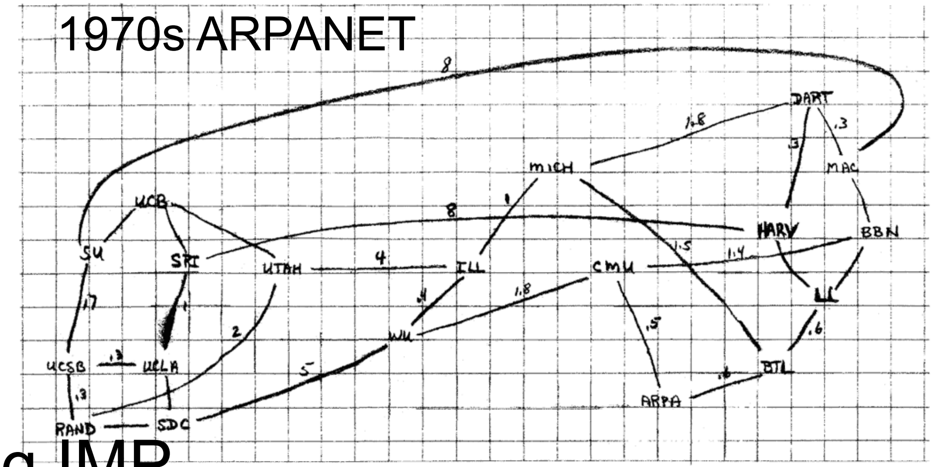
IPv6 -- If we build it, they will come... Or will they?

Owen DeLong
owend@he.net

A Brief History of the Internet

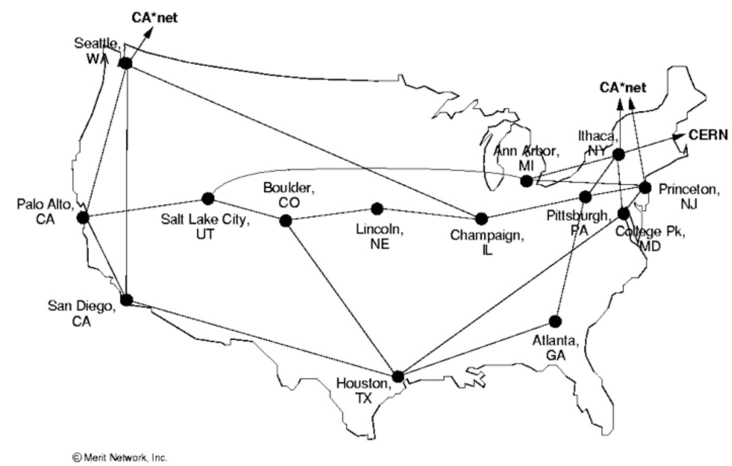


1970s ARPANET

- 1950 -- Computers
- 1961 -- Kleinrock Paper
- 1967 -- ARPANET
- 1969 -- BBN starts building IMP
- 1970 -- NCP released
- 1972 -- Email
- 1972 -- Open Architecture Networking
- 1973 -- Work begins on TCP/IP
- 1983 -- NCP->TCP/IP Flag Day
- 1984 -- Interactive Television pioneered by Orwell (jk)

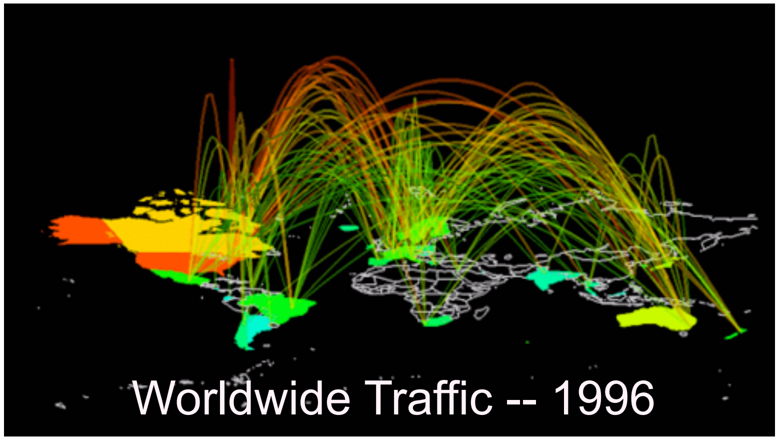
A Brief History of the Internet

NSFNET T1 Network 1991



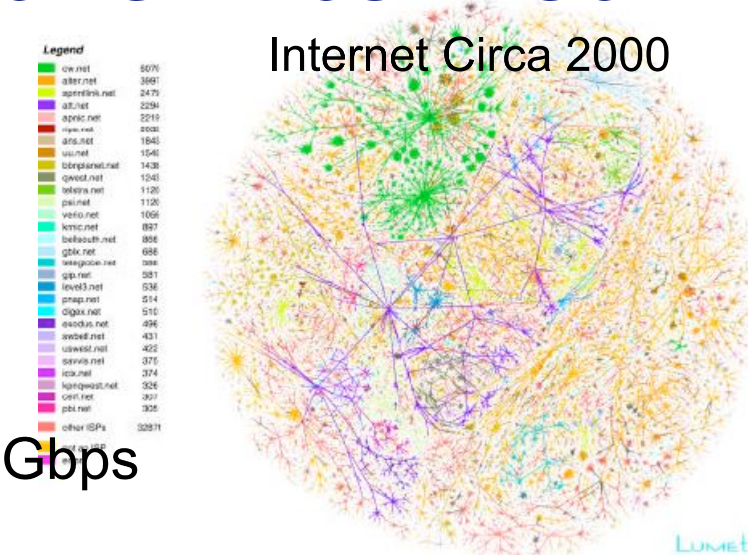
- 1984 -- DNS
- 1985 -- NSFNET Begins (0.991)
- 1986 -- IETF/IRTF Created
- 1987 -- First INTEROP
- 1988 -- NSFNET upgrades to T1
- 1989 -- RIPE Established
- 1990 -- Commercial ISPs
- 1990 -- ARCHIE 1st S.E.
- 1990 -- ARPANET Ends (0.844)
- 1991 -- WAIS, GOPHER
- 1991 -- Zero Wing released (Sega)

A Brief History of the Internet



- 1991 -- NSFNET upgrades to DS-3
- 1992 -- ISOC created
- 1992 -- APNIC Established
- 1994 -- Yahoo, Lycos
- 1994 -- IPNG Work Begins
- 1995 -- NSFNET Ends
- 1995 -- Alta Vista, Excite (0.562)
- 1996 -- Telcos notice VOIP, request congressional ban.
- 1996 -- HotBot, Ask Jeeves
- 1996 -- Backbone upgrades to OC3
- 1996 -- Dancing Baby (Oogachaka)

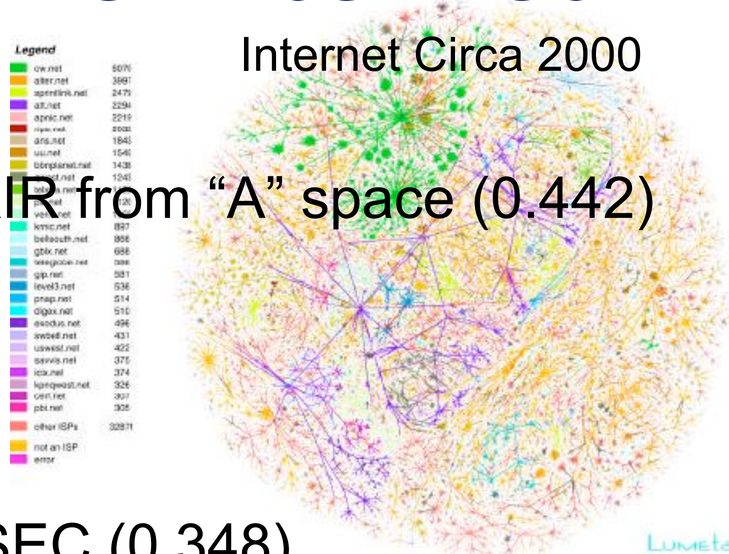
A Brief History of the Internet



Internet Circa 2000

- 1997 -- ARIN Established
- 1998 -- Google
- 1999 -- RFC 2663 Introduces NAT
- 1999 -- Backbone Upgraded to 2.4Gbps
- 1999 -- ICANN Tests SRS for DNS
- 2000 -- Internet 2 Backbone deploys IPv6 (.518)
- 2000 -- Renumbered to 19100 by UNSO (and others) OOPS!
- 2001 -- .biz and .info added
- 2001 -- LACNIC Established
- 2002 -- Internet 2 Backbone deploys Native IPV6 (Abilene)
- 2002 -- “All Your Bases” becomes biggest meme

A Brief History of the Internet

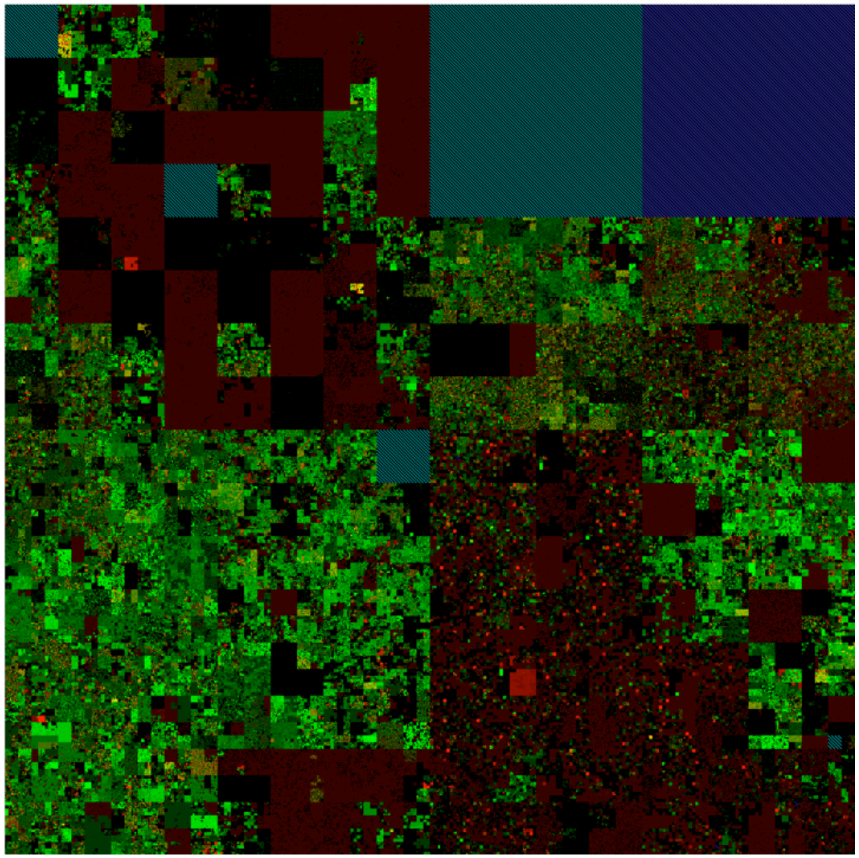


Internet Circa 2000

- 2002 -- First /8 (69/8) Allocated to RIR from “A” space (0.442)
- 2003 -- First Internet Flash Mobs
- 2004 -- US < 50% of Root Servers
- 2004 -- AfriNIC Established (0.375)
- 2005 -- .SE First CCTLD with DNSSEC (0.348)
- 2006 -- 6bone Testbed deprecated (0.312)
- 2007 -- Internet 2 Completes 100Gbps upgrade (0.286)
- 2008 -- First 6 root servers get IPv6 Addresses (0.219)
- 2009 -- RFC 5514 -- IPv6 over Social Networks (0.152)
- 2010 -- WikiLeaks (0.107)
- 2010 -- F*ckin’ Magnets, How do they work? (ICP Viral Video)

A Brief History of the Internet

- 2011 -- End of IPv4 Free Pool (0)
- 2011 -- APNIC Austerity
- 2011 -- World IPv6 Day (6/8)
- 2012 -- World IPv6 Launch (6/6)
- 2012 -- RIPE-NCC Austerity



IPv4 Address Space 2012

What are those Numbers in Parenthesis?

- 1985 (0.991)
- 1990 (0.884)
- 1995 (0.562)
- 2000 (0.518)
- 2002 (0.442)
- 2004 (0.375)
- 2005 (0.348)
- 2006 (0.312)
- 2007 (0.286)
- 2008 (0.219)
- 2009 (0.152)
- 2010 (0.107)

Remaining IANA Free Pool as a fraction of total usable address space (224 /8s)

0.00 as of February, 2011

What we’ll cover

- Basics of IPv6
- IPv6 Addressing Methods
  - SLAAC
  - DHCP
  - Static
  - Privacy
- Linux Configuration for Native Dual Stack
- IPv6 without a native backbone available
- Free IPv6?

Basics: IPv4 vs. IPv6

Property	IPv4 Address	IPv6 Address
Bits	32	128
Total address space	3,758,096,384 unicast 268,435,456 multicast 268,435,456 Experimental/other (Class E, F, G)	42+ Undecilion assignable[1] 297+ Undeciliion IANA reserved[2]
Most prevalent network size	/24 (254 usable hosts)	/64 (18,446,744,073,709,551,616 host addresses)
Notation	Dotted Decimal Octets (192.0.2.239)	Hexidecimal Quads (2001:db8:1234:9fef::1)
Shortening	Suppress leading zeroes per octet	Suppress leading zeroes per quad, longest group of zeroes replaced with ::

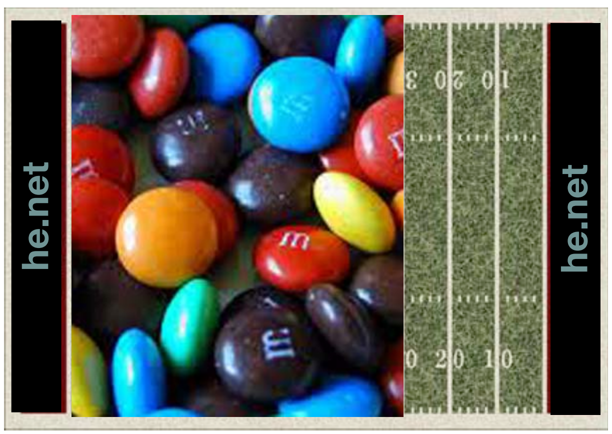
[1]42,535,295,865,117,307,932,921,825,928,971,026,432 assignable unicast (1/8th of total)

[2]297,747,071,055,821,155,530,452,781,502,797,185,024 IANA reserved (7/8th of total)

Network Size and Number of networks (The tasty version)

One IPv4 /24 -- 254 M&Ms

One IPv6 /64 -- Enough M&Ms to fill all 5 of the great lakes.



Full Address Space, One M&M per /24 covers 70% of a football field

Full Address Space, One M&M per /64 fills all 5 great lakes.

Comparison based on Almond M&Ms, not plain. Caution! Do not attempt to eat a /64 worth of any style of M&Ms.

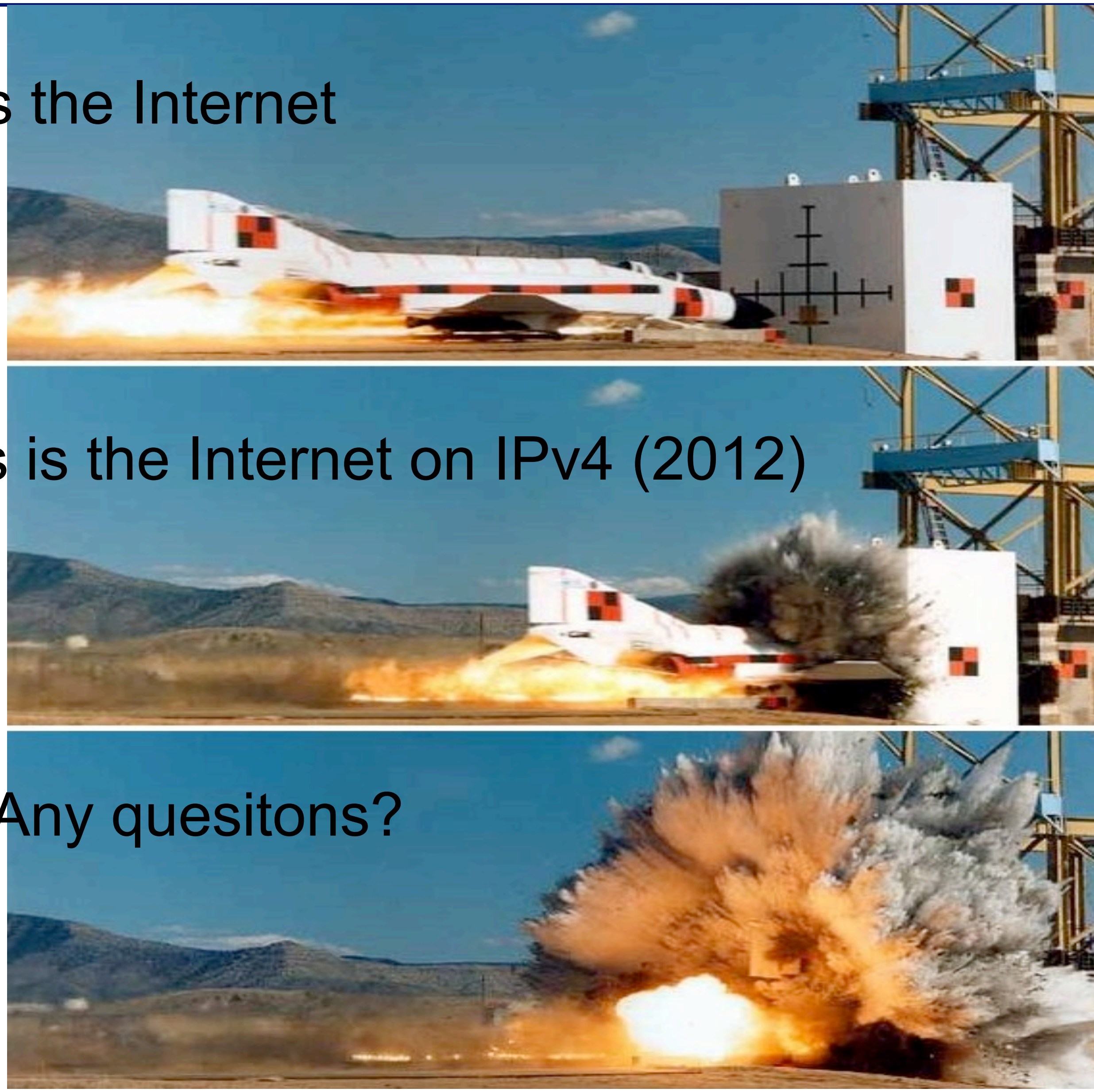
Basics: IPv4 vs. IPv6 thinking

Thought	IPv4 dogma	IPv6 dogma
Assignment Unit	Address (/32)	Network (/64)
Address Optimization	Tradeoff -- Aggregation, Scarcity	Aggregation (At least for this first 1/8th of the address space)
Address Issue Methodology	Sequential, Slow Start, frequent fragmentation	Bisection (minimize fragmentation), issue large, minimal requests for more, aggregate expansions.
NAT	Necessary for address conservation	Not supported, Not needed -- Breaks more than it solves (other than possible NAT64)
Address Configuration	Static, DHCP	Stateless Autoconf, Static, some DHCP (needs work), DHCP-PD (NEW!!)

This is the Internet

This is the Internet on IPv4 (2012)

Any quesitons?



Basics Address Scopes

- Link Local -- fe80::<UUVV:WW>ff:fe<XX:YYZZ> only valid on directly attached subnet.
- Site Local (deprecated) -- Only valid within site, use ULA or global as substitute.
- Unique Local Addresses (ULA) -- Essentially replaces IPv4 RFC-1918, but, more theoretical uniqueness.
- Global -- Pretty much any other address, currently issued from 2000::/3, globally unique and valid in global routing tables.

Basics: Stateless Autoconfiguration

- Easiest configuration
- No host configuration required
- Provides only Prefix and Router information, no services addresses (DNS, NTP, etc.)
- Assumes that all advertising routers are created equal, rogue RA can be pretty transparent to user (RA guard required on switches to avoid)

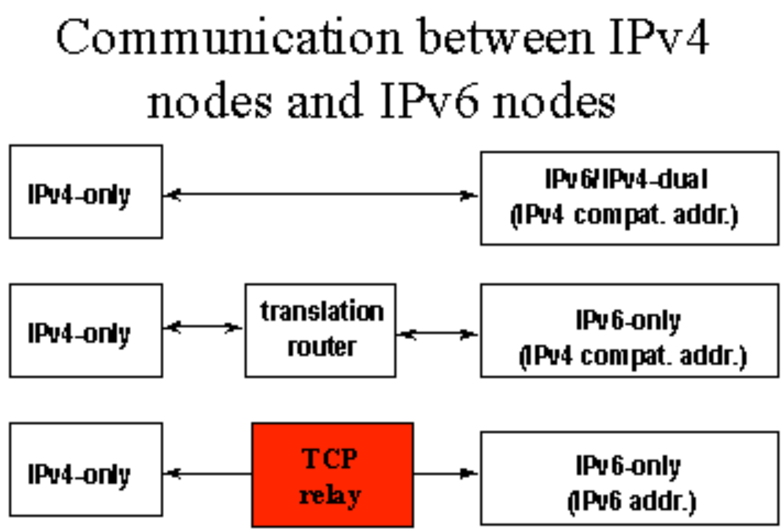
Stateless Autoconfiguration Process

- Host uses MAC address to produce Link Local Address. If MAC is EUI-48, convert to EUI-64 per IEEE process: invert 0x02 bit of first octet, insert 0xFFFE between first 24 bits and last 24 bits fe80::<EUI-64>
- IPv6 shutdown on interface if duplicate detected.
- ICMP6 Router Solicitation sent to All Routers Multicast Group

Stateless Autoconfigration Process (cont.)

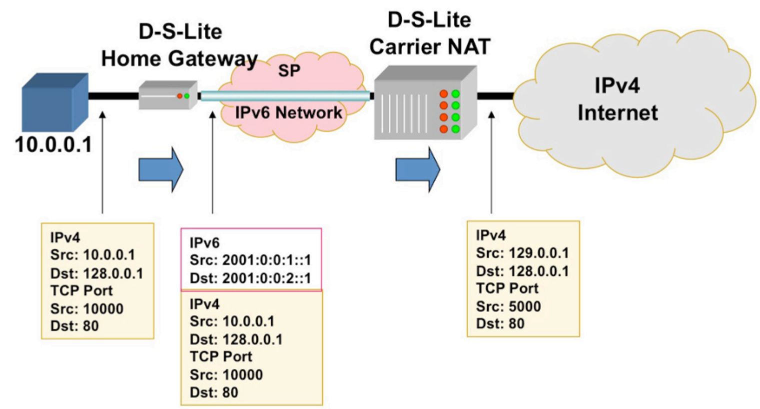
- Routers send ICMP6 Router Advertisement to link local unicast in response. Also sent to All Hosts Multicast group at regular intervals.
- Router Advertisement includes Prefix(es), Preference, Desired Lifetime, Valid Lifetime.
- Host resets applicable Lifetime counters each time valid RA received.
- Address no longer used for new connections after Desired lifetime expires.
- Address removed from interface at end of Valid lifetime.
- Prefix(es)+EUI-64 = Host EUI-64 Global Address, netmask always /64 for SLAAC.

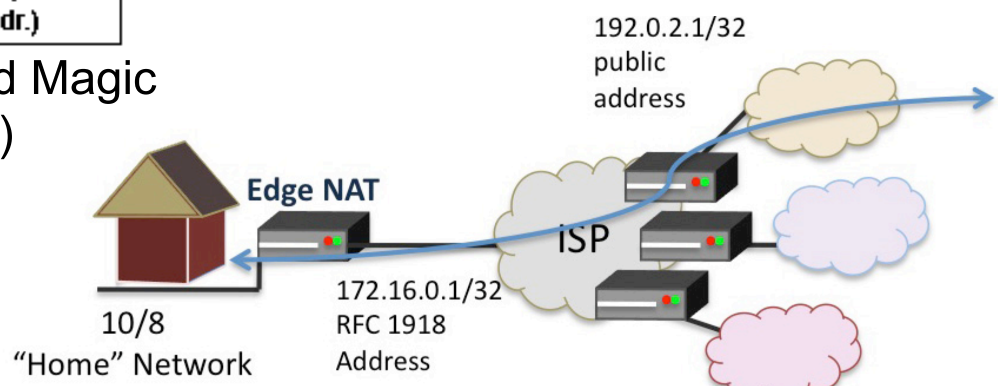
If you think IPv6 is hard, wait until you try any of these.



As yet undefined/unimplemented Magic (TCP relay could be SSH tunnel)

Dual Stack Lite (ISC)





Multiple Layer NAT (Carrier Grade NAT)

DHCPv6

- Can assign prefixes other than /64 -- Ideally larger (/48) prefixes to routers which then delegate various networks automatically downstream, a few limited implementations of this feature.
- Can assign addresses to hosts, cannot provide default router information.
- Can provide additional information about servers (DNS, Bootfile, NTP, etc.)
- Vendor support still lacking in some areas

Static Addressing

- IPv6 can be assigned statically, same as IPv4
- Common to use one of two techniques for IPv4 overlay networks:
  - Prefix::<addr> (first 12 bits of 64 bit <addr> must be 0)
  - Either <addr> is IPv4 last octet(s) expressed as BCD, or <addr> is IPv4 last octet(s) converted to hex.
  - e.g. 192.0.2.154/24 -> 2001:db8:cafe:beef::154/64 (BCD) or 2001:db8:cafe:beef::9a/64 (Hex)
  - These mappings won't conflict with autoconfigured addresses since autoconfigured addresses will never be 000x:xxxx:xxxx:xxxx.

Privacy Addresses

- Essentially a pathological form of Stateless Address Autoconfiguration which uses a new suffix for each flow and obfuscates the MAC address.
- RFC-3041
- Uses MD5 Hash with random component to generate temporary address
- Preferred and Valid lifetimes derived from SLAAC address
- Unfortunate default in Lion and Vista/later

Multiple addresses per interface

- IPv4 has some support for this in most implementations.
- IPv6 has full support for this in all implementations.
- IPv4, multiple addresses/interface are exception.
- IPv6, single address on an interface nearly impossible in useful implementation (link local required, global optional)

IPSEC

- In IPv4, IPSEC is add-on software.
- In IPv6, IPSEC is a required part of any IPv6 implementation
- IPv6 does NOT require IPSEC utilization
- IPSEC is considerably easier to configure in IPv6.
- IPSEC automation may be possible in future IPv6 implementations.

Configuring IPv6 Native on Linux

- Interface Configuration depends on your distro.
- Debian based distros (Debian, Ubuntu, etc.) use /etc/interfaces
- Red Hat based distros (RHEL, Fedora, CentOS) use /etc/sysconfig/network-scripts/ifcfg-<int>

/etc/network/interfaces

```
iface eth0 inet static
   address 192.0.2.127
   netmask 255.255.255.0
   gateway 192.0.2.1
```

IPv4 (Static)

```
iface eth0 inet6 static
   address 2001:db8:c0:0002::7f
   netmask 64
   gateway 2001:db8:c0:0002::1
```

IPv6 (Static)

```
iface eth1 inet6 auto
```

IPv6 (Autoconf)

/etc/sysconfig/network-scripts/ ifcfg-<int>

```
DEVICE=eth0
ONBOOT=yes
IPADDR=192.159.10.2
NETMASK=255.255.255.0
GATEWAY=192.159.10.254
```

IPv4 (Static)

```
IPV6INIT=yes
IPV6ADDR=2620:0:930::0200:1/64
IPV6_DEFAULTGW=2620:0:930::dead:beef
IPV6_AUTOCONF=no
IPV6ADDR_SECONDARIES="\
2001:470:1f00:3142::0200:1/64 \
2001:470:1f00:3142::0200:2/64"
```

IPv6 (Static)

```
IPV6INIT=yes
IPV6_AUTOCONF=yes
```

IPv6 (Autoconf)

IPv6 without a native connection

- Three options (In order of preference)
  - 6in4 -- Tunnel your IPv6 in an IPv4 GRE Tunnel
  - 6to4 -- Tunnel your IPv6 in an auto-tunnel using an any-casted IPv6 mapping service
  - Teredo -- Tunnel your IPv6 in an auto-tunnel using a multi-server auto-configured process defined by Microsoft.

Why 6in4

- GRE is well understood by most networkers
- Simple and deterministic
- No anycast magic -- Simplifies debugging
- Controlled by two endpoint adminsitrators -- Greatly simplifies debugging
- Disadvantage: Manual config, but, not hard.

Why 6to4

- Automatic configuration
- When it works, it's pretty clean and relatively self-optimizing.
- May be good option for mobile devices (laptop, cellphone, etc.)
- Hard to troubleshoot when it doesn't work.
- Disadvantage: Anycast == Non-deterministic debugging process.

Why Teredo?

- Autoconfiguration
- May bypass more firewalls than 6to4
- Enabled by default in Windows (whether you want it or not)
- Meredo available for Linux (client and server)
- Disadvantage: Complicated and tricky to debug if problems occur.

IPv6 For Free? YES!!

- Several tunnel brokers offer free IPv6.
  - My favorite is the HE Tunnelbroker at www.tunnelbroker.net
- If you or your organization has a presence at an exchange point with Hurricane Electric, we currently offer free IPv6 Transit.

Routing

- Usual suspects
  - OSPF (OSPFv3)
  - BGP (BGP4 Address Family inet6)
  - RA and RADVD
  - Support in Quagga and others

Firewalls

- ip6tables much like iptables
  - Excerpt from my ip6tables configuration

```
-A RH-Firewall-1-INPUT -d 2620:0:930::200:2/128 -m state --state NEW -m tcp -p tcp
                                                              --dport 3784 -j ACCEPT
-A RH-Firewall-1-INPUT -d 2620:0:930::200:1/128 -m state --state NEW -m udp -p udp
                                                              --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -d 2001:470:1f00:3142::200:1/128 -m state --state NEW -m udp
                                                              -p udp --dport 53 -j ACCEPT
-A RH-Firewall-1-INPUT -d 2620:0:930::200:2/128 -m state --state NEW -m udp
                                                              -p udp --dport 53 -j ACCEPT
```

DNS

- Forward DNS
  - Instant IPv6 -- Just add AAAA
- Reverse DNS
  - Slightly more complicated
  - ip6.arpa
  - 2620:0:930::200:2 -> 2620:0000:0930:0000:0000:0000:0200:0002
  - 2620:0000:0930:0000:0000:0000:0200:0002 -> 2000:0020:0000:0000:0000:0390:0000:0262
  - 2000:0020:0000:0000:0000:0390:0000:0262 -> 2.0.0.0.0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.3.9.0.0.0.0.0.0.2.6.2.ip6.arpa

DNS -- BIND Configuration

- Current BIND versions ship with IPv6 template zones (hints, rfc1912, etc.)
- IPv6 addresses valid in ACLs just like IPv4, same rules
- Zone configuration identical except reverse zones for IPv6 ranges called “ip6.arpa”:

```
zone "0.3.9.0.0.0.0.0.2.6.2.ip6.arpa" IN {
        type master;
        file "named.2620:0:930::-48.rev";
};
```

DNS -- BIND Configuration

- In IPv6 Reverse Zone files, $ORIGIN is your friend!
- Forward Zones A for IPv4, AAAA for IPv6, basically what you're used to:

```
mailhost                    IN      A       192.159.10.2
                            IN      AAAA    2620:0:930::200:2
```

- Reverse Zones PTR records, as described above:

```
$ORIGIN 0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.3.9.0.0.0.0.0.0.2.6.2.ip6.arpa.
1.0.0.0                     IN      PTR     ns.delong.sj.ca.us.
2.0.0.0                     IN      PTR     owen.delong.sj.ca.us.
4.0.0.0                     IN      PTR     irkutsk.delong.sj.ca.us.
```

DNS -- Reverse DNS Details

- In this example, we see:

```
$ORIGIN 0.0.2.0.0.0.0.0.0.0.0.0.0.0.0.0.3.9.0.0.0.0.0.0.2.6.2.ip6.arpa.
1.0.0.0                    IN        PTR       ns.delong.sj.ca.us.
2.0.0.0                    IN        PTR       owen.delong.sj.ca.us.
4.0.0.0                    IN        PTR       irkutsk.delong.sj.ca.us.
```

- $ORIGIN saves us lots of typing for 2620:0:930::200:
- Each entry contains the 4 hex digits for the last quad (0001, 0002, 0004)
- Note each nibble is a zone boundary

Troubleshooting

- Mostly like troubleshooting IPv4
- Mostly the same kinds of things go wrong
- Just like IPv4, start at L1 and work up the stack until it all works.
- If you are using IPv4 and IPv6 together, may be easier (due to familiarity) to troubleshoot L1-2 on IPv4.

A wee bit about Neighbor Discovery and other tools

- No broadcasts, no ARP
- This is one of the key differences with IPv6.
- Instead an solicited node multicast address is used.
- IPv4: arp 192.0.2.123
- IPv6: ip -f inet6 neigh show 2620:0:930::200:2
- ping -> ping6
- traceroute -> traceroute6
- telnet, ssh, wget, etc. just work

Cool SSH trick

- Special for those that made it through the whole presentation:
- If you have a dual stack host you can SSH to in between an IPv4 only and an IPv6 only host that need to talk TCP, then, you can do this from the client:
- ssh user@dshost -L <lport>:server:<dport>
- Then, from the client, connect to localhost:lport and the SSH tunnel will actually protocol translate the session.

SSH trick example

- myhost -- IPv6-only host 2620:0:930::200:f9
- dshost -- IPv4/v6 dual stack host: 192.159.10.2 and 2620:0:930::200:2
- desthost -- IPv4-only host 192.159.10.100
- On myhost I type:
  - ssh owen@2620:0:930::200:2 -L 8000:192.159.10.100:80
  - Then, I can browse to http://[::1]:8000
- My browser will connect to the ssh tunnel via IPv6, and, the SSH daemon at dshost will pass the contents along via IPv4.

Staff Training

- Hopefully this presentation works towards that.
- You'll need more.
- Plan for it.
- Budget for it.
- Allocate time for it.
- If possible, have the staff being trained leave their pagers/blackberries/iPhones/etc. in the car during training.

A word for Management

- This is _NOT_ a networking problem.
- This is _NOT_ an IT problem.
- This is an institution-wide issue.
  - It will touch every single department from executive management to academics, IT, accounting, marketing, and even physical logistics and facilities.
  - It is a major change to core infrastructure.
  - You are running out of time.
- If you are not in management, you need to be communicating this to your management.

Q&A

?

Contact:

Owen DeLong
IPv6 Evangelist
Hurricane Electric
760 Mission Court
Fremont, CA 94539, USA
http://he.net/

owend at he dot net
+1 (408) 890 7992