IP Next Generation (IPv6)

Shivkumar Kalyanaraman
Rensselaer Polytechnic Institute
shivkuma@ecse.rpi.edu
http://www.ecse.rpi.edu/Homepages/shivkuma

Overview

- Limitations of current Internet Protocol (IP)
- How many addresses do we need?
- IPv6 Addressing
- IPv6 header format
- IPv6 features: routing flexibility, plug-n-play, multicast support, flows

IP Addresses

- Example: 164.107.134.5
  - 1010 0100 : 0110 1011 : 1000 0110 : 0000 0101
  - A4:6B:86:05 (32 bits)

- Maximum number of address = 2^{32} = 4 Billion
- Class A Networks: 15 Million nodes
- Class B Networks: 64,000 nodes or less
- Class C Networks: 250 nodes or less

IP Address Format

- Three all-zero network numbers are reserved
- 127 Class A + 16,381 Class B + 2,097,151 Class C networks = 2,113,659 networks total
- Class B is most popular.
- 20% of Class B were assigned by 7/90 and doubling every 14 months \( \Rightarrow \) Will exhaust by 3/94

Question: Estimate how big will you become?
Answer: More than 256!
Class C is too small. Class B is just right.

How Many Addresses?

- 10 Billion people by 2020
- Each person has more than one computer
- Assuming 100 computers per person \( \Rightarrow \) 10^{12} computers
- More addresses may be required since
  - Multiple interfaces per node
  - Multiple addresses per interface
  - Some believe 2^{32} to 2^{48} addresses per host
- Safety margin \( \Rightarrow \) 10^{15} addresses

IPvng Requirements \( \Rightarrow \) 10^{12} end systems and 10^{10} networks. Desirable 10^{12} to 10^{15} networks

How big an address space?

- H Ratio \( = \log_{10}(\text{# of objects})/\text{available bits} \)
- \( 2^{n} \) objects with \( n \) bits: \( \text{H-Ratio} = \log_{10}2^{n} = 0.30103 \)
- French telephone moved from 8 to 9 digits at 10^{7} households \( \Rightarrow \) H = 0.25 (~3.3 bits/digit)
- US telephone expanded area codes with 10^{8} subscribers \( \Rightarrow \) H = 0.24
- Physics/space science net stopped at 15000 nodes using 16-bit addresses \( \Rightarrow \) H = 0.26
- 3 Million Internet hosts currently using 32-bit addresses \( \Rightarrow \) H = 0.20 \( \Rightarrow \) A few more years to go
IPv6 Addresses
- 128-bit long. Fixed size
- \(2^{128} = 3.4 \times 10^{38}\) addresses
- \(665 \times 10^{21}\) addresses per sq. m of earth surface
- If assigned at the rate of \(10^6\) /µs, it would take 20 years
- Expected to support \(8 \times 10^{17}\) to \(2 \times 10^{33}\) addresses
- \(8 \times 10^{17} \Rightarrow 1,564\) address per sq. m

IPv6 Addresses (Continued)
- Allows multiple interfaces per host.
- Allows multiple addresses per interface
- Allows unicast, multicast, anycast
- Allows provider based, site-local, link-local
- 85% of the space is unassigned

Colon-Hex Notation
- Dot-Decimal: 127.23.45.88
- Colon-Hex:
  - FEDC:0000:0000:0000:3243:0000:0000:ABCD
  - Can skip leading zeros of each word
  - Can skip one sequence of zero words, e.g., FEDC::3243:0000:0000:ABCD or :3243:0000:0000:ABCD
  - Can leave the last 32 bits in dot-decimal, e.g., ::127.23.45.88
  - Can specify a prefix by /length, e.g., 2345:BA23:7::/40

IPv6 vs IPv4
- IPv6 only twice the size of IPv4 header
- Only version number has the same position and meaning as in IPv4
- **Removed:**
  - Header length, fragmentation fields (identification, flags, fragment offset), header checksum
- **Replaced:**
  - Datagram length by payload length
  - Protocol type by next header
  - Time to live by hop limit
  - Type of service by “class” octet
- **Added:**
  - Flow label
  - All fixed size fields.

IPv6 vs IPv4 (Continued)
- No optional fields. Replaced by extension headers.
- **Idea:** avoid unnecessary processing by intermediate routers while not sacrificing the flexibility possible due to options
- Next Header = 6 (TCP), 17 (UDP), etc
Extension Headers

- Most extension headers are examined only at destination.
- Routing: Loose or tight source routing.
- Fragmentation: All IPv6 routers can carry 536 Byte payload.
- Authentication.
- Hop-by-Hop Options.
- Destination Options:

Extension Header (Continued)

- Only Base Header:
  - Base Header: TCP
    - Next = TCP
    - Segment

- Only Base Header and One Extension Header:
  - Base Header: Route Header
    - Next = TCP
    - Segment

- Only Base Header and Two Extension Headers:
  - Base Header: Route Header
    - Next = Auth
    - Next = TCP
    - Segment

Fragmentation

- Routers cannot fragment. Only source hosts can.
  ⇒ Need path MTU discovery or tunneling.
- Fragmentation requires an extension header.
- Payload is divided into pieces.
- A new base header is created for each fragment.

IPv6 Addressing and Routing

- Aggregatable Global Unicast Addresses.
- Link-local and Site-local addresses.
- Multicast and Anycast support.
- Provider-based inter-domain routing & IDRP.

Initial IPv6 Prefix Allocation

<table>
<thead>
<tr>
<th>Allocation</th>
<th>Prefix</th>
<th>Allocation</th>
<th>Prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td>00000000 Unassigned 101</td>
<td>Unassigned</td>
<td>110</td>
</tr>
<tr>
<td>NSAP</td>
<td>0000001 Unassigned 110</td>
<td>IPX</td>
<td>0000010 Unassigned 1111 0</td>
</tr>
<tr>
<td>Unassigned</td>
<td>0000011 Unassigned 1111 10</td>
<td>Unassigned</td>
<td>0001 Unassigned 1111110</td>
</tr>
<tr>
<td>Unassigned</td>
<td>001 Unassigned 1111110 0</td>
<td>Provider-based* 010</td>
<td>Link-Local 1111110 10</td>
</tr>
<tr>
<td>Unassigned</td>
<td>011 Site-Local 1111110 11</td>
<td>Geographic 100</td>
<td>Multicast 11111111</td>
</tr>
</tbody>
</table>

Aggregatable Global Unicast Addresses

- Address allocation: “provider-based” plan.
- Format: TLA + NLA + SLA + 64-bit interface ID.
- TLA = “Top level aggregator.” (13 bits).
- Ranges of TLA values allocated to various registries.
- For “backbone” providers or “exchange points.”

*Has been renamed as “Aggregatable global unicast.”
Aggregatable Global Unicast Addresses (Continued)
- NLA = “Next Level Aggregator” (32 bits)
- Second tier provider and a subscriber
- More levels of hierarchy possible within NLA
- SLA = “Site level aggregator” = 16 bits for link
- Renumbering after change of provider => change the TLA and NLA. But have same SLA & I/f ID

Local-Use Addresses
- Link Local: Not forwarded outside the link,
  FE:80::xxx
  - 10 bits n bits 118-n
  - 1111 1110 10 0 Interface ID
- Site Local: Not forwarded outside the site,
  FE:C0::xxx
  - 10 bits n bits m bits 18-n-m bits
  - 1111 1110 11 0 Subnet ID Interface ID
- Provides plug and play

Multicast Addresses
- 8 bits 4 bits 4 bits 112 bits
- Flags: 1111 1111 Scope Group ID
  - All routers recognize this format
  - 0 All routers can route multicast packets.
  - 1 Also IGMP part of ICMPv6 => required.
  - T = 0 => Permanent (well-known) multicast
    address, 1 => Transient

Multicast Addresses (Continued)
- Scope: 1 Node-local, 2 Link-local, 5 Site-local,
  8 Organization-local, E Global => routers reqd to
  honor this.
- Predefined: 1 => All nodes, 2 => Routers,
  1:0 => DHCP servers

Multicast & Anycast
- Scoping. Eg: 43 => NTP Servers
  - FF01::43 => All NTP servers on this node
  - FF02::43 => All NTP servers on this link
  - FF05::43 => All NTP servers in this site
  - FF08::43 => All NTP servers in this org.
  - FF0F::43 => All NTP servers in the Internet
- Structure of Group ID:
  - First 80 bits = zero (to avoid risk of group
    collision, because IP multicast mapping uses
    only 32 bits)
Address Auto-configuration
- Allows plug and play
- BOOTP and DHCP are used in IPv4
- DHCPng will be used with IPv6
- Two Methods: Stateless and Stateful
- Stateless:
  - A system uses link-local address as source and multicasts to "All routers on this link"
  - Router replies and provides all the needed prefix info
- Stateful:
  - All prefixes have a associated lifetime
  - System can use link-local address permanently if no router
  - Stateful:
    - Problem w stateless: Anyone can connect
    - Routers ask the new system to go DHCP server (by setting managed configuration bit)
    - System multicasts to "All DHCP servers"
    - DHCP server assigns an address

Address Auto-configuration (Continued)

ICMPv6: Neighbor Discovery
- ICMPv6 combines regular ICMP, ARP, Router discovery and IGMP.
- The “neighbor discovery” is a generalization of ARP & router discovery.
- Source maintains several caches:
  - destination cache: dest -> neighbor mapping
  - neighbor cache: neighbor IPv6 -> link address
  - prefix cache: prefixes learnt from router advertisements
  - router cache: router IPv6 addresses

Neighbor Discovery (Continued)
- Old destination => look up destination cache
- If new destination, match the prefix cache. If match => destination local!
- Else select a router from router cache, use it as the next-hop (neighbor).
- Add this neighbor address to the destination cache
- Solicitation-advertisement model:
  - Multicast solicitation for neighbor media address if unavailable in neighbor cache
  - Neighbor advertisement message sent to soliciting station.

Real-Time/QoS Support
- Flow label and the “class” octet field
- Flow = sequence of packets from a single source to a particular (unicast/multicast) destinations requiring special handling by intermediate routers
- Class field can simplify packet classification (data-plane support for QoS)
  - “Class” field assignments being worked upon by differentiated services group at IETF

Transition Mechanisms
- Dual-IP Hosts, Routers, Name servers
- Tunneling IPv6 over IPv4
- Nodes can be partially upgraded to IPv6
- It is better (though not required) to upgrade routers before upgrading hosts
Application Issues

- Most application protocols will have to be upgraded: FTP, SMTP, Telnet, Rlogin
- 27 of 51 Full Internet standards, 6 of 20 draft standards, 25 of 130 proposed standards will be revised for IPv6
- No checksum -> checksum at upper layer is mandatory, even in UDP
- non-IETF standards: X-Open, Kerberos, ... will be updated
- Should be able to request and receive new DNS records

Summary

- IPv6 uses 128-bit addresses
- Allows provider-based, site-local, link-local, multicast, anycast addresses
- Fixed header size, Extension headers instead of options for provider selection, security etc
- Allows auto-configuration
- Dual-IP router and host implementations for transition