Intrernetworking

An arbitrary collection of networks interconnected to provide host-to-host packet delivery.

The Internet Protocol (IP)

A common protocol that bridges the heterogeneity between hosts of different types of networks.

E.g. Interconnection of hosts H1 and H8.
“Best Effort” Networking

IP is a datagram connectionless protocol

Does not provide any type of guarantee about packet delivery
Out of order packet deliveries, duplicate packets, no error correction

If packets are lost, IP does not try to recover or retransmit (though lower or higher layer functionalities may do so)

IPoAC (IP over Avian Carriers) RFC 1149. Use pigeons to carry IP packets

IPoAC has been implemented successfully carrying 9 packets with 55% packet loss, and delay from 3,000-6,000 sec.

---

**IPV4 Packet Format**

<table>
<thead>
<tr>
<th>Field</th>
<th>Length (Bits)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4</td>
<td>Indicates the version of the IP protocol</td>
</tr>
<tr>
<td>Header length</td>
<td>4</td>
<td>Number of 32-bit words in the header</td>
</tr>
<tr>
<td>Type of Service (TOS)</td>
<td>8</td>
<td>Allow packets to be treated differently based on needs</td>
</tr>
<tr>
<td>Total Length (Bytes)</td>
<td>16</td>
<td>Total length of the packet</td>
</tr>
<tr>
<td>Identification</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Flags</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>13</td>
<td></td>
</tr>
<tr>
<td>Time to Live (TTL)</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>Protocol</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>Source IP Address</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Payload</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

**IP Packet Header Fields(1)**

- **Version number (4 bits)**
  - Indicates the version of the IP protocol
  - Necessary to know what other fields to expect
  - Typically "4" (for IPv4), and sometimes "6" (for IPv6)

- **Header length (4 bits)**
  - Number of 32-bit words in the header
  - Typically "5" (for a 20-byte IPv4 header)
  - Can be more when "IP options" are used

- **Type-of-Service (8 bits)**
  - Allow packets to be treated differently based on needs
  - E.g., low delay for audio, high bandwidth for bulk transfer
IP Packet Header Fields (2)

**Total length** (16 bits)
- Number of bytes in the packet
- Maximum size is 63,535 bytes \(2^{16} - 1\)
- though underlying links may impose harder limits

**Fragmentation information** (32 bits)
- Packet identifier, flags, and fragment offset
- Supports dividing a large IP packet into fragments
- in case a link cannot handle a large IP packet

**Time-To-Live** (8 bits)
- Used to identify packets stuck in forwarding loops
- and eventually discard them from the network

---

Time To Live (TTL) Field

Potential robustness problem
- Forwarding loops can cause packets to cycle forever
- Confusing if the packet arrives much later

**Time-to-live field in packet header**
- TTL field decremented by each router on the path
- Packet is discarded when TTL field reaches 0...
- ...and "time exceeded" message is sent to the source

---

Protocol Field

**Protocol** (8 bits)
- Identifies the higher-level protocol
  - E.g., "6" for the Transmission Control Protocol (TCP)
  - E.g., "17" for the User Datagram Protocol (UDP)
- Important for demultiplexing at receiving host
- Indicates what kind of header to expect next

<table>
<thead>
<tr>
<th>protocol=6</th>
<th>protocol=17</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP header</td>
<td>IP header</td>
</tr>
<tr>
<td>TCP header</td>
<td>UDP header</td>
</tr>
</tbody>
</table>
**Checksum Field**

- **Checksum (16 bits)**
  - Sum of all 16-bit words in the IP packet header
  - If any bits of the header are corrupted in transit
  - ... the checksum won’t match at receiving host
  - Receiving host discards corrupted packets
    - Sending host will retransmit the packet, if needed

\[
\begin{align*}
134 & \quad + \quad 212 \\
\hline
   & \quad = \quad 346
\end{align*}
\]

\[
\begin{align*}
134 & \quad + \quad 216 \\
\hline
   & \quad = \quad 350
\end{align*}
\]

Mismatch!

---

**IP Addresses Field**

- **Two IP addresses**
  - Source IP address (32 bits)
  - Destination IP address (32 bits)

- **Destination address**
  - Unique identifier for the receiving host
  - Allows each node to make forwarding decisions

- **Source address**
  - Unique identifier for the sending host
  - Recipient can decide whether to accept packet
  - Enables recipient to send a reply back to source

---

**Fragmentation and Reassembly**

- Various networks accept different packet size
  - E.g.

  ![Fragmentation Diagram](image)

  Each fragment is itself a self-contained IP datagram transmitted over various physical networks
  Each IP datagram is re-encapsulated for each physical networks
Header Fields

IPV4 Addresses

A unique 32-bit number
Identifies an interface (on a host, on a router, ...)
Represented in dotted-quad notation

A Flat Architecture

Suppose hosts had arbitrary addresses
Then every router would need a lot of information
...to know how to direct packets toward the host
IP Addressing
Routing protocols need addresses to find the destinations
IP addresses are hierarchical (Why?)
Network part
Host part
Class-based Addressing

IP Address Classes
Class A:
Initial byte: 0 – 127, 126 Nets; [0, 127 reserved]; 16,777,214 hosts per network
MIT IP: 18.9.22.169
Class B:
Initial byte: 128 – 191, 16,384 networks; 65,532 hosts per network
Arizona IP: 129.196.134.37
Class C:
Initial byte: 192 – 223; 2,097,152 networks; 254 hosts per network
Your home IP: 192.168.1.67

Think of Postal Addresses
Mail Addresses
Zip code: 85721
Street: E. Speedway Blvd.
Building on street: 1230
Room in building: 357
Name of occupant: John Smith

Delivering mail
Deliver letter to the post office in the zip code
Assign letter to mailman covering the street
Drop letter into mailbox for the building/room
Give letter to the appropriate person
Hierarchy in IP Addresses

Divided into network and host portions (left and right)
12.34.158.0/24 is a 24-bit prefix with 2^8 addresses

```
12 34 158 5
00001100 00100010 10011110 00000101
```

Network (24 bits) Host (8 bits)

Subnetting

<table>
<thead>
<tr>
<th>Network number</th>
<th>Host number</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class B address</td>
<td>00000000</td>
</tr>
<tr>
<td>Subnet mask (255.255.255.0)</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Network number</th>
<th>Subnet ID</th>
<th>Host ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subnetted address</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

An Example of Subnetting

Subnet mask: 255.255.255.128
Subnet number: 128.96.34.0

128.96.34.15
128.96.34.3
128.96.34.130
128.96.34.139

Subnet mask: 255.255.255.128
Subnet number: 128.96.34.128

128.96.34.129
128.96.34.131
128.96.34.133
128.96.34.135

Subnet mask: 255.255.255.0
Subnet number: 128.96.33.0

H1
H2
H3
H4
Easier to Route Packets

Number related hosts from a common subnet
1.2.3.0/24 on the left LAN
5.6.7.0/24 on the right LAN

Easy to Add New Hosts

No need to update the routers
E.g., adding a new host 5.6.7.213 on the right
Doesn’t require adding a new forwarding entry

Classless Inter-Domain Routing (CIDR)

Use two 32-bit numbers to represent a network.
Network number = IP address + Mask

IP Address : 12.4.0.0   IP Mask: 255.254.0.0

Address

Mask

Written as 12.4.0.0/15
CIDR: Hierarchical Address Allocation
Prefixes are key to Internet scalability
Address allocated in contiguous chunks (prefixes)
Routing protocols and packet forwarding based on prefixes
Today, routing tables contain ~150,000-200,000 prefixes

Obtaining a Block of Addresses
Separation of control
Prefix: assigned to an institution
Addresses: assigned by the institution to their nodes

Who assigns prefixes?
Internet Corporation for Assigned Names and Numbers
Allocates large address blocks to Regional Internet Registries
Regional Internet Registries (RIRs)
E.g., ARIN (American Registry for Internet Numbers)
Allocates address blocks within their regions
Allocated to Internet Service Providers and large institutions
Internet Service Providers (ISPs)
Allocate address blocks to their customers
Who may, in turn, allocate to their customers...

Are 32-bit Addresses Enough?
Not all that many unique addresses
\[ 2^{32} = 4,294,967,296 \text{ (just over four billion)} \]
Plus, some are reserved for special purposes
And, addresses are allocated in larger blocks

Many devices need IP addresses
Computers, PDAs, routers, tanks, toasters, ...

Long-term solution: a larger address space
IPv6 has 128-bit addresses \[ (2^{128} = 3.403 \times 10^{38}) \]

Short-term solutions: limping along with IPv4
Private addresses
Network address translation (NAT)
Dynamically-assigned addresses (DHCP)
Hard Policy Questions

How much address space per geographic region?
  Equal amount per country?
  Proportional to the population?
  What about addresses already allocated?

Address space portability
  Keep your address block when you change providers?
  Pro: avoid having to renumber your equipment
  Con: reduces the effectiveness of address aggregation

Keeping the address registries up to date
  What about mergers and acquisitions?
  Delegation of address blocks to customers?
  As a result, the registries are horribly out of date