Today's Lecture

I. IPv4 Overview
II. IP Fragmentation and Reassembly
III. IP and Routing
IV. IPv4 Options

IPv4 Overview

Internet Protocol v4 (RFC791) Functions

- A universal intermediate layer
- Routing
- Fragmentation and reassembly

“IP over Everything, Everything Over IP”

- Everything over IP
  - TCP, UDP
  - Appletalk
  - Netbios
  - SCSI
  - ATM
  - X.25
  - SNA
  - Sonet
  - Fibre Channel
  - Frame Relay...
  - Remote Direct Memory Access
  - Ethernet

- IP over everything
  - Dialup
  - ISDN
  - X.25
  - Ethernet
  - Wi-Fi
  - FDDI
  - ATM
  - Sonet
  - ...

- Even IP over IP!

IP = Basic Delivery Service

- Connectionless delivery simplifies router design and operation
  - Unreliable, best-effort delivery. Packets may be...
    - lost (discarded)
    - duplicated
    - reordered
    - and/or corrupted
IPv4 Datagram Format

<table>
<thead>
<tr>
<th>Field</th>
<th>Offset</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Header Length x4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Type of Service</td>
<td>8</td>
<td>8</td>
</tr>
<tr>
<td>Identification</td>
<td>16</td>
<td>16</td>
</tr>
<tr>
<td>Time-to-Live (hop count)</td>
<td>32</td>
<td>8</td>
</tr>
<tr>
<td>Protocol Identifier</td>
<td>36</td>
<td>8</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>44</td>
<td>8</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>52</td>
<td>32</td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>84</td>
<td>32</td>
</tr>
<tr>
<td>Options (if any)</td>
<td>116</td>
<td>≤320</td>
</tr>
<tr>
<td>Payload</td>
<td>156</td>
<td>≤65515</td>
</tr>
</tbody>
</table>

IPv4 "Universal Common Layer" Fields

- Version: 4 (i.e., IPv4)
- Header Length x4 (i.e., header length is always a multiple of 4 bytes)
  - normally = 5 (x4 = 20)
  - at most = 15 (x4 = 60)
- Total Length (incl. IP header) < 2^{16} - 1 (65535)
- Protocol Identifier: how to interpret the payload
  - e.g., TCP = 6, UDP = 17, ...
- Flags (3)
  - DF: Don't fragment
  - MF: More fragments

Header Checksum

- Only for detecting errors in the IP header
  - needed?
- Algorithm
  - add (ones-complement addition) consecutive 16-bit words to generate a 16 bit sum
  - then one's-complement this sum
  - (for purposes of computation, assume an "initial" checksum value of all zeros)

Checksum Code

```c
#include <stdio.h>
#include <stdint.h>

volatile u_short checksum(u_short *header, int length) {
    volatile register u_long sum = 0;
    while (length--) {
        sum += *header++;
        /* This is two's complement addition */
        if (sum & 0xFFFF0000) { /* carry occurred, wrap around */
            sum &= 0x0000FFFF;
            sum++;
        }
    }
    return ~(sum & 0x0000FFFF); /* 1's complement the sum */
}
```

Checksum Code (cont’d)

- Given...
  a) IP header
  b) length of the header (in units of 16 bit words)

- What type of errors is this guaranteed to detect?
Examples

<table>
<thead>
<tr>
<th>Original</th>
<th>Two bits corrupted</th>
<th>One bit corrupted</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 1 0 1</td>
<td>1 1 0 1</td>
<td></td>
</tr>
<tr>
<td>1 1 0 1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Type of Service (TOS) Field

- Purpose: tells routers about special service / handling needed by the application traffic
- Precedence (3 bits): affects queue service order
- TOS bits
  - D = “low delay” desired
  - T = “high throughput” desired
  - R = “high reliability” desired
- Good idea, but not widely used, obsoleted by RFC 2474

Example Values for TOS Field (RFC 1349)

<table>
<thead>
<tr>
<th>Application</th>
<th>Minimize delay</th>
<th>Maximize throughput</th>
<th>Maximize reliability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Telnet / Rlogin</td>
<td>1 0 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>FTP Data</td>
<td>0 1 0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SNMP</td>
<td>0 0 1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ICMP</td>
<td>0 0 0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fragmentation And Reassembly

Fragmentation (cont’d)

- With IPv4, the network splits large packets into fragments
  - each fragment is itself a properly-formed IP datagram
  - equal-sized? MTU-sized except last fragment?
- Fragments may themselves be fragmented at intermediate hops
- Routers must be able to handle fragments at least 576 bytes long

Fragmentation

- Each link layer technology has a maximum payload size
- Endpoints have no idea what link layers their traffic will encounter. Possible solutions?
Fragmentation Example

- Example below: path MTU = 620

Fragment payload size must be a multiple of 8 bytes, except for the last one

Fragmentation Fields

- Identification field uniquely identifies each datagram
  - allows fragments of a datagram to be matched together
- Each fragment has the same IP header as the original IP datagram, except for the following:
  - Fragment Offset
  - More Fragments flag
  - Options
  - IP Header Length
  - Total Length
  - Header Checksum

Fragmentation Fields (cont’d)

- IP Checksum will of course be different in fragment than for original datagram
- More Fragments flag = 0 if this is the last (or only) fragment of the datagram, 1 otherwise
- The Fragment Offset field gives offset of the data (payload) portion of the fragment relative to the start of data in the original IP datagram
  - in units of 8 bytes
  - 13 bits are enough to represent a maximum datagram length of \(2^{13} \times 8 = 2^{16}\)

Fragmentation Fields (cont’d)

- IP Options may or may not be included in fragment IP headers (option-dependent)
  - IP Header Length may therefore be different than in original datagram
- Total Length is length of the fragment, not length of the original datagram

Reassembly

- Fragments reassembled at final destination in a reassembly buffer
  - good? bad?
- What if some fragments never arrive? Problems?
  - ???
- What if two fragments overlap?
  - ???
Avoiding Fragmentation

• Is fragmentation even a good idea?

• Do Not Fragment flag “forbids” fragmentation by the network. If datagram exceeds MTU of the outgoing router interface, the router…
  – discards the datagram, and
  – sends ICMP error message back to the source

• Better approach: Path MTU Discovery (we’ll discuss later)

IP Routing Fields

Basic IP Routing Fields

• Source IP Address, Destination IP Address

• Time-to-Live (TTL) (max allowable “hop” count)
  – max of 255, usually initialized to 128 or greater
  – decremented by each router the datagram passes through

• When TTL=0…
  – datagram will be discarded
  – error message sent back to source by ICMP
  – purpose?

• What’s the longest valid IP path length???

IP Options

• Basic protocol property: extensibility

• IP options mainly used for testing / debugging
  – infrequently used; 40 bytes doesn’t give you much to work with

• Every IP option must start with:
  – Code (i.e., option type)
  – Option Length (maximum of 40 bytes)

What Options Are Used?

• Record Route [RFC791]

• Loose Source Route [RFC791]

• Strict Source Route [RFC791]

• Time Stamp [RFC791]

• MTU Probe and Reply [RFC1191, we’ll discuss in ICMP lecture]

• Router Alert [RFC2113]
Option #1: Record Route

- Function: to inform the endpoints of the path a packet takes through the network
- Source creates empty list for recording up to 9 router addresses on the path to the destination
- Option contains...
  - pointer to next available "slot" to record an address, and
  - an empty list of IP addresses
- Routers add outgoing interface to list, and increment Pointer
  - If Pointer > Option Length, no further addresses will be inserted

IP Record Route Option Format

- 32 bits
  - Code=7
  - Length
  - Pointer
  - First IP address (empty when started)
  - Second IP address (empty when started)
  - ...
  - Last IP address (empty when started)

- Not copied on fragmentation, appears in first fragment only

IP Record Route Option Example

- Code length Next available byte
- A: 128.23.0.0
- B: 138.6.25.96
- C: 140.10.0.0
- D: 140.10.52.101
- E: 200.14.7.60
- F: 200.14.7.43

Option #2: Source Route

- Use: source specifies the path to be taken by the packet
  - list specifies router incoming interfaces
- Two types: strict and loose
  - loose: any number of other hops may occur between the specified hops
  - strict: every hop must come directly from the list, in the order specified
    - if that interface is not directly connected, discard the packet and send back an error message

Source Route (cont'd)

- Router overwrites with address of outgoing interface
- This option must be copied to all fragments
  - so all fragments will take the same, specified route

IP Strict Source Route Option

- Code=137
  - Length
  - Pointer
  - First IP address (filled when started)
  - Second IP address (filled when started)
  - ...
  - Last IP address (filled when started)

- 32 bits
Option #3: Timestamping

- Allows intermediate routers to insert 32-bit (ms since midnight UT) timestamps in the option
  - right now: 6,480,000,000
- If IP addresses filled by source, only specified routers will insert timestamp

<table>
<thead>
<tr>
<th>Code</th>
<th>Length</th>
<th>Pointer</th>
<th>Overflow</th>
<th>Flags</th>
</tr>
</thead>
<tbody>
<tr>
<td>First IP address (may be filled by source)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>First timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last IP address (may be filled by source)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Last timestamp</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

32 bits

Option #3: Timestamping (cont’d)

- How many entries possible?
- Problem of unsynchronized clocks?
  - Not copied on fragmentation; in first fragment only

Option #5: Router Alert

- Alerts routers to more closely examine the contents of a “special” IP packet
  - example protocol benefiting from this: RSVP
- Value has only one interesting interpretation: “pay attention to this packet”
  - All fragments carry the option

<table>
<thead>
<tr>
<th>Code</th>
<th>Length</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>32 bits</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Summary

1. IP provides a universal intermediate layer, routing, and fragmentation and reassembly
   - IP is unreliable, best-effort delivery
   - Fragmentation is infrequent, undesirable; is it necessary?
   - IP is extensible through the 40-byte Options field

Next Lecture

- The Address Resolution Protocol (ARP)