Keep the names the same – just add “v6” in the appropriate place to differentiate
- Peer-groups are to be created within the appropriate address family

Next Steps

- Load all these templates into the routers across the backbone
  - Or simply upload them as each router has IPv6 iBGP deployed on it
- Originate the IPv6 address block on the chosen core routers within the backbone
  - Make sure there is more than one, and the prefix is originated in more than one PoP (for redundancy)
  - BGP network statement and matching static route to Null0 – same as for IPv4

Deploying: Core Router1 in PoP1

- Ensure that the IPv6 peer-groups are in place
  - Tftp load the configuration file from configuration server
- Full mesh iBGP
  - Set up configuration for all other core routers (those participating in the full mesh iBGP)
  - Don’t log into other routers yet – just work on CR1
- Route Reflector Clients
  - Set up the neighbor configuration for the route reflector clients in this PoP
- Insert any required prefixes into iBGP
  - Usually static LAN /64s (they do NOT go in IGP)

Deploying: Core Router1 in PoP1

- Example:

```
router bgp 100
 address-family ipv6
  neighbor corev6-ibgp peer-group
  neighbor corev6-ibgp remote-as 100
  neighbor corev6-ibgp next-hop-self
  neighbor corev6-ibgp update-source loopback0
  neighbor rrv6-client peer-group
  neighbor rrv6-client remote-as 100
  neighbor rrv6-client next-hop-self
  neighbor rrv6-client update-source loopback0
  neighbor rrv6-client route-reflector-client
  neighbor 2001:db8::2 peer-group corev6-ibgp
  neighbor 2001:db8::3 peer-group corev6-ibgp
  neighbor 2001:db8::10 peer-group rrv6-client
  neighbor 2001:db8::11 peer-group rrv6-client
```

Deploying: Gateway Router1 in PoP1

- Ensure that the IPv6 peer-groups are in place
  - Tftp load the configuration file from configuration server
- Route Reflector
  - Set up the neighbor configuration with the two route reflectors in the PoP
  - The two core routers (the route reflectors) have already been configured
  - So the IPv6 iBGP session should come up

Deploying: Gateway Router1 in PoP1

- Example:

```
router bgp 100
 address-family ipv6
  neighbor rrv6 peer-group
  neighbor rrv6 remote-as 100
  neighbor rrv6 next-hop-self
  neighbor rrv6 update-source loopback0
  neighbor rrv6 send-community
  neighbor 2001:db8::1 peer-group rrv6
  neighbor 2001:db8::1 description iBGP with CR1
  neighbor 2001:db8::2 peer-group rrv6
  neighbor 2001:db8::2 description iBGP with CR2
```

Deploying iBGP

- Repeat the previous strategy for all the routers in the first PoP
- And then repeat for all the PoPs
- No eBGP yet!!

Checks

- Are all the iBGP peers up?
  - Best to check on each route reflector
  - If peerings are still down investigate reasons - usually because a loopback address is missing from OSPFv3
- Are there the same number of IPv6 peers as there are IPv4 peers?
  - If not, what went wrong?
- Prefixes in iBGP
  - There probably will be none apart from the /32 aggregate block and any static LANs which have been introduced into iBGP

Seeking IPv6 Transit

Hello World, I'd like to talk to you…

Seeking Transit

- ISPs offering native IPv6 transit are still in the minority
- Next step is to decide:
  - whether to give transit business to those who will accept a dual stack connection

    or
  - Whether to stay with existing IPv4 provider and seek a tunnelled IPv6 transit from an IPv6 provider
- Either option has risks and challenges

Dual Stack Transit Provider

- Fall into two categories:
  A. Those who sell you a pipe over which you send packets
  B. Those who sell you an IPv4 connection and charge extra to carry IPv6
- ISPs in category A are much preferred to those in category B
- Charging extra for native IPv6 is absurd, given that this can be easily bypassed by tunnelling IPv6
  - IPv6 is simply protocol 41 in the range of IP protocol numbers

Dual Stack Transit Provider

- Advantages:
  - Can align BGP policies for IPv4 and IPv6 – perhaps making them more manageable
  - Saves money – they charge you for bits on the wire, not their colour
- Disadvantages:
  - Not aware of any

Separate IPv4 and IPv6 transit

- Retain transit from resolute IPv4-only provider
  - You pay for your pipe at whatever $ per Mbps
- Buy transit from an IPv6 provider
  - You pay for your pipe at whatever $ per Mbps
- Luck may uncover an IPv6 provider who provides transit for free
  - Getting more and more rare as more ISPs adopt IPv6

Separate IPv4 and IPv6 transit

- Advantages:
  - Not aware of any
  - But perhaps situation is unavoidable as long as main IPv4 transit provider can't provide IPv6
  - And could be a tool to leverage IPv4 transit provider to deploy IPv6 – or lose business
- Disadvantages:
  - Do the $$ numbers add up for this option?
  - Separate policies for IPv4 and IPv6 – more to manage

Forward and Reverse DNS

Connecting over IPv6 and fixing those traceroutes…

Forward and Reverse DNS

- Populating the DNS is an often omitted piece of an ISP operation
  - Unfortunately it is extremely vital, both for connectivity and for troubleshooting purposes
- Forward DNS for IPv6
  - Simply a case of including suitable AAAA records alongside the corresponding A records of a host
- Reverse DNS for IPv6
  - Requires getting the /32 address block delegated from the RIR, and then populating the ip6.arpa fields

Forward DNS

- Operators typically access the router by connecting to loopback interface address
  - Saves having to remember interface addresses or names - and these change anyway
- Setting up the IPv6 entries means adding a quad-A record beside each A record:

```
cr1.pop1 A      192.168.1.1
         AAAA   2001:db8::1:1
cr2.pop1 A      192.168.1.2
         AAAA   2001:db8::1:2
gw1.pop1 A      192.168.1.3
         AAAA   2001:db8::1:10
```

Forward DNS

- Completing the infrastructure zone file as per the example is sufficient
  - Update the SOA record
  - Reload the nameserver software
  - All set
- If connecting from an IPv6 enabled client
  - IPv6 transport will be chosen before the IPv4 transport
  - (Part of the transition process from IPv4 to IPv6)
  - For all connections to IPv6 enabled devices which have entries in the forward DNS zones
  - This could have positive as well as negative consequences!

Reverse DNS

- First step is to have the /32 address block delegated by the RIR
- Prepare the local nameservers to handle the reverse zone, for example in BIND:

```
zone ″8.b.d.0.1.0.0.2.ip6.arpa" in {
        type master;
        file "ip6.arpa-zones/db.2001.0db8;
        allow-transfer {"External"; "NOC-NET";};
};
```

- And then "create and populate the zone file"

Reverse DNS

- The db.2001.0db8 zone file heading:

```
$TTL 86400
@      IN     SOA     ns1.isp.net. hostmaster.isp.net. (
                      2008111000       ;serial
                      43200            ;refresh
                      3600             ;retry
                      608400           ;expire
                      7200)            ;minimum

             NS       ns1.isp.net.
             NS       ns2.isp.net.
;Hosts are list below here
```

Creating the reverse zone file

- IPv6 addresses are 128 bits long
  - Bits are grouped in 4 and represented in by a hexadecimal digit
  - Therefore and IPv6 address has 32 hexadecimal digits in it
  - Each one gets a field in IPv6’s reverse DNS
- 2001:db8::1:1 is the loopback address for cr1.pop1
  - We can omit leading zeros and padding zeros are replaced with a set of ::
  - This cannot be done in Reverse DNS ip6.arpa zone files
- Equivalent reverse value would be:
  - 1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa

Creating the reverse zone file

- Major chore filling up the zone file with entries such as
  - 1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.d.b.0.1.0.0.2.ip6.arpa
- Strategy needed!
  - Otherwise serious errors would result, reverse DNS wouldn't function, &c
  - Missing out a single "0" will have consequences
- Possible strategies:
  - Delegate infrastructure /48 to a separate zone file
  - Delegate PtP link /48 to a separate zone file
  - Each customer /48 is delegated to a separate zone file
  - Etc…

Creating the reverse zone file

- Reverse zone for the /32 could read like:

```
; header as previously
;
; Infrastructure /48
0.0.0.0   NS      ns1.isp.net.
0.0.0.0   NS      ns2.isp.net.
; Customer PtP link /48
1.0.0.0   NS      ns1.isp.net.
1.0.0.0   NS      ns2.isp.net.
; Customer One /48
2.0.0.0   NS      ns1.isp.net.
2.0.0.0   NS      ns2.isp.net.
; etc - fill in as we grow
f.f.f.f   NS      ns1.isp.net.
f.f.f.f   NS      ns2.isp.net.
```

Infrastructure reverse zone

- And now we have a /48 reverse zone delegated for infrastructure
  - How do we populate this file?? Entries could still be like this:

```
1.0.0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0      PTR    cr1.pop1.isp.net.
```

  - And we still would have to count zeroes!
- Suggestion 1:
  - Delegate loopbacks to their own /64
  - Keeps the loopback zone file separate, and perhaps easier to manage
- Suggestion 2:
  - Make use of the $ORIGIN directive

Example Infrastructure Reverse Zone

```
; Point to Point links
;
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1        PTR     ge0-1.cr1.pop1.isp.net.
2        PTR     ge0-0.br1.pop1.isp.net.
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.1.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1        PTR     ge0-1.cr1.pop1.isp.net.
2        PTR     ge0-1.br2.pop1.isp.net.
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.1.1.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1        PTR     ge0-1.cr2.pop1.isp.net.
2        PTR     ge0-1.br1.pop1.isp.net.
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.1.1.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1        PTR     ge0-1.cr2.pop1.isp.net.
2        PTR     ge0-0.br2.pop1.isp.net.
```

- Note the use of $ORIGIN and how it keeps the actual line with the PTR value simple

Example Loopback Reverse Zone

```
; PoP1
;
$ORIGIN 0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.b.d.0.1.0.0.2.ip6.arpa.
1.0             PTR     cr1.pop1.isp.net.
2.0             PTR     cr2.pop1.isp.net.
3.0             PTR     br1.pop1.isp.net.
4.0             PTR     br2.pop1.isp.net.
0.1             PTR     gw1.pop1.isp.net.
1.1             PTR     gw2.pop1.isp.net.
2.1             PTR     gw3.pop1.isp.net.
3.1             PTR     gw4.pop1.isp.net.
; etc
```

- Note again the use of $ORIGIN and how it keeps the actual lines with the PTR value **simple** for each loopback interface in the PoP

IPv6 DNS

- Previous examples show how to build forward and reverse DNS zone files
  - Forward is easy
  - Reverse can be troublesome unless care is applied and there is a good strategy in place
- There may well be tools out there which help build reverse DNS zone files from IPv6 address databases
  - Long term that will be a better approach!

Services Aggregation LANs

What about the servers…?

Services Aggregation LANs

- This is talking about the ISP content services
  - How to attach them to an IPv6 network
  - Not how to set up the services on them – that's coming later
- In IPv4 we had HSRP (or VRRP)
- For IPv6 we have GLBP
  - HSRP v2 is also usable, but GLBP allows for load balancing between default gateways

Setting up GLBP

- As with HSRP, GLBP operates a “virtual” default gateway managed by the two (or more) external routers on the LAN
- Need to set aside an IP address which all devices use as the default gateway
  - For IPv4, this was a real routable address
  - For IPv6, this has to be a link-local address
  - FE80::1 seems to be nice and short and doesn’t seem to be used for any particular purpose
  - Schema used is FE80::<glbp group number> as the FE80:: address has to be unique on the router

Setting up GLBP – Configuration

- Router 1:

```
interface GigabitEthernet0/3
 glbp 41 ipv6 FE80::41
 glbp 41 timers 5 10
 glbp 41 priority 150
 glbp 41 preempt
 glbp 41 load-balancing host-dependent
 glbp 41 name NOC-LAN
```

- Router 2:

```
 interface GigabitEthernet0/3
 glbp 41 ipv6 FE80::41
 glbp 41 timers 5 10
 glbp 41 load-balancing host-dependent
 glbp 41 name NOC-LAN
```

Checking GLBP status

```
cr2#sh glbp
GigabitEthernet0/3 - Group 41
  State is Standby
    4 state changes, last state change 00:44:30
  Virtual IP address is FE80::41
  Hello time 5 sec, hold time 10 sec
    Next hello sent in 1.996 secs
  Redirect time 600 sec, forwarder timeout 14400 sec
  Preemption disabled
  Active is FE80::219:E8FF:FE8B:5019, priority 150 (expires in 9.412 sec)
  Standby is local
  Priority 100 (default)
  Weighting 100 (default 100), thresholds: lower 1, upper 100
  Load balancing: host-dependent
  IP redundancy name is "NOC-LAN"
  Group members:
    0019.e873.8a19 (FE80::219:E8FF:FE73:8A19) local
    0019.e88b.5019 (FE80::219:E8FF:FE8B:5019)
  There are 2 forwarders (1 active)
  Forwarder 1
    State is Active
      1 state change, last state change 00:56:16
    MAC address is 0007.b400.2901 (default)
    Owner ID is 0019.e873.8a19
    Preemption enabled, min delay 30 sec
    Active is local, weighting 100
  Forwarder 2
    State is Listen
    MAC address is 0007.b400.2902 (learnt)
    Owner ID is 0019.e88b.5019
    Time to live: 14399.412 sec (maximum 14400 sec)
    Preemption enabled, min delay 30 sec
    Active is FE80::219:E8FF:FE8B:5019 (primary), weighting 100 (expires in 9.412 sec)
```

Default Gateway

Primary router

Setting up GLBP – FreeBSD server

- Configure the servers to use the virtual default gateway
- Because link local address is being used, one extra configuration line in /etc/rc.conf is needed specifying the default device:

```
ipv6_enable="YES"
ipv6_network_interfaces="em0"
ipv6_ifconfig_em0="2001:db8::1 prefixlen 64"
ipv6_defaultrouter="fe80::41%em0"
```

Required otherwise the link local address will not be accepted as default gateway

Setting up GLBP – Linux server

- Configure the servers to use the virtual default gateway
- Because link local address is being used, one extra configuration line in /etc/sysconfig/network is needed specifying the default device:

```
NETWORKING=yes
HOSTNAME=NOC-ALPHA
NETWORKING_IPV6=yes
IPV6_DEFAULTGW=FE80::41
IPV6_DEFAULTDEV=eth0
```

Required otherwise the link local address will not be accepted as default gateway

Services

Network is done, now let’s use it…!

Infrastructure complete

- This was the easy part
  - Network infrastructure generally is very simply to set up as dual stack IPv4 and IPv6
- The next steps are more complex
- Services?
  - Which to make available in IPv6 too?
- Customers?
  - Which can be offered services, and how?

ISP Services

- DNS, Mail, Web
  - Critical customer and Internet facing servers
  - Simple to transition to dual stack
- This involves:
  - Setting up appropriate IPv6 filters on hosting LANs (hint: replicate IPv4 filters)
  - Giving the servers IPv6 addresses
  - Ensuring that the server software is listening on both IPv4 and IPv6 ports
  - Publishing quad-A records along side the regular A records
  - Testing!

FTP Server

- Vsftpd is discussed here
  - Standard part of many Linux distributions now
- IPv6 is supported, but not enable by default
  - Need to run two vsftpd servers, one for IPv4, the other for IPv6
- IPv4 configuration file: /etc/vsftpd/vsftpd.conf

```
listen=YES
listen_address=<ipv4 addr>
```

- IPv6 configuration file: /etc/vsftpd/vsftpdv6.conf

```
listen=NO
listen_ipv6=YES
listen_address6=<ipv6 addr>
```

Other Servers and Services

Managing and Monitoring the Network

Watching the Infrastructure…

Managing and Monitoring the Network

- Existing IPv4 monitoring systems should not be discarded
  - IPv4 is not going away yet
- How to Monitor IPv6?
  - Netflow
  - MRTG
  - Others?

Netflow for IPv6

- Netflow Version 9 supports IPv6 records
- Configured on the router as:

```
interface fast 0/0
 ipv6 flow ingress
 ipv6 flow egress
```

- Displaying status is done by:

```
show ipv6 flow cache
```

- Which all gives the same on-router capability as with IPv4

Netflow for IPv6

- Public domain flow analysis tool NFSEN (and NFDUMP) support Netflow v5, v7 and v9 flow records
  - IPv6 uses v9 Netflow
  - NFSEN tools can be used to display and monitor IPv6 traffic
  - More information:
    - http://nfdump.sourceforge.net/
    - http://nfsen.sourceforge.net/
- ISPs using existing IPv4 netflow monitoring using NFSEN can easily extend this to include IPv6

MRTG

- MRTG is widely used to monitor interface status and loads on ISP infrastructure routers and switches
- Dual stack interface will result in MRTG reporting the combined IPv4 and IPv6 traffic statistics
- MRTG can use IPv6 transport (disabled by default) to access network devices

Other Management Features

- A dual stack network means:
  - Management of the network infrastructure can be done using either IPv4 or IPv6 or both
  - ISPs recognise the latter is of significant value
- If IPv4 network breaks (e.g. routing, filters, device access), network devices may well be accessible over IPv6
  - Partial "out of band" network
- IPv6 is preferred over IPv4 (by design) if AAAA and A records exist for the device
  - So remote logins to network infrastructure will use IPv6 first if AAAA record provided

Customer Connections

Network is done, now let's connect paying customers…

Customer Connections

- Giving connectivity to customers is the biggest challenge facing all ISPs
- Needs special care and attention, even updating of infrastructure and equipment
  - Cable/ADSL
  - Dial
  - Leased lines
  - Wireless Broadband

IPv6 to ADSL Customers

- Method 1: Use existing technology and CPE
  - This is the simplest option – it looks and feels like existing IPv4 service
  - PPPoE v6 + DHCPv6 PD
  - Used by ISPs such as Internode (AU) and XS4ALL (NL)
- Issues:
  - IPv6 CPE are generally more expensive (not the "throwaway" consumer devices yet)
  - Cheaper CPE have no IPv6 yet – need to be replaced/upgraded

IPv6 to ADSL Customers

- Method 2: use 6rd
  - This is for when BRAS cannot be upgraded to support IPv6
  - Used by ISPs such as FREE (FR)
  - Example:
    - 2001:db8:6000::/48 assigned to 6rd
    - Customer gets 192.168.4.5/32 by DHCP for IPv4 link
    - IPv6 addr is 2001:db8:6000:0405::/64 for their LAN (taking last 16 bits of IPv4 address)
    - DHCPv6 PD can be used here too (eg to give /56s to customers)
- Issues:
  - CPE needs to be replaced/upgraded to support 6rd

IPv6 to Dialup Customers

- Use existing technology:
  - Most dialup access routers are easily upgradable to support IPv6
  - Service looks and feels like the IPv4 service
  - PPPoEv6 with DHCPv6 PD (perhaps)
  - CPE is usually PC or laptop (and most OSes have supported IPv6 for many years)
  - Service already offered for several years by many ISPs

IPv6 to Fixed Link Customers

- Use existing technology:
  - Most access routers (PE) and Customer routers (CPE) are easily upgradeable or replaceable to include IPv6 support
  - Service looks and feels like existing IPv4 service
- Configuration options:
  - IPv6 unnumbered on point to point links (or address them)
  - Static routes, subnet size according to business size
  - Or use BGP with private or public (multihomed) ASN
  - Whatever is done for IPv4 should be repeated for IPv6
- Fixed link Customers are probably the easiest to roll IPv6 out to
  - Customer deploying IPv6 within their own networks is a separate discussion (rerun of this presentation!)

IPv6 to Customers

- What about addressing? Here is a typical strategy:
  - House:
    - /64 = 1 LAN
  - Small Organisation:
    - /60 = 16 LANs
    - Reserve the whole /56
    - Reserve a /48 for small orgs = 256 small orgs per /48
  - Medium Organisation:
    - /56 = 256 LANs
    - Reserve the whole /48
  - Large Organisation:
    - /48 = 65536 LANs

Customer Connections

- What about customer end systems?
  - Is IPv6 available on all their computers and other network connected devices?
  - How to migrate those which aren’t?
  - How to educate customer operations staff
  - What about their CPE?
  - What about the link between your edge device and their CPE?
  - What about security?

IOS Images for Cisco's Branch Office Routers

- Need AdvancedIPServices or IPPlus
  - Minimum specification is:

Router	RAM/Flash	IOS	Comments
2500	16M/16F	12.3(26)	No SSH
2600	64M/16F	12.3(26)	No OSPFv3
2600XM	96M/32F	12.3(26)	
2600XM	128M/32F	12.4(25d)	
1841	128M/32F	12.4(25d)	
1751/1760	64M/16F	12.3(26)	
1751/1760	96M/32F	12.4(25d)	

Conclusion

We are done…!

Conclusion

- When deploying IPv6 for the first time, a strategy and planning are of paramount importance
- Presentation has highlighted the steps in the planning and presentation process
  - Variations on the theme are quite likely - there is no single correct way of proceeding