Overview – Module 3

- **IPv6 Deployment in EGP (Case Study)**
- Basic Internet Service Delivery using IPv6 Transport

APNIC

IPv6 Deployment in EGP – Case Study

- Scenario:
  - BGP4 is used in Training ISP network
  - iBGP is used between internal routers in Training ISP to carry external prefixes (i.e Customer & Global Internet Prefixes)
  - Route Reflector is used to resolve iBGP full mesh scalability issue.

APNIC

IPv6 Deployment in EGP – Case Study

- Scenario:
  - Transit service with upstream ASes is configured with eBGP
  - Customer network from downstream can also be configured with eBGP/Static
  - Training ISP is having one native IPv6 transit and one tunnel IPv6 transit with AS45192 & AS131107 (2.35 as dot)

APNIC

IPv6 Deployment in EGP – Case Study

- Basic BGP Configuration:

```
router bgp 17821
address-family ipv6
no synchronization
```

APNIC

IPv6 Deployment in EGP – Case Study

Adding iBGP Neighbor:

```
router bgp 17821
address-family ipv6
!
neighbor 2406:6400:0000:0000::2 remote-as 17821
neighbor 2406:6400:0000:0000::2 update-source loopback 0
neighbor 2406:6400:0000:0000::2 activate
```

iBGP neighbor is always recommended with loopback interface

APNIC

IPv6 Deployment in EGP – Case Study

Announcing IPv6 Prefix:

```
router bgp 17821
address-family ipv6
!
neighbor 2406:6400:0000:0000::2 remote-as 17821
neighbor 2406:6400:0000:0000::2 update-source loopback 0
neighbor 2406:6400:0000:0000::2 activate
!
network 2406:6400:0100:0000::/48
```

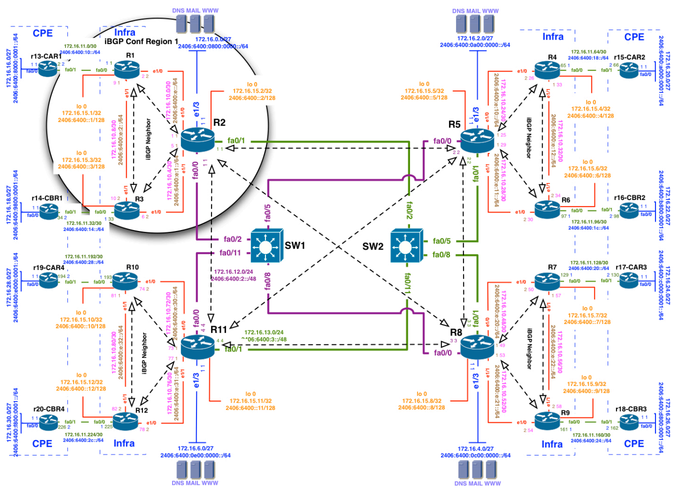
APNIC

IPv6 Deployment in EGP – Case Study

Add Pull-up route if needed:

```
router bgp 17821
address-family ipv6
!
neighbor 2406:6400:0000:0000::2 remote-as 17821
neighbor 2406:6400:0000:0000::2 update-source loopback 0
neighbor 2406:6400:0000:0000::2 activate
!
network 2406:6400:0100:0000::/48
exit
exit
ipv6 route 2406:6400:0100:0000::/48 null 0
```

iBGP Peering For Region 1



IPv4 iBGP Conf POP Router

- Router1

```
config t
router bgp 17821
address-family ipv4
no auto-summary
no synchronization
neighbor 172.16.15.2 remote-as 17821
neighbor 172.16.15.2 update-source loopback 0
neighbor 172.16.15.2 activate
neighbor 172.16.15.3 remote-as 17821
neighbor 172.16.15.3 update-source loopback 0
neighbor 172.16.15.3 activate
network 172.16.16.0 mask 255.255.254.0
exit
exit
ip route 172.16.16.0 255.255.254.0 null 0 permanent
exit
wr
```

APNIC

IPv4 iBGP Configuration Verification

- POP Router

```
sh bgp ipv4 unicast summary
sh bgp ipv4 unicast
sh ip route bgp
sh bgp ipv4 unicast neighbors [router 1……router12
loopback] advertised-routes
sh bgp ipv4 unicast neighbors [router 1……router12
loopback] received-routes
sh ip route [R2, R5, R8, R11 datacenter prefix]
```

APNIC

IPv6 iBGP Conf POP Router

- Router1

```
config t
router bgp 17821
address-family ipv6
no synchronization
neighbor 2406:6400:0000:0000::2 remote-as 17821
neighbor 2406:6400:0000:0000::2 update-source loopback 0
neighbor 2406:6400:0000:0000::2 activate
neighbor 2406:6400:0000:0000::3 remote-as 17821
neighbor 2406:6400:0000:0000::3 update-source loopback 0
neighbor 2406:6400:0000:0000::3 activate
network 2406:6400:0100:0000::/45
exit
exit
ipv6 route 2406:6400:0100:0000::/45 null 0
Exit
wr
```

APNIC

IPv6 iBGP Configuration Verification

- POP Router

```
sh bgp ipv6 unicast summary
sh bgp ipv6 unicast
sh ipv6 route bgp
sh bgp ipv6 unicast neighbors [router 1……router12
loopback] advertised-routes
sh bgp ipv6 unicast neighbors [router 1……router12
loopback] received-routes
sh ipv6 route [R2, R5, R8, R11 datacenter prefix]
```

APNIC

IPv4 iBGP Conf Core Router

- Router2 Configuration

```
config t
router bgp 17821
address-family ipv4
no auto-summary
no synchronization
neighbor 172.16.15.1 remote-as 17821
neighbor 172.16.15.1 update-source loopback 0
neighbor 172.16.15.1 activate
neighbor 172.16.15.3 remote-as 17821
neighbor 172.16.15.3 update-source loopback 0
neighbor 172.16.15.3 activate
neighbor 172.16.15.5 remote-as 17821
neighbor 172.16.15.5 update-source loopback 0
neighbor 172.16.15.5 activate
neighbor 172.16.15.8 remote-as 17821
neighbor 172.16.15.8 update-source loopback 0
neighbor 172.16.15.8 activate
neighbor 172.16.15.11 remote-as 17821
neighbor 172.16.15.11 update-source loopback 0
neighbor 172.16.15.11 activate
network 172.16.0.0 mask 255.255.254.0
exit
exit
ip route 172.16.0.0 255.255.254.0 null 0 permanent
Exit
wr
```

APNIC

IPv4 iBGP Configuration Verification

- Core Router

```
sh bgp ipv4 unicast summary
sh bgp ipv4 unicast
sh ip route bgp
sh bgp ipv4 unicast neighbors [router 1……router12 loopback] advertised-routes
sh bgp ipv4 unicast neighbors [router 1……router12 loopback] received-routes
sh ip route [R2, R5, R8, R11 datacenter prefix]
```

APNIC

IPv6 iBGP Conf Core Router

- Router2 Configuration

```
config t
router bgp 17821
address-family ipv6
no synchronization
neighbor 2406:6400:0000:0000::1 remote-as 17821
neighbor 2406:6400:0000:0000::1 update-source loopback 0
neighbor 2406:6400:0000:0000::1 activate
neighbor 2406:6400:0000:0000::3 remote-as 17821
neighbor 2406:6400:0000:0000::3 update-source loopback 0
neighbor 2406:6400:0000:0000::3 activate
neighbor 2406:6400:0000:0000::5 remote-as 17821
neighbor 2406:6400:0000:0000::5 update-source loopback 0
neighbor 2406:6400:0000:0000::5 activate
neighbor 2406:6400:0000:0000::8 remote-as 17821
neighbor 2406:6400:0000:0000::8 update-source loopback 0
neighbor 2406:6400:0000:0000::8 activate
neighbor 2406:6400:0000:0000::11 remote-as 17821
neighbor 2406:6400:0000:0000::11 update-source loopback 0
neighbor 2406:6400:0000:0000::11 activate
network 2406:6400:0001:0000::/48
exit
exit
ipv6 route 2406:6400:0001:0000::/48 null 0
exit
wr
```

APNIC

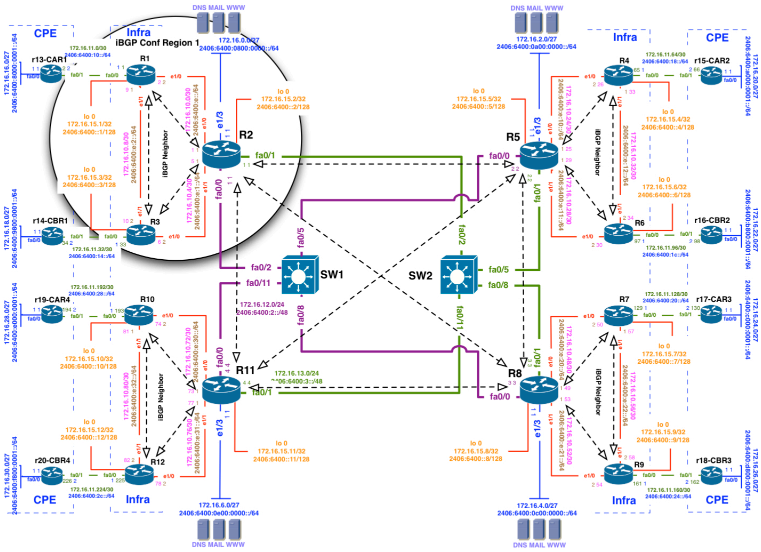
IPv6 iBGP Configuration Verification

- Core Router

```
sh bgp ipv6 unicast summary
sh bgp ipv6 unicast
sh ipv6 route bgp
sh bgp ipv6 unicast neighbors [router 1……router12 loopback] advertised-routes
sh bgp ipv6 unicast neighbors [router 1……router12 loopback] received-routes
sh ipv6 route [R2, R5, R8, R11 datacenter prefix]
```

APNIC

iBGP Full Mesh Issue



APNIC

iBGP Full Mesh Issue

- Route reflector configuration:

```
router bgp 17821
address-family ipv6
!
neighbor 2406:6400:0000:0000::1 remote-as 17821
neighbor 2406:6400:0000:0000::1 update-source loopback 0
neighbor 2406:6400:0000:0000::1 activate
!
neighbor 2406:6400:0000:0000::1 route-reflector-client
```

APNIC

Controlling IPV6 Route Aggregation

- IPv6 prefix filter configuration Customer:

```
config t
ipv6 prefix-list IPV6-CUST-OUT seq 5 permit ::/0 ge 32 le 32
ipv6 prefix-list IPV6-CUST-OUT seq 10 permit ::/0 ge 48 le 48
ipv6 prefix-list IPV6-CUST-IN seq 5 permit cust::/0 ge 32 le
32
ipv6 prefix-list IPV6-CUST-IN seq 10 permit cust::/0 ge 48 le
48

router bgp 17821
address-family ipv6
neighbor cust::2 prefix-list IPV6PREFIX out
exit
exit
exit
clear bgp ipv6 unicast cust::2 soft out
```

APNIC

Case Study- Deployment IPv6 in EGP

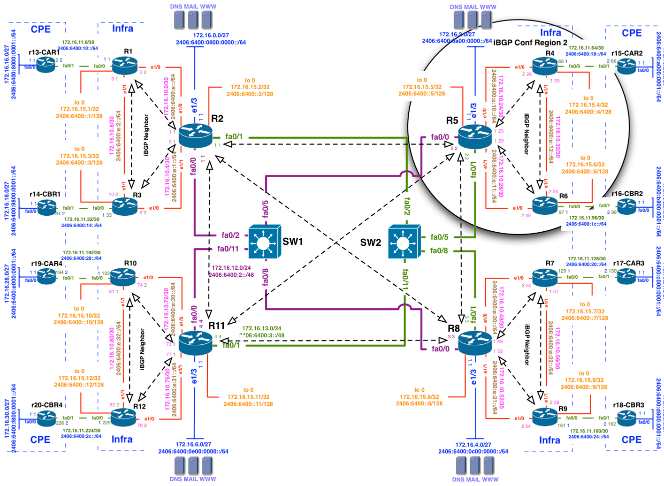
- IPv6 address summarization:

```
router bgp 17821
address-family ipv6
!
aggregate-address 2406:6400::/32
```

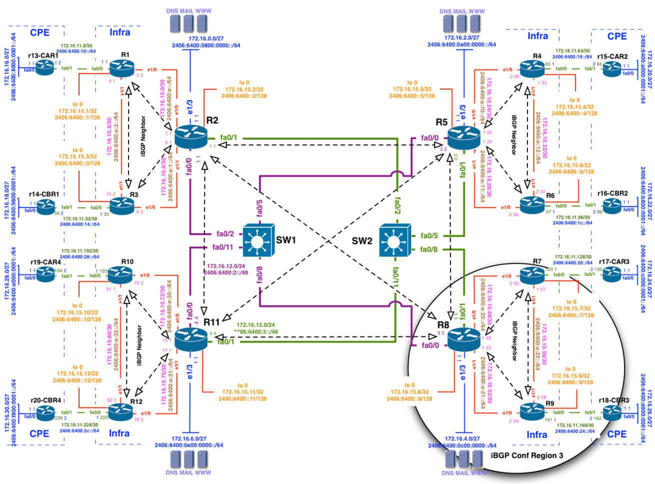
- Need to be very careful when you summarize address

APNIC

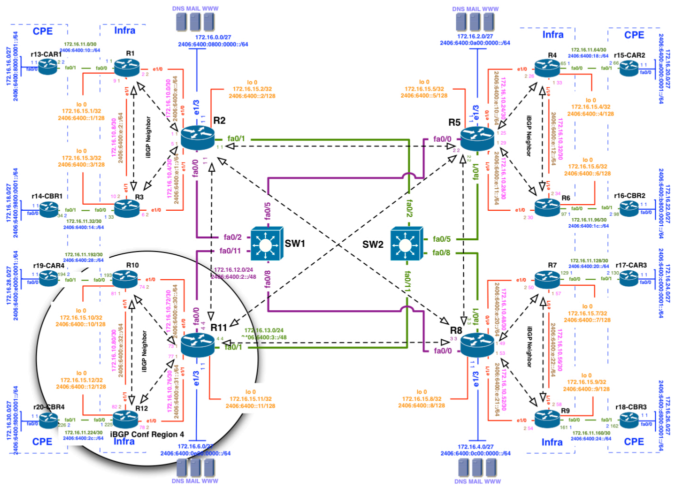
iBGP Peering For Region 2



iBGP Peering For Region 3

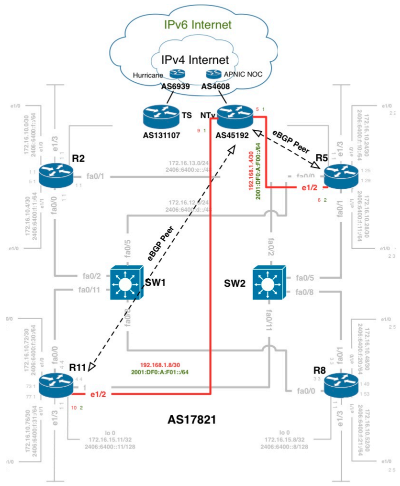


iBGP Peering For Region 4



APNIC

IPv6 Native Transit Conf Plan



APNIC

IPv6 IOS Command For eBGP

- Adding eBGP Neighbor:

```
router bgp 17821
address-family ipv6
!
neighbor 2406:6400:000D:0000::5 remote-as 45192
neighbor 2406:6400:000D:0000::5 activate
```

- eBGP neighbor is always recommended with directly connected interface

APNIC

IPv6 Native Transit Configuration

- Router5

```
config t
router bgp 17821
address-family ipv6
neighbor 2406:6400:000D:0000::5 remote-as 45192
neighbor 2406:6400:000D:0000::5 activate
neighbor 2406:6400:000E:0000::5 remote-as 45192
neighbor 2406:6400:000E:0000::5 activate
exit
exit
exit
Wr
```

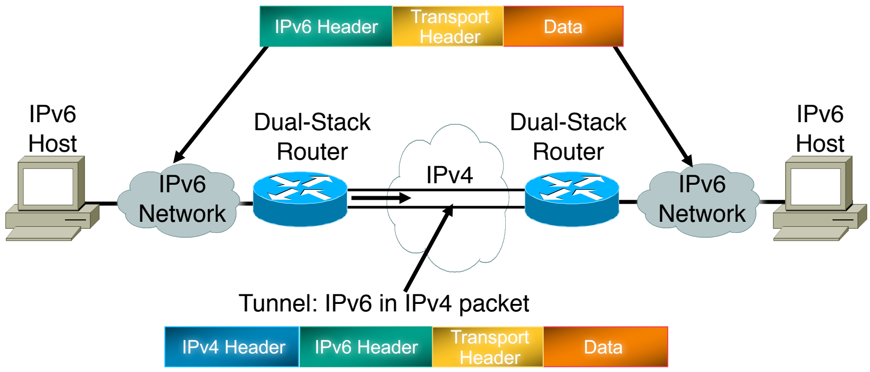
APNIC

Controlling IPV6 Route Aggregation

- IPv6 prefix filter configuration Native Transit:

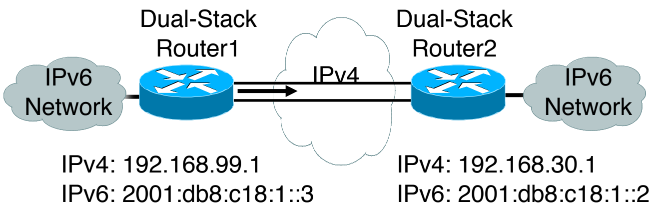
```
config t
ipv6 prefix-list IPV6-GLOBAL-IN seq 5 permit ::/0 ge 32 le 32
ipv6 prefix-list IPV6-GLOBAL-IN seq 10 permit ::/0 ge 48 le 48
!
ipv6 prefix-list IPV6-GLOBAL-OUT seq 5 permit ::/0 ge 32 le 32
ipv6 prefix-list IPV6-GLOBAL-OUT seq 10 permit ::/0 ge 48 le 48

router bgp 17821
address-family ipv6
neighbor 2406:6400:000D:0000::5 prefix-list IPV6-GLOBAL-IN in
neighbor 2406:6400:000D:0000::5 prefix-list IPV6-GLOBAL-OUT out
exit
exit
exit
clear bgp ipv6 unicast 2406:6400:000D:0000::5 soft in
clear bgp ipv6 unicast 2406:6400:000D:0000::5 soft out
```


IPv6 over IPv4 Tunnels



- Tunneling is encapsulating the IPv6 packet in the IPv4 packet
- Tunneling can be used by routers and hosts

Manually Configured Tunnel (RFC2893)



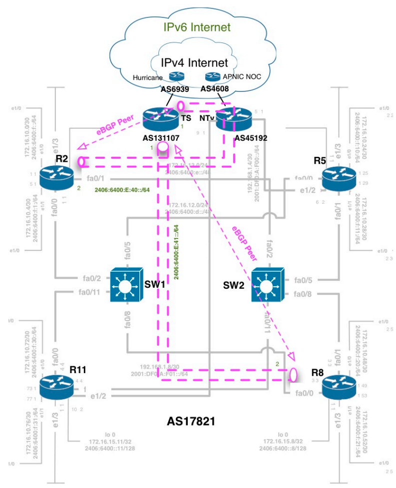

```
router1#

interface Tunnel0
 ipv6 address 2001:db8:c18:1::3/64
 tunnel source 192.168.99.1
 tunnel destination 192.168.30.1
 tunnel mode ipv6ip
```

```
router2#

interface Tunnel0
 ipv6 address 2001:db8:c18:1::2/64
 tunnel source 192.168.30.1
 tunnel destination 192.168.99.1
 tunnel mode ipv6ip
```

- Manually Configured tunnels require:
  - Dual stack end points
  - Both IPv4 and IPv6 addresses configured at each end

APNIC 29

IPv6 Tunnel Transit Configuration



APNIC

6to4 Tunnel Configuration

- IOS Command for Tunnel Interface:

- Router2

```
config t
interface Tunnel0
tunnel source 172.16.15.2
tunnel destination 192.168.1.1
tunnel mode ipv6ip
ipv6 address 2406:6400:F:40::2/64
ipv6 enable
```

APNIC

6to4 Tunnel Configuration

- IOS Command for Tunnel Peering:

```
router bgp 17821
address-family ipv6
neighbor 2406:6400:F:40::1 remote-as 23456
neighbor 2406:6400:F:40::1 activate
```

APNIC

Controlling IPv6 Route Aggregation

- IPv6 prefix filter configuration Tunnel Transit:

```
config t
ipv6 prefix-list IPV6-GLOBAL-IN seq 5 permit ::/0 ge 32 le 32
ipv6 prefix-list IPV6-GLOBAL-IN seq 10 permit ::/0 ge 48 le 48
!
ipv6 prefix-list IPV6-GLOBAL-OUT seq 5 permit ::/0 ge 32 le 32
ipv6 prefix-list IPV6-GLOBAL-OUT seq 10 permit ::/0 ge 48 le 48

router bgp 17821
address-family ipv6
neighbor 2406:6400:F:40::1 prefix-list IPV6-GLOBAL-IN in
neighbor 2406:6400:F:40::1 prefix-list IPV6-GLOBAL-OUT out
exit
exit
exit
clear bgp ipv6 unicast 2406:6400:F:40::1 soft in
clear bgp ipv6 unicast 2406:6400:F:40::1 soft out
```

APNIC

AS Numbers

- Two Ranges:
  - [0 – 65535] are the original 16 bit
  - [65536 – 4294967295] are the new 32 bit
- Usages
  - 0 and 65535 Reserved
  - 1 to 64495 Public Internet
  - 64496 to 64511 Documentation –RFC5398
  - 64512 to 65534 Private use
  - 23456 represent 32 Bit range in 16 bit world
  - 65536 to 65551 Documentation – RFC 5398
  - 65552 to 4294967295 Public Internet

APNIC 34

32 bit AS Number Representation

- AS DOT
  - Based upon 2-Byte AS representation
  - <Higher2bytes in decimal> . <Lower2bytes in decimal>
  - For example: AS 65546 is represented as 1.10
  - Easy to read, however hard for regular expressions
  - There is a meta character "." in regular expression
  - i.e For example, a.c matches "abc", etc., but [a.c] matches only "a", ".", or "c".

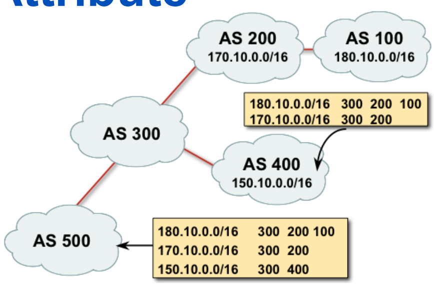
32 bit AS Number Representation

- AS PLAIN
  - ASPLAIN IETF preferred notation
  - Continuation on how a 2-Byte AS number has been represented historically
  - Notation: The 32 bit binary AS number is translated into a Single decimal value Example: AS 65546
  - Total AS Plain range (0 – 65535 - 65,536 - 4,294,967,295)

4 Byte AS Numbers

- 32 Bit range representation specified in RFC5396
- APNIC resource range:
  - In AS DOT: 2.0 ~ 2.1023
  - In AS PLAIN: 131072 ~ 132095
- AS number converter
- http://submit.apnic.net/cgi-bin/convert-asn.pl

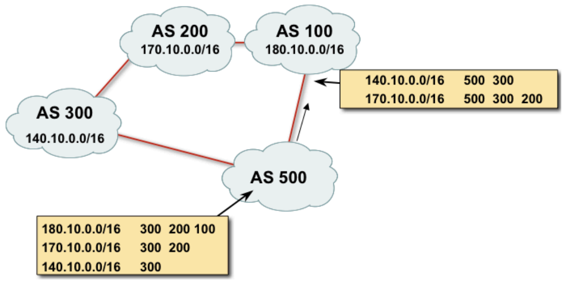
AS Path Attribute



- Sequence of ASes a route has traversed
- Used for
  - Loop detection
  - Path metrics where the length of the AS Path is used as in path selection

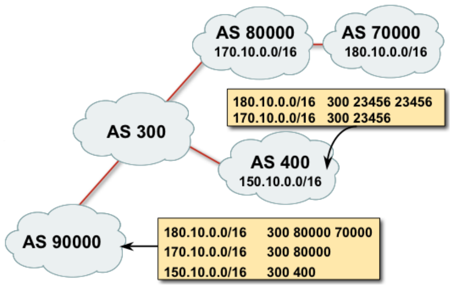
Source: www.cisco.com

AS Path Loop Detection



- 180.10.0.0/16 is not accepted by AS100 as the prefix has AS100 in its AS-PATH
- This is loop detection in action

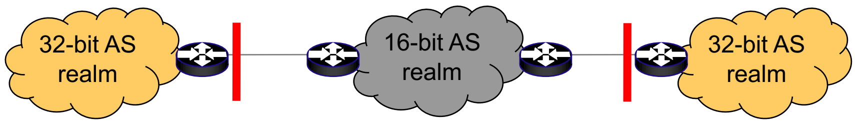
AS Path Attribute (2 byte and 4 byte)



- Internet with 16-bit and 32-bit ASNs
  - 32-bit ASNs are 65536 and above
  - AS-PATH length maintained

32-bit AS Transition

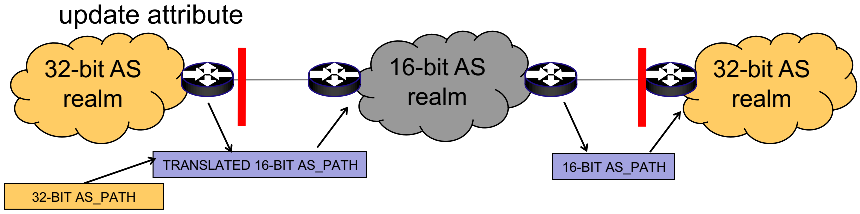
- Think about this space as a set of NEW / OLD boundaries
- Define the NEW / OLD and the OLD / NEW transitions
- Preserve all BGP information at the transition interfaces
  - Translate 32-bit AS Path information into a 16-bit representation
  - Tunnel 32-bit AS Path information through 16-bit AS domain as an update attribute



APNIC

32-bit AS Transition

- Think about this space as a set of NEW / OLD boundaries
- Define the NEW / OLD and the OLD / NEW transitions
- Preserve all BGP information at the transition interfaces
  - Translate 32-bit AS Path information into a 16-bit representation
  - Tunnel 32-bit AS Path information through 16-bit AS domain as an update attribute

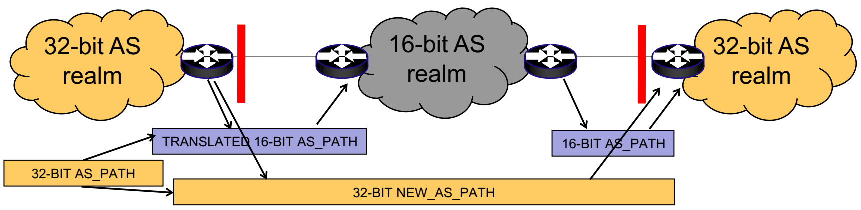


TRANSLATE all 32-bit-only AS numbers to **AS23456**

APNIC

32-bit AS Transition

- Think about this space as a set of NEW / OLD boundaries
- Define the NEW / OLD and the OLD / NEW transitions
- Preserve all BGP information at the transition interfaces
  - Translate 32-bit AS Path information into a 16-bit representation
  - Tunnel 32-bit AS Path information through 16-bit AS domain as an update attribute

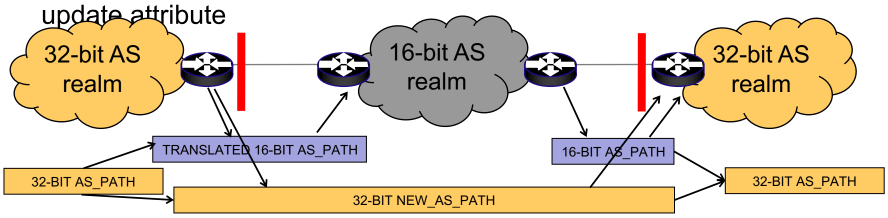


COPY 32-bit AS_PATH to NEW_AS_PATH

APNIC

32-bit AS Transition

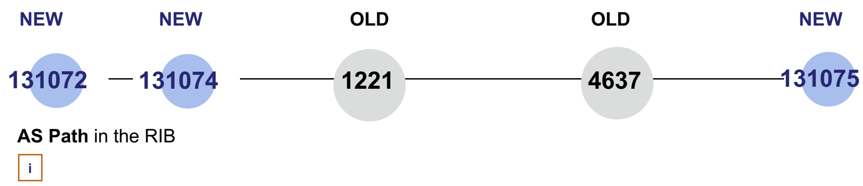
- Think about this space as a set of NEW / OLD boundaries
- Define the NEW / OLD and the OLD / NEW transitions
- Preserve all BGP information at the transition interfaces
  - Translate 32-bit AS Path information into a 16-bit representation
  - Tunnel 32-bit AS Path information through 16-bit AS domain as an update attribute



REASSEMBLE 32-bit AS_PATH

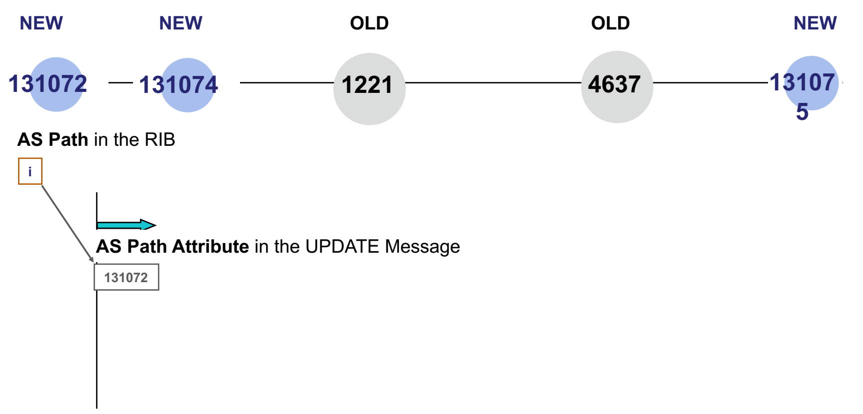
APNIC

32-bit / 16-bit BGP Example



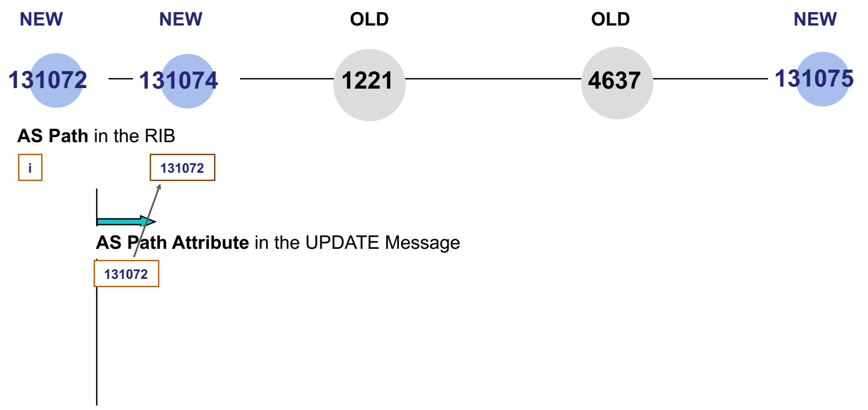
APNIC

32-bit / 16-bit BGP Example



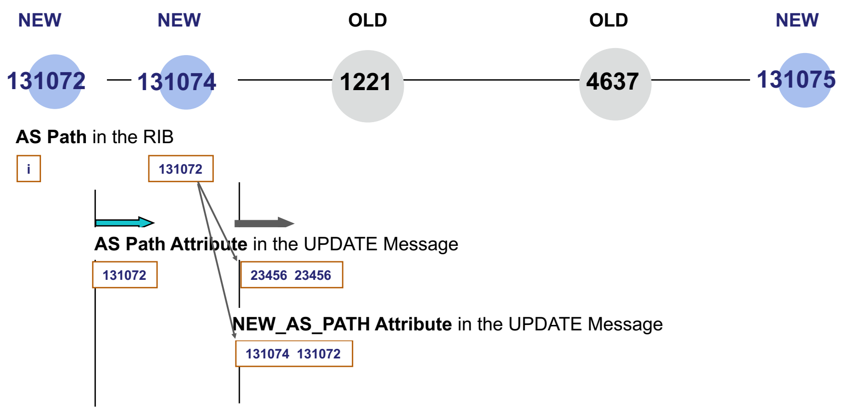
APNIC

32-bit / 16-bit BGP Example



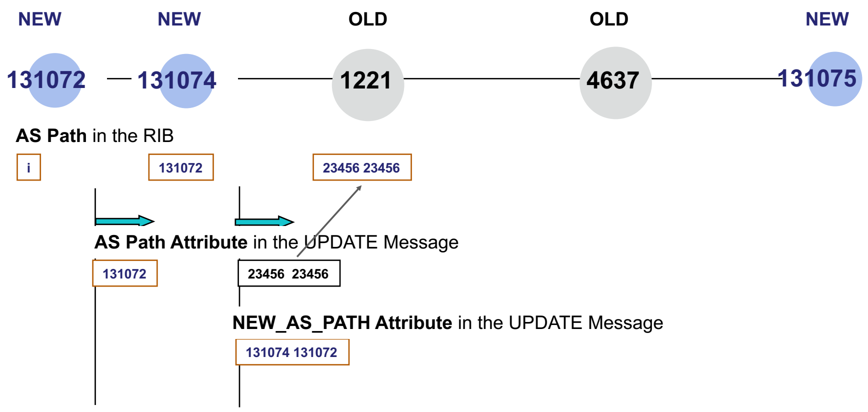
APNIC

32-bit / 16-bit BGP Example



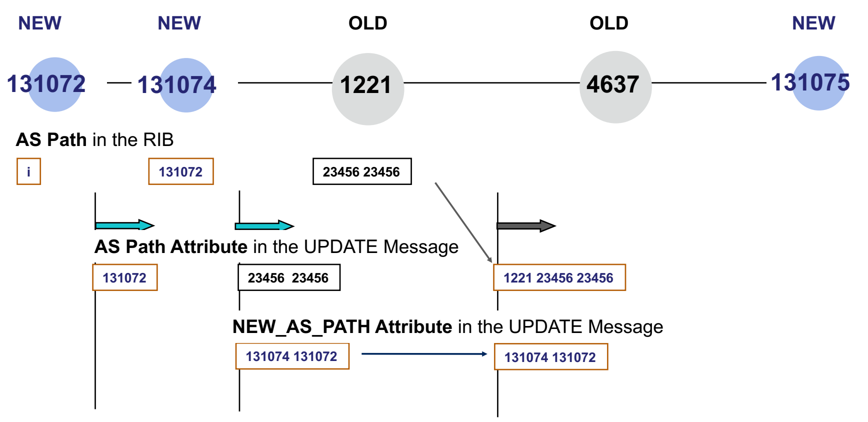
APNIC

32-bit / 16-bit BGP Example



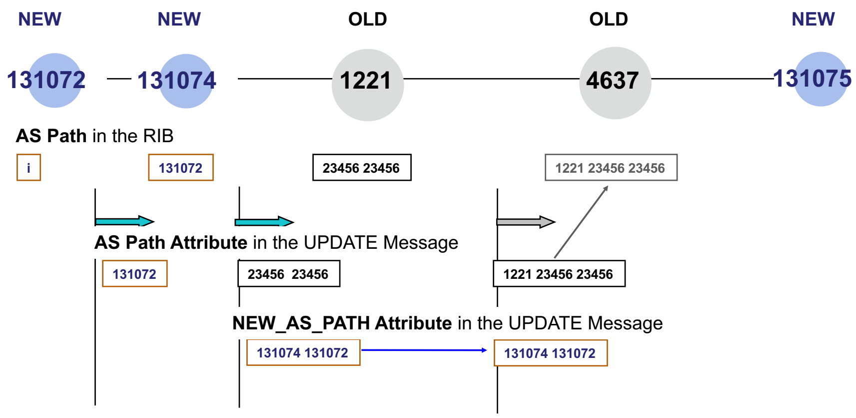
APNIC

32-bit / 16-bit BGP Example



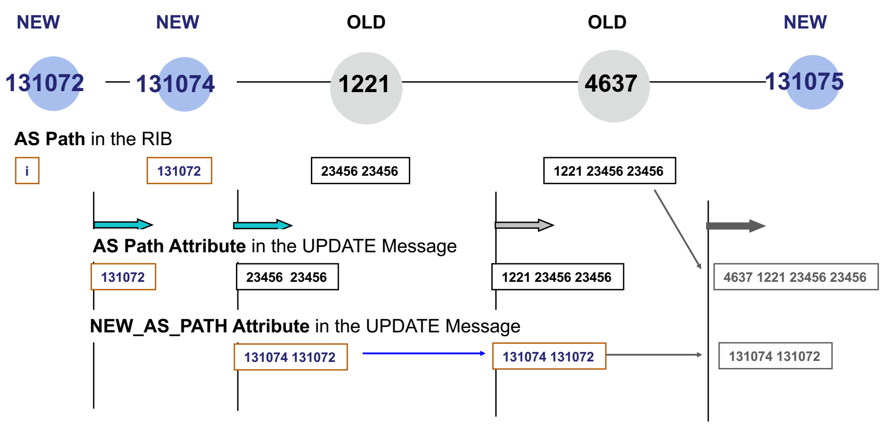
APNIC

32-bit / 16-bit BGP Example



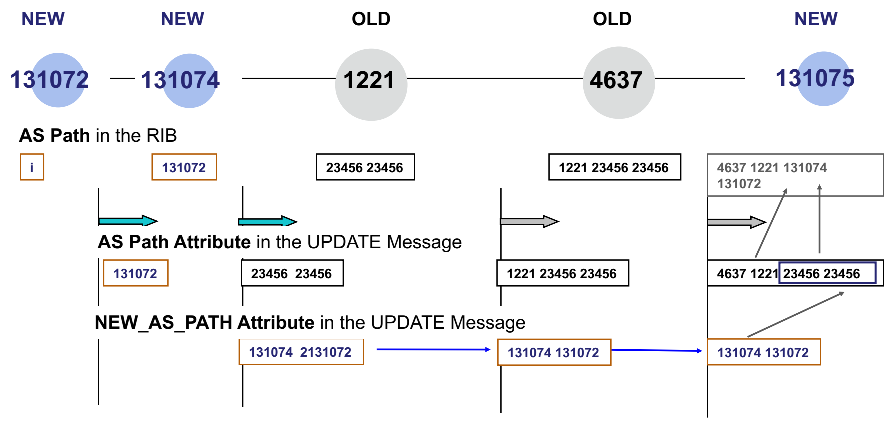
APNIC

32-bit / 16-bit BGP Example

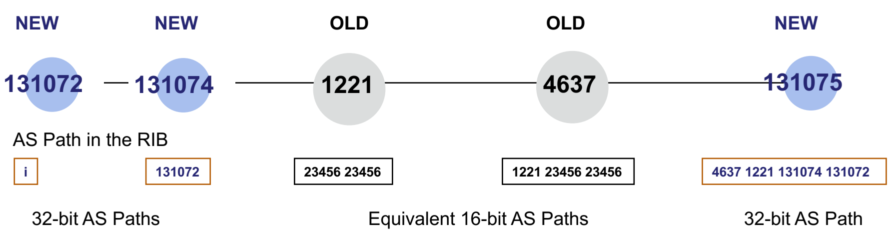


APNIC

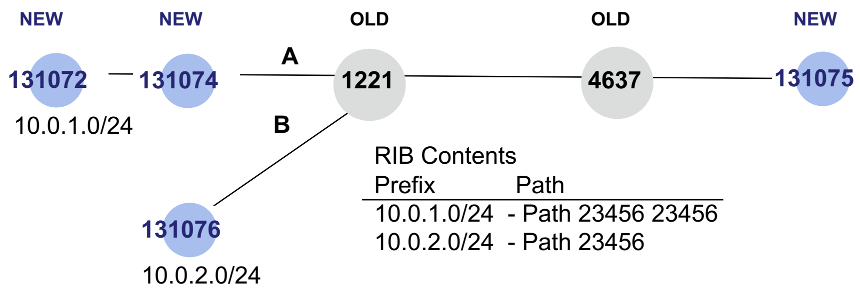
32-bit / 16-bit BGP Example



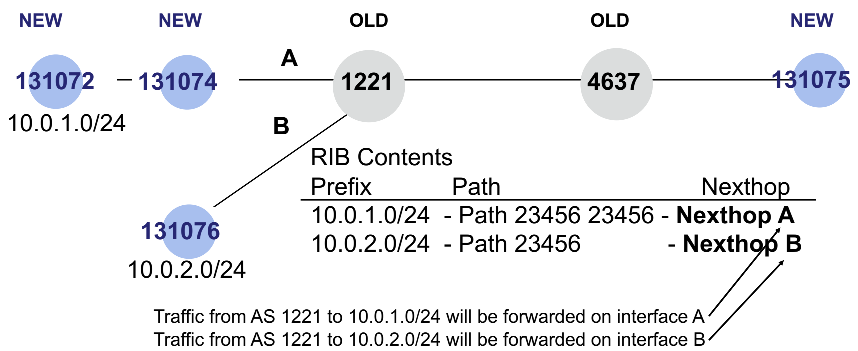
32-bit / 16-bit BGP Example



Can old-BGP get Confused?



NO! BGP Nexthop is the key!



This is standard BGP behaviour – nothing changes here for BGP as it is used today

NEW_AS_PATH Attribute

- BGP speakers in 16-bit AS domains should support NEW_AS_PATH as a transitive optional attribute in UPDATE messages
  – because that's where the 32-bit path is hiding
  – That's a "SHOULD" not a "MUST",
  – Its better if you do, but nothing fatally breaks if you don't
  – Mixed 2 / 4 Byte loops will get detected in the 16-bit world as a fallback
- Default BGP configurations will do the right thing here
  –

APNIC

NEW_AGGREGATOR Attribute

- BGP speakers in 16-bit AS domains should support NEW_AGGREGATOR as a transitive optional attribute in UPDATE messages
  – because that's where the 32-bit Aggregator AS is hiding
  – That's a "SHOULD" not a "MUST", by the way
  – Its better if you do, but nothing fatally breaks if you don't
- Default BGP configurations should do the right thing here
  –

APNIC

AS 23456

- AS 23456 is going to appear in many 16-bit AS paths – both origin and transit
- This is not an error – it's a 16-bit token holder for a 32-bit AS number

APNIC

AS Path and AS4 Path Example

- Router5:

Network Next Hop Metric LocPrf Weight Path
*> 2001::/32 2406:6400:F:41::1
0 23456 38610 6939 i
* i 2406:6400:D::5 0 100 0 45192 4608 4826 6939 i

*> 2001:200::/32 2406:6400:F:41::1
0 23456 38610 6939 2500 i
* i 2406:6400:D::5 0 100 0 45192 4608 4826 6939 2500 i

APNIC 60

Overview – Module 3

- IPv6 Deployment in EGP (Case Study)
- **Basic Internet Service Delivery using IPv6 Transport**

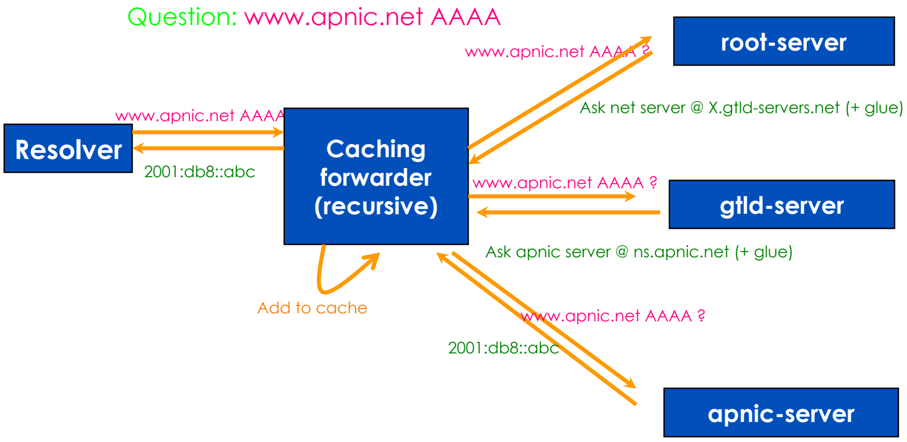
APNIC

DNS Basics

- DNS maps one resource to another resource
  - IP address to hostname (and vice versa)
  - Useful for long addresses (such as IPv6)
- Globally distributed, hierarchical tree structure
- Three components: namespace, resolvers, servers
- Resource records are the actual mappings
  - RR Types: A, AAAA, PTR, CNAME, etc

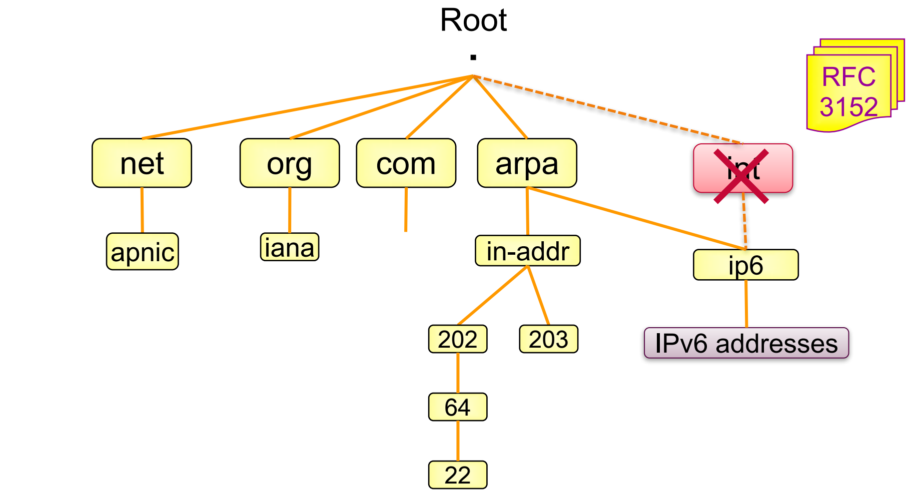
APNIC

DNS Overview (Lookup)



APNIC

Reverse DNS Tree – with IPv6



APNIC

RFCs

- RFC 3596 – DNS Extensions to Support IPv6
  - Introduced AAAA record
  - IP6.ARPA domain
  - Updates RFC1886 (uses IP6.INT domain)
- RFC 3152 – Delegation of IP6.ARPA
  - Used for reverse mapping
  - IP6.ARPA is analogous to IN-ADDR.ARPA zone for IPv4
- RFC 3901 – DNS IPv6 Transport Operational Guidelines
  - As a Best Common Practice

APNIC

IPv6 in the Root Servers

- http://www.internic.net/zones/named.root
- 11 of 13 root servers have IPv6 AAAA records
  - E and G root servers don't have IPv6 capability yet
  - root.hints file contains the IP address of the root servers

APNIC

IPv6 in TLDs

- (as of 15 April 2015)
- Total number of TLDs: 897
- TLDs with IPv6: 867 (96.7%)
- Registered domains with AAAA records: 6772609
  - COM: 3,063,152 of 116,996,997 domains
  - NET: 495,593 of 14,968,344 domains

Global IPv6 Deployment Progress Report
http://bgp.he.net/ipv6-progress-report.cgi

APNIC

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

APNIC

Forward DNS

- Operators typically access the router by connecting to loopback interface address
- Setting up the IPv6 entries means adding a quad-A record beside each A record:

```
r1.pop1    A      192.168.1.1
           AAAA   2001:db8::1:1
r2.pop1    A      192.168.1.2
           AAAA   2001:db8::1:2
gw1.pop1   A      192.168.1.3
           AAAA   2001:db8::1:10
```

APNIC

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
  - For all connections to IPv6 enabled devices which have entries in the forward DNS zones

APNIC

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”

APNIC

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
```

APNIC

Creating the reverse zone file

- IPv6 addresses are 128 bits long
  - Bits are grouped in 4 and represented in by a hexadecimal digit
  - Therefore an 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

APNIC

Creating the reverse zone file

- Major task is 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,
  - 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…

APNIC

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.
```

APNIC

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

APNIC

Example Infrastructure Reverse Zone

```
; Point to Point links
;
$ORIGIN 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.0.0.0.0.0.8.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 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.1.1.0.0.0.0.0.8.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 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.2.1.1.0.0.0.0.0.8.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 0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.3.1.1.0.0.0.0.0.8.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.
```

APNIC

Example Loopback Reverse Zone

```
; PoP1
;
$ORIGIN 0.0.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.8.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

APNIC

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!

APNIC

Dual Stack DNS Conf

- Both Master & Slave
  - DNS software bind-9.7.3.tar.gz [source ftp.isc.org/isc/bind9/9.7.3]
  - BIND root directory [/var/named/chroot] conf file path: /etc/sysconfig/named
  - [named.conf] file path: /var/named/chroot/etc/
  - Zone file path for master zone: /var/named/chroot/var/named/master/
  - Zone file path for slave zone: /var/named/chroot/var/named/slave/
  - Binary executable path: /usr/sbin/
  - Doc file path: /usr/share/doc/bind-9.7*

APNIC

Dual Stack DNS Conf

- #vi named.conf

```
options
{
        directory "/var/named";
        dump-file              "data/cache_dump.db";
        statistics-file        "data/named_stats.txt";
        memstatistics-file     "data/named_mem_stats.txt";
        listen-on-v6 { any; };
};
acl "slave-server-list" {
        203.176.189.29; 2001:0df0:a:100::1e;
        };
```

APNIC

Dual Stack DNS Conf

- Split DNS configuration:
  - 3 view need to configure
    - View "localhost_resolver
    - view "internal"
    - view "external"

APNIC

Dual Stack DNS Conf

- View "localhost_resolver"
  view "localhost_resolver"

{

match-clients { localhost; };

match-destinations { localhost; };

recursion yes;

include "/etc/named.root.hints";

include "/etc/named.rfc1912.zones";

};

- * rfc1912zones i.e. localhost, localdomain, 0.0.127 arpa, ::1 ipv6.arpa, 255 arpa, 0 arpa *

APNIC

Dual Stack DNS Conf

- view "internal"

view "internal"

{

match-clients { localnets; };

match-destinations { localnets; };

recursion yes;

include "/etc/named.root.hints";

APNIC

Dual Stack DNS Conf

- view "internal"

```
zone "romlab.net" {
type master;
file "master/romlab.net.db";
allow-update      { none; };
allow-transfer { slave-server-list; };
};
```

APNIC

Dual Stack DNS Conf

- view "internal"

```
zone "189.176.203.in-addr.arpa" {
type master;
file "master/189.176.203.in-addr.arpa.db";
allow-update      { none; };
allow-transfer { slave-server-list; };
};
```

APNIC

Dual Stack DNS Conf

- view "internal"

```
zone " a.0.0.0.0.f.d.0.1.0.0.2.ip6.arpa" {
type master;
file " master/a.0.0.0.0.f.d.0.1.0.0.2.ip6.arpa.db";
allow-update      { none; };
allow-transfer { slave-server-list; };
};
         };
```

APNIC

Dual Stack DNS Conf

- view ”external”

```
view "external"
{
   match-clients         { any; };
   match-destinations    { any; };
   recursion no;
allow-query-cache { none; };
```

APNIC

Dual Stack DNS Conf

- view ”external”

```
zone "romlab.net" {
type master;
file "master/romlab.net.db";
allow-update        { none; };
allow-transfer { slave-server-list; };
    };
```

APNIC

Dual Stack DNS Conf

- view ”external”

```
zone "189.176.203.in-addr.arpa" {
type master;
file "master/189.176.203.in-addr.arpa.db";
allow-update        { none; };
allow-transfer { slave-server-list; };
};
```

APNIC

Dual Stack DNS Conf

- view ”external”

```
zone " a.0.0.0.0.f.d.0.1.0.0.2.ip6.arpa" {
type master;
file " master/a.0.0.0.0.f.d.0.1.0.0.2.ip6.arpa.db";
allow-update        { none; };
allow-transfer { slave-server-list; };
};
};
```

APNIC

Dual Stack DNS Conf

- Zone file ”ipv6.arpa”

```
$TTL    86400
@             IN SOA  ns1.romlab.net. root.romlab.net. (
                            2011032801      ; serial
                            3H              ; refresh
                            15M             ; retry
                            1W              ; expiry
                            1D )            ; minimum
           IN NS          ns1.romlab.net.
           IN NS          ns2.romlab.net.
f.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0        IN PTR       ns1.romlab.net.
e.1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0        IN PTR       ns2.romlab.net.
```

APNIC

Configuring DHCPv6 on Linux

- Server Configuration [dhcp6s]
  - First need install DHCPv6 RPM on the server
    - # yum -y install dhcpv6
  - Enable IPv6 networking and IPv6 forwarding
    - # vi /etc/sysconfig/network

    NETWORKING_IPV6=yes
    IPV6FORWARDING=yes

APNIC

Configuring DHCPv6 on Linux

- Configure IPv6 on interface
  - # vi /etc/sysconfig/network-scripts/ifcfg-eth0

  IPV6INIT=yes
  IPV6ADDR=" 2406:6400:a000::1/64"
- Specify interface for DHCP server
  - # vi /etc/sysconfig/dhcp6s

  DHCP6SIF=eth0
  DHCP6SARGS=

APNIC

Configuring DHCPv6 on Linux

– Edit the DHCPv6 server configuration file as follows:

```
# cp /usr/share/doc/dhcpv6-*/dhcp6s.conf /etc/
```

- # vi /etc/dhcp6s.conf

```
interface eth0 {
        server-preference 255;
        renew-time 60;
        rebind-time 90;
```

APNIC

Configuring DHCPv6 on Linux

```
option dns_servers 2406:6400:800::2 example.com;
link AAA {
pool{
        range 2406:6400:800::20 to 2406:6400:800::40/64;
        prefix 2406:6400:800::/64;
                };
        };
};
```

Start DHCPv6 server daemon:

```
# service network restart && service dhcp6s start && chkconfig dhcp6s on
```

APNIC

Unix Webserver

- Apache 2.x supports IPv6 by default
- Simply edit the httpd.conf file
  - HTTPD listens on all IPv4 interfaces on port 80 by default
  - For IPv6 add:

```
Listen [2001:db8:10::1]:80
```

    - So that the webserver will listen to requests coming on the interface configured with 2001:db8:10::1/64

APNIC

Unix Sendmail

- Sendmail 8 as part of a distribution is usually built with IPv6 enabled
  - But the configuration file needs to be modified
- Then edit /etc/mail/sendmail.mc thus:
  - Remove the line which is for IPv4 only and enable the IPv6 line thus (to support both IPv4 and IPv6):
  - DAEMON_OPTIONS('Name=IPv4, Family=inet' Addr=203.176.189.2')dnl
  - DAEMON_OPTIONS('Name=IPv6, Family=inet6, Addr=3ffe:b00:1:1::1')dnl
  - configuration files such as mailertable, access, and relay-domains
  - IPV6:3ffe:b00:1:1::1
  - Remake sendmail.cf, then restart sendmail

APNIC

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>
```

IPv6@APNIC

Community

- Policy development
- Participation
- Community activities
- IANA transition
- Internet ecosystem
- Security@APNIC
- IPv6@APNIC
  - Key IPv6 messages
  - IPv6 data and statistics
  - IPv6 transition stories
  - IPv6 for governments
  - IPv6 for mobile networks
  - IPv6 Best Current Practices
  - IPv6 for Decision Makers
  - IPv6 for CTOs
  - About CGN

IPv6@APNIC

IPv6 is a top issue for the Asia Pacific Internet community. APNIC engages in activities throughout the region to help facilitate a smooth transition. The greater goal is to support the Asia Pacific in deploying IPv6 to maintain a scalable Internet for everyone.

APNIC reached the last /8 of IPv4 addresses in April 2011, and now delegates IPv4 resources according to the "last /8 policy". The scarcity of IPv4 makes IPv6 deployment critical for all networks and organizations in the Asia Pacific. Here's what APNIC is doing to support the community in achieving real and tangible IPv6 deployment:

Distributing IPv6 addresses

Getting an IPv6 block is the first step in your transition, and the process is very simple.
Kickstart IPv6 - one click to IPv6

IPv6 training and education

Is your technical staff ready to deploy IPv6? Gaining technical knowledge does not happen overnight. Plan and implement training for your personnel. APNIC Training is constantly updating our IPv6 content, to reflect the industry's best current practices.
Upcoming training events

APNIC 101

---

www.facebook.com/APNIC

www.twitter.com/apnic

www.youtube.com/apnicmultimedia

www.flickr.com/apnic

www.weibo.com/APNICrir

APNIC 102


Thank You!
END OF SESSION
APNIC
103