Basic IPv6 Concepts

David Fernández (david@dit.upm.es)
Dpto. Ingeniería de Sistemas Telemáticos
Universidad Politécnica de Madrid
Contents

- New IPv6 Features
- Header Formats
- Autoconfiguration
- ICMPv6
- Neighbour Discovery
New IPv6 Features

- IPv6 is an evolution from IPv4 (not a revolution)
  - IPv6 = IPv4 redesigned incorporating more than 20 years of operational experience

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Addressing</td>
<td>128 bits addresses hierarchically assigned</td>
</tr>
<tr>
<td>Routing</td>
<td>Strongly Hierarchical (route aggregation)</td>
</tr>
<tr>
<td>Performance</td>
<td>Simple datagram header aligned to 64 bits</td>
</tr>
<tr>
<td>Extensibility</td>
<td>Improved support for extensions and options</td>
</tr>
<tr>
<td></td>
<td>New flexible option headers format</td>
</tr>
<tr>
<td>Multimedia</td>
<td>Better support for QoS (flow label)</td>
</tr>
<tr>
<td>Multicast</td>
<td>Compulsory. Better scope control</td>
</tr>
<tr>
<td>Security</td>
<td>Built-in security: authentication/encryption (IPSEC)</td>
</tr>
<tr>
<td>Autoconfiguration</td>
<td>Stateless and stateful address configuration</td>
</tr>
<tr>
<td>Mobility</td>
<td>Better support (efficiency and security)</td>
</tr>
</tbody>
</table>
Header Format: IPv4 vs. IPv6

IPv4

- Some fields maintained
  - with some names changed
- Some fields disappear
- Some fields moved to optional headers

Header Format: IPv4 vs. IPv6

<table>
<thead>
<tr>
<th>Field</th>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
<td>6</td>
</tr>
<tr>
<td>Internet Header Length</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Type of Service</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Protocol</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Source Address</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>Destination Address</td>
<td>16</td>
<td>128</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16</td>
<td>20</td>
</tr>
<tr>
<td>Options Padding</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Some fields maintained with some names changed. Some fields disappear. Some fields moved to optional headers.
## Header Format: IPv4 vs. IPv6

### IPv4

- Some fields maintained
  - with some names changed
- Some fields disappear
- Some fields moved to optional headers

### Header Format

<table>
<thead>
<tr>
<th>Field</th>
<th>IPv4</th>
<th>IPv6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version (Ver)</td>
<td>1 byte</td>
<td>1 byte</td>
</tr>
<tr>
<td>Total Length</td>
<td>1 byte</td>
<td>1 byte</td>
</tr>
<tr>
<td>Identification</td>
<td>2 bytes</td>
<td>16 bytes</td>
</tr>
<tr>
<td>Flags</td>
<td>1 byte</td>
<td>1 byte</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>1 byte</td>
<td>1 byte</td>
</tr>
<tr>
<td>Time to Live</td>
<td>1 byte</td>
<td>1 byte</td>
</tr>
<tr>
<td>Protocol</td>
<td>1 byte</td>
<td>2 bytes</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>2 bytes</td>
<td>2 bytes</td>
</tr>
<tr>
<td>Source Address</td>
<td>4 bytes</td>
<td>16 bytes</td>
</tr>
<tr>
<td>Destination Address</td>
<td>4 bytes</td>
<td>16 bytes</td>
</tr>
<tr>
<td>Options</td>
<td>4 bytes</td>
<td>4 bytes</td>
</tr>
<tr>
<td>Padding</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

IPv6 Tutorial – Basic IPv6 Concepts
Header Format: IPv4 vs. IPv6

IPv4
- Some fields maintained
  - with some names changed
- Some fields disappear
- Some fields moved to optional headers

IPv6
- 32 bits
- 40 bytes
- 20 bytes
- 20 bytes

Some fields maintained
- with some names changed
- Some fields disappear
- Some fields moved to optional headers

IPv6 Tutorial – Basic IPv6 Concepts
Basic IPv6 Header Format

- **Vers**: 64 bits
- **Traffic Class**: 40 bytes
- **Flow Label**: 47 bits
- **Payload Length**: 45 bits
- **Next Header**: 47 bits
- **Hop Limit**: 55 bits

**Source Address**: 64 bits

**Destination Address**: 64 bits
Summary of Header Changes

- Address length increased to 128 bits
- Fragmentation and options fields removed from base header
- Header checksum removed
- Header length field removed (fixed length header)
- New flow label field
- Fields renamed:
  - ToS -> Traffic Class
  - Protocol -> Next header
  - Time To Live -> Hop Limit
- Alignment changed to 64 bits
IPv6 Extension Headers

- Option headers format and treatment greatly improved:

| Base Header | Extension Header 1 | ... | Extension Header N | Data |

- Advantages:
  - Number and size of options not limited
  - Extension headers strictly ordered to simplify processing by routers
  - Definition of behavior for unknown options
Examples of IPv6 Packets

- IPv6 Header NH = TCP
- TCP Header + DATA

- IPv6 Header NH = Routing
- Routing Header NH = TCP
- TCP Header + DATA

- IPv6 Header NH = Routing
- Routing Header NH = Fragment
- Fragment Header NH = TCP
- TCP Header + DATA

- IPv6 Header NH = Fragment
- Fragment Header NH = Security
- Security Header NH = TCP
- TCP Header + DATA
Extension Headers Currently Defined

- **Hop-by-Hop Options:**
  - Information to be processed by every node
    - Exs: Jumbogram (RFC 2675): packets up to 4 GB!
    - Router Alert (RFC 2711)

- **Routing:**
  - similar to IPv4 Source Route option

- **Fragment:**
  - Fragmentation and reassembly

- **Authentication:**
  - Digital Signatures

- **Encapsulating Security Payload:**
  - Encryption Information

- **Destination Options:**
  - Information to be processed by destination
**Fragmentation in IPv6 (I)**

- **End-to-end fragmentation:**
  - Only the source can fragment
  - Routers DO NOT fragment

- **Path MTU Discovery (RFC 1981)** used to discover MTU for each destination
  - Used at present in most IPv4 implementations

- **Simple technique: trial and error**
  - Send packets using link MTU
  - If a router discards a packet due to MTU:
    - Sends an **ICMPv6 Packet Too Big** message to the source informing about the MTU causing the discarding
    - Source adapts PMTU for that destination (PMTU cache)
Fragmentation in IPv6 (II)

- Fragment header:
  - NH | Reserved | Offset | MF
  - Datagram Id.

- Link layer must support a 1280-byte MTU
  - Otherwise, link layer must use a transparent fragmentation and reassembly scheme

- For configurable MTU link layers, MTU size of at least 1500 bytes
  - Example: Maximum Receive Unit (MRU) of a Point-to-Point Protocol (PPP) link
QoS Support in IPv6 Header

- Two fields related with QoS:
  - Traffic Class:
    - Differentiated Services Field (RFC 2474)
    - Aligned with IPv4 TOS field
  - Flow label:
    - Enables efficient IPv6 flow classification based only on IPv6 main header fields in fixed positions
    - Work in progress: draft-ietf-ipv6-flow-label-07.txt
Autoconfiguration in IPv6

- Key feature of IPv6
- Main autoconfiguration goals:
  - Designed for hosts
  - Plug & play
  - Avoid manual configuration of addresses
  - Isolated networks
  - Graceful renumbering
- Three autoconfiguration types defined:
  - Stateless (Router Advertisement)
  - Stateful (DHCPv6)
  - Both
- Besides, link-local addresses for communication between nodes attached to the same link
**Link-Local Address**

- A node forms a link-local address whenever an interface becomes enabled
- Formed by:
  - Fixed 64-bit address prefix (FE80::/64)
  - Unique interface identifier (typically a EUI-64 identifier built from the MAC addr)
- Duplicate Address Detection (DAD) mechanism guarantees uniqueness
- Example: link-local addresses on 802 LANs

![Diagram illustrating link-local address formation](image-url)
**Stateless Autoconfiguration**

- Based on ICMPv6 Router Advertisements (RA) sent by routers
- Formed by:
  - Address prefix exported in RA
  - Unique interface identifier (same as in link-local)

```
A
fe80:: 250:4ff:fe5c:b3f4
3ffe:3328:1:: 250:4ff:fe5c:b3f4

Global Prefix

Token (Dir. MAC)

B
fe80:: 250:4ff:fe5e:aaff

C

S

ff02::1

250:4ff:fe5c:b3f4

3ffe:3328:1::3ffe:3328:1:: /64

Prefix:
3ffe:3328:1::/64

RA

ROUTER

B

C

S

fe80:: 250:4ff:fe5e:aaff

```

© DIT-UPM, May, 2003 IPv6 Tutorial – Basic IPv6 Concepts
## Router Advertisement

<table>
<thead>
<tr>
<th>Source</th>
<th>Router address</th>
<th>Destination</th>
<th>link-local multicast or node link-local</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Type = 13</strong></td>
<td>Code</td>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td>Cur Hop Limit</td>
<td>M</td>
<td>Reserved</td>
<td>Router Lifetime</td>
</tr>
<tr>
<td>Reachable Time</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Retrans Timer</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Options</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prefix Information options</td>
<td>- Subnet number and mask</td>
<td>- Node Subnets addresses</td>
<td></td>
</tr>
<tr>
<td>Link MTU definition</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Stateless Autoconfiguration Process

1. **Assign link-local address**
   - Duplicate detection

2. **IF fails THEN**
   - Autoconfiguration stops

3. **Find routers**
   - Wait periodic RA (from few seconds to 30 minutes)
   - Send RS

4. **IF no receive any RA THEN**
   - Isolated network

5. **Finish autoconfiguration**

---

© DIT-UPM, May, 2003
IPv6 Tutorial – Basic IPv6 Concepts
Privacy Issues in Autoconfiguration

- **RFC 3041:** Privacy Extensions for Stateless Address Autoconfiguration in IPv6
  - Proposes extensions to generate global-scope addresses from interface identifiers that change over time
  - **Objective:**
    - difficult eavesdroppers and other information collectors to identify when different addresses used in different transactions actually correspond to the same node.
Stateful Autoconfiguration

- Standard DHCP based autoconfiguration
  - Not integrated with Pv4 (independent autoconfiguration)
  - New packet design (no BOOTP legacy)
  - Clients may obtain many addresses
- Clients use link-local addresses to speak with DHCP server/relays
  - No more 255.255.255.255’s!!
Stateless or Stateful?

- Decision made by the network or site administrator:
  - **Managed Address Configuration** bit in RA
    - 0 = Do Stateless configuration.
    - 1 = Do not do Stateless configuration

- Stateless and Stateful can Coexist. For example:
  - **Other stateful configuration** bit in RA
    - Addresses could come from Stateless
    - Additional Configuration Information could be provided by DHCPv6
More Autoconfiguration in IPv6

- Proposals to automate the delegation of address prefixes to customer sites:
- Requirements defined:
  - “Requirements for IPv6 prefix delegation”. draft-ietf-ipv6-prefix-delegation-requirement-01.txt
- Several proposals:
  - Extensions to DHCP
  - New ICMP Prefix Delegation messages
  - Router renumbering (RFC2894)
**ICMP in IPv6**

- A new version of Internet Control Message Protocol (ICMPv6) has been defined.
- Apart from ICMP original functionalities (error messages and diagnostic) it provides the framework for:
  - Autoconfiguration
  - Address resolution (Neighbor Discovery)
  - Path MTU Discovery
  - Multicast Group Management (Multicast Listener Discovery - MLD)
  - Duplicate Address Detection (DAD)
  - IPv6 mobility
Neighbor Discovery Overview
Neighbor Discovery Overview

- Defined in RFC2461
Neighbor Discovery Overview

- Defined in RFC2461
- **Used by IPv6 nodes on the same link:**
  - to discover each other's presence,
  - to determine each other's link-layer addresses,
  - to find routers, and
  - to maintain reachability information about the paths to active neighbors.
Neighbor Discovery Overview

- Defined in RFC2461
- Used by IPv6 nodes on the same link:
  - to discover each other's presence,
  - to determine each other's link-layer addresses,
  - to find routers, and
  - to maintain reachability information about the paths to active neighbors.
- Replaces ARP, ICMPv4 Router Discovery, and ICMPv4 Redirect
Neighbor Discovery Overview

- Defined in RFC2461
- Used by IPv6 nodes on the same link:
  - to discover each other's presence,
  - to determine each other's link-layer addresses,
  - to find routers, and
  - to maintain reachability information about the paths to active neighbors.
- Replaces ARP, ICMPv4 Router Discovery, and ICMPv4 Redirect
- Eliminates the use of broadcast for address resolution (intelligent use of multicast)
**Address Resolution**

- Requests sent to “Solicited-node multicast address”
  - Most probably only the node queried will receive the Neighbor Solicitation message

```
3ffe:3328:1::250:4ff:fe5c:b3f4
3ffe:3328:1::250:4ff:fe5e:aaff
```

```
ff02::1:ff5e:aaff
```

```
(1)
```

```
Neighbour Solicitation
3ffe:3328:1::250:4ff:fe5e:aaff
```

```
(2)
```

```
Neighbour Advertisement
3ffe:3328:1::250:4ff:fe5e:aaff
00-50-04-5c-b3-f4
```

```
S
```

```
3ffe:3328:1::250:4ff:fe5e:aaff
```

```
Solicited-node multicast address
```

```
ROUTER
```

```
dit© DIT© DIT-UPM, May, 2003
```

```
IPv6 Tutorial – Basic IPv6 Concepts
```

```
© DIT-UPM, May, 2003
```

```
IP6 FORUM
```

```
24
```
Neighbor Discovery Overview (II)

- **ND is used by nodes:**
  - For address resolution
  - To determine link-layer address changes
  - To determine neighbor reachability

- **ND is used by hosts:**
  - To discover neighboring routers
  - Autoconfigure addresses, address prefixes, and other configuration parameters

- **ND is used by routers:**
  - To advertise their presence, host configuration parameters, and on-link prefixes
  - To inform hosts of a better next-hop address to forward packets for a specific destination
Neighbor Discovery Processes & Messages

**ND Processes**
- Router discovery
- Prefix discovery
- Parameter discovery
- Address autoconfiguration
- Address resolution
- Next-hop determination
- Neighbor unreachability detection
- Duplicate address detection
- Redirect function

**Messages**
- Router Solicitation
- Router Advertisement
- Neighbor Solicitation
- Neighbor Advertisement
- Redirect
References

- **Multicast Listener Discovery (MLD) for IPv6.** S. Deering, W. Fenner, B. Haberman. October 1999. RFC 2710