CS244a: An Introduction to Computer Networks

Handout 4: Layer 3 and the Internet Protocol (IP)

Nick McKeown
Professor of Electrical Engineering and Computer Science, Stanford University
nickm@stanford.edu
http://www.stanford.edu/~nickm

Outline

• IP: The Internet Protocol
  - Service characteristics
  - The IP Datagram format
  - IP addresses
  - Classless Interdomain Routing (CIDR)
  - An aside: Turning names into addresses (DNS)
  - Forwarding IP Datagrams
The Internet Protocol (IP)

- Characteristics of IP:
  - CONNECTIONLESS: mis-sequencing
  - UNRELIABLE: may drop packets...
  - BEST EFFORT: ... but only if necessary
  - DATAGRAM: individually routed

Protocol Stack:

- App
- Transport
- Network
- Link

TCP / UDP

IP

TCP Segment

IP Datagram

Source: A

Destination: B

Architectures, Links, Topology

Transparent
The IP Datagram

<table>
<thead>
<tr>
<th>vers</th>
<th>HLen</th>
<th>TOS</th>
<th>Total Length</th>
<th>Hop count</th>
<th>Offset within original packet</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>Flags</td>
<td>FRAG Offset</td>
<td>TTL</td>
<td>Protocol</td>
<td>checksum</td>
</tr>
<tr>
<td>&lt;=64 KBytes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Fragmentation

**Problem:** A router may receive a packet larger than the maximum transmission unit (MTU) of the outgoing link.

**Solution:** R1 fragments the IP datagram into multiple, self-contained datagrams.

- Offset=0 More Frag=0
- Offset=0 More Frag=1
Fragmentation

- Fragments are re-assembled by the destination host; not by intermediate routers.
- To avoid fragmentation, hosts commonly use path MTU discovery to find the smallest MTU along the path.
- Path MTU discovery involves sending various size datagrams until they do not require fragmentation along the path.
- Most links use MTU > 1500 bytes today.
- Try: `traceroute -f berkeley.edu 1500` and `traceroute -f berkeley.edu 1501`
- (DF=1 set in IP header, routers send "ICMP" error message, which is shown as "!F").
- Bonus: Can you find a destination for which the path MTU < 1500 bytes?

IP Addresses

- IP (Version 4) addresses are 32 bits long
- Every interface has a unique IP address:
  - A computer might have two or more IP addresses
  - A router has many IP addresses
- IP addresses are hierarchical
  - They contain a network ID and a host ID
  - E.g. Stanford addresses start with: 171.64...
- IP addresses are assigned statically or dynamically (e.g. DHCP)
- IP (Version 6) addresses are 128 bits long
IP Addresses

Originally there were 5 classes:

CLASS "A" 0 Net ID 16 Host-ID
CLASS "B" 10 Net ID 8 Host-ID
CLASS "C" 110 Net ID 8 Host-ID
CLASS "D" 1110 Multicast Group ID
CLASS "E" 11110 Reserved

A B C D 2^{32-1}

IP Addresses

Examples

Class "A" address: www.mit.edu
18.181.0.31
(18<128 => Class A)

Class "B" address: mekong.stanford.edu
171.64.74.155
(128<171<128+64 => Class B)
IP Addressing

Problem:

- Address classes were too "rigid". For most organizations, Class C were too small and Class B too big. Led to inefficient use of address space, and a shortage of addresses.
- Organizations with internal routers needed to have a separate (Class C) network ID for each link.
- And then every other router in the Internet had to know about every network ID in every organization, which led to large address tables.
- Small organizations wanted Class B in case they grew to more than 255 hosts. But there were only about 16,000 Class B network IDs.

Two solutions were introduced:

- **Subnetting** within an organization to subdivide the organization's network ID.
- **Classless Interdomain Routing (CIDR)** in the Internet backbone was introduced in 1993 to provide more efficient and flexible use of IP address space.

- CIDR is also known as "supernetting" because subnetting and CIDR are basically the same idea.
Subnetting

- Subnetting is a form of hierarchical routing.
- Subnets are usually represented via an address plus a subnet mask or “netmask”.
  - e.g.
    
    ```
    nickm@elaine17.Stanford.EDU > ifconfig hme0
    hme0: flags=863<UP,BROADCAST,NOTRAILERS,RUNNING,MULTICAST> mtu 1500
    inet 171.64.15.82 netmask ffffff00 broadcast 171.64.15.255
    ```
  - Netmask ffffff00: the first 24 bits are the subnet ID, and the last 8 bits are the host ID.
  - Can also be represented by a "prefix + length", e.g. 171.64.15.0/24, or just 171.64.15/24.
Example of subnetting at Stanford

Classless Interdomain Routing (CIDR) Addressing

- The IP address space is broken into line segments.
- Each line segment is described by a prefix.
- A prefix is of the form \( x/y \) where \( x \) indicates the prefix of all addresses in the line segment, and \( y \) indicates the length of the segment.
- e.g. The prefix 128.9/16 represents the line segment containing addresses in the range: 128.9.0.0 \ldots 128.9.255.255.
**Classless Interdomain Routing (CIDR)**

**Addressing**

- Most specific route = "longest matching prefix"

Prefix aggregation:
- If a service provider serves two organizations with prefixes, it can (sometimes) aggregate them to form a shorter prefix. Other routers can refer to this shorter prefix, and so reduce the size of their address table.
- E.g. ISP serves 128.9.14.0/24 and 128.9.15.0/24, it can tell other routers to send it all packets belonging to the prefix 128.9.14.0/23.

**ISP Choice:**
- In principle, an organization can keep its prefix if it changes service providers.
Size of the Routing Table at the core of the Internet

Source: http://www.cidr-report.org/

Prefix Length Distribution

Source: Geoff Huston, Jan 2006
Mapping Computer Names to IP addresses
The Domain Naming System (DNS)

Names are hierarchical and belong to a domain:
- e.g. elaine17.stanford.edu
- Common domain names: .com, .edu, .gov, .org, .net, .uk (or other country-specific domain).
- Top-level names are assigned by the Internet Corporation for Assigned Names and Numbers (ICANN).
- A unique name is assigned to each organization.

DNS Client-Server Model
- DNS maintains a hierarchical, distributed database of names
- Servers are arranged in a hierarchy.
- Each domain has a “root” server.
- An application needing an IP address is a DNS client.

A DNS Query
1. Client asks local server.
2. If local server does not have address, it asks the root server of the requested domain.
3. Addresses are cached in case they are requested again.

E.g. www.eecs.berkeley.edu

Example: On elaine machines, try ’host www.mit.edu’
An example of names and addresses

Mapping the path between two hosts

nickm@yuba.Stanford.EDU > host yuba
yuba.Stanford.EDU has address 171.64.74.58

nickm@yuba.Stanford.EDU > host bbr2-rtr.stanford.edu | sort -n
bbr2-rtr.stanford.edu has address 171.64.0.126
bbr2-rtr.stanford.edu has address 171.64.1.133
bbr2-rtr.stanford.edu has address 171.64.1.86
bbr2-rtr.stanford.edu has address 171.64.1.19
bbr2-rtr.stanford.edu has address 171.64.1.97
bbr2-rtr.stanford.edu has address 171.64.1.242
bbr2-rtr.stanford.edu has address 171.64.1.24
bbr2-rtr.stanford.edu has address 171.64.1.19
bbr2-rtr.stanford.edu has address 171.64.1.124
bbr2-rtr.stanford.edu has address 171.66.16.1
bbr2-rtr.stanford.edu has address 171.67.1.93
bbr2-rtr.stanford.edu has address 171.67.20.1
bbr2-rtr.stanford.edu has address 171.67.254.224
bbr2-rtr.stanford.edu has address 171.67.255.126
bbr2-rtr.stanford.edu has address 172.244.0
bbr2-rtr.stanford.edu has address 172.27.20.1
bbr2-rtr.stanford.edu has address 192.168.2.129
bbr2-rtr.stanford.edu has address 192.168.7.154

Example

Mapping the path between two hosts

nickm@yuba.Stanford.EDU > host bbr2-rtr.stanford.edu | sort -n
bbr2-rtr.stanford.edu has address 171.64.0.126
bbr2-rtr.stanford.edu has address 171.64.1.133
bbr2-rtr.stanford.edu has address 171.64.1.86
bbr2-rtr.stanford.edu has address 171.64.1.19
bbr2-rtr.stanford.edu has address 171.64.1.97
bbr2-rtr.stanford.edu has address 171.64.1.242
bbr2-rtr.stanford.edu has address 171.64.1.24
bbr2-rtr.stanford.edu has address 171.64.1.19
bbr2-rtr.stanford.edu has address 171.64.1.124
bbr2-rtr.stanford.edu has address 171.66.16.1
bbr2-rtr.stanford.edu has address 171.67.1.93
bbr2-rtr.stanford.edu has address 171.67.20.1
bbr2-rtr.stanford.edu has address 171.67.254.224
bbr2-rtr.stanford.edu has address 171.67.255.126
bbr2-rtr.stanford.edu has address 172.244.0
bbr2-rtr.stanford.edu has address 172.27.20.1
bbr2-rtr.stanford.edu has address 192.168.2.129
bbr2-rtr.stanford.edu has address 192.168.7.154
**Example**

*Mapping the path between two hosts*

![Diagram](image)

As of Jan 2006

Winter 2008   CS244a Handout 4  25

**An aside:**

*Error Reporting (ICMP) and traceroute*

**Internet Control Message Protocol:**

- Used by a router/end-host to report some types of error,
  - E.g. Destination Unreachable: packet can't be forwarded to/towards its destination.
  - E.g. Time Exceeded: TTL reached zero, or fragment didn't arrive in time. Traceroute uses this error to its advantage.
  - An ICMP message is an IP datagram, and is sent back to the source of the packet that caused the error.
**How a Router Forwards Datagrams**

- Every datagram contains a destination address.
- The router determines the prefix to which the address belongs, and routes it to the “Network ID” that uniquely identifies a physical network.
- All hosts and routers sharing a Network ID share same physical network.

### Forwarding/routing table

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next-hop</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>65/8</td>
<td>128.17.16.1</td>
<td>3</td>
</tr>
<tr>
<td>128.9/16</td>
<td>128.17.14.1</td>
<td>2</td>
</tr>
<tr>
<td>128.9/16</td>
<td>128.17.14.1</td>
<td>2</td>
</tr>
<tr>
<td>128.9/16/20</td>
<td>128.17.10.1</td>
<td>7</td>
</tr>
<tr>
<td>128.9/25/24</td>
<td>128.17.14.1</td>
<td>2</td>
</tr>
<tr>
<td>128.9/176/20</td>
<td>128.17.20.1</td>
<td>1</td>
</tr>
<tr>
<td>142.12/19</td>
<td>128.17.16.1</td>
<td>3</td>
</tr>
</tbody>
</table>

*Example:* 128.9.16.14 => Port 2
Forwarding Datagrams

- Is the datagram for a host on a directly attached network?
- If no, consult forwarding table to find *next-hop*.

Inside a router
Inside a router

Forwarding in an IP Router

- Lookup packet DA in forwarding table.
  - If known, forward to correct port.
  - If unknown, drop packet.
- Decrement TTL, update header Checksum.
- Forward packet to outgoing interface.
- Transmit packet onto link.

Question: How is the address looked up in a real router?
Making a Forwarding Decision
Class-based addressing

Class-based addressing allows for the categorization of IP addresses into different classes (A, B, C, D, E). This categorization helps in making efficient forwarding decisions.

**Routing Table:** The routing table contains entries for different IP address ranges to determine the next hop for traffic.

**Exact Match:** There are many well-known ways to find an exact match in a table.

Direct Lookup

IP Address

Memory

Next-hop, Port

**Problem:** With $2^{32}$ addresses, the memory would require 4 billion entries.
**Associative Lookups**

"Contents addressable memory" (CAM)

<table>
<thead>
<tr>
<th>Network Address</th>
<th>Port Number</th>
<th>Hit?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Advantages:**
- Simple

**Disadvantages:**
- Slow
- High Power
- Small
- Expensive

Search data is compared with every entry in parallel.

---

**Hashed Lookups**

Input: 32-bit Search Data
Output: 16-bit Hashed Address

Input: 16-bit Hashed Address
Output: 32-bit Associated Data

Hit?
Lookups Using Hashing

**An example**

![Diagram](image)

**Advantages:**
- Simple
- Expected lookup time can be small

**Disadvantages**
- Non-deterministic lookup time
- Inefficient use of memory
Trees and Tries

Binary Search Tree:

```
<   <   <   >
  >
  >
```

\[ \text{Nentries} \]

Binary Search Trie: ("reTRIEval")

```
0 1 0 1
0 1 0 1
```

Requires 32 memory references, regardless of number of addresses

Search Tries

Multiway tries reduce the number of memory references

16-ary Search Trie

```
0000, ptr
0000, 0
1111, ptr
0000, 0
```

```
1111, ptr
00001110000
11111111111
```

Question: Why not just keep increasing the degree of the trie?
**Classless Addressing**

**CIDR**

```
128.9.19/24
128.9.25/24
128.9.16/20 128.9.176/20
0 128.9/16 2^{32-1}
```

128.9.16.14

*Most specific route = "longest matching prefix"*

**Question:** How can we look up addresses if they are not an exact match?

---

**Ternary CAMs**

**Associative Memory**

<table>
<thead>
<tr>
<th>Value</th>
<th>Mask</th>
<th>Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.1.1.32</td>
<td>255.255.255.255</td>
<td>1</td>
</tr>
<tr>
<td>10.1.1.0</td>
<td>255.255.255.0</td>
<td>2</td>
</tr>
<tr>
<td>10.1.0.0</td>
<td>255.255.255.0</td>
<td>3</td>
</tr>
<tr>
<td>10.0.0.0</td>
<td>255.0.0.0</td>
<td>4</td>
</tr>
</tbody>
</table>

*Note:* Most specific routes appear closest to top of table
Longest prefix matches using Binary Tries

Example Prefixes:
- a) 00001
- b) 00010
- c) 00011
- d) 001
- e) 0101
- f) 011
- g) 10
- h) 1010
- i) 111
- j) 111100
- k) 11110001

Lookup Performance Required

<table>
<thead>
<tr>
<th>Line</th>
<th>Line Rate</th>
<th>Pkt-size=40B</th>
<th>Pkt-size=240B</th>
</tr>
</thead>
<tbody>
<tr>
<td>T1</td>
<td>1.5Mbps</td>
<td>4.68 Kpps</td>
<td>0.78 Kpps</td>
</tr>
<tr>
<td>OC3</td>
<td>55Mbps</td>
<td>480 Kpps</td>
<td>80 Kpps</td>
</tr>
<tr>
<td>OC12</td>
<td>622Mbps</td>
<td>1.94 Mpps</td>
<td>323 Kpps</td>
</tr>
<tr>
<td>OC48</td>
<td>2.5Gbps</td>
<td>7.81 Mpps</td>
<td>1.3 Mpps</td>
</tr>
<tr>
<td>OC192</td>
<td>10 Gbps</td>
<td>31.25 Mpps</td>
<td>5.21 Mpps</td>
</tr>
</tbody>
</table>
Discussion

• Why was the Internet Protocol designed this way?
  - Why connectionless, datagram, best-effort?
  - Why not automatic retransmissions?
  - Why fragmentation in the network?

• Must the Internet address be hierarchical?
• What address does a mobile host have?
• Are there other ways to design networks?