IPv6

The New Internet Protocol
Outline

- The Protocol (new ICMP)
- Addressing and Routing (provider addressing)
- Autoconfiguration
- Security
- Support of Real-time Communication
- Deployment Strategy
The Design of IPv6

- IPv4 design was very good IPv6 should keep most of it
- It could only increase the size of addresses and keep everything the same
- Experience brought lessons for improvement
# IPv6 Header (40 bytes)

<table>
<thead>
<tr>
<th></th>
<th>Version</th>
<th>Prio</th>
<th>Flow Label</th>
<th>Payload Length</th>
<th>Next Header</th>
<th>Hop Limit</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>0</td>
<td>16</td>
<td>0</td>
<td>128 bits</td>
<td>128 bits</td>
<td>128 bits</td>
</tr>
<tr>
<td>0</td>
<td>31</td>
<td>16</td>
<td>16</td>
<td>128 bits</td>
<td>128 bits</td>
<td>128 bits</td>
</tr>
</tbody>
</table>

- **Source Address**: 128 bits
- **Destination Address**: 128 bits
# IPv4 Header (20 bytes)

<table>
<thead>
<tr>
<th>Bitwise Location</th>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-3</td>
<td>Version</td>
<td>4-bit Version</td>
</tr>
<tr>
<td>4-7</td>
<td>HD len</td>
<td>Header Length</td>
</tr>
<tr>
<td>8-15</td>
<td>TOS</td>
<td>Type of Service</td>
</tr>
<tr>
<td>16-19</td>
<td>Identification</td>
<td>Identification</td>
</tr>
<tr>
<td>20-27</td>
<td>Tot len</td>
<td>Total Length</td>
</tr>
<tr>
<td>28-31</td>
<td>HD chksum</td>
<td>Header Checksum</td>
</tr>
<tr>
<td>0-23</td>
<td>Protocol</td>
<td>Protocol Type</td>
</tr>
<tr>
<td>24-31</td>
<td>TTL</td>
<td>Time-to-Live</td>
</tr>
<tr>
<td>Source Address</td>
<td>32 bits</td>
<td>Source Address</td>
</tr>
<tr>
<td>Destination Address</td>
<td>32 bits</td>
<td>Destination Address</td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
<td>Options (if any)</td>
</tr>
</tbody>
</table>
Note that while the IPV6 address are four times as large as the IPV4 address, the header length is only twice as big.
Notations of IPv6 Addresses

- 128 bit is represented as:
  - 8 integers (16-bit) separated by colons
    - each integer is represented by 4 hex digits

Example:

Simplifications

- Skip leading zeros
  - Example: 1080:0000:0000:0000:0008:0800:200C:417A
  - is reduced to: 1080:0:0:0:8:800:200C:417A

- A set of consecutive nulls is replaced by ::
  (at most one :: inside an address)
  - the above address is reduced to:
    - 1080::8:800:200C:417A
Comparison of Headers

- V6: 6 fields + 2 addr
- V4: 10 fields + 2 addr + options
- Deleted:
  - Header length
  - type of service
  - identification, flags, fragment offset
  - Header Checksum
- Added:
  - Priority
  - Flow label
- Renamed:
  - length -> Payload length
  - Protocol -> Next header
  - time to live -> Hop Limit
- Redefined: Option mechanism
Simplifications

- **Fixed format headers**
  - no options -> no need for header length
  - options expressed as Extension headers

- **No header checksum**
  - reduce cost of header processing, no checksum updates at each router
  - minimal risk as encapsulation of media access protocols (e.g., Ethernet, PPP) have checksum

- **No segmentation**
  - hosts should use path MTU discovery
  - otherwise use the minimum MTU (536 bytes)
Renaming

- Total Length ➔ Payload Length
  - not include header length
  - max length 64Kbytes with provision for larger packets using “jumbo gram” option

- Protocol Type ➔ Next header, can be set to:
  - Protocol type (UDP, TCP, etc.)
  - Type of first extension header

- TTL ➔ Hop limit
  - “Truth in advertising!
  - number of hops NOT number of seconds
New Fields

- Flow label & Priority
  - to facilitate the handling of real time traffic
Options  ➔  Extension Headers

Routers treats packets with options as “second class citizens” because it is slow to process, thus programmers tend not use them and options almost became obsolete.
Daisy Chain of Headers

IPv6 Header
Next Header= TCP

TCP Header + Data

IPv6 Header
Next Header= Routing

Routing Header
Next Header= TCP

TCP Header + Data
IPv6 extension headers

- Hop-by-hop options
- Routing
- Fragment
- Destination options
- Authentication
- Encryption Security Payload
## Protocol & Header Types

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Keyword</th>
<th>Header Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>HBH</td>
<td>hop-by-hop</td>
</tr>
<tr>
<td>3</td>
<td>ICMP</td>
<td>Inet Control</td>
</tr>
<tr>
<td>4</td>
<td>IP</td>
<td>v4 encapsul.</td>
</tr>
<tr>
<td>6</td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>UDP</td>
<td></td>
</tr>
<tr>
<td>43</td>
<td>RH</td>
<td>Routing hdr</td>
</tr>
</tbody>
</table>
Routers will only look at the routing header if they recognize one of their addresses in the destination field of the main header.

<table>
<thead>
<tr>
<th>Next Header</th>
<th>0</th>
<th>Num addr</th>
<th>Next Addr</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reserved</td>
<td></td>
<td>strict/loose bit mask (24)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Address[0]</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Address[Num Addr - 1]</td>
<td></td>
</tr>
</tbody>
</table>
Fragment Header

- Routers do not fragment oversized packets
- Sender is to fragment & Receiver is to reassemble

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Reserved</th>
<th>Fragment offset</th>
<th>Res</th>
<th>M</th>
</tr>
</thead>
<tbody>
<tr>
<td>Identification</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
**Destination Option Header**

Will only be examined by the station specified in the destination address.

<table>
<thead>
<tr>
<th>Next Header</th>
<th>Hdr Ext Len</th>
<th>Options</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Option Type</th>
<th>Opt Data Len</th>
<th>Option Data</th>
</tr>
</thead>
</table>
Hop-by-Hop Option Header

Will be examined by each router.
Has same form as destination options hdr.

To satisfy networking requirement of supercomputers, the Jumbo payload option is used to send very large packets (the IPv6 length field is set to zero):

<table>
<thead>
<tr>
<th>194</th>
<th>Opt Len = 4</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Jumbo Payload Length</td>
</tr>
</tbody>
</table>
ICMP.... Streamlined

- Removed unused functions in ICMP of v4
- Incorporate IGMP of v4

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Checksum</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Message Body
ICMP Error Messages

1  Destination Unreachable. Codes:
   0  No route to destination
   1  destination prohibited
   3  Address unreachable
   4  Port unreachable

2  Packet Too Big
   contain next hop MTU.
   used for path MTU discovery

3  Time Exceeded. Codes:
   0  Hop limit exceeded
   1  Fragment reassembly time exceed

4  Parameter Problem
   No error message in response to multicast or ICMP packets
Other ICMP messages

128  Echo Request
129  Echo Reply

130  Group Membership Query
131  Group Membership Report
132  Group Membership Termination

133  Router Solicitation
134  Router Advertisement
135  Neighbor Solicitation
136  Neighbor Advertisement
137  Redirect
Impact on Upper Layers

Upper-layer Checksums : Mandatory (even UDP)

| pseudoheader | transport header | transport data |

| Source Address |
| Destination Address |
| 0 | Next header | Payload Length |
Domain Name Service

32-bit address to 128-bit address

Programming interface

Address data structures

AF_INET6, PF_INET6, in_addr6, sockaddr_in6

Name-to-address translation functions

Address conversion functions
Points of Controversy

- Do we need more than 255 Hops?
  - allowing hop count to be very large, looping packets will be relayed many times before being discarded

- Should packets be larger than 64K?
  - allowing very large packets increase the size of queues and the variability of queuing delays

- Can we live without checksum?
  - Some IPv4 routers started to cut corners by not verifying checksums to gain advantage over competition. By removing checksum altogether offers all routers the same advantage.
Real-time Support & Flows

- A proper handling of *flows* is required for high-quality *multimedia communications* in the new Internet.
- A *flow* is a sequence of packets sent from a particular *source* to a particular (unicast or multicast) *destination* for which the source desires *special handling* by the intervening routers.
The flow label & source address are used to assert which packets belong to what flows.

In IPv6 port numbers deep inside due to daisy chaining.
Even may not be visible due to encryption.
Security

- If security is provided *at the IP level* it becomes standard service that all applications can use.

- It is absolutely necessary to implement if we want to develop *of commercial use* the Internet, e.g. to deter sniffing attacks on passwords and credit card numbers.
Headers

 Authentication header
 Guarantee that the *source address is authentic* & the packet has *not been altered* during transmission.

 Encryption header
 Guarantee that only *legitimate receivers* will be able to *read the content* of the packet
Transitioning the Internet

At the beginning, all IPv6-capable hosts will also be IPv4-capable so as to retain connectivity with the existing Internet.

To transform IPv4 into a dual-stack IPv6-capable host, it should include:
- The IPv6 basic code
- Handling IPv6 within TCP & UDP
- Modify socket interface to support new addresses
- Handling the interface with the name service
The 6-Bone

- The Similar to the M-Bone, initially the connectivity is achieved by *tunneling*.
- IPv6 packet will be *encapsulated* within IPv4 packets.