

# IPv6 Deployment at clara.net

**David Freedman**

Network Manager – clara.net

**UKNOF 8** – September 2007

# Why IPv6?

For us, back in 2002, a number of things were happening...

- Globally, concerns about IPv4 exhaustion propagating (usual FUD)
- Globally, IPv6 deployment was happening around us
- Customers were asking “what are you doing about this?”
- Engineers were asking “what are we doing about this?”
- UK6X (BT Exact IPv6 IX) just formed (2001)
- **\*\*insert usual stuff about emerging mobile applications here\*\***
- “One day your fridge will want to talk to your mp3 player” and other silly things
- **More importantly , we already had a customer offering a v6 service...**

# IPNG

- **IPNG were a small organisation dedicated to providing IPv6 connectivity to end-users via tunnels.**
- **They had membership of the UK6X and an /48 allocation from UK6X space.**
- **They also had a connection to 6bone and some 6bone space.**
- **End users would be allocated a /64 for connectivity**
- **System was fully automated for end user provisioning and change requests.**
- **At its peak it had 4000 users sustaining 2Mbit/Second of traffic.**

# Getting ClaraNET on the Native IPv6 Internet - So where to start?

- Well, firstly one should obtain an IPv6 allocation from one's RIR.
- In 2002, RIPE NCC had the following requirements for obtaining an IPv6 allocation:

To qualify for an initial allocation of IPv6 address space, an organisation must:

- a) be an LIR;
- b) not be an end site;
- c) plan to provide IPv6 connectivity to organisations to which it will assign /48s, by advertising that connectivity through its single aggregated address allocation; and
- d) have a plan for making at least 200 /48 assignments to other organisations within two years.

From: [hostmaster@clara.net](mailto:hostmaster@clara.net)

Date: 05/08/02

To: [hostmaster@ripe.net](mailto:hostmaster@ripe.net)

Dear hostmaster,

Can we please have some IPv6 Space.

We would like a /35 for our customer IPNG , some /48s for customers and believe that we will be allocating around 100 /48s a year, making our two year target of 200 /48s.

Lots of Love

Claranet.

From: [hostmaster@ripe.net](mailto:hostmaster@ripe.net)

Date: 06/08/02

To: [hostmaster@clara.net](mailto:hostmaster@clara.net)

Dear Claranet,

You can have 2001:0a88::/32

Lots of Love

RIPE NCC.

# So where to start?

- Well, of course it wasn't quite as simple as this. Having looked back at the original hostmaster ticket, there were 35 emails exchanged between our hostmaster and RIPE NCC!
- Since then the policy has changed somewhat:

## 5.1. Initial allocation

### 5.1.1. Initial allocation criteria

To qualify for an initial allocation of IPv6 address space, an organisation must:

- a. be an LIR;
- b. advertise the allocation that they will receive as a single prefix if the prefix is to be used on the Internet;
- c. have a plan for making sub-allocations to other organisations and/or End Site assignments within two years.

### 5.1.2. Initial allocation size

Organisations that meet the initial allocation criteria are eligible to receive a minimum allocation of /32.

Organisations may qualify for an initial allocation greater than /32 by submitting documentation that reasonably justifies the request. If so, the allocation size will be based on the number of existing users and the extent of the organisation's infrastructure.

Source: [http://www.ripe.net/ripe/docs/ipv6policy.html#initial\\_allocation](http://www.ripe.net/ripe/docs/ipv6policy.html#initial_allocation)

# So what next?

Well, the /32 needed to be added to our IP provisioning and management systems.

We hadn't at this point thought of the development work involved in changing the code for this such to support IPv6 addressing and subnetting so it lived in a text file for a while 😊

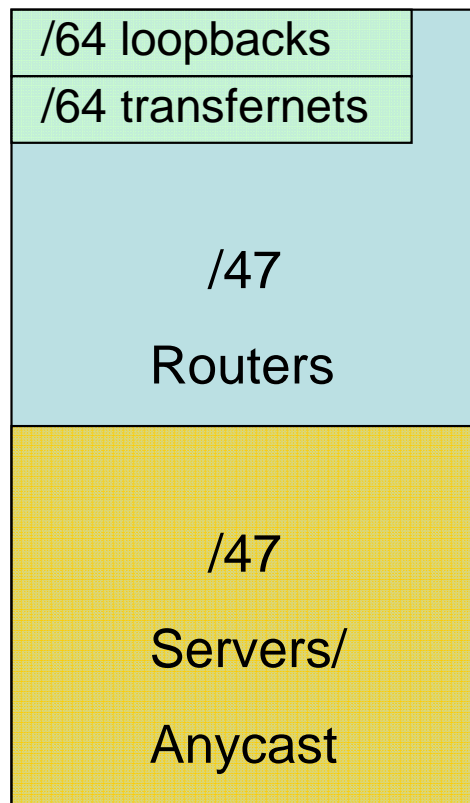


# Carving it up

- A /32 gave us 8 /35s
- We chose the first /35 to be our own
- We allocated the next to IPNG as promised
- From our first /35, we designated the first /48 as infrastructure
- The following /48s would be allocated to customers.

# Carving it up

- So, our first infrastructure /48, how did we carve that one up?



- Two /47s, one for routers the other for servers or anycast IPv6 addresses
- Router /47 split into /64s, with the first reserved for loopbacks, the rest for transfer networks

# Carving it up

- Why /64 router transfer networks?
- Well, /64 is minimum for stateless autoconfiguration
- Why would you want autoconfiguration between routers?
- We opted for /126 size subnets for point-to-point transfer networks and /64s for shared LAN segments (ex IPv4 broadcast domains)
- Alternatively, you may wish to use /112 size subnets

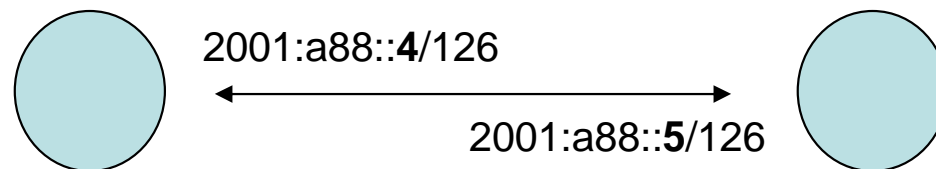
# /126

- With /126 we closely mirror the /30 IPv4 model.
- 4 usable addresses per subnet

```
$ sipcalc 2001:a88::/64 --v6split=126 | head -13  
-[ipv6 : 2001:a88::/64] - 0
```

```
[Split network]
```

```
Network          - 2001:0a88:0000:0000:0000:0000:0000:0000 -  
                  2001:0a88:0000:0000:0000:0000:0000:0003  
Network          - 2001:0a88:0000:0000:0000:0000:0000:0004 -  
                  2001:0a88:0000:0000:0000:0000:0000:0007  
Network          - 2001:0a88:0000:0000:0000:0000:0000:0008 -  
                  2001:0a88:0000:0000:0000:0000:0000:000b  
Network          - 2001:0a88:0000:0000:0000:0000:0000:000c -  
                  2001:0a88:0000:0000:0000:0000:0000:000f  
Network          - 2001:0a88:0000:0000:0000:0000:0000:0010 -  
                  2001:0a88:0000:0000:0000:0000:0000:0013
```



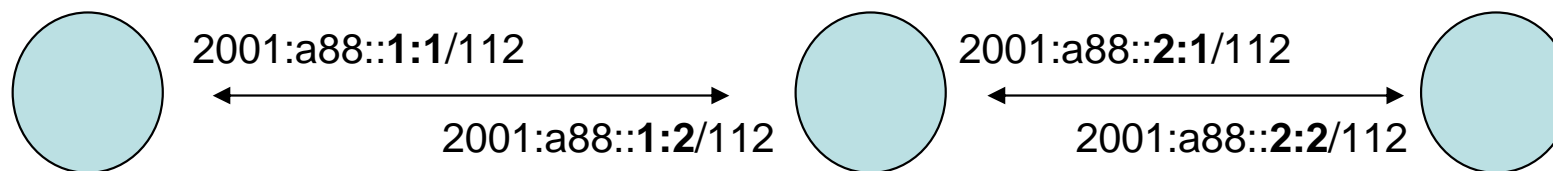
# /112

- /112 is another one of these, a /112 occupies from ::0000 to ::FFFF so is perhaps easier to look at

```
$ sipcalc 2001:a88::/64 --v6split=112 | head -13  
-[ipv6 : 2001:a88::/64] - 0
```

[Split network]

```
Network          - 2001:0a88:0000:0000:0000:0000:0000:0000 -  
                  2001:0a88:0000:0000:0000:0000:0000:ffff -  
Network          - 2001:0a88:0000:0000:0000:0000:0001:0000 -  
                  2001:0a88:0000:0000:0000:0000:0001:ffff -  
Network          - 2001:0a88:0000:0000:0000:0000:0002:0000 -  
                  2001:0a88:0000:0000:0000:0000:0002:ffff -  
Network          - 2001:0a88:0000:0000:0000:0000:0003:0000 -  
                  2001:0a88:0000:0000:0000:0000:0003:ffff -  
Network          - 2001:0a88:0000:0000:0000:0000:0004:0000 -  
                  2001:0a88:0000:0000:0000:0000:0004:ffff -
```



# /64

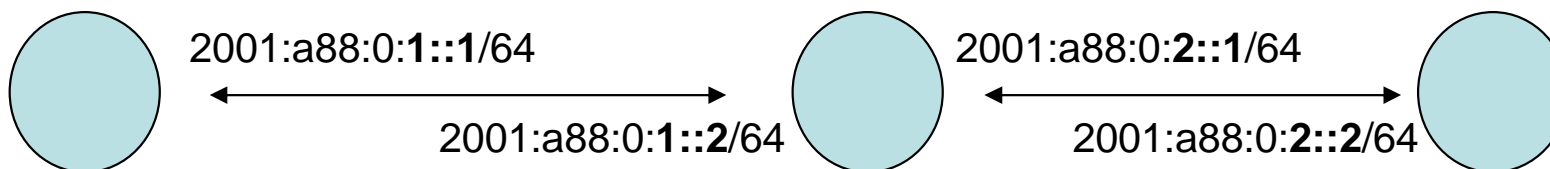
- /64 is the accepted normal, occupying from :: to ::FFFF:FFFF:FFFF:FFFF

```
$ sipcalc 2001:a88::/48 --v6split=64 | head -13  
-[ipv6 : 2001:a88::/48] - 0
```

[Split network]

```
Network          - 2001:0a88:0000:0000:0000:0000:0000:0000 -  
                  2001:0a88:0000:0000:ffff:ffff:ffff:ffff  
Network          - 2001:0a88:0000:0001:0000:0000:0000:0000 -  
                  2001:0a88:0000:0001:ffff:ffff:ffff:ffff  
Network          - 2001:0a88:0000:0002:0000:0000:0000:0000 -  
                  2001:0a88:0000:0002:ffff:ffff:ffff:ffff  
Network          - 2001:0a88:0000:0003:0000:0000:0000:0000 -  
                  2001:0a88:0000:0003:ffff:ffff:ffff:ffff  
Network          - 2001:0a88:0000:0004:0000:0000:0000:0000 -  
                  2001:0a88:0000:0004:ffff:ffff:ffff:ffff
```

**If you are unsure as to which scheme to use then my personal recommendation would be to use /64 link addresses as it will simplify your life greatly.**



# Rolling it out – The routers

- From 2002, the point of conception, to present, we are an almost entirely Cisco powered network
- Four basic models of routers present:
  - 12000 series (GSR)
  - 7500 series
  - 7200 series
  - 6500 / ( now 7600 series as well)
- IPv6 AND stable IOSsen required for all!
- No easy task

# Rolling it out – The routers

- **No good telling you what worked then, code was buggy and unstable. Best talk about what works now:**

- **12000 series**

**GSRs limited to 12.0(S) and now newer 12.0(SY).**

Both are the only 12.0 code train to have IPv6 support, chances are you are running a supported release, but check for bugs first if its not recent.

IPv6 support on Engine 0 and Engine 1 linecards exists but performance is terrible. Don't do it.

- **7500 series**

No 12.0(S) IOS IPv6 support. **12.3 Mainline worked for us** but is not very functional especially when it comes to modern features.

**Would not suggest deploying IPv6 code to 75xx series unless you really have to**

It will consume RAM and flash that you probably don't have and won't want to buy.

- **7200 series**

**12.2S and the newer 12.2SB** are the current recommendations. Again YMMV

Don't even \*think\* about trying this on anything less than NPE-400, there is no official support for it.

- **6500/7600 series**

**12.2 SX (6500) and 12.2SR (7600) are the current favourites**, would recommend minimum SupervisorII if you have old 6500 chassis

As with all of these **READ THE RELEASE NOTES FIRST**,  
I can't stress how important this is!!!



# Rolling it out – The routers

We assumed, from day one, that all IPv6 routers would run IPv6 CEF, there would be no reason not to other than bugs and in such case we would attempt to find a stable release:

```
!  
ipv6 unicast-routing  
ipv6 cef (distributed)  
!
```

*Finding a stable release actually proved quite hard, we were using 12.2S at the time which had newly introduced CEF code (CSSR), this was quite buggy and the CEF consistency checker became a vital feature for us at the time, in this day and age code is more stable.*

---

Next to configure a loopback address for routing protocol purpose. We decided that we would overload the IPv6 loopback address over the top of the existing IPv4 loopback interface, such

```
!  
interface loopback 0  
  ipv6 address 2001:a88::1/128  
  ipv6 enable  
!
```

# Rolling it out – The routers

Now, the problem here is, that for TCP/UDP listeners that the router maintains for management , likely we now have an IPv6 listener socket that exists.

The most important priority should be to secure the router, therefore securing the VTY lines is a good start:

Designate authorised source addresses for management (such as your NOC) and apply such to protect the VTYs

```
!  
ipv6 access-list vty6  
  permit ipv6 2001:A88:1234:FFFE::/64 any  
!  
line vty 0 4  
  ipv6 access-class vty6 in  
!
```

There are many others, beyond the scope of this presentation, the Cisco document “**Managing IOS applications over IPv6**” lists these and is available at the following URL:

[http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6\\_c/sa\\_mgev6.htm](http://www.cisco.com/univercd/cc/td/doc/product/software/ios123/123cgcr/ipv6_c/sa_mgev6.htm)

# Rolling it out – The interfaces

So, next we standardised on what an IPv6 interface configuration would look like, which options would be configured or deconfigured and how this would look like from a LAN or Point-to-Point perspective.

We settled on the following configuration:

```
!  
interface GigabitEthernet 1/1/2  
  ipv6 address 2001:a88:1234::1/126 ← set address  
  ipv6 enable ← enable the IPv6 protocol on the interface  
  ipv6 nd suppress-ra ← turn off router advertisements  
  no ipv6 redirects ← disallow redirects  
!
```

# Rolling it out – The routing

- We opted not to use tunnels between our own routers, even as a migration technique, they would never be removed and the routing would be suboptimal and constrained.
- If you do use tunnels, Cisco routers tend to set the tunnel MTU up to be the egress interface which can be fun where tunnel leaves over 4470 byte POS interface, it of course inherits a 4470 byte MTU which could break things when tunnel travels over a 1500 byte Ethernet interface.
- Remember to watch your tunnel MTU if you do this

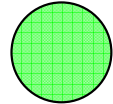
# Rolling it out – The routing

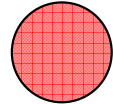
- Our network has MPLS deployed throughout. It was tempting to use 6PE at one point, negating the need to have IPv6 running on the core (P) routers (since it uses LSPs to communicate) but 6PE is an ugly hack and would only need ripping out at some stage since the core would remain IPv4.
- We opted to do label switched IPv4 alongside routed IPv6, since there are no native IPv6 label signaling protocols it is not possible to build IPv6 LSPs across pure IPv6 paths, IPv6 TE is therefore also impossible.

# Rolling it out – The routing

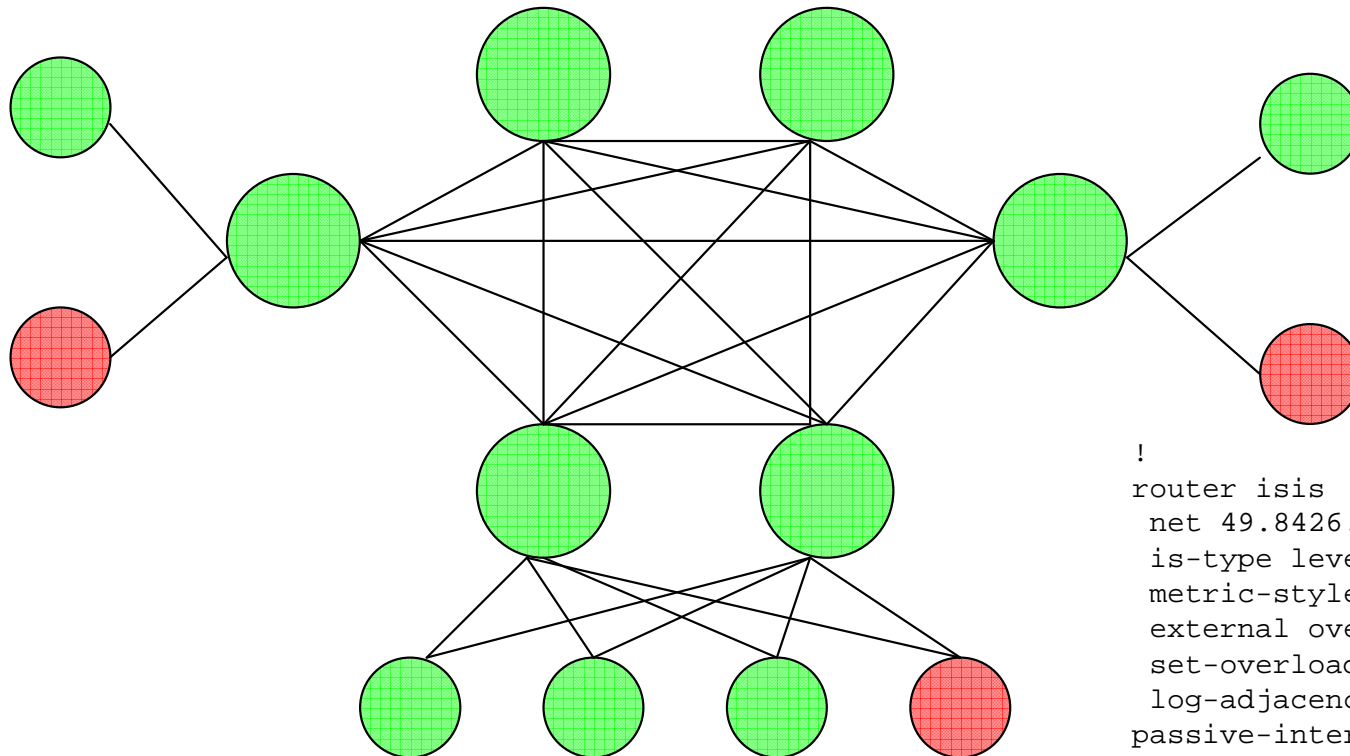
- Our network uses IS-IS as an IGP and iBGP to carry both customer and internet routing.
- We made the decision to deploy multitoplogy IS-IS since integrated IS-IS is TLV based, routers not participating in the migration will ignore the TLVs for the IPv6 topology and it will not affect them
- IPv6 BGP adjacencies built over the top of IPv4 BGP adjacencies, same topology and routing policy

# Rolling it out – The routing

 V6 Participating node

 Non-V6 Participating node

Start with standard single topology IS-IS

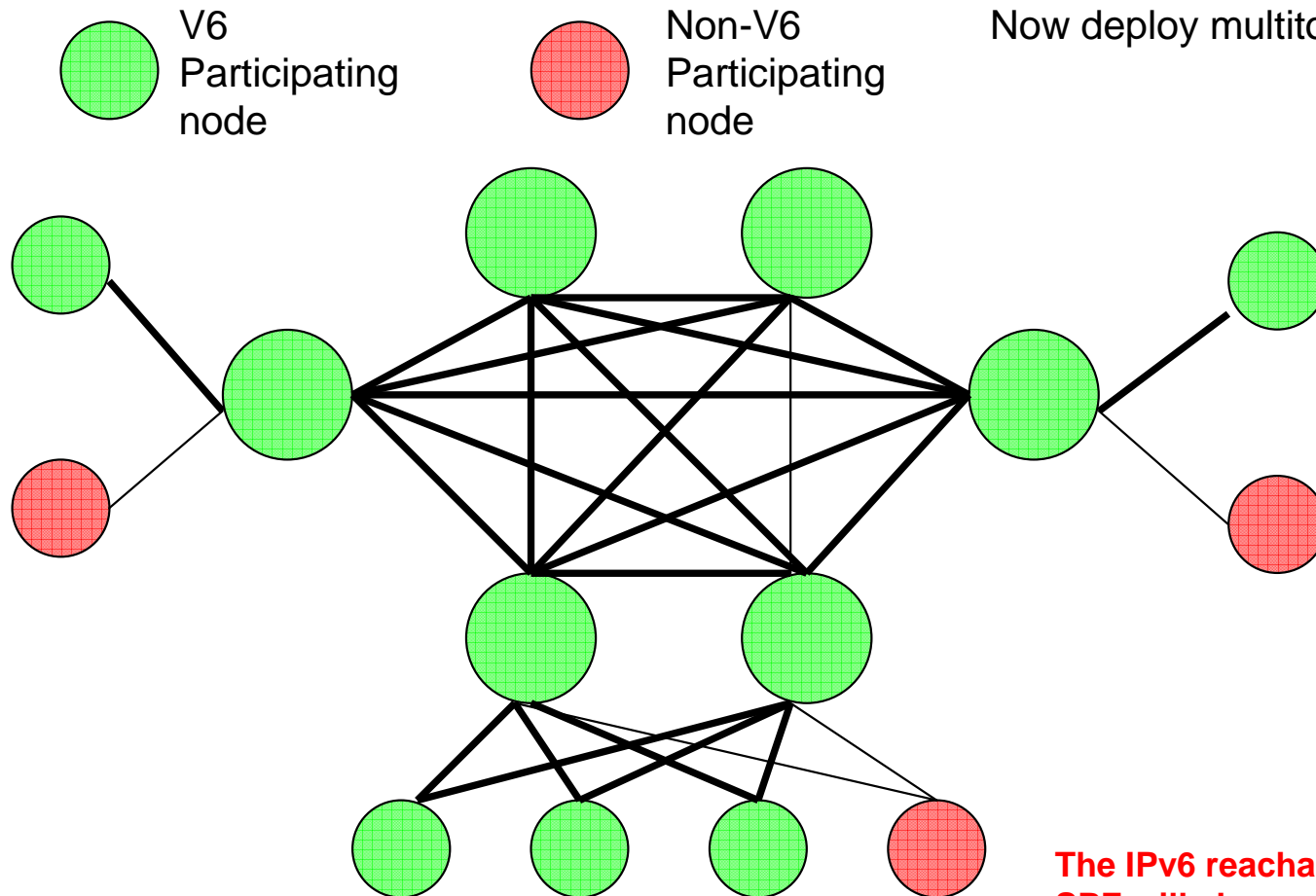


```
!  
router isis  
  net 49.8426.0200.0801.6800.0015.00  
  is-type level-2-only  
  metric-style wide  
  external overload signalling  
  set-overload-bit on-startup wait-for-bgp  
  log-adjacency-changes all  
  passive-interface Loopback0  
!  
interface gigabitethernet 0/0/0  
  ip router isis  
  isis metric 100 level-2  
!
```

```
#show clns neighbors
```

System Id	Interface	SNPA	State	HT	Type	Protocol
foo	Gi0/0/0	0019.2f8c.dacc	Up	9	L2	IS-IS
bar	Gi1/1/3	0004.8084.d602	Up	29	L2	IS-IS

# Rolling it out – The routing



Now deploy multitopology on participating boxes

```
!  
router isis  
  net 49.8426.0200.0801.6800.0015.00  
  is-type level-2-only  
  metric-style wide  
  external overload signalling  
  set-overload-bit on-startup  
  wait-for-bgp  
  log-adjacency-changes all  
  passive-interface Loopback0  
!  
  address-family ipv6  
    multi-topology  
    exit-address-family  
!  
interface gigabitethernet 0/0/0  
ip router isis  
  ipv6 router isis  
  isis metric 100 level-2  
  isis ipv6 metric 100 level-2  
!
```

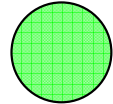
**The IPv6 reachability TLV will carry its OWN metric, SPF will also run separately to build an IPv6 topology, so remember to configure an IPv6 metric on the interface if an IPv4 one exists and you wish to keep the active topology the same.**

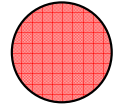
```
#show clns neighbors
```

System Id	Interface	SNPA	State	HT	Type	Protocol
foo	Gi0/0/0	0019.2f8c.dacc	Up	9	L2	<b>M-ISIS</b>
bar	Gi1/1/3	0004.8084.d602	Up	29	L2	IS-IS



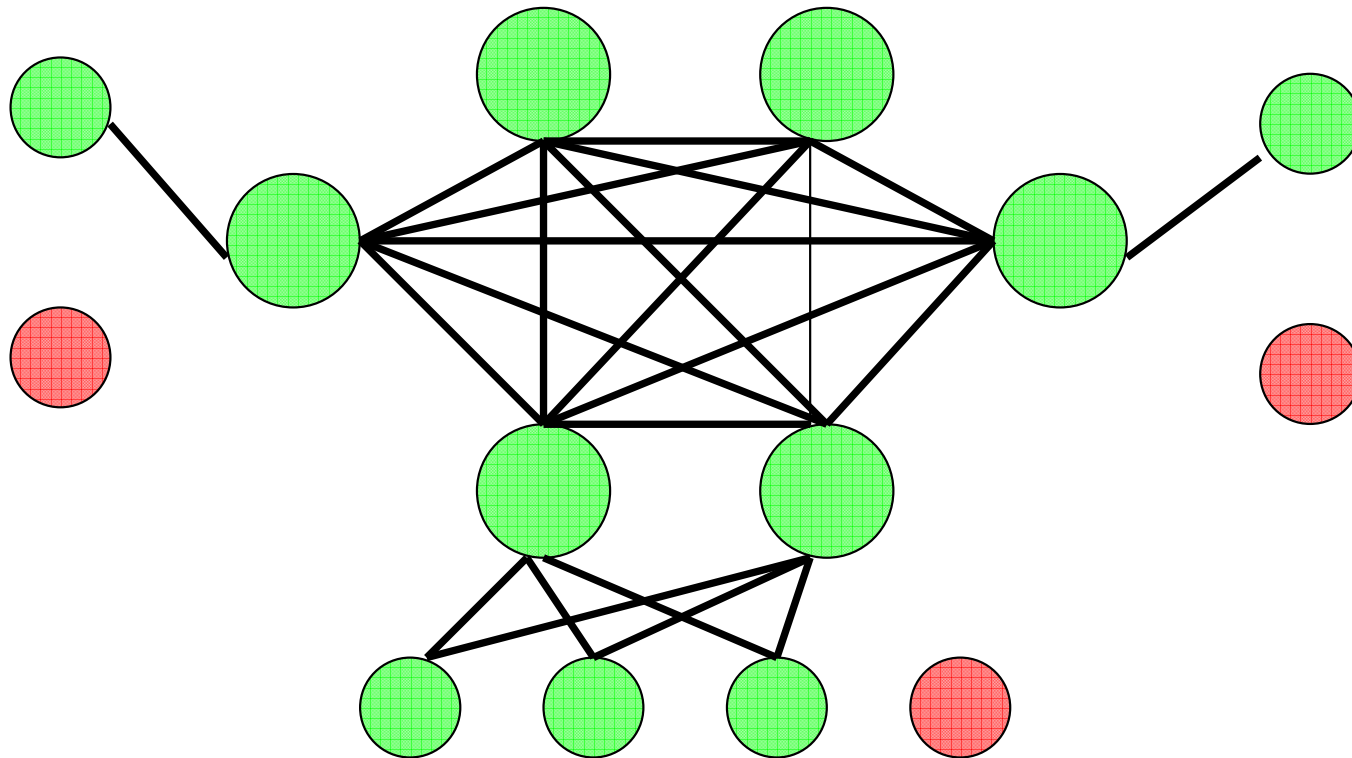
# Rolling it out – The routing

 V6 Participating node

 Non-V6 Participating node

Now deploy additional MP-BGP sessions

Using the same policy



```
!  
router bgp 8426  
neighbor 1.2.3.4 description FOO  
neighbor 1.2.3.4 peer-group POP  
neighbor 2001:a88::1 description FOO  
neighbor 2001:a88::1 peer-group POP6  
neighbor POP peer-group  
neighbor POP remote-as 8426  
neighbor POP6 peer-group  
neighbor POP6 remote-as 8426  
!  
address-family ipv4 unicast  
neighbor POP activate  
neighbor POP remote-as 8426  
neighbor POP send-community  
neighbor POP next-hop-self  
neighbor 1.2.3.4 peer-group POP  
network 1.0.0.0 mask 255.255.255.0  
address-family ipv6 unicast  
neighbor POP6 activate  
neighbor POP6 send-community  
neighbor POP6 next-hop-self  
neighbor 2001:a88::1 peer-group POP6  
network 2001:a88::/32  
!
```

```
#sh bgp ipv6 summary
```

```
BGP router identifier 5.6.7.8, local AS number 8426
```

2001:A88::1	4	8426	1021453	853089	806576	0	0	1d	1
2001:A88::2	4	8426	437681	1186200	806576	0	0	6d	1

# Connecting it to the internet

- We opted for the quickest way to get reachable

Back then, our upstream provider was IPv6 enabled and would give us a dual stack session, we opted for this.

# Connecting it to the internet

- Next we attached to the UK6X, an IPv6 only internet exchange set up by BT exact in 2001 such to facilitate IPv6 internetworking in the UK, we obtained our own dedicated connection alongside IPNG
- UK6X is more like a route server model and provided us with a full table
- Full table swaps in this day and age are generally bad news as they create problems later on with regards to routing policy and traffic exchange, we opted on this to get us up and running and moved to a partial (i.e peering model) later on
- We also waited until other exchanges (such as LINX) allowed for IPv6 traffic and began IPv6 peering over these IXPs when the functionality became available

# Connecting it to the internet

- In order to peer or receive transit, we decided to ensure that our routing policy was published in the RIPE database, this meant creating a **ROUTE6** object for our /32

```
route6:      2001:A88::/32
descr:      CLARA-IPV6-AGG1
origin:     AS8426
mnt-by:     AS8426-MNT
source:     RIPE
```

# Connecting it to the internet

- The next step was to describe our relationships with IPv6 peers.
- Here we hit a snag.
- The current RPSL language that our AUT-NUM object is written did not have the syntax to describe IPv6 adjacencies

# RPSL-NG

- We would have to migrate our AUT-NUM object to **RPSL-NG**, the new language designed to represent objects in the database and be forwardly compatible with emerging addressing schemes.
- Unfortunately, this would be a lot of work, not just training but operational support systems which both parse and make changes to the AUT-NUM object would also have to be re-written

# RPSL-NG

<http://www.ripe.net/ripe/meetings/ripe-43/presentations/ripe43-routing-rpslNg/>

- We opted for a staged migration to an RPSL-NG AUT-NUM object, by embedding the RPSL-NG information in our **remarks** fields, so we could begin to develop new, alongside existing code

```
aut-num:      AS8426
as-name:      CLARANET-AS
descr:        ClaraNET
descr:        UK AS of European ISP
import:       from AS42 action pref=300; community.append(8426:700,8426:799); accept AS42 AND
              {0.0.0.0/0^1-24}
export:       to AS42 announce AS-CLARANET
```

```
...
remarks:      Current IPv6 policy, ready for the RPSL-NG object
remarks:      --
remarks:      mp-import:    afi ipv6.unicast ← Import for IPv6 peers
remarks:      {
remarks:      from 2001:A88:0:2::26 at 2001:A88:0:2::25 action pref=1;
              community.append(8426:2000,8426:2030,8426:2998); accept AS29452;
remarks:      from 2001:7F8:4::312:1 at 2001:7F8:4::20EA:1 action pref=300;
              community.append(8426:700,8426:799); accept AS-JANETPLUS;
...
remarks:      }
remarks:
remarks:
remarks:      mp-export:    afi ipv6.unicast ← Export for IPv6 peers
remarks:      {
remarks:
remarks:      to 2001:7f8:4::999e:1 at 2001:7F8:4::20EA:1 announce AS-CLARANET;
remarks:      to 2001:7f8:4:1::999e:2 at 2001:7F8:4:1::20EA:1 announce AS-CLARANET;
...
remarks:      }
              Remote peer  Us  AS-SET (remains the same)
```

# RPSL-NG

<http://www.ripe.net/ripe/meetings/ripe-43/presentations/ripe43-routing-rpslNg/>

- Adding new IPv6 peers would mean using specific tools which updated the remarks fields.
- Sadly, RPSL-NG has not been widely adopted. The IRRToolset since v4.8.2 has had support for RPSL-NG , but of course, you need to get it to compile first 😊
- Due to the lack of widespread adoption, we have not currently migrated our AUT-NUM object to RPSL-NG.
- We do , however use our own object to build our peer filter policies.



# BGP Challenges - IPv6 BOGONS

- Well, as one would have expected, an IPv6 internet does indeed contain bogon networks, i.e networks not to be accepted from peers.
- The list of these is quite extensive at present, I would like to direct your attention to the document titled

**“Packet Filter and Route Filter Recommendation for IPv6 at xSP routers “**

Available here:

<http://www.cymru.com/Bogons/ipv6.txt>

- If you just want to get yourself up and running quickly, Gert Döring maintains an up to date list of what should currently be filtered, based on both a relaxed and strict model.

Available here:

<http://www.space.net/~gert/RIPE/ipv6-filters.html>

# BGP Challenges – Ghost routes

- Various releases of Cisco IOS software out there in the field have a bug whereby when reachability to destinations becomes lost, a BGP withdrawal is not sent
- This bug is mostly apparent when dealing with V6 prefixes
- IPv6 global table is riddled with these so called “Ghost” routes which end up blackholing traffic to non-existent destinations.
- SIXXS have a “Ghost Route Hunter” (GRH) project currently ongoing to monitor these, I would heartily recommend perusal of the SIXXS site as its list of tools for providing information about the V6 Global table are both extensive and useful

<http://www.sixxs.net/tools/grh/>

# Server Infrastructure

- We deployed /64 server LANs
- No stateless autoconfiguration (disabling router advertisement)
- IPv6 HSRP not available at the time but now available
- Don't forget your security policy!! IPv6 ACLs to be in place before your boxes go in to a reachable subnet!

# DNS

- Next on the list to tackle was the DNS.

For reverse queries, We opted on using IP6.ARPA as IP6.INT was being deprecated in 2006.

For forward queries, AAAA records were used.

Router<->Router Link nomenclature would use the “ipv6.router.uk.clara.net” domain for the time being, to highlight the new stack was in use.



# DNS

The forward is even simpler....

```
$ cat ipv6.router.uk.clara.net.zone
$TTL                                18000
@          IN          SOA      ns0.clara.net.      hostmaster.clara.net. (
                                2004120604         ; Serial number
                                17280              ; Refresh every 2 days
                                3600               ; Retry every hour
                                1728000            ; Expire every 20 days
                                172800 )           ; Minimum 2 days
                                IN          NS       ns0.clara.net.
                                IN          NS       ns1.clara.net.
                                IN          NS       ns2.clara.net.

routera          IN          AAAA     2001:a88::1
routerb          IN          AAAA     2001:a88::2
```

Don't forget.

If you want to be authoritative for zones via IPv6 , your DNS servers must have IPv6 reachability – Your named process must have an reachable IPv6 listener

This is common sense!!!

# Other Service Infrastructure

- Speak to your systems administrators. Many modern operating systems have fully featured IPv6 stacks and service applications (such as web servers, mail servers etc..) are now usually built with decent IPv6 operational support.
- The only risks you run if you plan to use older machines running older software that you believe are operationally functional are instability and possible lack of security and/or auditing features you may need.
- Don't forget, **security is everybody's responsibility** at the end of the day. When rolling out IPv6 services make sure your network security is adequate, don't leave it to be a sysadmin problem, provide network security before they move in!

# Offering the service to customers :- producing an IPv6 service offering

- **Tunnels to end users**
  - do this from a dedicated machine (router or \*nix box) – NOT FROM YOUR CORE NETWORK!
  - Remember , with tunnels, register the assignments in RIPE
  - Don't forget MTU issues
  - Ensure tunnel is carried along **appropriate routing**, have tunnel path follow infrastructure path.
- **IPv6 asynchronous access services**
  - It is unlikely that you will have dialup equipment in your network running stable and functional IPv6 code (although possible), look to L2TP to tunnel people to an LNS device
  - Provide IPv6 DSL services to end users if you can, this however will mean changes to your RADIUS and potentially billing and provisioning systems.
- **IPv6 dedicated access services**
  - Provide access to customer's IPv6 assignments over dedicated access
  - Provide IPv6 transit or partials



# Afterthoughts

- Deploy IPv6 to your office and NOC LANs. (with appropriate security of course), give your staff a taste of working with it, some of them will end up supporting it!
- Many versions of Windows have IPv6 support, but its most mature under Vista / W2K3 server, in fact under Vista its configured and operational by default.
- Don't charge customers for IPv6, make it an additional part of your standard service offering. I wouldn't buy from an upstream that charged extra for this. At the end of the day bandwidth is bandwidth. Think carefully before making any service level offerings (if at all) on dedicated IPv6 based services.
- People will ask "Is IPv6 the right way forward?" and scream about the potential FIB size.  
**IPv6 is here to stay** and quite extensively deployed now, it should now be our job to make it both usable and supportable.

Questions?