Where we are in the Course

- Starting the Network Layer!
  - Builds on the link layer. **Routers** send packets over multiple networks

<table>
<thead>
<tr>
<th>Application</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transport</td>
</tr>
<tr>
<td>Network</td>
</tr>
<tr>
<td>Link</td>
</tr>
<tr>
<td>Physical</td>
</tr>
</tbody>
</table>
Why do we need a Network layer?

- We can already build networks with links and switches and send frames between hosts ...
Shortcomings of Switches

1. Don’t scale to large networks
   - Blow up of routing table, broadcast

Table for all destinations in the world!

Broadcast new destinations to the whole world!
Shortcomings of Switches (2)

2. Don’t work across more than one link layer technology
   - Hosts on Ethernet + 3G + 802.11 …

Can we play too?  Go away!
Shortcomings of Switches (3)

3. Don’t give much traffic control
   – Want to plan routes / bandwidth

That was lame.
Network Layer Approach

• Scaling:
  – Hierarchy, in the form of prefixes

• Heterogeneity:
  – IP for internetworking

• Bandwidth Control:
  – Lowest-cost routing
  – Later QOS (Quality of Service)
Routing vs. Forwarding

- Routing is the process of deciding in which direction to send traffic
  - Network wide (global) and expensive
Routing vs. Forwarding (2)

- **Forwarding** is the process of sending a packet on its way
  - Node process (local) and fast
Topic

• How do we connect different networks together?
  – This is called **internetworking**
  – We’ll look at how IP does it
How Networks May Differ

• Basically, in a lot of ways:
  – Service model (datagrams, VCs)
  – Addressing (what kind)
  – QOS (priorities, no priorities)
  – Packet sizes
  – Security (whether encrypted)

• Internetworking hides the differences with a common protocol. (Uh oh.)
Connecting Datagram and VC networks

- An example to show that it’s not so easy
  - Need to map destination address to a VC and vice-versa
  - A bit of a “road bump”, e.g., might have to set up a VC
Internetworking – Cerf and Kahn

• Pioneered by Cerf and Kahn, the “fathers of the Internet”
  – In 1974, later led to TCP/IP

• Tackled the problems of interconnecting networks
  – Instead of mandating a single network technology
Internet Reference Model

- IP is the “narrow waist” of the Internet
  - Supports many different links below and apps above

<table>
<thead>
<tr>
<th>1. Link</th>
<th>Ethernet</th>
<th>3G</th>
<th>802.11</th>
</tr>
</thead>
<tbody>
<tr>
<td>2. Internet</td>
<td>IP</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3. Transport</td>
<td>TCP</td>
<td>UDP</td>
<td></td>
</tr>
<tr>
<td>4. Application</td>
<td>SMTP</td>
<td>HTTP</td>
<td>RTP</td>
</tr>
</tbody>
</table>

CSE 461 University of Washington
IP as a Lowest Common Denominator

• Suppose only some networks support QOS or security etc.
  – Difficult for internetwork to support

• Pushes IP to be a “lowest common denominator” protocol
  – Asks little of lower-layer networks
  – Gives little as a higher layer service
IPv4 (Internet Protocol)

- Various fields to meet straightforward needs
  - Version, Header (IHL) and Total length, Protocol, and Header Checksum
IPv4 (2)

- Network layer of the Internet, uses datagrams
  - Provides a layer of addressing above link addresses (next)
IPv4 (3)

- Some fields to handle packet size differences (later)
  - Identification, Fragment offset, Fragment control bits
IPv4 (4)

• Other fields to meet other needs (later, later)
  – Differentiated Services, Time to live (TTL)

Later, with QOS

Later, with ICMP

Payload (e.g., TCP segment)
Topic

- How do routers **forward** packets?
  - We’ll look at how IP does it
  - (We’ll cover routing later)
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

```
0001 0010 0001 1111 0000 0000 0000 0001
↔
aaaa aaaa abbbbb bbbb ccccc cccc cccc dddd dddd dddd dddd
```

↔ A.B.C.D
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

```
00010010|00011111|00000000|xxxxxxxxxx ↔
```

```
128.13.0.0/16 ↔
```

CSE 461 University of Washington
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 the breakdown of IP addresses into network and host portions, with Class A, B, and C allocations.]
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>

Diagram:

A -- B -- C -- D
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 → 

192.24.63.255

More specific

192.24.12.0/22

IP address

192.24.0.0

192.24.15.255/22

192.24.12.0

192.24.6.0

Prefix

Next Hop
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

<table>
<thead>
<tr>
<th>Ethernet</th>
<th>IP</th>
<th>UDP</th>
<th>DHCP</th>
</tr>
</thead>
</table>
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

One link

Server
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?
ARP (Address Resolution Protocol)

- Node uses to map a local IP address to its Link layer addresses.

```
| Source Ethernet | Dest. Ethernet | Source IP | Dest. IP | Payload ...
```

- From NIC
- From DHCP
- From ARP
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

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>ARP</td>
<td></td>
</tr>
<tr>
<td>Ethernet</td>
<td></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
Topic

• How do we connect networks with different maximum packet sizes?
  – Need to split up packets, or discover the largest size to use
Packet Size Problem

• Different networks have different maximum packet sizes
  – Or MTU (Maximum Transmission Unit)
  – E.g., Ethernet 1.5K, WiFi 2.3K

• Prefer large packets for efficiency
  – But what size is too large?
  – Difficult because node does not know complete network path
Packet Size Solutions

• Fragmentation (now)
  – Split up large packets in the network if they are too big to send
  – Classic method, dated

• Discovery (next)
  – Find the largest packet that fits on the network path and use it
  – IP uses today instead of fragmentation
IPv4 Fragmentation

- Routers fragment packets that are too large to forward
- Receiving host reassembles to reduce load on routers
IPv4 Fragmentation Fields

- Header fields used to handle packet size differences
  - Identification, Fragment offset, MF/DF control bits
IPv4 Fragmentation Procedure

• Routers split a packet that is too large:
  – Typically break into large pieces
  – Copy IP header to pieces
  – Adjust length on pieces
  – Set offset to indicate position
  – Set MF (More Fragments) on all pieces except last

• Receiving hosts reassembles the pieces:
  – Identification field links pieces together, MF tells receiver when it has all pieces
IPv4 Fragmentation (2)

Before
MTU = 2300

ID = 0x12ef
Data Len = 2300
Offset = 0
MF = 0

After
MTU = 1500

ID =
Data Len =
Offset =
MF =

(Ignore length of headers)
IPv4 Fragmentation (3)

**Before**
- MTU = 2300
- ID = 0x12ef
- Data Len = 2300
- Offset = 0
- MF = 0

**After**
- MTU = 1500
- ID = 0x12ef
- Data Len = 1500
- Offset = 0
- MF = 1
- ID = 0x12ef
- Data Len = 800
- Offset = 1500
- MF = 0
IPv4 Fragmentation (4)

• It works!
  – Allows repeated fragmentation

• But fragmentation is undesirable
  – More work for routers, hosts
  – Tends to magnify loss rate
  – Security vulnerabilities too
Path MTU Discovery

• Discover the MTU that will fit
  – So we can avoid fragmentation
  – The method in use today

• Host tests path with large packet
  – Routers provide feedback if too large; they tell host what size would have fit
Path MTU Discovery (2)
Path MTU Discovery (3)
Path MTU Discovery (4)

- Process may seem involved
  - But usually quick to find right size
- Path MTU depends on the path and so can change over time
  - Search is ongoing
- Implemented with ICMP (next)
  - Set DF (Don’t Fragment) bit in IP header to get feedback messages
Topic

• What happens when something goes wrong during forwarding?
  – Need to be able to find the problem

What happened?

XXXXXXX

Yikes!
Internet Control Message Protocol

- ICMP is a companion protocol to IP
  - They are implemented together
  - Sits on top of IP (IP Protocol=1)

- Provides error report and testing
  - Error is at router while forwarding
  - Also testing that hosts can use
ICMP Errors

• When router encounters an error while forwarding:
  – It sends an ICMP error report back to the IP source address
  – It discards the problematic packet; host needs to rectify
ICMP Message Format

• Each ICMP message has a Type, Code, and Checksum
• Often carry the start of the offending packet as payload
• Each message is carried in an IP packet
ICMP Message Format (2)

- Each ICMP message has a Type, Code, and Checksum
- Often carry the start of the offending packet as payload
- Each message is carried in an IP packet

![Diagram of ICMP message format]

- **Src**=router, **Dst**=A
  - Protocol = 1
  - **Type**=X, **Code**=Y
- **Src**=A, **Dst**=B
  - XXXXXXXXXXXXXXXX

Portion of offending packet, starting with its IP header
Example ICMP Messages

<table>
<thead>
<tr>
<th>Name</th>
<th>Type / Code</th>
<th>Usage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dest. Unreachable (Net or Host)</td>
<td>3 / 0 or 1</td>
<td>Lack of connectivity</td>
</tr>
<tr>
<td>Dest. Unreachable (Fragment)</td>
<td>3 / 4</td>
<td>Path MTU Discovery</td>
</tr>
<tr>
<td>Time Exceeded (Transit)</td>
<td>11 / 0</td>
<td>Traceroute</td>
</tr>
<tr>
<td>Echo Request or Reply</td>
<td>8 or 0 / 0</td>
<td>Ping</td>
</tr>
</tbody>
</table>

Testing, not a forwarding error: Host sends Echo Request, and destination responds with an Echo Reply
Traceroute

- IP header contains TTL (Time to live) field
  - Decremented every router hop, with ICMP error if it hits zero
  - Protects against forwarding loops
Traceroute (2)

- Traceroute repurposes TTL and ICMP functionality
  - Sends probe packets increasing TTL starting from 1
  - ICMP errors identify routers on the path