Packet and Packet Header

Protocols are rules ("syntax" and "grammar") governing communication between nodes.

Just as with the postal system, the "content" you want to send must be put into an envelope and the envelope must be addressed.

The "envelope" in this case is the packet header. The format of a packet header is part of the protocol.

For the Internet, the network-layer protocol is the Internet Protocol (IP).

Encapsulation

Each protocol has its own "envelope".

- Each protocol attaches its header to the packet.
- So we have a protocol wrapped/encapsulated inside another protocol.
- Each layer of header contains a protocol de-multiplexing field to identify the "packet handler" the next layer up, e.g.,
  - protocol number
  - port number

Previously... the Internet is a packet switched network:

- Data is parcelled into packets.
- Each packet carries a destination address.
- Each packet is routed independently.

Network Layer

Where are we now?

Application
  - DNS
  - Socket API

Transport
  - Network

Network
  - Link

Link
  - Physical

Packet and Packet Header

Protocols are rules ("syntax" and "grammar") governing communication between nodes.

Just as with the postal system, the "content" you want to send must be put into an envelope and the envelope must be addressed.

The "envelope" in this case is the packet header. The format of a packet header is part of the protocol.

For the Internet, the network-layer protocol is the Internet Protocol (IP).

Encapsulation

Each protocol has its own "envelope".

- Each protocol attaches its header to the packet.
- So we have a protocol wrapped/encapsulated inside another protocol.
- Each layer of header contains a protocol de-multiplexing field to identify the "packet handler" the next layer up, e.g.,
  - protocol number
  - port number
IPv4 Packet Header Format

<table>
<thead>
<tr>
<th>Field</th>
<th>Length</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>4 bits</td>
<td>Version number of IP version</td>
</tr>
<tr>
<td>Header Length (bytes)</td>
<td>4 bits</td>
<td>Size of remaining header (length header + remaining data)</td>
</tr>
<tr>
<td>Type of Service (TOS)</td>
<td>8 bits</td>
<td>Differentiated services</td>
</tr>
<tr>
<td>Total Length (bytes)</td>
<td>16 bits</td>
<td>Total length of header + data (including options)</td>
</tr>
<tr>
<td>Identification</td>
<td>20 bytes</td>
<td>Identification number</td>
</tr>
<tr>
<td>Time to Live (TTL)</td>
<td>8 bits</td>
<td>Time to live before discarding the packet</td>
</tr>
<tr>
<td>Protocol</td>
<td>16 bits</td>
<td>Protocol number of upper layer protocol</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>16 bits</td>
<td>Checksum of header</td>
</tr>
<tr>
<td>Source IP Address</td>
<td>32 bits</td>
<td>Source IP address</td>
</tr>
<tr>
<td>Destination IP Address</td>
<td>32 bits</td>
<td>Destination IP address</td>
</tr>
<tr>
<td>Options (if any)</td>
<td></td>
<td>Options (e.g., TCP/UDP packet, max size?)</td>
</tr>
<tr>
<td>Payload (e.g., TCP/UDP packet, max size?)</td>
<td></td>
<td>Payload</td>
</tr>
</tbody>
</table>

Packet Forwarding

Goal: deliver packets through routers from source to destination

- **source node** puts destination address in packet header
- **each router** node on the Internet:
  - looks up destination address in its routing/forwarding table
  - we’ll study several path selection (i.e., routing) algorithms
  - sends the packet to the next hop towards the destination
  - routes may change during session
  - analogy: driving, asking directions

IPv4 Addressing: Introduction

IPv4 address: 32-bit identifier for host/router interface

- interface (i.f.): connection between host/router and physical link
- **routers** typically have multiple interfaces
- host **may** have multiple interfaces

Flat vs. Hierarchical Addressing

Flat addressing:
- each router needs 10 entries in its routing table

Hierarchical addressing:
- **hosts** only need to know the default router, usually its border router
- each **border router** keeps in its routing table:
  - addresses of all hosts within its own network
  - next hop address of other networks
  - (Cf. forwarding table stores the outgoing interface number)
IPv4 Addressing

Independent of physical hardware address

32-bit number represented as dotted decimal:
• for ease of reference
• each # is the decimal representation of an octet

Divided into two parts:
• network prefix, globally assigned
  • route to network first
• host ID, assigned locally

Example: 12.34.158.0/24
is a 24-bit network prefix with 2^8 host addresses

Classfull Addresses

For the example network prefix: 12.34.158.0/24
• how many hosts can the network have?

What is a good partition of the 32-bit address space between the network and host parts?

Historically ... classfull addresses:

Class A: 0[10] *, very large /8 blocks (e.g., MIT has 18.0.0.0/8)
Class B: 10[10] *, large /16 blocks (e.g., UM has 141.213.0.0/16)
Class C: 110[10] *, small /24 blocks (e.g., AT&T Labs has 192.20.225.0/24)
Class D: 1110[10] *, multicast groups
Class E: 11110[10] *, reserved for future use

Subnets

A network can be further divided into subnets

What makes a subnet?
• the IP addresses of all interfaces within a subnet have the same network prefix
• hosts within a subnet can physically reach each other without intervening router

Classfull Addresses

Problems:
1. everybody wanted a Class B address (the Goldilock problem)
2. address space usage became inefficient
3. routing table explosion
4. and then, address space became scarce...
   • by 1992, half of Class B had been allocated, would have been exhausted by 3/94

Solution:
Make network portion of address arbitrary length, determined by a prefix mask
• uses two 32-bit numbers to represent a network address
  • network address = IP address & prefix mask
Classless InterDomain Routing (CIDR)

Usually written as a.b.c.d/x, where x is number of bits in the network portion of the address: 12.4.0.0/15

<table>
<thead>
<tr>
<th>IP address:</th>
<th>12.4.0.0</th>
<th>00001000 00000010 00000000 00000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>prefix mask:</td>
<td>255.254.0.0</td>
<td>11111111 11111111 00000000 00000000</td>
</tr>
</tbody>
</table>

Another example (not at octet boundary!):

200.23.16.0/23

CIDR: Route Aggregation

Hierarchical addressing allows efficient advertisement of routing information:

CIDR: Hierarchical Address Allocation

Prefixes are key to Internet routing scalability

- address allocation by ICANN, ARIN/RIPE/APNIC and by ISPs
- routing protocols and packet forwarding based on prefixes
- today, routing tables contain ~150,000-200,000 prefixes

Longest Prefix Match: More Specific Routes

ISPs-R-Us has a more specific route to **Organization 1**
Packet Forwarding Summary

Forwarding is destination-based:
• packet has a destination address
• router identifies longest-matching prefix

Forwarding table:
• maps each IP prefix to next-hop link(s)
• entries can be statically configured
  • e.g., "map 12.34.158.0/24 to Serial0/0.1"

But, this doesn’t adapt to
• failures
• new equipment
• the need to balance load
  • ...

That is where routing protocols come in...
[more on routing later in the term]

Special IPv4 Addresses

Network identification:
• 0s on host part, e.g., 141.212.0.0 (cannot be used to send packets)

Directed broadcast:
• 0xffff on host part, e.g., 141.212.255.255
• broadcast to all hosts on a network (e.g., 141.212) (not implemented)

Limited broadcast:
• 0xffffffff, received by all hosts on LAN, not forwarded beyond LAN

This computer:
• 0.0.0.0 used at startup to solicit IP address with RARP (deprecated)

Loopback address:
• 127.*.*.*, (usually 127.0.0.1), named localhost
  • packets sent to localhost traverse down the kernel networking code to the network layer and back up to the application without traversing the network, useful for testing networking code

Internet Control Message Protocol

ICMP is used by hosts and routers to communicate network-level information
• error reporting: unreachable host, network, port, protocol
• echo request/reply (used by ping)
• ICMP error message does not trigger another ICMP message

ICMP runs on network-layer, but “above” IP:
• ICMP messages are carried inside IP datagrams

ICMP message format:

<table>
<thead>
<tr>
<th>Type</th>
<th>Code</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>echo reply (ping)</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>dest network unreachable</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>dest host unreachable</td>
</tr>
<tr>
<td>3</td>
<td>2</td>
<td>dest protocol unreachable</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>dest port unreachable</td>
</tr>
<tr>
<td>3</td>
<td>4</td>
<td>frag needed but DF set</td>
</tr>
<tr>
<td>3</td>
<td>6</td>
<td>dest network unknown</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
<td>dest host unknown</td>
</tr>
<tr>
<td>8</td>
<td>0</td>
<td>echo request (ping)</td>
</tr>
<tr>
<td>9</td>
<td>0</td>
<td>route advertisement</td>
</tr>
<tr>
<td>10</td>
<td>0</td>
<td>router discovery</td>
</tr>
<tr>
<td>11</td>
<td>0</td>
<td>TTL expired</td>
</tr>
<tr>
<td>12</td>
<td>0</td>
<td>bad IP header</td>
</tr>
</tbody>
</table>
Traceroute and ICMP

How to discover the routers along a path?

Source sends a series of UDP packets to destination
- first 3 packets have TTL set to 1, spaced 3 secs apart
- next 3 packets have TTL set to 2, etc.
- packets are all sent to an unused port number

When packet’s TTL expired:
- router discards packet
- and sends to source an ICMP message (type 11, code 0)
- message includes IP address of router

When ICMP message arrives back at the source, source calculates round-trip time (RTT)

Stopping criterion:
- UDP packets eventually arrive at destination host
- destination returns ICMP “destination port unreachable” message (type 3, code 3)
- when source gets this ICMP message, it stops sending UDP packets

“Real” Internet Delays and Routes

traceroute to eurocom.fr (92.243.13.96), 64 hops max, 40 byte packets

<table>
<thead>
<tr>
<th>Hop</th>
<th>IP Address</th>
<th>RTT (ms)</th>
<th>Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>141.212.110.1</td>
<td>1.386</td>
<td>0.560</td>
</tr>
<tr>
<td>2</td>
<td>vss-cse-engin.umich.edu (141.213.127.134)</td>
<td>0.486</td>
<td>0.291</td>
</tr>
<tr>
<td>3</td>
<td>case-bin-rab-r-bin-rab1.umnet.umich.edu (192.12.80.177)</td>
<td>0.373</td>
<td>0.348</td>
</tr>
<tr>
<td>4</td>
<td>v-barb-baslab-2-r-bin-sab1.umnet.umich.edu (192.12.80.11)</td>
<td>0.952</td>
<td>0.576</td>
</tr>
<tr>
<td>5</td>
<td>v-bin-sab-internet-sab1.mich.net (150.12.80.37)</td>
<td>0.527</td>
<td>0.546</td>
</tr>
<tr>
<td>6</td>
<td>ae0x76.wsu5.mich.net (198.108.23.9)</td>
<td>1.476</td>
<td>1.928</td>
</tr>
<tr>
<td>7</td>
<td>test1.xfdor-123net.mich.net (198.108.23.20)</td>
<td>2.234</td>
<td>2.261</td>
</tr>
<tr>
<td>8</td>
<td>ae-8-3.0.1018.asbn0.tr-cps.internet2.edu (198.71.47.25)</td>
<td>22.498</td>
<td>22.594</td>
</tr>
<tr>
<td>9</td>
<td>64.57.20.106</td>
<td>22.737</td>
<td>22.698</td>
</tr>
<tr>
<td>10</td>
<td>* * *</td>
<td>* no response (probe lost, router not replying, 3 secs timer)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ae1x19.xfdor-123net.mich.net (198.108.23.20)</td>
<td>2.234</td>
<td>2.261</td>
</tr>
<tr>
<td>12</td>
<td>ae-8-3-0.1018.asbn0.tr-cps.internet2.edu (198.71.47.25)</td>
<td>22.498</td>
<td>22.594</td>
</tr>
<tr>
<td>13</td>
<td>64.57.20.106</td>
<td>22.737</td>
<td>22.698</td>
</tr>
<tr>
<td>14</td>
<td>* * *</td>
<td>* no response (probe lost, router not replying, 3 secs timer)</td>
<td></td>
</tr>
</tbody>
</table>

Transatlantic link

Traceroute to eurocom.fr (92.243.13.96), 64 hops max, 40 byte packets

<table>
<thead>
<tr>
<th>Hop</th>
<th>IP Address</th>
<th>RTT (ms)</th>
<th>Delay (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>141.212.110.1</td>
<td>1.386</td>
<td>0.560</td>
</tr>
<tr>
<td>2</td>
<td>vss-cse-engin.umich.edu (141.213.127.134)</td>
<td>0.486</td>
<td>0.291</td>
</tr>
<tr>
<td>3</td>
<td>case-bin-rab-r-bin-rab1.umnet.umich.edu (192.12.80.177)</td>
<td>0.373</td>
<td>0.348</td>
</tr>
<tr>
<td>4</td>
<td>v-barb-baslab-2-r-bin-sab1.umnet.umich.edu (192.12.80.11)</td>
<td>0.952</td>
<td>0.576</td>
</tr>
<tr>
<td>5</td>
<td>v-bin-sab-internet-sab1.mich.net (150.12.80.37)</td>
<td>0.527</td>
<td>0.546</td>
</tr>
<tr>
<td>6</td>
<td>ae0x76.wsu5.mich.net (198.108.23.9)</td>
<td>1.476</td>
<td>1.928</td>
</tr>
<tr>
<td>7</td>
<td>test1.xfdor-123net.mich.net (198.108.23.20)</td>
<td>2.234</td>
<td>2.261</td>
</tr>
<tr>
<td>8</td>
<td>ae-8-3.0.1018.asbn0.tr-cps.internet2.edu (198.71.47.25)</td>
<td>22.498</td>
<td>22.594</td>
</tr>
<tr>
<td>9</td>
<td>64.57.20.106</td>
<td>22.737</td>
<td>22.698</td>
</tr>
<tr>
<td>10</td>
<td>* * *</td>
<td>* no response (probe lost, router not replying, 3 secs timer)</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>ae1x19.xfdor-123net.mich.net (198.108.23.20)</td>
<td>2.234</td>
<td>2.261</td>
</tr>
<tr>
<td>12</td>
<td>ae-8-3-0.1018.asbn0.tr-cps.internet2.edu (198.71.47.25)</td>
<td>22.498</td>
<td>22.594</td>
</tr>
<tr>
<td>13</td>
<td>64.57.20.106</td>
<td>22.737</td>
<td>22.698</td>
</tr>
<tr>
<td>14</td>
<td>* * *</td>
<td>* no response (probe lost, router not replying, 3 secs timer)</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>srv4.feitlen.bis (92.243.13.96)</td>
<td>116.906</td>
<td>116.925</td>
</tr>
</tbody>
</table>