Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
- Packet forwarding

So far you know how to build a Local Area Network

How do we get them to talk to each other?

Logical Structure of an Internet

- Interconnection of separately managed networks using routers
  - Individual networks can use different (layer 1-2) technologies
- Packet travels from source to destination by hopping through networks
  - "Network" layer responsibility
- How do routers connect heterogeneous network technologies?
Solution: Internet Protocol (IP)

- Inter-network connectivity provided by the Internet protocol
- Hosts use Internet Protocol to send packets destined across networks.
- IP creates abstraction layer that hides underlying technology from network application software
  - Allows range of current & future technologies
  - WiFi, traditional and switched Ethernet, personal area networks, ...

The Packet as an Envelope

| Local Address Header (Ethernet) | To: Destination Host
| From: Sender Host |

| IP Header | To: 123.45.67.89 (Destination Host)
| From: 169.229.49.157 (Sender Host) |

HTTP Packet Payload:
GET nyan.cat….

Traveling through the Internet

- Source adds all headers (HTTP, transport -> PHY)
- Each router:
  - Removes datalink layer
  - Adds IP header to make forwarding decision
  - Adds data link layer header for next network
  - Destination removes all headers (PHY -> HTTP)
Protocol Demultiplexing

- What layers do not need a protocol field?

What are the Goals?

- LANs: “Connect hosts” → switching:
  - “Wire” abstraction: behaves like Ethernet – helps manageability
  - Only has to scale up a “LAN size”
  - Availability
- Internet: “Connect networks” → routing:
  - Scalability
  - Manageability of individual networks – contributes to scalability
  - Availability
  - Affects addressing, protocols, routing

Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
- Packet forwarding

Addressing and Forwarding

- Flat address space with smart routers
  - Packets carry destination
  - Routers know location of every host
- Flat address space with dumb routers
  - Packet carries a path
  - Hierarchical Routing Space
    - What we actually do in IP
    - (Table of virtual circuits ids)
    - More on this later, but not today
Flat Address Forwarding

- Bridge/switch has a table that shows for each MAC Address which port to use for forwarding.
- For every packet, the bridge “looks up” the entry for the packets destination MAC address and forwards the packet on that port.
- Other packets are broadcast – why?
- Timer is used to flush old entries.

<table>
<thead>
<tr>
<th>MAC Address</th>
<th>Port</th>
<th>Age</th>
</tr>
</thead>
<tbody>
<tr>
<td>8711C98900AA</td>
<td>2</td>
<td>15</td>
</tr>
<tr>
<td>99A323C90842</td>
<td>2</td>
<td>01</td>
</tr>
<tr>
<td>301B2369011C</td>
<td>2</td>
<td>16</td>
</tr>
<tr>
<td>695519001190</td>
<td>3</td>
<td>36</td>
</tr>
</tbody>
</table>

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 and forward.
- Defined for IPv4 but rarely used.
- End points need to know a lot about network.
- Economic and security concerns.
- Variable header size.

Hierarchical Addressing

- Flat addresses – one address for every host.
  - Peter Steenkiste: 123-45-6789
  - Does not scale – router table size explodes.
  - 630M (1/09) entries, doubling every 2.5 years.
  - Why does it work for Ethernet?
- Hierarchical – add structure.
  - Pennsylvania / Pittsburgh / Oakland / CMU / Gates / 9th fl / Steenkiste.
  - Common “trick” to simplify forwarding, reduce forwarding table.
- What type of Hierarchy do we need for the Internet?
  - How many levels?
  - Same hierarchy depth for everyone?
  - Who controls the hierarchy?

Why is this not a good solution for the Internet?

- Each router tracking $2^{32}$ addresses = scalability nightmare.
- B bridge/switch has a table that shows for each MAC Address which port to use for forwarding.
- For every packet, the bridge “looks up” the entry for the packets destination MAC address and forwards the packet on that port.
- Other packets are broadcast – why?
- Timer is used to flush old entries.
IP Addresses (IPv4)

- Unique 32-bit number associated with a host
  - Represented with the "dotted quad" notation
  - e.g., 12.34.158.5

Hierarchical in IP Addressing

- 32 bits are partitioned into a prefix and suffix components
- Prefix is the network component: CMU
- Suffix is host component: Prof. Sherry's laptop at CMU

00011100 00100010 10011110 00000101

Network (23 bits)  Host (9 bits)

History of Internet Addressing

- Always dotted-quad notation
- Always network/host address split
- But nature of that split has changed over time

Original Internet Addresses

- First eight bits: network component
- Last 24 bits: host component

*Assumed 256 networks were more than enough!*
IP Address Structure, ca 1981

Routers know how to get to network ID, but not individual hosts.

<table>
<thead>
<tr>
<th>Network ID</th>
<th>Host ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class A</td>
<td>1</td>
</tr>
<tr>
<td>Class B</td>
<td>10</td>
</tr>
<tr>
<td>Class C</td>
<td>100</td>
</tr>
<tr>
<td>Class D</td>
<td>110</td>
</tr>
<tr>
<td>Class E</td>
<td>111</td>
</tr>
</tbody>
</table>

Class A: 32 bits, 255.0.0.0
Class B: 24 bits, 255.255.0.0
Class C: 16 bits, 255.255.255.0

Multicast Addresses:
Class D: 24 bits, 255.255.255.0

Reserved for experiments:
Class E: 32 bits, 255.255.255.255

IP Route Lookup, ca 1981

- Address specifies prefix for forwarding table
- Extract address type and network ID
- Forwarding table structure reflects address structure
  - Logically, a separate forwarding table for each address class
  - For unicast address (classes A-C) entries contain
    - The prefix for a destination network (length 8/16/24)
    - Information on how to forward the packet, e.g., exit port, ...
- www.cmu.edu address 128.2.11.43
- Class B address – class + network is 128.2
- Lookup 128.2 in forwarding table for class B
- Tables are still large!
  - 2 Million class C networks

Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
  - Packet forwarding

Subnetting

- Add another layer to hierarchy
- Variable length subnet masks
  - Could subnet a network internally into several chunks
- Subnetting is done internally in the organization
  - It is not visible outside – important for management
Subnet Addressing

RFC917 (1984)

- Some “LANs” are very big
  - Large companies, universities, …
  - Internet became popular quickly
- Cannot manage this as a single LAN
  - Hard to manage, becomes inefficient
- Need simple way to partition large networks
  - Partition into multiple IP networks that share the same prefix – called a "subnet", part of a network
- CMU case study in RFC
  - Chose not to adopt – concern that it would not be widely supported

IP Address Problem (1991)

- Address space depletion
  - Suppose you need $2^{16} + 1$ addresses?
  - Class A too big for all but a few domains
  - Class C too small for many domains but they don’t need a class B address
  - Class B address pool allocated at high rate
  - Many allocated address block are sparsely used
- Developed a strategy based on a three solutions
  - Switch to a “classless” addressing model – this lecture
  - Network address translation (NAT) – later in the course
  - Definition of IPv6 with larger IP addresses – next lecture

Today’s Addressing: CIDR

- CIDR = Classless Interdomain Routing
- Idea: Flexible division between network and host addresses
  - Not limited to three sizes 8/16/24
  - Prefix can be any size
- Motivation: offer a better tradeoff between size of the forwarding table and efficient use of the IP address space

CIDR (example)

- Suppose a network has fifty computers
  - allocate 6 bits for host addresses (since $2^5 < 50 < 2^6$)
  - remaining 32 - 6 = 26 bits as network prefix
- Flexible boundary means the boundary must be explicitly specified with the network address!
  - Informally, "slash 26" → 128.23.9/26
  - Formally, prefix represented with a 32-bit mask: 255.255.255.192 where all network prefix bits set to “1” and host suffix bits to “0”
Classful vs. Classless addresses

- Example: an organization needs 500 addresses.
  - A single class C address not enough (254 hosts).
  - Instead a class B address is allocated (~65K hosts)
  - That's overkill, a huge waste!
- CIDR allows an arbitrary prefix-suffix boundary
  - Hence, organization allocated a single /23 address (equivalent of 2 class C's)
- Maximum waste: 50%

Hence, IP Addressing: Hierarchical

- CIDR allows more efficient use of the IP address space
  - Helps (at least for a while) with the high demand for IP addresses
  - But how does this help with the growth of forwarding tables?
  - Number of destination networks is growing as well!
- Solution has two complementary parts:
  - Allocation of IP addresses is done hierarchically
  - Routers will combine forwarding entries for destinations "in the same general direction"

Allocation Done Hierarchically

- Historically assignment of prefixes was “first come first serve”
- With CIDR: Internet Corporation for Assigned Names and Numbers (ICANN) gives large blocks to…
- Regional Internet Registries, such as the American Registry for Internet Names (ARIN), which give blocks to…
- Large institutions (ISPs), which give addresses to …
- Individuals and smaller institutions
- FAKE Example:
  
  ICANN ➔ ARIN ➔ AT&T ➔ UCB ➔ EECS

CIDR: Addresses allocated in contiguous prefix chunks

Recursively break down chunks as get closer to host

\[
\begin{align*}
12.0.0.0/8 & : 12.0.0.0/15 \\
12.2.0.0/16 & : 12.2.0.0/15 \\
12.3.0.0/16 & : 12.3.0.0/22 \\
12.3.4.0/24 & : 2.3.254.0/23 \\
12.253.0.0/16 & : 12.253.0.0/19 \\
& 12.253.32.0/19 \\
& 12.253.64.0/19 \\
& 12.253.64.109/30 \\
12.253.96.0/18 & : 12.253.96.0/18 \\
& 12.253.128.0/17 \\
& 12.253.128.0/17 \\
\end{align*}
\]
IP Addressing → Scalable Forwarding?

- How many forwarding entries does France Telecom need for LBL/UCB destinations?
- How about if all a.0.0.0/8 addresses are served by AT&T - it "own" a.0.0.0/8, right?
- In practice, it is complicated ...

How LPM Works

- Routing protocols aggregate forwarding entries to reduce table size
  - E.g., 3 forwarding entries A/B/C 01010011.xy/10 can be combined into 01010011/8 if they forward through the same port
  - A fourth entry D that uses a different egress port has its own entry
  - Works correctly because of longest prefix match (LPM)
    - Packets to A/B/C will match only the 01010011/8
    - Packets to D will match entries but will prefer the short “/10” entry
  - Legacy prefixes (e.g., 128.2) also often have their own entry

How LPM Works

CIDR Implication: Longest Prefix Match

- How to deal with multi-homing, legacy addresses, ...

Filling in Some Router Details

- How do routing protocols learn the prefix size?
  - Routing advertisements include the prefix size; for destination addresses in packets, the prefix size is not relevant
  - For stub networks (subnetting): routers are configured by admin
  - But a router now needs ~30 forwarding tables?
  - No – forwarding uses a single tree data structure (called a trie)
    - Very efficient algorithms exist for look up both in HW and SW
  - How do routers know the prefix size for destination addresses?
    - They do not need them because of how LPM look up works
Outline

- IP design goals
- Traditional IP addressing
  - Addressing approaches
  - Class-based addressing
  - Subnetting
  - CIDR
- Packet forwarding

Host Routing Table Example

<table>
<thead>
<tr>
<th>Destination</th>
<th>Gateway</th>
<th>Genmask</th>
<th>Iface</th>
</tr>
</thead>
<tbody>
<tr>
<td>128.2.209.100</td>
<td>0.0.0.0</td>
<td>255.255.255.0</td>
<td>eth0</td>
</tr>
<tr>
<td>128.2.0.0</td>
<td>0.0.0.0</td>
<td>255.255.0.0</td>
<td>eth0</td>
</tr>
<tr>
<td>127.0.0.0</td>
<td>0.0.0.0</td>
<td>255.0.0.0</td>
<td>lo</td>
</tr>
<tr>
<td>0.0.0.0</td>
<td>128.2.254.36</td>
<td>0.0.0.0</td>
<td>eth0</td>
</tr>
</tbody>
</table>

- From “netstat –rn”
- Host 128.2.209.100 when plugged into CS ethernet
- Dest 128.2.209.100 \(\rightarrow\) routing to same machine
- Dest 128.2.0.0 \(\rightarrow\) other hosts on same ethernet
- Dest 127.0.0.0 \(\rightarrow\) special loopback address
- Dest 0.0.0.0 \(\rightarrow\) default route to rest of Internet
- Main CS router: gigrouter.net.cs.cmu.edu (128.2.254.36)

Routing to the Network

- Five subnets (yellow)
  1. 10.1.8/24
  2. 10.1.2/23
  3. 10.1.0/24
  4. 10.1.0/24
  5. 10.1.1.3/31
- Packet to 10.1.1.3 arrives from Internet
- Path is R2 \(\rightarrow\) R1 \(\rightarrow\) H1 \(\rightarrow\) H2
- H1 serves as a router for the 10.1.1.2/31 network (2 IP addresses)

Routing Within the Subnet

- Packet to 10.1.1.3
- Matches 10.1.0.0/23

Routing table at R2

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Egress Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>-</td>
<td>lo0</td>
</tr>
<tr>
<td>Default or 0/0</td>
<td>provider</td>
<td>10.1.16.1</td>
</tr>
<tr>
<td>10.1.0.0/24</td>
<td>-</td>
<td>10.1.1.2</td>
</tr>
<tr>
<td>10.1.2.0/23</td>
<td>-</td>
<td>10.1.2.1</td>
</tr>
<tr>
<td>10.1.0.23</td>
<td>10.1.2.2</td>
<td>10.1.2.1</td>
</tr>
</tbody>
</table>

Each router port has an IP address:
Routing Within the Subnet

• Packet to 10.1.1.3  
• Matches 10.1.1.2/31  
• Longest prefix match

Routing table at R1

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Egress Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>-</td>
<td>lo0</td>
</tr>
<tr>
<td>Default or 0/0</td>
<td>10.1.2.1</td>
<td>10.1.2.2</td>
</tr>
<tr>
<td>10.1.1.0/24</td>
<td>-</td>
<td>10.1.0.1</td>
</tr>
<tr>
<td>10.1.5.0/24</td>
<td>-</td>
<td>10.1.0.1</td>
</tr>
<tr>
<td>10.1.1.2/31</td>
<td>10.1.1.4</td>
<td>10.1.1.1</td>
</tr>
</tbody>
</table>

Routing table at H1

<table>
<thead>
<tr>
<th>Destination</th>
<th>Next Hop</th>
<th>Egress Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>127.0.0.1</td>
<td>-</td>
<td>lo0</td>
</tr>
<tr>
<td>Default or 0/0</td>
<td>10.1.1.1</td>
<td>10.1.1.4</td>
</tr>
<tr>
<td>10.1.1.2/31</td>
<td>-</td>
<td>10.1.1.1</td>
</tr>
</tbody>
</table>

• Packet to 10.1.1.3  
• Direct route

• Longest prefix match

Important Concepts

• Hierarchical addressing critical for scalable system
• Don’t require everyone to know everyone else
• Reduces number of updates when something changes
• Classless inter-domain routing supports more efficient use of address space
• Adds complexity to routing, forwarding, …
• But it is Scalable!

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?
  • Assigned by sys admin (static or dynamic)
  • 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
IP Address Availability Remains a Major Challenge

- Some are in big trouble!
- APNIC: Asia
- AFRINIC: Africa
- ARIN: North America
- LACNIC: Latin America
- RIPE NCC: Europe, Middle East, parts of central Asia