

IPv6 DEPLOYMENT AFRICA ONLINE ZIMBABWE

Presented by Mark Tinka
Chief Network Architect
Global Transit
Kuala Lumpur, Malaysia



IPv6: The Interest

IPv6: The Interest

- Main reason; readiness for effects of IPv4 exhaustion.
- Understand the technology so we aren't caught flat-footed.
- To be ahead of the game.
- Dual-stack IP network deployed May, 2007.

IPv6: The Interest

- Knowledge primarily acquired from the “6NET IPv6 Deployment Guide” of 2005.
- Although not very current at the time, this 442-page document served as an excellent base.
- It became easier to build upon this information with present-day concepts and designs related to IPv6 (as well as IPv4).

Design Choice

Design Choice

- For our IPv6 deployment, a dual-stack design was the obvious choice.
- A simple, scalable, clean IPv6 design was our strongest motivation for a dual-stack deployment.
- 6PE promised IPv4 forwarding rates, but this was not a concern at the time; our kit still had a lot of “juice” left in it.

Pre-IPv6 Network

Pre-IPv6 Network

- 95% Cisco backbone.
- MPLS-enabled network.
- OSPFv2 carrying Loopback IPv4 infrastructure addresses.
- iBGP carrying customer prefixes.
- MP-iBGP carrying customer VPNv4 prefixes.
- BGP-free core; core routing 100% MPLS-based.
- Routers were all software-based (7200-VXR's, 3600's, 2800's, e.t.c.).
- Production code was current at the time, IOS 12.3 (GD) and 12.4 (LD), mainline.

First things first

First things first

- We needed IPv6 addresses (obviously).
- Initially worried about the potential complexity and cost in acquiring these.
- We were pleasantly surprised...
- AfriNIC only required us to be a paid-up IPv4 LIR member, and indicate a reasonable number of potential IPv6 customers from our current IPv4 ones.
- 20 was a reasonable number :-).

First things first

- And 5 days later...

First things first

[AFRINIC #21886] afrinic-ipv6-first-alloc - KMail

File Edit View Message Settings Help

Courier 24

[AFRINIC #21886] afrinic-ipv6-first-alloc

From: "Afrinic Hostmaster " <hostmaster@afarinic.net>
To: mtinka@africaonline.co.zw
Date: Tue Apr 3 17:05:02 2007

Hello,

This is to inform you that Afrinic has allocated the following range of IPv6 address space to your organisation:

2001:42b0::/32

The Afrinic whois database shows the following information:

```
inet6num:      2001:42b0::/32
netname:       AfOL-Zw-NETv6
descr:         Africa Online Zimbabwe IPv6 Networks
descr:         In case of abuse, please contact abuse@africaonline.co.zw
country:       Zw
org:           ORG-AOZ1-AFRINIC
admin-c:       AOZ1-AFRINIC
tech-c:        MT991-AFRINIC
status:        ALLOCATED-BY-RIR
mnt-by:        AFRINIC-HM-MNT
mnt-lower:     AfOL-Zw-MNT
notify:        mtinka@africaonline.co.zw
changed:       hostmaster@afarinic.net 20070403
source:        AFRINIC
parent:        2001:4200::/23
```

You can assign from this range according to the policies described in the document: "IPv6 Address Allocation and Assignment Policy in the Afrinic region", available at:

<http://www.afarinic.net/docs/policies/afpol-v6200407-000.htm>

1 2 3 4 Local Folders/Afrinic [AFRINIC #21886] afrinic-ipv6-first-alloc tinka@the-host: ~ - S tinka@the-host: ~ - S 11:10 Casablanca

Deployment

Deployment

- Verified IPv6 support on installed kit.
- Nothing special to be done at Layer 2; this was all transparent to the IPv6 protocol.
- IOS code on the routers supported all the IPv6 features we planned to deploy.
- IPv6-signaled MPLS core not supported; means IPv6 core would be BGP-aware.

Deployment

- Servers and stations running openSuSE-10.2 had native IPv6 support, enabled by default.
- Servers running FreeBSD-6.0, 6.1 and 6.2 had native IPv6 support, enabled with 3 lines in `‘/etc/rc.conf’`.

Addressing

Addressing

- Yes, IPv6 provides millions of addresses and can satisfy even the most insatiable and wasteful of networks...
- But we did not see the need to “waste” address space if we did not have to... did we not once think 4.2 billion IPv4 addresses were more than enough :-).
- A debate for another day, perhaps...

Addressing

- We assigned our address space like so:
 - /48 per PoP; included both infrastructure + customer point-to-point WAN links.
 - Each /48 broken down further to individual /64's.
 - /64 assigned to backbone-wide Loopbacks (taken from main PoP's /48 assignment, for ease of administration).
 - /64 assigned to routing and switch infrastructure.
 - /64 assigned to servers/services.
 - /64 assigned to WAN point-to-point links

Addressing

- On more specific addressing:
 - /128 used for Loopback addresses.
 - /112 used for routing and switching kit subnets.
 - /112 used for server/services subnets.
 - /126 used for WAN point-to-point links.
- For customers, our plan was to assign /48's, per the AfriNIC policy (do they really need that much?).

Configuration

Configuration

- Dual-stack configuration was very straight forward on the routers.
- Support for IPv6 routing in the switches was not very obvious. For those, the management plane would remain in IPv4 land.

Configuration

- A few tips on IPv6 Cisco IOS configuration:
 - ipv6 cef - enable IPv6 CEF switching; it's not enabled by default (I wonder why).
 - ipv6 unicast-routing - also not enabled by default... hmmmh... could this hurt deployment :-\ ?.
 - no ipv6 source-route - disable source routing for IPv6.
 - no ipv6 redirects - disable IPv6 ICMP Redirects.
 - ipv6 nd supress-ra - disable Neighbor Discovery Router Advertisements.

Configuration

□ OSPFv3

- Much easier to configure compared to OSPFv2.
- OSPF area and process ID configured at interface.
- General OSPF configurations still done under main router configuration mode:

```
interface Loopback0
  ipv6 ospf 1 area 0
!
ipv6 router ospf 1
  passive-interface default
  no passive-interface GigabitEthernet0/1
```

Configuration

□ BGP

- Requires MP (Multi-protocol) BGP.

```
router bgp 27576
  no bgp default ipv4-unicast
  !
  address-family ipv6
```

- Configuration similar to IPv4 BGP setup.
- Only difference is 'network' statements are written using CIDR notation.

```
router bgp 27576
  network 2001:42B0::/32
```


Configuration

- ❑ Static routing, ACL's and prefix lists also utilize CIDR notation when describing IPv6 addresses.
- ❑ No more inverse mask calculations :-).

```
ipv6 route 2001:42B0::2:0/112 Null0
ipv6 route ::/0 Serial4/0
!
ipv6 prefix-list upstreams-in6 seq 10 deny 3FFE::/16 le 128
ipv6 prefix-list upstreams-in6 seq 300 permit ::/0 le 48
!
ipv6 access-list filter-incoming6
deny ipv6 3FFE::/16 any
permit ipv6 any 2001:42B0::/32
```

Configuration

- openSuSE-10.2
 - Compiled by default within Linux kernel
 - Enabled by default after install.
 - IPv6 address and routing added via the ``ifconfig`` and ``route`` commands.

```
ifconfig eth0 inet6 add 2001:42b0::6
```

```
route -A inet6 add default gw 2001:42b0::ff7f
```

Configuration

- FreeBSD-6.0, 6.1 and 6.2
 - Compiled by default within kernel.
 - Enabled via `/etc/rc.conf` with 3 lines.

```
ipv6_enable="YES"
```

```
ipv6_defaultrouter="2001:42B0:0000:0000:0000:0000:0000:FF7F"
```

```
ipv6_ifconfig_dc1="2001:42B0:0000:0000:0000:0000:0000:0006/112"
```

DNS

DNS

- Dual-stack hosts had no problem resolving AAAA and PTR records via an IPv4-only DNS server.
- BIND-9 running on FreeBSD-6.2 had no problems listening on an IPv6 interface.
- No recompilation needed, simply add the following to `/etc/named.conf`:

```
options {  
    listen-on-v6 { any; }  
}
```

DNS

- DNS name resolution using an IPv6-only resolver on our openSuSE-10.2 stations worked fine too.
- Creation of fully functional AAAA forwarding records worked fine.
- The 'ip6.arpa' reverse records worked with no problems, for both our ::1 (localhost) and 2001:42b0::/32 zones.
- We used 'sipcalc' to create full 'ip6.arpa' domains (command was 'sipcalc -r')

IP Transit

IP Transit

- With our core network functional as a dual-stack platform, it was time to talk to the outside world.
- First stop was our satellite service provider; they could support neither native nor tunneled IPv6 connectivity.
- Needless to say, our DVB satellite receiver/router had no IPv6 support either (which would only have been a problem for native connectivity).

IP Transit

- The solution was to create an IPv6-in-IPv4 tunnel to our peering router at LINX.
- From here, we received native IPv6 connectivity from Global Crossing via a private Gigabit Ethernet circuit.
- 'aut-num' and 'route' objects were successfully created in the RIPE WHOIS database.
- At the time, we received ~790 IPv6 prefixes on a full BGP feed with Global Crossing.

IP Transit

route6: 2001:42b0::/32

descr: Africa Online Zimbabwe Networks - 2001:42b0::/32

descr: In case of abuse, please contact
abuse@africaonline.co.zw

origin: AS27576

mnt-by: AfOL-Zw-MNT

source: RIPE # Filtered

Basic security

Basic security

□ Cisco IOS

- SSH over IPv6 was supported.
- ACL's for IPv6 are supported.
- The equivalent of BCP-38 and RFC 3330 for IPv6 at the time included filtering of the following addresses:

3FFE::/16

2001:DB8::/32

FE00::/9

FF00::/8

- Application of the ACL's to an interface uses a slightly different command than that of IPv4.

```
interface GigabitEthernet0/1
```

```
ipv6 traffic-filter acl-name in|out
```

Basic security

- openSuSE-10.2
 - 'ip6tables' included within this distribution's Linux kernel.
 - 'ip6tables' application ships with the distribution.
 - SSH over IPv6 worked with no problems.

Basic security

- FreeBSD-6.0, 6.1 and 6.2
 - 'IPfw' and 'Pf' are supported.
 - SSH over IPv6 worked with no problems.

Issues

Issues

□ Cisco

- Limited to no-support for IPv6 in the management plane for some of the desktop switches.
- OSPFv3 IPsec Authentication & Encryption only supported in IOS releases we did not feel were ready for production, e.g., 12.3T.
- SNMP over IPv6 was not supported, although pollers such as MRTG had support.

Issues

- openSuSE-10.2
 - `ip6tables' contributed scripts not as rife as those for IPv4 (`iptables').
 - We found the `ping6' and `traceroute6' commands rather annoying.

Issues

□ FreeBSD-6.0

- Enabling 'IPfw' for IPv4 also enables firewall filtering for IPv6 if an IPv6-enabled interface is detected.
- Remains so even when the IPfw firewall rules do not apply to an IPv6-only interface.
- IPfw support for IPv6 seriously broken in FreeBSD-6.0.
- Fixed in FreeBSD-6.1 and later.
- 'IP6fw' deprecated and will be removed from future FreeBSD releases following 6.1_RELEASE.
- We found the 'ping6' and 'traceroute6' commands equally annoying.

Issues

- Bandwidth manager
 - We had a bandwidth manager in our core providing bandwidth management services to our IPv4 addresses.
 - We were concerned IPv6 traffic would have problems traversing this device.
 - We bypassed the device with a separate core router.
 - Months later, it turns out IPv6 traffic can flow through the device after all, but no bandwidth management services are possible at this time, as no IPv6 support is included in the code.

Note

Note

- ❑ Should you ever need to configure Cisco IOS with IPv6 only, you still need a 32-bit Router-ID.
- ❑ Essentially, an IPv4 address on a Loopback interface.
- ❑ The IPv6 routing protocols on an IPv6-only IOS router require an IPv4 Router-ID.

Concerns

Concerns

- ❑ A scalable transition mechanism for IPv4-only and/or IPv6-only sites.
- ❑ Router forwarding rates in native IPv6 deployments, especially in software-based platforms.
- ❑ The readiness of essential ISP ancillary software, e.g., billing, provisioning, management, e.t.c.
- ❑ Current IPv6 address allocation and assignment policies.

Helpful resources

Helpful Resources

<http://www.civil-tongue.net/clusterf>

http://www.getipv6.info/index.php/Main_Page

<http://www.cisco.com/en/US/products/sw/iossw>

Thank you!

Q&A

mtinka@globaltransit.net