

# IPv6

## Present and Future

Chris Cappuccio  
[chris@nmedia.net](mailto:chris@nmedia.net)

BendTel  
*Systems Architect*  
OpenBSD  
*Developer*

# What is IPv6?

- A new version of the base Internet Protocol
- Replaces IPv4

# What is IPv6?

- IPv6 development started with the IETF Internet Engineering Steering Group (IESG) as the Internet Protocol Next Generation (IPng) working group
- Protocol development happens through the Internet Engineering Task Force ([www.ietf.org](http://www.ietf.org))
- Currently an IETF Internet working group

# What is IPv6?

- Core protocol development was based on the work of Robert S. Hinden (Nokia) and Stephen E. Deering (Cisco)
- Current protocol development is based on work of the current IPv6 implementers, although the key architects are still the sole authors of RFC 3513 (issued April, 2003) which defines the core IPv6 standard

# What is IPv6?

- Early IPv6 software development was led by the US Naval Research Laboratory (NRL), the French Institut National de Recherche en Informatique et en Automatique (INRIA), and the Japanese Widely Integrated Distributed Environment (WIDE, a group of over 100 Japanese corporations)
- Current IPv6 software development is led by the Japanese KAME (a collection of researchers and programmers sponsored by six Japanese corporations: Fujitsu, Hitachi, Internet Initiative Japan, NEC, Toshiba, Yokogawa Electric), and to some extent, two US corporations, Cisco and Microsoft

# Core upgrades in IPv6

- Scalability; many more addresses
- Security; mandatory IPsec
  - We all know how standards are ignored
  - Key IPv6 implementers do implement IPsec
- Flow label mechanism; layer 3 identifier
  - Quick, efficient traffic flow identification
  - MPLS
- Plug and Play; dhcp not an afterthought
- Clear and refined specifications

# IP address shortage?

- The rumored shortage is a myth today (but probably not after the next 10 years)
- Total percentage of available IPv4 space announced on the internet today: 35.2%
- Available space announced per RIR
  - APNIC: 66.9%
  - ARIN: 74.3%
  - RIPE: 56.5%
  - LACNIC: 44.4%
  - AfriNIC: 11%

<http://mailman.apnic.net/mailman/listinfo/bgp-stats>

# IP address shortage?

- An IPv4 address is 32 bits long
- IPv4 provides  $2^{32}$ , or 4,294,967,296 addresses
- That's four billion, two hundred ninety-four million, nine hundred sixty-seven thousand, two hundred ninety-six addresses
- Not enough for each person on the planet to control a single IP address

# IP address shortage solution?

- An IPv6 address is 128 bits long
- IPv6 provides  $2^{128}$ , or 340,282,366,920,938,463,463,374,607,431,768,211,456 addresses
- That's three hundred forty undecillion, two hundred eighty-two decillion, three hundred sixty-six nonillion, nine hundred twenty octillion, nine hundred thirty-eight septillion, four hundred sixty-three sextillion, four hundred sixty-three quintillion, three hundred seventy-four quadrillion, six hundred seven trillion, four hundred thirty-one billion, seven hundred sixty-eight million, two hundred eleven thousand, four hundred fifty-six addresses

# IPv4 allocation strategy

- Early Internet IP allocation was based on classes
  - Class A - Y.0.0.0 – Y.255.255.255
  - Class B - Y.Y.0.0 – Y.Y.255.255
  - Class C - Y.Y.Y.0 – Y.Y.Y.255

# IPv4 allocation strategy

- The original internet number crunch was created by the allocation of classful networks
- By providing for only three sizes of allocations to internet participants, the original scheme became unworkable as the Internet grew

# IPv4 allocation strategy

- Using the Class A, B, and C addressing scheme, the Internet could support:
  1. 126 Class A networks with up to 16,777,214 hosts each
  2. 65,000 Class B networks with up to 65,534 hosts each
  3. 1,589,248 Class C networks with up to 256 hosts each
  4. Class D for IP multicast with up to 1,040,400 addresses
  5. Class E for future use with up to 1,040,400 addresses

# IPv4 allocation strategy

- Classful allocations proved to be inefficient, and so classful subnet allocation started (RFC 917, RFC 950, RFC 1219)
- A sub-net is allocated on byte boundaries within a classful network

# IPv4 allocation strategy

- Classless Inter-Domain Routing (CIDR) was created due to routing table size/aggregation problems with subnets
- Following the introduction of CIDR, IP addresses have been allocated by prefix length
  - CIDR allocates networks sized any power of 2, from 1 to 32 bits
  - Hence, slash notation: 192.168.1.0/24
  - RFC 1517, RFC 1518, RFC 1519, RFC 1520

# IPv4 allocation strategy

- CIDR is an addressing scheme which allows for more efficient allocation of IP addresses than the old Class A, B, and C address schemes or subnet schemes.
- CIDR replaces classful networking and classful subnets, and makes the concepts of Class A, B, and C obsolete
- Subnets which match CIDR allocations (powers of 2) continue to exist in our minds, although ISPs generally refer to them by their CIDR notation and not subnet notation
  - These standard subnets defined in RFC 1878

# IPv4 allocation strategy

- By eliminating Class A, B, C, and subnets, and by turning network announcements into generalized prefixes, CIDR allows for network route announcements to be aggregated and thus IP space is conserved more highly
- Today, IPv4 networks are allocated upon demonstrated need by ARIN, APNIC, RIPE, LACNIC, and AfriNIC according to CIDR network prefix lengths, that is, /8, /9, ... /16, /17, /18, /19, /20, /21 ...
- Prefix lengths are efficient because they are easy to type, easy to remember, easy for computers and routers to work with, and easy to write software for

# IPv6 allocation strategy

- Like IPv4+CIDR, IPv6 networks have a prefix length. In fact, the concept of a netmask is deprecated under IPv6. IPv6 has no netmask, only a prefix length (Note: you might see it called bitlength or other names)
- You could use a netmask with IPv6, but since IPv6 addresses are so long, it's much simpler to use *address-slash-prefixlength* notation
  - *No software will come with a place to enter a netmask for an IPv6 address*

# IPv6 allocation strategy

- The current IPv6 allocation guidelines are defined in RFC 3177
- It looks something like a combination of CIDR and Classful networking. I like to think of it as the best of IPX (Yes, I said IPX) and the best of CIDR
- These guidelines are not set in stone, and could change in the next 5 to 10 years if people find a better way

# IPv6 allocation strategy

- Unlike IPv4 CIDR, IPv6 networks are assigned in fixed amounts
  - That sounds like classful networking all over, doesn't it?
- ISPs are allocated IP space such that every ISP has a /32
- That means 128-32 (96) bits are left for the ISP to allocate. That's  $2^{96}$  addresses for each ISP!
- It gets even better

# IPv6 allocation strategy

- Because we have so many IPs under IPv6, things get interesting
  - Each ISP customer gets a /48 allocation
  - As a customer, each network that you allocate internally gets a /64
  - So, as a customer, you get  $2^{(64-48)}$  internal networks to allocate (That comes out to 65536 networks)
  - Each device you allocate gets a /128 allocation (that's one single IPv6 address)

# IPv6 allocation strategy

- There is a point to all of this
- By having networks of /64 size, the next 64 bits get filled with the 48 bit IEEE 802 MAC address of each of your devices (and 16 bits of whatever)
- No longer do you have to setup an IP address
- Now you get to use IPv6 auto configuration (stateless, RFC 2462) or DHCPv6 (stateful, RFC 3315)
- DHCPv6 can manage your name server on top of everything else, and you can just refer to everything by name
  - Of course, you may be doing this already with IPv4

# IPv6 allocation strategy

- You will use /64 networks because you want to use IPv6 auto configuration
- Neighbor Discovery (RFC 2461) replaces ARP
- Every computer plugged into the network uses Neighbor Discovery to find other routers, hosts, and to determine the network address

# IPv6 auto configuration example

- Here's where “the best of IPX” comes in (It's obviously where this idea came from. This idea is practical now that IPv6 has more than 48 bits to play with)
  - Let's say you have a host with a MAC address of 00:AC:80:12:41:E1
  - Let's also say that your IPv6 ethernet starts with 2424:1101:1011:FFFF
  - Once this host talks with neighbor discovery (RFC 2461), it could auto-configure its IP address as:
    - 2424:1101:1011:FFFF::00AC:8012:41E1
    - More on IPv6 address style soon

# IPv6 allocation strategy

- The summary here is:
  - Each provider will have enough addresses to assign 65,536 /48 IP ranges to as many customers
  - Those customers will have enough addresses to assign 65,536 /64 IP ranges to as many networks.
  - Each /64 network will auto-configure itself
  - Renumbering if you move between providers will be easy (in theory)

# IPv4 and IPv6 addresses

- An IPv4 address is a 32 bit number that can be represented in many ways:
  - Decimal Number: 3506561041
  - Decimal Octets: 209.1.224.17
  - Hex Number: d101e011
- You could even use binary
- *The point here is that they are all the same thing.* That is, a 32 bit number that identifies an internet address, usually a host (naturally, they can also identify a network address, netmask, broadcast address or other special address)

# IPv4 and IPv6 addresses

- IPv6 could also be represented in many different ways. For instance, you could try the decimal octet method
- Don't forget that IPv6 addresses are numbers that are 128 bits long
- Decimal Octet
  - IPv4: 128.252.135.4
  - IPv6:

252.19.208.145.232.121.212.114.35.42.104.142.12.0.0.1

# IPv6 addresses

- Using a decimal octet notation for IPv6 would be highly cumbersome
- IPv6 addressing uses a hex format so that now **each group conveys 16 bits** of information instead of **just 8 bits per octet**
- RFC 3513 Section 2 defines the accepted text representation for IPv6 addresses using 16 bit hex groups

# IPv6 addresses

- RFC 3513 says:

“The preferred form is x:x:x:x:x:x:x:x, where the 'x's are the hexadecimal values of the eight 16-bit pieces of the address.

Examples:

**FEDC:BA98:7654:3210:FEDC:BA98:7654:3210**

**1080:0:0:0:8:800:200C:417A”**

- Even with hex representation, the address is still rather long and unwieldy

# IPv6 addresses

- There are other ways to shorten IPv6 addresses, particularly since many allocations will include zeros in several groups
  - For instance, FFFF::0101:3030 fills the middle :: portion with an appropriate number of groups with all zeros
- This is in the RFC 3513. Section 2 defines this behavior (and is easy to read)

# IPv6 transition

- IPv6 defines many other new standards besides DHCPv6 and Neighbor Discovery
  - It is similar enough to IPv4 that it does not require huge changes to application logic to support IPv6
  - It is different enough from IPv4 that it generally requires a rewrite of the IP layer in a TCP/IP stack to add IPv6 support

# IPv6 transition

- There are several transition mechanisms
  - Dual stack support: IPv4 and IPv6
    - Changes required on every single router and host that wants to participate in IPv6
    - The ultimate goal is for the entire Internet to be dual stack
  - Tunnel Broker (RFC 3053)
    - Much of the IPv6 internet runs through tunnels today
    - Changes not required on every single router, only those participating in IPv6 networks
    - ISATAP
      - SRI RAND
  - Translation
    - IPv4-v6 IPv6-v4 NAT

# IPv6 users today

- US Government
  - NSF
    - vBNS+
  - DoD
    - October, 2003, Full Implementation 2008 ?
- Tunnel Networks
  - 6bone <http://www.6bone.net>
  - ISP
- Asia
- Europe

# NAT

- You probably use NAT today
  - NAT breaks things
  - It interferes with the end-to-end nature of the Internet and its protocols
    - Or else you wouldn't have to work around it all the time with proxies and strange firewall rule sets
  - NAT is no longer necessary with IPv6
    - Now you have plenty of IP addresses

# Stateful Packet Filtering

- Stateful Packet Filtering
  - Be prepared to understand it
    - Google is a start
  - Make sure your firewall supports it
    - The consumer fare of Linksys, D-Link, Netgear, and others often do not
  - Any good NAT implementation already makes use of a quality connection state engine
    - Once again this excludes Linksys, D-Link, and Netgear because they often don't use a real state engine at all

# Stateful Packet Filtering

- Point your firewall vendor to this URL if they do not understand these concepts
  - [http://home.iae.nl/users/guido/papers/tcp\\_filtering.ps.gz](http://home.iae.nl/users/guido/papers/tcp_filtering.ps.gz)
- The game is already lost if your vendor doesn't understand this
  - Wait, this is a talk on IPv6
- Make sure you have this so you can avoid NAT in the future

# IPv6 Applications

- You will have enough IP addresses on your networks for every toaster, refrigerator and light switch to participate
  - Do you really want your toaster on the Internet?
- “Next Generation” applications
- Fixes several problems of the current generation as well

# IPsec

- Now IPsec is ground-up, not an outside layer
  - Every device must implement IPsec to be IPv6 compliant
  - Properly implemented end-to-end cryptography solves confidentiality issues
  - Application layer vulnerabilities remain the target

You can find a copy of this  
presentation online at:

<http://www.bendtel.net/ipv6/bug-ipv6-may14.pdf>