Network Layer (II)

IP Design Goals
- Should have fields for source and destination IP addresses
- Needs means for error control:
  - detection of packet header corruption (L2 does the heavy lifting)
  - limiting the lifespan of a packet (TTL not in TCP)
- Fragmentation
  - carrying transport layer messages that are longer than what L2 can support

IPv4 Header

- version: v4 or v6; v6 has a different format, so decide how to parse the binary sequence
- TOS: distinguish different types of IP datagrams.
  - e.g. it might be useful to distinguish real-time datagrams (such as those used by an IP telephony application) from non-real-time traffic (e.g. FTP). The specific level of service to be provided is a policy issue determined and configured by the network administrator for that router.
- upper layer: used only when an IP datagram reaches its final destination, indicating the transport layer protocol, 6 - TCP, 17 - UDP, 1 - ICMP, so that the destination will know how to parse the data.
Fragmentation
- Problem:
  • network layer needs to deliver a PDU (protocol data unit) that is longer than what the link layer permits
- Solutions:
  • drop and inform (default behavior of IPv6)
  • break the PDU to smaller units (fragments) and reassemble them at the destination (default behavior of IPv4)

IPv4 Fragmentation
- Identification (16 bits)
  • identifies the original fragmented packet
  • sender generates it and increments the number for each packet it sends
- Fragment Offset (13 bits)
  • specifies the location (in 8-byte blocks) of the fragment in the packet
- M (1 bit) - More Fragments
- D (1 bit) - Do not Fragments

Fragmentation Example
- Transport layer Packet length 4,500 bytes, MTU 2,500 bytes:

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Total bytes</th>
<th>Header bytes</th>
<th>Data bytes</th>
<th>&quot;More fragments&quot; flag</th>
<th>Fragment offset (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2500</td>
<td>20</td>
<td>2480</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>2040</td>
<td>20</td>
<td>2020</td>
<td>0</td>
<td>310</td>
</tr>
</tbody>
</table>

- 0.
- $0 + 2480/8 = 310$

- Please note that the MUT is not static across the entire network.
- It is possible that the above two fragments reach another router connecting a link with smaller MTU. Then, the above two fragments will be further fragmented at that router.

- Suppose the above two fragments reach a router connecting a link with MTU 1,500 bytes. These two fragments will be further fragmented:
<table>
<thead>
<tr>
<th>Fragment</th>
<th>Total bytes</th>
<th>Header bytes</th>
<th>Data bytes</th>
<th>&quot;More fragments&quot; flag</th>
<th>Fragment offset (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1500</td>
<td>20</td>
<td>1480</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1020</td>
<td>20</td>
<td>1000</td>
<td>1</td>
<td>185</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>20</td>
<td>1480</td>
<td>1</td>
<td>310</td>
</tr>
<tr>
<td>4</td>
<td>560</td>
<td>20</td>
<td>540</td>
<td>0</td>
<td>495</td>
</tr>
</tbody>
</table>

- 0.
- 0 + 1480/8 = 185
- 185 + 1000/8 = 310
- 310 + 1480/8 = 495

### IPv6 - Protocol Design

- Keep the good stuff …
  - unreliable datagram service
  - TTL, TOS (for compatibility)
- Eliminate the unnecessary …
  - no fragmentation (only as an option)
  - no header checksums
- Address the issues …
  - longer addresses and more

### IPv6 Header

- **version**: 4-bit, 6 indicate v6
- **pri**: 8-bit traffic class field, like the TOS field in v4, used to give priority to certain datagram
- **flow label**: 20-bit, identify datagrams in same “flow”
  - concept of “flow” not well defined: audio and video are flow, or file transfer and email are not; it is possible that traffic carried by a high-priority user (someone paying for better service for their traffic) might be also treated as flow.
- **payload len**: 16-bit integer, number of bytes in the data part (not including the 40-byte header)
- next header: identify upper layer protocol for data (TCP, UDP)

- Discussions
  - **Fragmentation**: handle by the source and destination, if the router receives a too large packet to be forwarded, it drops the packet, and send a “Packet Too Big” ICMPv6 message.
  - Header checksum: TCP/UDP and link layer protocols have checksum

---

**IPv6 Address Representation**

- An IPv6 address is represented by 8 groups of 16-bit hexadecimal values separated by colons (:)
- Can be abbreviated:
  - **omit leading zeroes** in a 16-bit value
  - **replace one group of consecutive zeros** by a double colon
- Example:
  - `2606:4100:38c0:9::5` vs `2606:4100:38c0:0009:0000:0000:0000:0005`

- ping IPv6: `gloin:~> ping6 fe80::da9e:f3ff:fe41:f985%eno1`

---

**Neighbor Discovery Protocol**

- **Replacing ARP in IPv6** (multicast vs broadcast)
- **Built into ICMPv6**, defines ICMPv6 message
  - **Router Solicitation** (Type 133)
    - Hosts inquire with Router Solicitation messages to locate routers on an attached link (all routers multicast ff02::2). Routers which forward packets not addressed to them generate Router Advertisements immediately upon receipt of this message rather than at their next scheduled time.
  - **Router Advertisement** (Type 134)
    - Routers advertise their presence together with various link an Internet parameters either periodically (all nodes multicast ff02::1), or in response to a Router Solicitation message sender. It includes information like available link prefixes, MTU, and etc.
  - **Neighbor Solicitation** (Type 135)
    - Neighbor solicitations are used by nodes to determine the link layer address of a neighbor, or to verify that a neighbor is still reachable via a cached link layer address.
  - **Neighbor Advertisement** (Type 136)
    - Neighbor advertisements are used by nodes to respond to a Neighbor Solicitation message.
  - **Redirect** (Type 137)
    - Router may inform hosts of a better first hop router for a destination.
- **Router Discovery**
  - When first joining a link, an IPv6 host multicasts a router solicitation to the *all routers* multicast group, and each router active on the link responds by sending a router advertisement with its address to the *all nodes* group.

- **Address Resolution**
  - A host seeking the MAC address of a neighbor multicasts a neighbor solicitation and the neighbor (if online) responds with its link layer address in a neighbor advertisement.

- **Redirection**
  - Routers use redirect ICMPv6 message to either point hosts toward a more preferable router, or to indicate that the destination actually resides on link.
• Diagram REF: https://packetlife.net/blog/2008/aug/28/ipv6-neighbor-discovery/
  - RFC 4861

Disclaimer: Notes adapted from the textbook and online resources.