Internet Protocol (IP): Packet Format, Fragmentation, Options

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

Overview

- IP features
- IP datagram format
- Fragmentation
- IP options
Ref: RFC 791, Chap 3, 11.5-11.8, Lab 1

IP Features

- Connectionless service
- Addressing
- Data forwarding
- Fragmentation and reassembly
- Supports variable size datagrams
- Best-effort delivery: Delay, out-of-order, corruption, and loss possible. Higher layers should handle these.
- Provides only “Send” and “Delivery” services
  Error and control messages generated by Internet Control Message Protocol (ICMP)
What IP does NOT provide

- End-to-end data reliability & flow control (done by TCP or application layer protocols)
- Sequencing of packets (like TCP)
- Error detection in payload (TCP, UDP or other transport layers)
- Error reporting (ICMP)
- Setting up route tables (RIP, OSPF, BGP etc)
- Connection setup (it is connectionless)
- Address/Name resolution (ARP, RARP, DNS)
- Configuration (BOOTP, DHCP)
- Multicast (IGMP, MBONE)

IP Datagram Format

- First Word purpose: info, variable size header & packet.
- Version (4 bits)
- Internet header length (4 bits): units of 32-bit words. Min header is 5 words or 20 bytes.
- Type of service (TOS: 8 bits): Reliability, precedence, delay, and throughput. Not widely supported
- Total length (16 bits): header + data. Units of bytes. Total must be less than 64 kB.
IP Header (Cont)
- 2nd Word Purpose: fragmentation
- Identifier (16 bits): Helps uniquely identify the datagram between any source, destination address
- Flags (3 bits): More Flag (MF):more fragments
  Don't Fragment (DF)
  Reserved
- Fragment offset (13 bits): In units of 8 bytes

IP Header (Cont)
- Third word purpose: demuxing, error/looping control, timeout.
- Time to live (8 bits): Specified in router hops
- Protocol (8 bits): Next level protocol to receive the data: for de-multiplexing.
- Header checksum (16 bits): 1's complement sum of all 16-bit words in the header.
- Change header => modify checksum using 1's complement arithmetic.
- Source Address (32 bits): Original source. Does not change along the path.

Header Format (contd)
- Destination Address (32 bits): Final destination. Does not change along the path.
- Options (variable length): Security, source route, record route, stream id (used for voice) for reserved resources, timestamp recording
- Padding (variable length):
  Makes header length a multiple of 4
- Payload Data (variable length): Data + header \leq 65,535 bytes
Maximum Transmission Unit
- Each subnet has a maximum frame size:
  - Ethernet: 1518 bytes
  - FDDI: 4500 bytes
  - Token Ring: 2 to 4 kB
- Transmission Unit = IP datagram (data + header)
- Each subnet has a maximum IP datagram length (header + payload) = MTU

Fragmentation
- Datagrams larger than MTU are fragmented
- Original header is copied to each fragment and then modified (fragment flag, fragment offset, length,...)
- Some option fields are copied (see RFC 791)

Fragmentation Example
- Payload size 452 bytes needs to be transmitted across a Ethernet (MTU=1500B) and a SLIP line (MTU=280B)
- Length = 472B, Header = 20B => Payload = 452B
- Fragments need to be multiple of 8-bytes.
  - Nearest multiple to 260 (280 - 20B) is 256B
  - First fragment length = 256B + 20B = 276B
  - Second fragment length = (452B - 256B) + 20B = 216B
Reassembly
- Reassembly only at the final destination
- Partial datagrams are discarded after a timeout
- Fragments can be further fragmented along the path. Subfragments have a format similar to fragments.
- Minimum MTU along a path ⇒ Path MTU

Further notes on Fragmentation
- Performance: single fragment lost ⇒ entire packet useless. Waste of resources all along the way. Ref: Kent & Mogul, 1987
- Don’t Fragment (DF) bit set ⇒ datagram discarded if need to fragment. ICMP message generated: may specify MTU (default = 0)
- Used to determine Path MTU (in TCP & UDP)
- The transport and application layer headers do not appear in all fragments. Problem if you need to peep into those headers.

IP Protocol Numbers

<table>
<thead>
<tr>
<th>Decimal</th>
<th>Key word</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>ICMP</td>
<td>Internet Control Message Protocol</td>
</tr>
<tr>
<td>2</td>
<td>IGMP</td>
<td>Internet Group Management Protocol</td>
</tr>
<tr>
<td>4</td>
<td>ST</td>
<td>Stream Protocol</td>
</tr>
<tr>
<td>5</td>
<td>TCP</td>
<td>Transmission Control Protocol</td>
</tr>
<tr>
<td>8</td>
<td>EGP</td>
<td>Exterior Gateway Protocol</td>
</tr>
<tr>
<td>9</td>
<td>IGP</td>
<td>Interior Gateway Protocol</td>
</tr>
<tr>
<td>17</td>
<td>UDP</td>
<td>User Datagram Protocol</td>
</tr>
</tbody>
</table>
IP Options Coding

<table>
<thead>
<tr>
<th>Type</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1B</td>
<td>1B</td>
<td>nB</td>
</tr>
</tbody>
</table>

Flag Copy  Class  Number
1b  2b  5b

- Flag Copy: 0 = Copy the option only into the first fragment of a fragmented datagram
- 1 = Copy into all fragments
- Class: 0 = User or control, 1 = Reserved, 2 = Diagnostics, 3 = reserved

IP Options

<table>
<thead>
<tr>
<th>Class</th>
<th>Number</th>
<th>Length</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>End of Options</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>No Op</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>11</td>
<td>Security</td>
</tr>
<tr>
<td>0</td>
<td>3</td>
<td>Var</td>
<td>Loose Source Routing</td>
</tr>
<tr>
<td>0</td>
<td>7</td>
<td>Var</td>
<td>Record Route</td>
</tr>
<tr>
<td>0</td>
<td>8</td>
<td>4</td>
<td>Stream ID (obsolete)</td>
</tr>
<tr>
<td>0</td>
<td>9</td>
<td>Var</td>
<td>Strict Source Routing</td>
</tr>
<tr>
<td>2</td>
<td>4</td>
<td>Var</td>
<td>Internet Time-Stamp</td>
</tr>
</tbody>
</table>

IP Source Routing

<table>
<thead>
<tr>
<th>Code</th>
<th>Length</th>
<th>Pointer</th>
<th>Router Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>128.2.3.4</td>
<td>128.7.8.9</td>
<td>128.10.4.12</td>
</tr>
<tr>
<td>P</td>
<td>128.2.3.4</td>
<td>128.7.8.9</td>
<td>128.10.4.12</td>
</tr>
</tbody>
</table>

- Loose Source Routing (LSSR): Specify partial route list
- Strict Source Routing: Specify full route.
Route Recording

- Need to allow enough space to record IP addresses on route. Datagram size does not change as it goes through internet.

```
<table>
<thead>
<tr>
<th>Code</th>
<th>Length</th>
<th>Pointer</th>
<th>Route Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>P</td>
<td>128.2.3.4</td>
<td>Empty</td>
<td>Empty</td>
</tr>
<tr>
<td>P</td>
<td>128.2.3.4</td>
<td>128.7.8.9</td>
<td>Empty</td>
</tr>
</tbody>
</table>
```

Timestamp Option

- Record timestamps along route
- Overflow (Oflw) counter incremented if out of space
- Flags: allows some further options for flexibility

```
<table>
<thead>
<tr>
<th>Code</th>
<th>Length</th>
<th>Pointer</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oflw</td>
<td>Flags</td>
<td>IP Address 1</td>
<td>Timestamp 1</td>
</tr>
<tr>
<td>IP Address n</td>
<td>Timestamp n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

Discussion on IP Header Design

- If fragmentation is going to be avoided all the time, why not have the 4-bytes of fragmentation info as an IP option?
- Is 32-bit addresses going to be enough?
- Why mess with variable length headers? Can the variability in header length be controlled to allow better encoding?
- Are the IP options really that useful? Why variable length option headers?
- Many of these issues addressed in IPv6.
Summary

- IP header: supports connectionless delivery, variable length pkts/headers/options, fragmentation/reassembly,
- Fragmentation/Reassembly, Path MTU discovery.
- Options: Source routing, Record route, Timestamp