IPv6 Tutorial

ICANN
Sao Paulo
December, 2006

Jordi Palet
(jordi.palet@consulintel.es)
Why a New IP?

Only compelling reason: more addresses!

– for billions of new devices, e.g., cell phones, PDAs, appliances, cars, etc.

– for billions of new users, e.g., in China, India, etc.

– for “always-on” access technologies, e.g., xDSL, cable, ethernet-to-the-home, etc.
But Isn’t There Still Lots of IPv4 Address Space Left?

• ~ Half the IPv4 space is unallocated
  – if size of Internet is doubling each year, does this mean only one year’s worth?!
• No, because today we deny unique IPv4 addresses to most new hosts
  – we make them use methods like NAT, PPP, etc. to share addresses
• But new types of applications and new types of access need unique addresses!
Why Are NAT’s Not Adequate?

• They won’t work for large numbers of “servers”, i.e., devices that are “called” by others (e.g., IP phones)
• They inhibit deployment of new applications and services
• They compromise the performance, robustness, security, and manageability of the Internet
Incidental Benefits of Bigger Addresses

- Easy address auto-configuration
- Easier address management/delegation
- Room for more levels of hierarchy, for route aggregation
- Ability to do end-to-end IPsec (because NATs not needed)
Incidental Benefits of New Deployment

• Chance to eliminate some complexity, e.g., in IP header
• Chance to upgrade functionality, e.g., multicast, QoS, mobility
• Chance to include new enabling features, e.g., binding updates
Summary of Main IPv6 Benefits

• Expanded addressing capabilities
• Server-less autoconfiguration (“plug-n-play”) and reconfiguration
• More efficient and robust mobility mechanisms
• Built-in, strong IP-layer encryption and authentication
• Streamlined header format and flow identification
• Improved support for options / extensions
Why Was 128 Bits Chosen as the IPv6 Address Size?

• Some wanted fixed-length, 64-bit addresses
  – easily good for $10^{12}$ sites, $10^{15}$ nodes, at .0001 allocation efficiency (3 orders of mag. more than IPng requirement)
  – minimizes growth of per-packet header overhead
  – efficient for software processing

• Some wanted variable-length, up to 160 bits
  – compatible with OSI NSAP addressing plans
  – big enough for autoconfiguration using IEEE 802 addresses
  – could start with addresses shorter than 64 bits & grow later

• Settled on fixed-length, 128-bit addresses
  – (340,282,366,920,938,463,463,374,607,431,768,211,456 in all!)
### What Ever Happened to IPv5?

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0–3</td>
<td>unassigned</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>IPv4</td>
<td>(today’s widespread version of IP)</td>
</tr>
<tr>
<td>5</td>
<td>ST</td>
<td>(Stream Protocol, not a new IP)</td>
</tr>
<tr>
<td>6</td>
<td>IPv6</td>
<td>(formerly SIP, SIPP)</td>
</tr>
<tr>
<td>7</td>
<td>CATNIP</td>
<td>(formerly IPv7, TP/IX; deprecated)</td>
</tr>
<tr>
<td>8</td>
<td>PIP</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>9</td>
<td>TUBA</td>
<td>(deprecated)</td>
</tr>
<tr>
<td>10-15</td>
<td>unassigned</td>
<td></td>
</tr>
</tbody>
</table>
IPv6 Tutorial

Header Formats
RFC2460

- Internet Protocol, Version 6: Specification
- Changes from IPv4 to IPv6:
  - Expanded Addressing Capabilities
  - Header Format Simplification
  - Improved Support for Extensions and Options
  - Flow Labeling Capability
  - Authentication and Privacy Capabilities
**IPv4 Header Format**

- 20 Bytes + Options

<table>
<thead>
<tr>
<th>bits:</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>20</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>H. Length</td>
<td>TOS</td>
<td>Total Length</td>
<td>Identification</td>
<td>Flags</td>
</tr>
<tr>
<td>Time To Live</td>
<td>Protocol</td>
<td></td>
<td></td>
<td>Header Checksum</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>32 bits Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>32 bits Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Options</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Modified Field
- Deleted Field
IPv6 Header Format

- From 12 to 8 Fields (40 bytes)

<table>
<thead>
<tr>
<th>bits:</th>
<th>4</th>
<th>12</th>
<th>16</th>
<th>24</th>
<th>32</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>Class of Traffic</td>
<td>Flow Label</td>
<td>Payload Length</td>
<td>Next Header</td>
<td>Hop Limit</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- 128 bits Source Address
- 128 bits Destination Address

- Avoid checksum redundancy
- Fragmentation end to end
Summary of Header Changes

- 40 bytes
- Address increased from 32 to 128 bits
- Fragmentation and options fields removed from base header
- Header checksum removed
- Header length is only payload (because fixed length header)
- New Flow Label field
- TOS -> Traffic Class
- Protocol -> Next Header (extension headers)
- Time To Live -> Hop Limit
- Alignment changed to 64 bits
Extension Headers

• “Next Header” Field

IPv6 Header
Next Header = TCP

TCP Header
DATA

IPv6 Header
Next Header = Routing

Routing Header
Next Header = TCP

TCP Header
DATA

IPv6 Header
Next Header = Security

Security Header
Next Header = Fragmentation

Fragmentation Header
Next Header = TCP

TCP Header
DATA

The IPv6 Portal
Consulting in Telecommunications
Extension Headers Goodies

• Processed Only by Destination Node
  – Exception: Hop-by-Hop Options Header

• No more “40 byte limit” on options (IPv4)

• Extension Headers defined currently:
  – Hop-by-Hop Options
  – Routing
  – Fragment
  – Authentication (RFC 2402, next header = 51)
  – Encapsulating Security Payload (RFC 2406, next header = 50)
  – Destination Options
Control Plane IPv4 vs. IPv6

IPv6

ND
ICMPv6
MLD

IPv4

ICMP
IGMPv2

ARP

Ethernet

Multicast
Broadcast
Multicast
IPv6 Tutorial

Addressing and Routing
Text Representation of Addresses

“Preferred” form: 1080:0:FF:0:8:800:200C:417A
Compressed form: FF01:0:0:0:0:0:0:43
IPv4-compatible: 0:0:0:0:0:0:13.1.68.3
or ::13.1.68.3

URL: http://[FF01::43]/index.html
Address Types

Unicast (one-to-one)
  – global
  – link-local
  – site-local (deprecated)
  – Unique Local (ULA)
  – IPv4-compatible
Multicast (one-to-many)
Anycast (one-to-nearest)
Reserved
## Address Type Prefixes

<table>
<thead>
<tr>
<th>address type</th>
<th>binary prefix</th>
</tr>
</thead>
<tbody>
<tr>
<td>IPv4-compatible</td>
<td>0000...0 (96 zero bits)</td>
</tr>
<tr>
<td>Global unicast</td>
<td>001</td>
</tr>
<tr>
<td>Link-local unicast</td>
<td>1111 1110 10</td>
</tr>
<tr>
<td>Site-local unicast</td>
<td>1111 1110 11 (deprecated)</td>
</tr>
<tr>
<td>ULA</td>
<td>1111 110x (1= Locally assigned)</td>
</tr>
<tr>
<td>Multicast</td>
<td>1111 1111</td>
</tr>
</tbody>
</table>

- All other prefixes reserved (approx. 7/8ths of total)
- Anycast addresses allocated from unicast prefixes
### Aggregatable Global Unicast Addresses (RFC2374)

(Deprecated)

<table>
<thead>
<tr>
<th>001</th>
<th>TLA</th>
<th>NLA*</th>
<th>SLA*</th>
<th>Interface ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Public Topology</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(45 bits)</td>
</tr>
</tbody>
</table>

- TLA = Top-Level Aggregator
- NLA* = Next-Level Aggregator(s)
- SLA* = Site-Level Aggregator(s)

- TLAs may be assigned to ISPs and IX

Addresses (RFC2374)

• Deprecation notice: This allocation is no longer recommended for new deployments.
Global Unicast Addresses (RFC3587)

- The global routing prefix is a value assigned to a zone (site, a set of subnetworks/links)
  - It has been designed as an hierarchical structure from the Global Routing perspective
- The subnetwork ID, identifies a subnetwork within a site
  - Has been designed to be an hierarchical structure from the site administrator perspective
- The Interface ID is build following the EUI-64 format
Global Unicast Addresses in Production Networks

<table>
<thead>
<tr>
<th>Sub-network ID (16 bits)</th>
<th>Interface ID (64 bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Routing Prefix (45 bits)</td>
<td>Sub-network ID (16 bits)</td>
</tr>
</tbody>
</table>

- LIRs receive by default /32
  - Production addresses today are from prefixes 2001, 2003, 2400, 2800, etc.
  - Can request for more if justified
- /48 used only within the LIR network, with some exceptions for critical infrastructures
- /48 to /128 is delegated to end users
  - Recommendations following RFC3177 and current policies
    - /48 general case, /47 if justified for bigger networks
    - /64 if only and only one network is required
    - /128 if it is sure that only and only one device is going to be connected
### Global Unicast Addresses for the 6Bone

**TLA 1FFE (hex) assigned to the 6Bone**

- Thus, 6Bone addresses start with 3FFE:
  - (binary 001 + 1 1111 1111 1110)

- Next 12 bits hold a “pseudo-TLA” (pTLA)
  - Thus, each 6Bone pseudo-ISP gets a /28 prefix

- Not to be used for production IPv6 service

![Diagram](attachment:image.png)

<table>
<thead>
<tr>
<th>001</th>
<th>TLA</th>
<th>pTLA</th>
<th>NLA*</th>
<th>SLA*</th>
<th>Interface ID</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>64 bits</td>
</tr>
</tbody>
</table>

- 6Bone: experimental IPv6 network used for testing only

**Until 06/06/06!**
Link-Local & Site-Local Unicast Addresses

Link-local addresses for use during auto-configuration and when no routers are present:

```
1111111010 0 interface ID
```

Site-local addresses for independence from changes of TLA / NLA* (deprecated !):

```
1111111011 0 SLA* interface ID
```
Unique Local IPv6 Unicast Addresses
IPv6 ULA (RFC4193)

• Globally unique prefix with high probability of uniqueness
• Intended for local communications, usually inside a site
• They are not expected to be routable on the Global Internet
• They are routable inside of a more limited area such as a site
• They may also be routed between a limited set of sites
• Locally-Assigned Local addresses
  – vs Centrally-Assigned Local addresses
IPv6 ULA Characteristics

• Well-known prefix to allow for easy filtering at site boundaries

• ISP independent and can be used for communications inside of a site without having any permanent or intermittent Internet connectivity

• If accidentally leaked outside of a site via routing or DNS, there is no conflict with any other addresses

• In practice, applications may treat these addresses like global scoped addresses
IPv6 ULA Format

- Format:

<table>
<thead>
<tr>
<th>Prefix</th>
<th>L</th>
<th>global ID</th>
<th>subnet ID</th>
<th>interface ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 bits</td>
<td>1</td>
<td>40 bits</td>
<td>16 bits</td>
<td>64 bits</td>
</tr>
</tbody>
</table>

- FC00::/7 Prefix identifies the Local IPv6 unicast addresses
- L = 1 if the prefix is locally assigned
- L = 0 may be defined in the future
- ULA are created using a pseudo-randomly allocated global ID
  - This ensures that there is not any relationship between allocations and clarifies that these prefixes are not intended to be routed globally
Centrally Assigned Unique Local IPv6 Unicast Addresses (1)

- Centrally-Assigned Local addresses
  - vs Locally-Assigned Local addresses

- Latest Draft:
  - draft-ietf-ipv6-ula-central-01.txt
  - February 2005
  - No longer active
  - It defines the characteristics and requirements for Centrally assigned Local IPv6 addresses in the framework defined in IPv6 ULA – RFC4193
Centrally Assigned Unique Local IPv6 Unicast Addresses (2)

• The major difference between both assignments:
  – the Centrally-Assigned is uniquely assigned and the assignments can be escrowed to resolve any disputes regarding duplicate assignments

• It is recommended that sites planning to use Local IPv6 addresses use a centrally assigned prefix as there is no possibility of assignment conflicts. Sites are free to choose either approach

• The allocation procedure for creating global-IDs for centrally assigned local IPv6 addresses is setting L=0. Remember that the allocation procedure for locally assigned local IPv6 addresses is thru L=1, as is defined in RFC4193
Interface IDs

The lowest-order 64-bit field of unicast addresses may be assigned in several different ways:

- auto-configured from a 48-bit MAC address (e.g., Ethernet address), expanded into a 64-bit EUI-64
- assigned via DHCP
- manually configured
- auto-generated pseudo-random number (to counter some privacy concerns)
- possibly other methods in the future
# IPv6 in Ethernet

<table>
<thead>
<tr>
<th>48 bits</th>
<th>48 bits</th>
<th>16 bits</th>
<th>IPv6 Header and Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet Destination Address</td>
<td>Ethernet Source Address</td>
<td>100011011011101 (86DD)</td>
<td></td>
</tr>
</tbody>
</table>
Some Special-Purpose Unicast Addresses

• The unspecified address, used as a placeholder when no address is available:

  0:0:0:0:0:0:0:0

• The loopback address, for sending packets to self:

  0:0:0:0:0:0:0:1
Multicast Addresses

- Low-order flag indicates permanent/transient group; three other flags reserved
- Scope field:
  1 - node local
  2 - link-local
  5 - site-local
  8 - organization-local
  B - community-local
  E - global
  (all other values reserved)
Routing

• Uses same “longest-prefix match” routing as IPv4 CIDR

• Straightforward changes to existing IPv4 routing protocols to handle bigger addresses
  – unicast: OSPF, RIP-II, IS-IS, BGP4+, …
  – multicast: MOSPF, PIM, …

• Can use Routing header with anycast addresses to route packets through particular regions
  – e.g., for provider selection, policy, performance, etc.
IPv6 Tutorial

Mobility
IPv6 Mobility

• A mobile host has one or more home address(es)
  – relatively stable; associated with host name in DNS
• When it discovers it is in a foreign subnet (i.e., not its home subnet), it acquires a foreign address
  – uses auto-configuration to get the address
  – registers the foreign address with a home agent, i.e, a router on its home subnet
• Packets sent to the mobile’s home address(es) are intercepted by home agent and forwarded to the foreign address, using encapsulation
Mobile IPv4

correspondent host

foreign agent

home agent

home location of mobile host
Mobile IPv6

correspondent host

mobile host

home agent

home location of mobile host
IPv6 Tutorial

IPv4-IPv6 Coexistence & Transition
Transition / Co-Existence Techniques

A wide range of techniques have been identified and implemented, basically falling into three categories:

(1) dual-stack techniques, to allow IPv4 and IPv6 to co-exist in the same devices and networks

(2) tunneling techniques, to avoid order dependencies when upgrading hosts, routers, or regions

(3) translation techniques, to allow IPv6-only devices to communicate with IPv4-only devices

Expect all of these to be used, in combination
Dual-Stack Approach

• When adding IPv6 to a system, do not delete IPv4
  – this multi-protocol approach is familiar and well-understood (e.g., for AppleTalk, IPX, etc.)
  – note: in most cases, IPv6 will be bundled with new OS releases, not an extra-cost add-on

• Applications (or libraries) choose IP version to use
  – when initiating, based on DNS response:
    • if (destination has AAAA record) use IPv6, else use IPv4
  – when responding, based on version of initiating packet

• This allows indefinite co-existence of IPv4 and IPv6, and gradual app-by-app upgrades to IPv6 usage
• A6 record as experimental
Tunnels to Get Through IPv6-Ignorant Routers

- Encapsulate IPv6 packets inside IPv4 packets (or MPLS frames)
- Many methods exist for establishing tunnels:
  - manual configuration
  - “tunnel brokers” (using web-based service to create a tunnel)
  - “6-over-4” (intra-domain, using IPv4 multicast as virtual LAN)
  - “6-to-4” (inter-domain, using IPv4 addr as IPv6 site prefix)
- Can view this as:
  - IPv6 using IPv4 as a virtual link-layer, or
  - an IPv6 VPN (virtual public network), over the IPv4 Internet (becoming “less virtual” over time, we hope)
Translation

• May prefer to use IPv6-IPv4 protocol translation for:
  – new kinds of Internet devices (e.g., cell phones, cars, appliances)
  – benefits of shedding IPv4 stack (e.g., serverless autoconfig)
• This is a simple extension to NAT techniques, to translate header format as well as addresses
  – IPv6 nodes behind a translator get full IPv6 functionality when talking to other IPv6 nodes located anywhere
  – they get the normal (i.e., degraded) NAT functionality when talking to IPv4 devices
  – methods used to improve NAT functionality (e.g., RSIP) can be used equally to improve IPv6-IPv4 functionality
Thanks!

Contact:
– Jordi Palet Martínez (Consulintel): jordi.palet@consulintel.es

The IPv6 Portal:
• http://www.ipv6tf.org
IPv6 as Innovation Opportunity for Developing Countries

• Easy to deploy IPv6 in new infrastructures

• Lack of applications which take advantage of IPv6
  – Software development is not expensive, actually becomes cheaper with IPv6
  – Easy to be done in developing countries