Lecture 12: Aggregation
Lecture 12 Overview

- Subnetting
  - Classless addressing

- Route aggregation
Class-based Addressing

Most significant bits determines “class” of address

<table>
<thead>
<tr>
<th>Class A</th>
<th>Network</th>
<th>Host</th>
<th>127 nets, 16M hosts</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class B</td>
<td>1 0</td>
<td></td>
<td>16K nets, 64K hosts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Class C</td>
<td>1 1 0</td>
<td></td>
<td>2M nets, 254 hosts</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Special addresses

- Class D (1110) for multicast, Class E (1111) experimental
- 127.0.0.1: local host (a.k.a. the loopback address)
- Host bits all set to 0: network address
- Host bits all set to 1: broadcast address
Individual networks may be composed of several LANs
- Only want traffic destined to local hosts on physical network
- Routers need a way to know which hosts on which LAN

Networks can be arbitrarily decomposed into subnets
- Each subnet is simply a prefix of the host address portion
- Subnet prefix can be of any length, specified with netmask
Subnet Addresses

Every (sub)network has an address and a netmask
- Netmask tells which bits of the network address is important
- Convention suggests it be a proper prefix

Netmask written as an all-ones IP address
- E.g., Class B netmask is 255.255.0.0
- Sometimes expressed in terms of number of 1s, e.g., /16

Need to size subnet appropriately for each LAN
- Only have remaining bits to specify host addresses
IP Address Problem (1991)

Address space depletion
- In danger of running out of classes A and B

Why?
- Class C too small for most organizations (only ~250 addresses)
- Very few class A – very careful about giving them out (who has 16M hosts anyway?)
- Class B – greatest problem
Classless Inter-Domain Routing (1993)

- Networks described by variable-length prefix and length
- Allows arbitrary allocation between network and host address

<table>
<thead>
<tr>
<th>Network</th>
<th>Host</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prefix</td>
<td>Mask=# significant bits representing prefix</td>
</tr>
</tbody>
</table>

- e.g. 10.95.1.2 contained within 10.0.0.0/8:
  » 10.0.0.0 is network and remainder (95.1.2) is host

Pro: Finer grained allocation; aggregation
Con: More expensive lookup: longest prefix match
Longest Matching Prefix

- Forwarding table contains many prefix/length tuples
  - They *need not* be disjoint!
  - E.g. 200.23.16.0/20 and 200.23.18.0/23
  - What to do if a packet arrives for destination 200.23.18.1?
  - Need to find the longest prefix in the table which matches it (200.23.18.0/23)

- Not a simple table, requires multiple memory lookups
  - Lots and lots of research done on this problem
  - Lots of this work was historically done by UCSD faculty
PATRICIA Tree (Radix tree)

(Practical Algorithm to Retrieve Information Coded In Alphanumeric)

- Straightforward way to look up LMP
  - Arrange route entries into a series of bit tests
  - Worst case = 32 bit tests
  - Problem: memory speed is a bottleneck

```
0
  ↓
10
  ↓
16
  ↓
19
  ↓
128.32.130/24
  ↓
128.32.150/24
```

Bit to test – 0 = left child, 1 = right child

CSE 123 – Lecture 12: Aggregation
Route Aggregation

Combine adjacent networks in forwarding tables

- Helps keep forwarding table size down

```
Organization 0
  200.23.16.0/23
Organization 1
  200.23.18.0/23
Organization 2
  200.23.20.0/23
Organization 7
  200.23.30.0/23
```

```
Fly-By-Night-ISP

“Implicit in BGP specification”
```

```
ISPs-R-Us

“Send me anything with addresses beginning 200.23.16.0/20”
```

```
Internet

“Send me anything with addresses beginning 199.31.0.0/16”
```
But what if address range is not contiguous?

```
Organization 0: 200.23.16.0/23
Organization 2: 200.23.20.0/23
Organization 7: 200.23.30.0/23
Organization 1: 200.23.18.0/23

Fly-By-Night-ISP

Send me anything with addresses beginning 200.23.16.0/20

ISP's-R-Us

Send me anything with addresses beginning 199.31.0.0/16 or 200.23.18.0/23

Internet
```
The space crunch…

Still running out of IP addresses… what to do?

Two solutions

❖ Network Address Translation – multiple multiple hosts on a single IP address (future class)
❖ Get bigger addresses -> IPv6

IPv6: 128bit addresses… we won’t run out

❖ 64bit routing prefix, 64bit host id

An IPv6 address (in hexadecimal)

\[
2001:0DB8:AC10:FE01:0000:0000:0000:0000
\]

Zeroes can be omitted

CSE 123 – Lecture 12: Aggregation
### IPv6 vs IPv4 header

<table>
<thead>
<tr>
<th>IPv6</th>
<th>IPv4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ver.</td>
<td>4</td>
</tr>
<tr>
<td>Traffic Class</td>
<td>0</td>
</tr>
<tr>
<td>Flow Label</td>
<td>20</td>
</tr>
<tr>
<td>Payload Length</td>
<td>20</td>
</tr>
<tr>
<td>Next Header</td>
<td>8</td>
</tr>
<tr>
<td>Hop Limit</td>
<td>16</td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
</tr>
<tr>
<td>Options...</td>
<td></td>
</tr>
<tr>
<td>Ver.</td>
<td>4</td>
</tr>
<tr>
<td>Hdr Len</td>
<td>10</td>
</tr>
<tr>
<td>Type of Service</td>
<td>8</td>
</tr>
<tr>
<td>Total Length</td>
<td>32</td>
</tr>
<tr>
<td>Identification</td>
<td>8</td>
</tr>
<tr>
<td>Flg</td>
<td>1</td>
</tr>
<tr>
<td>Fragment Offset</td>
<td>16</td>
</tr>
<tr>
<td>Time to Live</td>
<td>8</td>
</tr>
<tr>
<td>Protocol</td>
<td>4</td>
</tr>
<tr>
<td>Header Checksum</td>
<td>12</td>
</tr>
<tr>
<td>Source Address</td>
<td></td>
</tr>
<tr>
<td>Destination Address</td>
<td></td>
</tr>
</tbody>
</table>

- Gray bits are unique to each header
- Changes
  - Eliminate fragmentation-related fields
  - Eliminate header checksum
  - Added flow label
  - Quadruple size of addresses
  - IPv6 header (40 bytes) vs IPv4 (20 bytes)
IPv6 Transition is slow

Need to support both protocols at the same time
- Complicated… if a destination has both a IPv4 and IPv6 address which to use?

Less need in developed world -> slower adoption

That said
- All major operating systems now support IPv6
- All major router vendors
- US Mobile carriers (e.g., Tmobile, Verizon, etc)
- Offered as option by many US ISPs

In your lifetime it is likely that IPv6-based addressing will start to dominate
Summary

You can’t route efficiently on flat address spaces
- You’d need a table the size of all hosts on the Internet
- You’d need to send updates about that table to everyone

Network-layer addressing is done hierarchically
- Routing prefix + host suffix
- Originally, this split was done statically (class-based addressing)
- Now it is done dynamically (CIDR)
- Requires more complex forwarding table lookup
- Allows contiguous chunks of address space to be aggregated (for the purposes of routing) into fewer prefixes
For Next Time

• Read 3.2.6, 9.3.1

• Finish up Project 1!