15-441 Computer Networking

Lecture 8 – IP Addressing and Forwarding

Internet Protocol (IP)
• Hour Glass Model
  • Create abstraction layer that hides underlying technology from network application software
  • Make as minimal as possible
  • Allows range of current & future technologies
  • Can support many different types of applications

Logical Structure of Internet
• Ad hoc interconnection of networks
  • No particular topology
  • Vastly different router & link capacities
  • Send packets from source to destination by hopping through networks
    • Router connect one network to another
    • Different paths to destination may exist

What is an Internetwork?
• Multiple incompatible LANs can be physically connected by specialized computers called routers
• The connected networks are called an internetwork
  • The “Internet” is one (very big & successful) example of an internetwork

LAN 1 and LAN 2 might be completely different, totally incompatible LANs (e.g., Ethernet and ATM)
Designing an Internetwork

- How do I designate a distant host?
  - Addressing / naming
- How do I send information to a distant host?
  - Underlying service model
    - What gets sent?
    - How fast will it go?
    - What happens if it doesn’t get there?
  - Routing
- Challenges
  - Heterogeneity
  - Assembly from variety of different networks
  - Scalability
    - Ensure ability to grow to worldwide scale

Outline

- Methods for packet forwarding
  - Traditional IP addressing
  - CIDR

Getting to a Destination

- How do you get driving directions?
- Intersections $\rightarrow$ routers
- Roads $\rightarrow$ links/networks
- Roads change slowly

Forwarding Packets

- Table of virtual circuits
  - Connection routed through network to setup state
  - Packets forwarded using connection state
- Source routing
  - Packet carries path
- Table of global addresses (IP)
  - Routers keep next hop for destination
  - Packets carry destination address
Simplified Virtual Circuits

- Connection setup phase
  - Use other means to route setup request
  - Each router allocates flow ID on local link
- Each packet carries connection ID
  - Sent from source with 1st hop connection ID
- Router processing
  - Lookup flow ID – simple table lookup
  - Replace flow ID with outgoing flow ID
  - Forward to output port

Virtual Circuits

- Advantages
  - Efficient lookup (simple table lookup)
  - More flexible (different path for each flow)
  - Can reserve bandwidth at connection setup
  - Easier for hardware implementations
- Disadvantages
  - Still need to route connection setup request
  - More complex failure recovery – must recreate connection state
- Typical use → fast router implementations
  - ATM – combined with fix sized cells
  - MPLS – tag switching for IP networks

Source Routing

- List entire path in packet
  - Driving directions (north 3 hops, east, etc..)
- Router processing
  - Strip first step from packet
  - Examine next step in directions
  - Forward to next step

Example

Sender
R1
R2
R3
Receiver

Packet
conn 5 → 3
conn 5 → 4
conn 5 → 3

Sender
R1
R2
R3
Receiver
Source Routing Example

Receiver

Packet

Sender

Source Routing

• Advantages
  • Switches can be very simple and fast

• Disadvantages
  • Variable (unbounded) header size
  • Sources must know or discover topology (e.g., failures)

• Typical uses
  • Ad-hoc networks (DSR)
  • Machine room networks (Myrinet)

Global Addresses (IP)

• Each packet has destination address
• Each router has forwarding table of destination → next hop
  • At v and x: destination → east
  • At w and y: destination → south
  • At z: destination → north
• Distributed routing algorithm for calculating forwarding tables

Global Address Example
Global Addresses

- Advantages
  - Stateless – simple error recovery

- Disadvantages
  - Every switch knows about every destination
    - Potentially large tables
  - All packets to destination take same route
  - Need routing protocol to fill table

Router Table Size

- One entry for every host on the Internet
  - 300M entries, doubling every 18 months

- One entry for every LAN
  - Every host on LAN shares prefix
  - Still too many and growing quickly

- One entry for every organization
  - Every host in organization shares prefix
  - Requires careful address allocation

Comparison

<table>
<thead>
<tr>
<th></th>
<th>Source Routing</th>
<th>Global Addresses</th>
<th>Virtual Circuits</th>
</tr>
</thead>
<tbody>
<tr>
<td>Header Size</td>
<td>Worst</td>
<td>OK – Large address</td>
<td>Best</td>
</tr>
<tr>
<td>Router Table Size</td>
<td>None</td>
<td>Number of hosts (prefix)</td>
<td>Number of circuits</td>
</tr>
<tr>
<td>Forward Overhead</td>
<td>Best</td>
<td>Prefix matching</td>
<td>Pretty Good</td>
</tr>
<tr>
<td>Setup Overhead</td>
<td>None</td>
<td>None</td>
<td>Connection Setup</td>
</tr>
<tr>
<td>Error Recovery</td>
<td>Tell all hosts</td>
<td>Tell all routers</td>
<td>Tell all routers and tear down circuit and re-route</td>
</tr>
</tbody>
</table>

Outline

- Methods for packet forwarding

  - Traditional IP addressing

  - CIDR
Addressing in IP

- IP addresses are names of interfaces
  - E.g., 128.2.1.1
- Domain Name System (DNS) names are names of hosts
  - E.g., www.cmu.edu
- DNS binds host names to interfaces
- Routing binds interface names to paths

Addressing Considerations

- Hierarchical vs. flat
  - Pennsylvania / Pittsburgh / Oakland / CMU / Seshan vs.
    Srinivasan Seshan: 123-45-6789
- What information would routers need to route to Ethernet addresses?
  - Need hierarchical structure for designing scalable binding from
    interface name to route!
- What type of Hierarchy?
  - How many levels?
  - Same hierarchy depth for everyone?
  - Same segment size for similar partition?

IP Addresses

- Fixed length: 32 bits
- Initial classful structure (1981)
- Total IP address size: 4 billion
  - Class A: 128 networks, 16M hosts
  - Class B: 16K networks, 64K hosts
  - Class C: 2M networks, 256 hosts

<table>
<thead>
<tr>
<th>High Order Bits</th>
<th>Format</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>7 bits of net, 24 bits of host</td>
<td>A</td>
</tr>
<tr>
<td>10</td>
<td>14 bits of net, 16 bits of host</td>
<td>B</td>
</tr>
<tr>
<td>110</td>
<td>21 bits of net, 8 bits of host</td>
<td>C</td>
</tr>
</tbody>
</table>

IP Address Classes (Some are Obsolete)

- Class A:
  - 8 bits for network ID
  - 24 bits for host ID
- Class B:
  - 16 bits for network ID
  - 16 bits for host ID
- Class C:
  - 21 bits for network ID
  - 8 bits for host ID
- Class D:
  - Multicast Addresses
- Class E:
  - Reserved for experiments
Original IP Route Lookup

- Address would specify prefix for forwarding table
  - Simple lookup
- www.cmu.edu address 128.2.11.43
  - Class B address – class + network is 128.2
  - Lookup 128.2 in forwarding table
  - Prefix – part of address that really matters for routing
- Forwarding table contains
  - List of class+network entries
  - A few fixed prefix lengths (8/16/24)
- Large tables
  - 2 Million class C networks

Subnet Addressing

RFC917 (1984)

- Class A & B networks too big
  - Very few LANs have close to 64K hosts
  - For electrical/LAN limitations, performance or administrative reasons
- Need simple way to get multiple “networks”
  - Use bridging, multiple IP networks or split up single network address ranges (subnet)
- CMU case study in RFC
  - Chose not to adopt – concern that it would not be widely supported

Subnetting

- Add another layer to hierarchy
- Variable length subnet masks
  - Could subnet a class B into several chunks

Subnetting Example

- Assume an organization was assigned address 150.100
- Assume < 100 hosts per subnet
- How many host bits do we need?
  - Seven
- What is the network mask?
  - 1111111 1111111 1111111 1000000
  - 255.255.255.128
**Forwarding Example**

- Assume a packet arrives with address 150.100.12.176
- Step 1: AND address with class + subnet mask

```
150.100.12.128
H1
150.100.12.154
H2
150.100.10.1
To Internet
```

**Aside: Interaction with Link Layer**

- How does one find the Ethernet address of a IP host?
- **ARP**
  - Broadcast search for IP address
    - E.g., "who-has 128.2.184.45 tell 128.2.206.138" sent to Ethernet broadcast (all FF address)
  - Destination responds (only to requester using unicast) with appropriate 48-bit Ethernet address
    - E.g., "reply 128.2.184.45 is-at 0:d0:bc:f2:18:58" sent to 0:c0:4f:d:ed:c6

**IP Address Problem (1991)**

- Address space depletion
  - In danger of running out of classes A and B
  - Why?
    - Class C too small for most domains
    - Very few class A – very careful about giving them out
    - Class B – greatest problem
  - Class B sparsely populated
    - But people refuse to give it back
  - Large forwarding tables
    - 2 Million possible class C groups

**IP Address Utilization (‘97)**

[Link to CAIDA website for IPv4 space utilization data]
Outline

• Methods for packet forwarding

• Traditional IP addressing

• CIDR

Classless Inter-Domain Routing (CIDR) – RFC1338

• Allows arbitrary split between network & host part of address
  • Do not use classes to determine network ID
  • Use common part of address as network number
  • E.g., addresses 192.4.16 - 192.4.31 have the first 20 bits in common. Thus, we use these 20 bits as the network number → 192.4.16/20

• Enables more efficient usage of address space (and router tables) → How?
  • Use single entry for range in forwarding tables
  • Combined forwarding entries when possible

CIDR Example

• Network is allocated 8 class C chunks, 200.10.0.0 to 200.10.7.255
  • Allocation uses 3 bits of class C space
  • Remaining 20 bits are network number, written as 201.10.0.0/21
  • Replaces 8 class C routing entries with 1 combined entry
  • Routing protocols carry prefix with destination network address
  • Longest prefix match for forwarding

IP Addresses: How to Get One?

Network (network portion):

• Get allocated portion of ISP’s address space:

<table>
<thead>
<tr>
<th>ISP’s block</th>
<th>11001000 00010111 00010000 00000000</th>
<th>11001000 00010111 00010000 00000000</th>
<th>11001000 00010111 00010100 00000000</th>
<th>11001000 00010111 00011110 00000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organization 0</td>
<td>11001000 00010111 00010000 00000000</td>
<td>200.23.16.0/20</td>
<td>200.23.16.0/23</td>
<td>200.23.20.0/23</td>
</tr>
<tr>
<td>Organization 1</td>
<td>11001000 00010111 00010100 00000000</td>
<td>200.23.18.0/23</td>
<td>200.23.20.0/23</td>
<td>200.23.30.0/23</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Organization 7</td>
<td>11001000 00010111 00011110 00000000</td>
<td>200.23.30.0/23</td>
<td>200.23.30.0/23</td>
<td>200.23.30.0/23</td>
</tr>
</tbody>
</table>
IP Addresses: How to Get One?

- How does an ISP get block of addresses?
  - From Regional Internet Registries (RIRs)
    - ARIN (North America, Southern Africa), APNIC (Asia-Pacific), RIPE (Europe, Northern Africa), LACNIC (South America)

- How about a single host?
  - Hard-coded by system admin in a file
  - DHCP: Dynamic Host Configuration Protocol: dynamically get address: "plug-and-play"
    - Host broadcasts "DHCP discover" msg
    - DHCP server responds with "DHCP offer" msg
    - Host requests IP address: "DHCP request" msg
    - DHCP server sends address: "DHCP ack" msg

CIDR Illustration

Provider is given 201.10.0.0/21

- 201.10.0.0/22
- 201.10.4.0/24
- 201.10.5.0/24
- 201.10.6.0/23

CIDR Implications

- Longest prefix match!!

Important Concepts

- Hierarchical addressing critical for scalable system
  - Don't require everyone to know everyone else
  - Reduces amount of updating when something changes
- Non-uniform hierarchy useful for heterogeneous networks
  - Initial class-based addressing too coarse
  - CIDR helps
Routing Through Single Network

- Path Consists of Series of Hops
  - Source – Router
  - Router – Router (typically high-speed, point-to-point link)
  - Router – Destination
- Each Hop Uses Link-Layer Protocol
  - Determine hop destination
    - Based on destination
  - Send over local network
    - Put on header giving MAC address of intermediate router (or final destination)

How is IP Design Standardized?

- IETF
  - Voluntary organization
  - Meeting every 4 months
  - Working groups and email discussions
  - “We reject kings, presidents, and voting; we believe in rough consensus and running code” (Dave Clark 1992)
  - Need 2 independent, interoperable implementations for standard

Addressing Considerations

- Fixed length or variable length?
- Issues:
  - Flexibility
  - Processing costs
  - Header size
- Engineering choice: IP uses fixed length addresses
Virtual Circuits/Tag Switching

- Connection setup phase
  - Use other means to route setup request
  - Each router allocates flow ID on local link
  - Creates mapping of inbound flow ID/port to outbound flow ID/port
- Each packet carries connection ID
  - Sent from source with 1st hop connection ID
- Router processing
  - Lookup flow ID – simple table lookup
  - Replace flow ID with outgoing flow ID
  - Forward to output port

Virtual Circuits Examples

Virtual Circuits

- Advantages
  - More efficient lookup (simple table lookup)
  - More flexible (different path for each flow)
  - Can reserve bandwidth at connection setup
  - Easier for hardware implementations
- Disadvantages
  - Still need to route connection setup request
  - More complex failure recovery – must recreate connection state
- Typical uses
  - ATM – combined with fix sized cells
  - MPLS – tag switching for IP networks

Hierarchical Addressing Details

- Flat → would need router table entry for every single host… way too big
- Hierarchy → much like phone system…

- Hierarchy
  - Address broken into segments of increasing specificity
    - 412 (Pittsburgh area) 268 (Oakland exchange) 8734 (Seshan’s office)
    - Pennsylvania / Pittsburgh / Oakland / CMU / Seshan
  - Route to general region and then work toward specific destination
  - As people and organizations shift, only update affected routing tables
Hierarchical Addressing Details

- **Uniform Hierarchy**
  - Segment sizes same for everyone
  - 412 (Pittsburgh area) 268 (Oakland exchange) 8734 (Seshan’s office)
  - System is more homogeneous and easier to control
    - Requires more centralized planning
- **Nonuniform Hierarchy**
  - Number & sizes of segments vary according to destination
    - Pennsylvania / Pittsburgh / Oakland / CMU / Seshan
    - Delaware / Smallville / Bob Jones
  - System is more heterogenous & decentralized
    - Allows more local autonomy

Some Special IP Addresses

- 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

CIDR

- **Supernets**
  - Assign adjacent net addresses to same org
  - Classless routing (CIDR)
- How does this help routing table?
  - Combine forwarding table entries whenever all nodes with same prefix share same hop

Aggregation with CIDR

- Original Use: Aggregate Class C Addresses
- One organization assigned contiguous range of class C’s
  - e.g., Microsoft given all addresses 207.46.192.X – 207.46.255.X
- Specify as CIDR address 207.46.192.0/18
  - Represents $2^{18}$ = 64 class C networks
  - Use single entry in routing table
    - Just as if were single network address
14

9-22-05 Lecture 8: Addressing/Forwarding

Size of Complete Routing Table

- Source: www.cidr-report.org
- Shows that CIDR has kept # table entries in check
  - Currently require 124,894 entries for a complete table
  - Only required by backbone routers

Finding a Local Machine

- Routing Gets Packet to Correct Local Network
  - Based on IP address
  - Router sees that destination address is of local machine
- Still Need to Get Packet to Host
  - Using link-layer protocol
  - Need to know hardware address
- Same Issue for Any Local Communication
  - Find local machine, given its IP address

Address Resolution Protocol (ARP)

- Diagrammed for Ethernet (6-byte MAC addresses)
- Low-Level Protocol
  - Operates only within local network
  - Determines mapping from IP address to hardware (MAC) address
  - Mapping determined dynamically
    - No need to statically configure tables
    - Only requirement is that each host know its own IP address

ARP Request

- op: Operation
  - 1: request
  - 2: reply
- Sender
  - Host that wants to determine MAC address of another machine
- Target
  - Other machine
- Requestor
  - Fills in own IP and MAC address as “sender”
  - Why include its MAC address?
- Mapping
  - Fills desired host IP address in target IP address
- Sending
  - Send to MAC address ff:ff:ff:ff:ff:ff
    - Ethernet broadcast
ARP Reply

- **Op:** Operation
  - **2:** reply
- **Sender**
  - Host with desired IP address
- **Target**
  - Original requestor

- Responder becomes “sender”
  - Fill in own IP and MAC address
  - Set requestor as target
  - Send to requestor’s MAC address

ARP Example

- **Exchange Captured with windump**
  - Windows version of tcpdump
- **Requestor:**
  - `blackhole-ad.scs.cs.cmu.edu` (128.2.194.66)
  - MAC address: 0:2:b3:8a:35:bf
- **Desired host:**
  - `bryant-tp2.vlsi.cs.cmu.edu` (128.2.222.198)
  - MAC address: 0:3:47:b8:e5:f3

ARP Cache Example

- **Show using command “arp -a”**

```
Interface: 128.2.222.198 on Interface 0x1000003
Internet Address Physical Address Type
128.2.20.218 00-b0-8e-83-df-50 dynamic
128.2.102.129 00-b0-8e-83-df-50 dynamic
128.2.194.66 00-02-b3:8a-35-bf dynamic
128.2.198.34 00-06-5b-f3-5f-42 dynamic
128.2.203.3 00-90-27-3c-41-11 dynamic
128.2.203.61 08-00-20-a6-ba-2b dynamic
128.2.205.192 00-60-08-1e-9b-fd dynamic
128.2.206.125 00-b0-b7-c5-b3-f3 dynamic
128.2.206.139 00-a0-c9-98-2c-46 dynamic
128.2.222.180 08-00-20-a6-ba-c3 dynamic
128.2.242.182 08-00-20-a7-19-73 dynamic
128.2.254.36 00-b0-8e-83-df-50 dynamic
```
ARP Cache Surprise

• How come 3 machines have the same MAC address?

<table>
<thead>
<tr>
<th>Internet Address</th>
<th>Physical Address</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.2.20.218</td>
<td>00-b0-8a-83-df-50</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.102.129</td>
<td>00-b0-8a-83-df-50</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.194.66</td>
<td>00-02-b3-8a-35-bf</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.198.34</td>
<td>00-06-5b-c3-57-42</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.203.3</td>
<td>00-90-27-3c-41-11</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.203.61</td>
<td>08-00-20-a6-ba-2b</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.205.192</td>
<td>00-60-08-1e-9b-fd</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.206.125</td>
<td>00-db-fb-8b-f3</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.206.139</td>
<td>00-ac-c9-98-2c-46</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.222.180</td>
<td>08-00-20-a6-ba-c3</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.242.182</td>
<td>08-00-20-a7-9f-73</td>
<td>dynamic</td>
</tr>
<tr>
<td>128.2.254.36</td>
<td>00-b0-8a-83-df-50</td>
<td>dynamic</td>
</tr>
</tbody>
</table>

CMU’s Internal Network Structure

• CMU Uses Routing Internally
• Maintains forwarding tables using OSPF
• Most CMU hosts cannot be reached at link layer

Proxy ARP

• Provides Link-Layer Connectivity Using IP Routing
  • Local router (gigrouter) sees ARP request
  • Uses IP addressing to locate host
  • Becomes “Proxy” for remote host
  • Using own MAC address
  • Requestor thinks that it is communicating directly with remote host

Monitoring Packet Traffic

• Experiment
  • Ran windump for 15 minutes connected to CMU network
  • No applications running
  • But many background processes use network
  • Lots of ARP traffic (71% of total)
  • Average 37 ARP requests / second (why all from CS hosts?)
  • Only see responses from own machine (why?)
Monitoring Packet Traffic

- Other Traffic
  - Mostly UDP
    - Encode low-level protocols such as bootp
    - Nothing very exciting (why?)
  
- Answers for UDP and ARP
  - On a switched network you only see broadcast traffic or traffic sent to/from you
  - TCP is never sent broadcast

![Pie chart showing the distribution of IP messages](chart.png)