Internet Protocol (IP)

CPSC 527
UBC
Overview

- Internetworking
- IP Address format
- IP data forwarding
- Fragmentation and reassembly
Internetworking

- Internetwork = Collection of networks
  Connected via routers
Internet = Virtual Network

- Any computer can talk to any other computer
Class A: (1+3 bytes)
- Network: 0
- Local: 7, 24 bits

Class B: (2+2 bytes)
- Network: 10
- Local: 2, 14, 16 bits

Class C: (3+1 bytes)
- Network: 110
- Local: 3, 21, 8 bits

Class D:
- Network: 1110
- Host Group (Multicast): 4, 28 bits

Local = Subnet + Host (Variable length)
## Classes and Dotted Decimal Notation

- Binary: 11000000 00000101 00110000 00000011
- Hex Colon: C0:05:30:03
- Dotted Decimal: 192.5.48.3

<table>
<thead>
<tr>
<th>Class</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0 through 127</td>
</tr>
<tr>
<td>B</td>
<td>128 through 191</td>
</tr>
<tr>
<td>C</td>
<td>192 through 223</td>
</tr>
<tr>
<td>D</td>
<td>224 through 239</td>
</tr>
<tr>
<td>E</td>
<td>240 through 255</td>
</tr>
</tbody>
</table>
An Addressing Example

- All hosts on a network have the same network prefix
Subnetting

- With classes, the network part is 1-byte, 2-byte, or 3-byte long. You need class B address space just for 257 addresses.
- Any number of bits can be treated as one “subnetwork”
- Example: First 23 bits = subnet
  Address: 10010100 10101000 00010000 11110001
  Mask: 11111111 11111111 11111110 00000000
  .AND. 10010100 10101000 00010000 00000000

Network

Subnet 1  Subnet 2  ...  Subnet n
Supernetting

- Subnetting = subset of a network
- Supernet = superset of networks
  \[ \text{Supernet} = \sum \text{Class C addresses} \]

- Example:
  - Class C 1: 11010100 10101000 00010000
  - Class C 2: 11010100 10101000 00010001
  - Supernet: 11010100 10101000 00010000

- First 23 bits = subnet
  - Address: 11010100 10101000 00010000 11110001
  - Mask: 11111111 11111111 11111110 00000000
  - .AND. 10010100 10101000 00010000 00000000
Special IP Addresses

- All-0 host suffix ⇒ Network Address
- All-0s ⇒ This computer
  (In some old networks: 0.0.0.0 = broadcast. Not used.)
- All-0s network ⇒ This network.
  E.g., 0.0.0.2 = Host 2 on this network
- All-1 host suffix ⇒ All hosts on the destination net
  (directed broadcast),
  All-0 host suffix ⇒ Berkeley directed broadcast address
- All-1s ⇒ All hosts on this net (limited broadcast)
  ⇒ Subnet number cannot be all 1
- 127.*.*.* ⇒ Loopback through IP layer
Private Addresses

- Any organization can use these inside their network. Can’t go on the internet. [RFC 1918]
- 10.0.0.0 - 10.255.255.255 (10/8 prefix)
- 172.16.0.0 - 172.31.255.255 (172.16/12 prefix)
- 192.168.0.0 - 192.168.255.255 (192.168/16 prefix)
Classless Interdomain Routing (CIDR)

- Pronounced “Cider”
- Classless ⇒ Forget classes.
  Use Addresses and prefix lengths [RFC1517-1520]
- All routing table entries have prefix lengths
  Example: 164.107.61.0/26
Multi-Homed Hosts

- Each interface has an address.
  Two or more interfaces ⇒ Multi-homed hosts
- Multihoming is for reliability or performance
Routers and the IP Addressing Principle

- Routers have two or more addresses. One for each interface.

Diagram:
- Ethernet: 131.108.0.0, 131.108.99.5
- Token Ring: 223.24.129.2, 223.240.129.0
- WAN 78.0.0.0
- 223.240.129.17

- 78.0.0.17
IP Features

- Connectionless service
- Variable size datagrams
- Best-effort delivery: Delay, out-of-order, corruption, and loss possible. Higher layers should handle these.
- Handles only data forwarding
  Uses routing tables prepared by other protocols, e.g., Open Shortest Path First (OSPF), Routing Information Protocol (RIP)
- Provides only “Send” and “Delivery” services
  Error and control messages generated by Internet Control Message Protocol (ICMP)
Forwarding an IP Datagram

- Delivers datagrams to destination network (subnet)
- Routers maintain a “routing table” of “next hops”
- Next Hop field does not appear in the datagram

Table at R2:

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>Net 1</td>
<td>Forward to R1</td>
</tr>
<tr>
<td>Net 2</td>
<td>Deliver Direct</td>
</tr>
<tr>
<td>Net 3</td>
<td>Deliver Direct</td>
</tr>
<tr>
<td>Net 4</td>
<td>Forward to R3</td>
</tr>
</tbody>
</table>

Fig 16.2
IP Addresses and Routing Table Entries

- IF ((Mask[i] & Destination Addr) = = Destination[i])

  *Forward to NextHop[i]*

<table>
<thead>
<tr>
<th>Destination</th>
<th>Mask</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>30.0.0.0</td>
<td>255.0.0.0</td>
<td>40.0.0.7</td>
</tr>
<tr>
<td>40.0.0.0</td>
<td>255.0.0.0</td>
<td>Deliver direct</td>
</tr>
<tr>
<td>128.1.0.0</td>
<td>255.255.0.0</td>
<td>Deliver direct</td>
</tr>
<tr>
<td>192.4.10.0</td>
<td>255.255.255.0</td>
<td>128.1.0.9</td>
</tr>
</tbody>
</table>
### Sample Routing Table

<table>
<thead>
<tr>
<th>Network-Address</th>
<th>Netmask</th>
<th>Gateway-Address</th>
<th>Interface</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.0.0.0</td>
<td>0.0.0.0</td>
<td>24.93.104.1</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>24.93.104.0</td>
<td>255.255.248.0</td>
<td>24.93.104.238</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>24.93.104.238</td>
<td>255.255.255.255</td>
<td>127.0.0.1</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>24.255.255.255</td>
<td>255.255.255.255</td>
<td>24.93.104.238</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td>255.0.0.0</td>
<td>127.0.0.1</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>128.146.0.0</td>
<td>255.255.0.0</td>
<td>164.107.61.254</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>164.107.61.0</td>
<td>255.255.255.255</td>
<td>164.107.61.210</td>
<td>164.107.61.210</td>
<td>1</td>
</tr>
<tr>
<td>164.107.61.210</td>
<td>255.255.255.255</td>
<td>164.107.61.210</td>
<td>164.107.61.210</td>
<td>1</td>
</tr>
<tr>
<td>164.107.255.255</td>
<td>255.255.255.255</td>
<td>164.107.61.210</td>
<td>164.107.61.210</td>
<td>1</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td>224.0.0.0</td>
<td>24.93.104.238</td>
<td>24.93.107.238</td>
<td>1</td>
</tr>
<tr>
<td>224.0.0.0</td>
<td>224.0.0.0</td>
<td>164.107.61.210</td>
<td>164.107.61.210</td>
<td>1</td>
</tr>
<tr>
<td>255.255.255.255</td>
<td>255.255.255.255</td>
<td>164.107.61.210</td>
<td>164.107.61.210</td>
<td>1</td>
</tr>
</tbody>
</table>
## IP Datagram Format

<table>
<thead>
<tr>
<th>Vers</th>
<th>H. Len</th>
<th>Service Type</th>
<th>Total Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Identification</th>
<th>Flags</th>
<th>Fragment Offset</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time to live</td>
<td>Type</td>
<td>Header Checksum</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Source IP Address</th>
<th>Destination IP Address</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>IP Options (May be omitted)</th>
<th>Padding</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Data</th>
</tr>
</thead>
</table>
IP Header Format

- **Version (4 bits)**
- **Internet header length (4 bits):** in 32-bit words. Min header is 5 words or 20 bytes.
- **Type of service (8 bits):** Reliability, precedence, delay, and throughput
- **Total length (16 bits):** header + data in bytes. Total must be less than 64 kB.
- **Identifier (16 bits):** Helps uniquely identify the datagram during its life for a given source, destination address
IP Header (Cont)

- Flags (3 bits): More flag - used for fragmentation
  No-fragmentation
  Reserved
- Fragment offset (13 bits): In units of 8 bytes
- Time to live (8 bits): Specified in router hops
- Protocol (8 bits): Next level protocol to receive the data
- Header checksum (16 bits): 1’s complement sum of all 16-bit words in the header
IP Header (Cont)

- Source Address (32 bits): Original source. Does not change along the path.
- Destination Address (32 bits): Final destination. Does not change along the path.
- Options (variable): Security, source route, record route, stream id (used for voice) for reserved resources, timestamp recording
- Padding (variable): Makes header length a multiple of 4
- Data (variable): Data + header ≤ 65,535 bytes
Transmission Across An Internet

- Datalink/network header changes at every hop

Diagram:

- Source (S) on Net 1
- Router R1 on Net 2
- Router R2 on Net 3
- Destination (D)

Headers:
- Header 1: IP header
- Header 2
- Header 3
- Datagram
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: MTU

![Diagram showing MTU values of different networks]

Net 1: MTU=1500
Net 2: MTU=1000
<table>
<thead>
<tr>
<th>Decimal</th>
<th>Keyword</th>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Reserved</td>
<td>Reserved Protocol</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 Forwarding Process

- **Packet Received**
  - Header checksum valid?
    - No
      - Discard Packet
    - Yes
      - Decrement TTL
        - TTL\(\geq 0\)?
          - No
            - Send ICMP Error Message to source
          - Yes
            - Route table lookup
              - Route found?
                - No
                  - Build new packet
                    - Transmit the packet
                - Yes
                  - Send ARP Request
                    - No
                      - ARP Response Received?
                        - Yes
                          - Send ARP Request
                        - No
                          - MAC Address Available?
                            - Yes
                              - Send ARP Request
                            - No
                              - Default Route Available?
                                - Yes
                                  - Build new packet
                                    - Transmit the packet
                                - No
                                  - Discard Packet
## 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>

<table>
<thead>
<tr>
<th>Flag Copy</th>
<th>Class</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>1b</td>
<td>2b</td>
<td>5b</td>
</tr>
</tbody>
</table>

- **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>
</tbody>
</table>

| P    | 128.2.3.4 | 128.7.8.9 | 128.10.4.12 |
## Route Recording

<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>
</tbody>
</table>

| P    | 128.2.3.4 | 128.7.8.9 | Empty      |
### Timestamp Option

<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></td>
<td></td>
<td>IP Address n</td>
<td>Timestamp n</td>
</tr>
</tbody>
</table>

2-31
Fragmentation

- Datagrams larger than MTU are fragmented
- Original header is copied to each fragment and then modified (fragment flag, fragment offset, length,...)

```
IP Header + Original Datagram

IP Hdr 1  |  Data 1  |
         |         |
         |         | IP Hdr 2  |  Data 2  |
         |         |           |         |
         |         | IP Hdr 3  |  Data 3  |
```
Fragmentation

MTU = 1500B

ID = 12345, More = 1
Offset = 0W, Len = 1500B

MTU = 512B

ID = 12345, More = 1
Offset = 0W, Len = 512B

MTU = 256B

ID = 12345, More = 1
Offset = 0W, Len = 256B

ID = 12345, More = 1
Offset = 32W, Len = 256B

ID = 12345, More = 1
Offset = 64W, Len = 256B

ID = 12345, More = 1
Offset = 96W, Len = 256B

ID = 12345, More = 1
Offset = 128W, Len = 256B

ID = 12345, More = 1
Offset = 160W, Len = 220B

ID = 12345, More = 1
Offset = 64W, Len = 512B

ID = 12345, More = 1
Offset = 64W, Len = 256B

ID = 12345, More = 1
Offset = 96W, Len = 256B

ID = 12345, More = 1
Offset = 128W, Len = 476B

ID = 12345, More = 1
Offset = 128W, Len = 256B

ID = 12345, More = 1
Offset = 160W, Len = 220B
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. It is not possible to tell how many times fragmented.
- Minimum MTU along a path ⇒ Path MTU
IPv4 uses 32-bit addresses organized as network prefix and host suffix.

Four classes of networks: A, B, C, D

Routers determine next hop using routing tables

IP provides connectionless unreliable service