Recap

• We want the network layer to:
  – Scale to large networks
    • Using addresses with hierarchy
  – Support diverse technologies
    • Internetworking with IP
  – Use link bandwidth well
    • Lowest-cost routing

This lecture
More later
Next time
IP Addresses

• IPv4 uses 32-bit addresses
  – Later we’ll see IPv6, which uses 128-bit addresses

• Written in “dotted quad” notation
  – Four 8-bit numbers separated by dots

\[
\begin{align*}
\text{aaaaaaaabbbbbbbbcccccccccddddd} & \leftrightarrow \text{A.B.C.D} \\
00010010000111110000000000000001 & \leftrightarrow
\end{align*}
\]
IP Prefixes

- Addresses are allocated in blocks called **prefixes**
  - Addresses in an L-bit prefix have the same top L bits
  - There are $2^{32-L}$ addresses aligned on $2^{32-L}$ boundary
IP Prefixes (2)

- Written in “IP address/length” notation
  - Address is lowest address in the prefix, length is prefix bits
  - E.g., 128.13.0.0/16 is 128.13.0.0 to 128.13.255.255
  - So a /24 (“slash 24”) is 256 addresses, and a /32 is one address

\[
\begin{array}{c}
00010010|00011111|00000000|xxxxxxxxx \\
\hline
\hline
\hline
| | | | \leftrightarrow 128.13.0.0/16
\end{array}
\]
Classful IP Addressing

- Originally, IP addresses came in fixed size blocks with the class/size encoded in the high-order bits
  - They still do, but the classes are now ignored

![Diagram showing classful IP addressing with network and host portions and their respective ranges for Class A, Class B, and Class C addresses.]
IP Forwarding

All addresses on one network belong to the same prefix
• Node uses a table that lists the next hop for prefixes

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.24.0.0/19</td>
<td>D</td>
</tr>
<tr>
<td>192.24.12.0/22</td>
<td>B</td>
</tr>
</tbody>
</table>

![Network Diagram]

A: 192.24.0.0/19
B: 192.24.12.0/22
C: 192.24.0.0/19
D: 192.24.12.0/22
Longest Matching Prefix

- Prefixes in the table might overlap!
  - Combines hierarchy with flexibility

- **Longest matching prefix** forwarding rule:
  - For each packet, find the longest prefix that contains the destination address, i.e., the most specific entry
  - Forward the packet to the next hop router for that prefix
Longest Matching Prefix (2)

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>192.24.0.0/19</td>
<td>D</td>
</tr>
<tr>
<td>192.24.12.0/22</td>
<td>B</td>
</tr>
</tbody>
</table>

192.24.6.0 ➔
192.24.14.32 ➔
192.24.54.0 ➔
Host/Router Distinction

- In the Internet:
  - Routers do the routing, know which way to all destinations
  - Hosts send remote traffic (out of prefix) to nearest router

Not for my network? Send it to the router

It’s my job to know which way to go ...
Host Forwarding Table

- Give using longest matching prefix
  - 0.0.0.0/0 is a default route that catches all IP addresses

<table>
<thead>
<tr>
<th>Prefix</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>My network prefix</td>
<td>Send to that IP</td>
</tr>
<tr>
<td>0.0.0.0/0</td>
<td>Send to my router</td>
</tr>
</tbody>
</table>
Flexibility of Longest Matching Prefix

• Can provide default behavior, with less specifics
  – To send traffic going outside an organization to a border router

• Can special case behavior, with more specifics
  – For performance, economics, security, …
Performance of Longest Matching Prefix

• Uses hierarchy for a compact table
  – Relies on use of large prefixes

• Lookup more complex than table
  – Used to be a concern for fast routers
  – Not an issue in practice these days
Topic

• Filling in the gaps we need to make for IP forwarding work in practice
  – Getting IP addresses (DHCP) »
  – Mapping IP to link addresses (ARP) »

What’s my IP?

What link layer address do I use?
Getting IP Addresses

• Problem:
  – A node wakes up for the first time ...
  – What is its IP address? What’s the IP address of its router? Etc.
  – At least Ethernet address is on NIC

Hey, where am I?
Getting IP Addresses (2)

1. Manual configuration (old days)
   – Can’t be factory set, depends on use

2. A protocol for automatically configuring addresses (DHCP) »
   – Shifts burden from users to IT folk
DHCP

- DHCP (Dynamic Host Configuration Protocol), from 1993, widely used
- It leases IP address to nodes
- Provides other parameters too
  - Network prefix
  - Address of local router
  - DNS server, time server, etc.
DHCP Protocol Stack

• DHCP is a client-server application
  – Uses UDP ports 67, 68

```
+---+     +---+     +---+     +---+
| DHCP   | UDP   | IP    | Ethernet |
+---+     +---+     +---+     +---+
DHCP Addressing

• **Bootstrap issue:**
  – How does node send a message to DHCP server before it is configured?

• **Answer:**
  – Node sends broadcast messages that delivered to all nodes on the network
  – Broadcast address is all 1s
  – IP (32 bit): 255.255.255.255
  – Ethernet (48 bit): ff:ff:ff:ff:ff:ff
DHCP Messages

Client

Server

One link
DHCP Messages (2)
DHCP Messages (3)

• To renew an existing lease, an abbreviated sequence is used:
  – REQUEST, followed by ACK

• Protocol also supports replicated servers for reliability
Sending an IP Packet

**Problem:**

- A node needs Link layer addresses to send a frame over the local link
- How does it get the destination link address from a destination IP address?

Uh oh ...

My IP is 1.2.3.4
ARP (Address Resolution Protocol)

- Node uses to map a local IP address to its Link layer addresses
ARP Protocol Stack

- ARP sits right on top of link layer
  - No servers, just asks node with target IP to identify itself
  - Uses broadcast to reach all nodes

<table>
<thead>
<tr>
<th>ARP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ethernet</td>
</tr>
</tbody>
</table>
ARP Messages

Node

One link

Target
ARP Messages (2)

Who has IP 1.2.3.4?

I do at 1:2:3:4:5:6
Discovery Protocols

- Help nodes find each other
  - There are more of them!
    - E.g., zeroconf, Bonjour

- Often involve broadcast
  - Since nodes aren’t introduced
  - Very handy glue
END