IPv6 & Linux
About Me

- Work at Jumping Bean
  - Developer & Trainer
- Contact Info:
  - Twitter @mxc4
  - Twitter @jumpingbeansa
  - mark@jumpingbean.co.za
Goals & Motivation

Why?

- Why IPv6?
- Why this talk?
  - Information on the internet fragmented and confusing,
  - No single how-to to get hands dirty

What?

- Understanding of IPv6 concepts, protocol vis-a-vis IPv4,
- How to set up a Linux LAN to use IPv6,
  - Part 1 – Setting up your LAN for IPv6
  - Part 2 – Connecting to the Internet with IPv6
Why IPv6?
Why IPv6?

- Replacement for IPv4,
- 128 bit IP address
  - IPv4 allowed for 4.3 billion possible addresses,
  - IPv6 allows for 340 undecillion addresses 3.40E38,
  - 7.9E28 more than IPv4 addresses,
  - ~ 4.8x10^28 addresses for every human on earth (7 billion people).
  - 1E32 – number of stars in the universe (estimated)
  - 1E82 – number of atoms in the universe (estimated)
IPv6 Benefits

- No need for NAT,
  - Unique, publicly routable, address per device,
- Devices can have more than one address,
- Eliminates network address collision when merging networks,
- “Simplified” auto-configuration,
- Better handling for mobile devices,
- Better multicast support,
- IPSec was mandatory, now optional,
- Simplified router processing
  - No support for router fragmentation,
  - Packet header processing more efficient
- No broadcast traffic
IPv6 History

- RFC 791 (IPv4) published 1981
- RFC 2460 (IPv6) published 1998
- A long time ago ...
- Not backwardly compatible with IPv4
IPv6 Addresses
IPv6 Address Notation

- 128 bit address written in hexadecimal,
  - Written as 8 groups of 16 bits separated by a colon:
    - 2001:0db8:85a3:0000:0000:8a2e:0370:7334

- Abbreviation rules:
  - Drop leading zeros in 16 bit group,
  - If 16 bits all zero replace with empty string “::”
  - If there are sequential groups of 0 replaced by empty string then collapse into a single double colon ::
    - 2001:db8:85a3::8a2e:370:7334
IPv6 Routing Prefix & Interface ID

- “Network mask” is fixed at 64 most significant bits
  - no CIDR,
- Interface identifier (host portion) is fixed at 64 least significant bits
- Common to see IPv6 address with prefix mask that don't match 64 bits,
  - Used in routing,
  - Used in address block assignment,
  - Used in slicing up blocks for special usage
IPv6 Address Allocation

- Internet Assigned Numbers Authority (IANA) assigned Regional Internet Registrars 23/12 bit blocks,

- Regional Internet registrars (Afrinic) assign blocks 19/32 to local Internet registrars,

- End User recommended to get a /48 block which means 65335 subnets but now recommended 56 subnet only 256 subnets.
IPv6 Address Allocation

- Entities can apply for own, provider independent, IPv6 address block with Regional registrar
- Great for ISP independence,
- Why such large allocations?
  - IPv4 routing tables size (current) - 545K,
  - IPv6 routing table size (current) - 22K,
  - Generous allocation policy to avoid routing table explosion
IPv6 How it Works

- Every interface has a link-local address,
  - Network segment only,
- Additional address obtain via
  - Manual configuration, or
  - Automatic configuration,
  - SLAAC
  - DHCP
- Other addresses
  - Unique local address (ULA) - site routable,
  - Global address - internet routable,
IPv6 Link Local

• Each interface auto-assigned a link-local ip address – fe80::/10,
  – Actual assigned link local is fe80::/64
  – replaces layer 2 arp protocols with layer 3,
    • Neighbourhood discovery → map IP to Mac via Neighbour solicitation,
  – Unique only on local network segment,
  – Used to boot strap other IPv6 protocols and addresses
  – Interface prefix is generated from mac address on ethernet NICs using EUI64:
    • Mac address is 48 bits long,
    • Interface identifier is 64 bits long
  – Not forwarded by routers
Unique Local Address/Global Addresses

- **Stateless Automatic Address Configuration** - allows IPv6 networks to auto-configure themselves via ICMPv6 packets
- **Link-Local address** allows for
  - the issuing of router solicitation packets,
  - Receipt of router advertisement packets,
- **Routers**
  - Receive solicitation packets,
  - Send advertisement packets
  - Provide node with one or more network prefix and router address
  - Network prefix can be a ULA or global address
  - Client does duplicate address detection (DAD)
IPv6 - Configurations

- **SLAAC** can be used in a number of ways:
  - Stateless without DHCPv6,
  - Stateless with DHCPv6
  - Stateful with DHCPv6

- **Stateless** -
  - Router/DHCP server does not track ip address,
  - Simply provides network prefix,
  - Node not guaranteed to get same IPv6 address,
  - Node configures host identifier,

- **Stateful** -
  - DHCP server keeps track of addresses handed out (leases),
  - DHCP can assign same IPv6 address to returning node (DUID),
IPv6 - SLAAC

**Pros**
- Automatic configurations,
- No configuration required by client,

**Cons**
- No updating of DNS for nodes, fixed with RFC6106,
- Limited set of configurations options for auto configuration of nodes
IPv6 – ULA/Global Configurations

• **Without DHCP** - Router can also send
  - DNS server information,
  - Router IPv6 address (default gateway),
  - Flags

• **With DHCP** – Node can obtain
  - Fixed IP address,
  - Additional configuration information
  - **DUID** – device unique id,
    - DHCPv6 does not use mac address for unique identification,
    - Each address assigned based on DUID and interface Association identifier,
    - Designed to prevent updating DHCP server when network card changes
    - DUID is created by OS or DHCPClient,
    - **IAID** – from mac
Unique Local Address

• ULA – similar to private addresses in IPv4,
• Can route traffic across network segments,
• Used for company or home lan,
• Should not be routed by gateway devices,
• Network prefix fc00::/7. As 8th bit is always 1 will see fd00 for ula address
• You can create your own ULA or use sites such as http://unique-local-ipv6.com/
Global Addresses

- Assigned by ISP or Afrinic etc,
- Globally routable,
- Similar to IPv4 public addresses,
- For ISP router will need to receive IPv6 prefix for use in configuring IP addresses for nodes,
- Global addresses currently start with 2001::
How to do this on Linux?
IPv6 on Linux

- How to set up a basic IPv6 network for lan,
- What we will need:
  - `radvd` – router advertisement daemon,
    - “apt-get install radvd”
    - or a router on your network with a router advertisement daemon running and configured with your DHCP server details,
  - `isc-dhcp-server` – dhcpv6 capable server,
    - “apt-get install isc-dhcp-server”
  - `bind9` – DNS server for Dynamic DNS updates
    - “apt-get install bind9”
**IPv6 RADVD Configuration**

- **Enable IPv6 forwarding**
  - `net.ipv6.conf.default.forwarding=1`

- **Edit /etc/radvd.conf**
  
  - Prefix – the network prefix to advertise, can have more than one,
  
  - Options
    - **AdvOnLink** – on or off link
    - **AdvAutonomous** – whether this prefix can be used for auto config
    - Enable DHCPv6 lookup
      - **AdvManagementFlag** – use stateful IP assignment
      - **AdvOtherConfigFlag** – get additional config from DHCP server

```plaintext
interface eth0
{
  AdvSendAdvert on;
  prefix fd45:2222:0:1::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
  };
}
interface eth0
{
  AdvSendAdvert on;
  prefix fd45:2222:0:1::/64
  {
    AdvOnLink on;
    AdvAutonomous on;
  };
}
```

**JUMPINGBEAN.CO.ZA**
IPv6 – DHCPv6 Set up

- Isc-dhcp-server can run both IPv4 and IPv6 DHCP services,
- IPv6 DHCP uses different ports to IPv4,
- Most options same as for IPv4 with 6 appended,
  - subnet6, range6
- Use DUID instead of MAC for static address assignment,
- Need to setup keys for dynamic DNS update
- Ubuntu 14.04 – has a bug cannot start dhcp server with “-6” option to enable ipv6.
  - Usually edit /etc/default/isc-dhcp-server and add “-6” to options
  - Need to add to rc.local for now
  - "sudo dhcpcd -6 -cf /etc/dhcp/dhcpcd.conf -lf /var/lib/dhcp/dhcpcd.leases wlan0"
ddns-update-style interim;
ddns-updates on;

update-conflict-detection false;
update-optimization false;

option domain-name "jozilug.co.za";
option dhcp6.name-servers fd5d:12c9:2201:1::2;

default-lease-time 600;
max-lease-time 7200;
include "/etc/dhcp/rndc.key";

zone jozilug.co.za. { 
  primary 127.0.0.1;
  key rndc-key;
}

zone 1.0.0.0.1.0.2.2.c.9.2.1.d.5.d.f { 
  primary 127.0.0.1;
  key rndc-key;
}

subnet6 fd5d:12c9:2201:1::/64 { 
  range6 fd5d:12c9:2201:1::100 fd5d:12c9:2201:1::200;
DHCPv6

- Can operate in several modes
  - Stateless mode → router advertisements assign ip address, DHCP provides DNS, time servers etc
  - Stateful mode → DHCP assigns ip addresses and network services,
  - DHCPv6-PD – prefix delegation obtains network prefix from upstream provider

- Router solicitation →
  - O flag → get configuration information,
  - M flag → get IP address
DHCPv6

- Client uses DUID to identify itself (mac address in DHCPv4)
  - DUID – unique per server/client,
  - Should not be changed in products lifetime,
  - Must be globally unique

- IAID – Interface association ID unique per interface and IP address
DUID

- 4 ways to generate DUID
  - Link layer address + time,
  - Vendor assigned unique id based on enterprise number,
  - Link layer address,
  - UUID – used for SIP devices

- Different devices will have different capabilities → e.g. no persistent storage therefore different ways to generate a unique id

- Problem to detect DUIDs → put on label?

```
hexdump -e "\"%07.7_ax \" 1/2 \"%04x\" \" 14/1 \"%02x:\" \"\n\"
/var/lib/dhcpv6/dhcp6c_duid
```
IPv6 - Bind Set up

- Bind works as for IPv4,
- Bind hosts IPv4 and IPv6 addresses in same zone file,
- Bind will answer queries with the available address. I.e IPv4 host can query for an IPv6 address
- On Ubuntu place zone files in /var/lib/bind otherwise apparmor will prevent updating of zone files
IPv6 - Bind9 Zone File

$ORIGIN .
$TTL 604800 ; 1 week
jozilug.co.za IN SOA jozilug.co.za. admin.jozilug.co.za. ( 150 ; serial 604800 ; refresh (1 week) 86400 ; retry (1 day) 2419200 ; expire (4 weeks) 604800 ; minimum (1 week) )
NS ns.jozilug.co.za.
A 127.0.0.1
AAAA ::1

$ORIGIN jozilug.co.za.
gateway AAAA fd5d:12c9:2201:1::2
ns AAAA fd5d:12c9:2201:1::2
IPv6 – Bind Reverse Zone File

; BIND reverse data file for broadcast zone

$TTL 604800
@ IN SOA ns.jozilug.co.za. admin.jozilug.co.za ( 
    1 ; Serial
    604800 ; Refresh
    86400 ; Retry
    2419200 ; Expire
    604800 ) ; Negative Cache TTL

@ IN NS ns.jozilug.co.za.

2.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.1.0.0.0.1.0.2.2.9.c.2.1.d.5.d.f.ip6.arpa. IN PTR ns.jozilug.co.za
Connecting to the Outside World
Way too many options

- There are a plethora of “transition mechanisms”
  - IPv4 and IPv6 incompatibility
  - Initially IPv6 over IPv4
  - Then IPv4 over IPv6
- Some are focused on Service provider
  - CG-NAT, NAT444, 464XLAT
- Others for LANS,

**Approaches**

- Dual stack
- Encapsulation,
  - Tunnels,
  - A+P,
  - DS-Lite
- Translation,
  - NAT64
  - DNS64,
What to use to connect your LAN?
NAT64/DNS64

- Your ISP gives you an IPv4 address,
- Use only IPv6 internally and use NAT64(tagya),
- Configure bind9 to return all IPv4 addresses as “fake” ipv6 addresses,

Bind9 Additions to options

dns64 fd5d:12c9:2201:1:1::/96 {
  clients {
    any;
  };
  exclude {
    any;
  };
};
**NAT64/DNS64**

- **Pros** – can use Iptables v4 to managed internet connection on Nat64 IPV4 pool,
  - Use only IPv6 internally,
  - Easy to set up
- **Cons** – No access to global IPv6 network. IPv6 only hosts will remain dark
  - Not every type of service is accessible
    - Skype,
    - Web Sockets,
    - SIP
Tunnels 6in4

- Set up DHCPv4 along with DHCPv6,
- Static or automatic tunnels
- Static
  - Create IPV6 SIT tunnel (6in4) to router IPv6 traffic
  - Use a tunnel broker like Hurricane Electric or SixX
- Dynamic
  - Teredo
  - ISATAP
DS-Lite

• Used by ISPs
• IPv4 over IPv6 and IPv4 natting
• DS-Lite – Dual Stack light
  – CPE provides private IPv4 addresses to LAN,
  – CPE encapsulates IPv4 addresses in IPv6,
  – Delivers packet to ISP Carrier Grade Nat (CGN) with public IPv4 address,
    • Recovers IPv4 packets,
    • Nat its,
• Return traffic is mapped to IPv4 then encapsulated in IPv6 and back to client
MAP & A+P

- Proposal for ISPs to extend IPv4 address space,
- Address + Port → Single Ipv4 address shared amongst several clients.
  - Client identified by address and port,
  - Each client assigned a port range,
- MAP →
  - Mapping and Address Port → CISCO Ipv6 transition proposal
  - Combined A+P with tunnelling IPV4 packets over ISP Ipv6 network.
Miscellaneous
Privacy Extensions

- RFC 4941 "Privacy Extensions for Stateless Address Autoconfiguration in IPv6".

- Sysctl use_tempaddr=
  - <= 0 : disable Privacy Extensions
  - == 1 : enable Privacy Extensions, but prefer public addresses over temporary addresses.
  - > 1 : enable Privacy Extensions and prefer temporary addresses over public addresses.

- net.ipv6.conf.eth0.use_tempaddr=2 → /etc/sysctl.conf
- net.ipv6.conf.default.use_tempaddr → only sets network addresses assigned after boot up
- net.ipv6.conf.all.use_tempaddr = 2
- net.ipv6.conf.default.use_tempaddr = 2
- net.ipv6.conf.nic0.use_tempaddr = 2
Disable IPv6

- Remember iptables protects IPv4 addresses only!
- Temporarily disable
  - `sudo sh -c 'echo 1 > /proc/sys/net/ipv6/conf/<interface-name>/disable_ipv6'`
- Edit `/etc/sysctl.conf`
  - `net.ipv6.conf.all.disable_ipv6 = 1`
  - `net.ipv6.conf.default.disable_ipv6 = 1`
  - `net.ipv6.conf.lo.disable_ipv6 = 1`
- Edit `/etc/default/grub`
  - `GRUB_CMDLINE_LINUX="ipv6.disable=1"`
Mark Clarke @mxc4
www.Jumping Bean.co.za
Training, Development & Support