Last Time

• **Focus:**
  – What to do when one shared LAN isn’t big enough?

• **Interconnecting LANs**
  – Bridges and LAN switches
  – But there are limits ...
This Time: Internetworks

• Set of interconnected networks, e.g., the Internet
  – Scale and heterogeneity
The Protocol Stack

- **Thinking about roles:**
  - **Transport:** Process to Process
    - Example: TCP
      Reliable bytestream
  - **Network:** Host to Global Host
    - Example: IP
      Unreliable datagram
  - **Data Link/Physical:** Host to Local Host
    - Example: Ethernet
      Pretty reliable frame delivery
As a picture

- **IP** is the network layer protocol used in the Internet
- Routers are network level gateways
- **Packet** is the term for network layer protocol data units (PDUs)
Layers and Addressing

• Link layer address required to deliver along next hop
  – Example: 00:50:56:c0:00:01

• IP address required to deliver host-host
  – Example: 128.208.1.137

• Link layer addresses can be assigned (more or less) arbitrarily
  – Why?

• IP address assignment is more constrained
  – Why?
Packet formats: encapsulation

- View of a packet on the (Ethernet) wires

  Ethernet Header | IP Header | Higher layer headers and Payload

- (In a pure world) Routers work with IP header, not higher
  - Higher would be a “layer violation”

- Routers strip and add link layer headers
Network Layer Goals

- Run over heterogeneous Link/Physical layers
  - Motivates minimizing promises about the service

- Global delivery
  - Must be scalable

- Arbitrary Topology
  - Hard to get it wrong!

- Low overhead switching
  - Minimal processing of IP packet
    - E.g., don’t have to rewrite IP header (much…)

- Network control / diagnosis
  - Routers have IP addresses, just like everyone else
    - Ping / traceroute
Hop-by-Hop, not Paths

• Datagram delivery: postal service
  – connectionless, best-effort or unreliable service
  – Network can’t guarantee delivery of the packet
  – Each packet from a host is routed independently
  – Example: IP

• Virtual circuit models: telephone
  – connection-oriented service
  – Signaling: connection establishment, data transfer, teardown
  – All packets from a host are routed the same way (router state)
  – Example: ATM, Frame Relay, X.25
Internet Protocol (IP)

• IP (RFC791) defines a datagram “best effort” service
  – May be loss, reordering, duplication, and errors!
  – Currently IPv4 (IP version 4), IPv6 “on the way”

• Routers forward packets using periodically updated routes
  – Routing protocols (RIP, OSPF, BGP) run between routers to maintain routes (routing table, forwarding information base)
  – Over medium term, one path from host A to host B

• Global, hierarchical addresses, not flat addresses
  – 32 bits in IPv4 (128 bits in IPv6)
  – ARP (Address Resolution Protocol) maps IP to MAC addresses for final delivery
The IP Narrow Waist

Model

Application
Transport
Network
Link

Protocols

Many
(HTTP, SMTP)
TCP / UDP
IP
Many
(Ethernet, …)

The “narrow waist”
IPv4 Header (and Select Fields)

- Length of packet
- Min 20 bytes, max 65K bytes (limit to packet size)
Fragmentation: What, Why, and Why Not

• Different networks may have different frame limits (MTUs)
  – Ethernet 1.5KB, FDDI 4.5KB

• Don’t know if packet will be too big for path beforehand
  – Could fragment on demand inside the network
    • IPv4
  – Could return an error to sending host
    • IPv6
Fragmentation and Reassembly

• **Strategy**
  – fragment only when necessary (MTU < Datagram size)
    • try to avoid fragmentation at source host

  – this implies that refragmentation must be possible
    • fragments are self-contained IP datagrams

  – delay reassembly until destination host

  – do not recover from lost fragments
Avoiding Fragmentation

• Always send small datagrams
  – Might be too small
    • Why does that matter?

• “Guess” MTU of path
  – Use DF flag. May have large startup time

• Discover actual MTU of path
  – One RT delay w/help, much more w/o
    • Hosts send packets, routers return error if too large
IPv4 Header Fields...

- **Time To Live**
- **Decremented by router and packet discarded if = 0**
- **Prevents immortal packets**
- **traceroute**
IPv4 Header Fields ... 

- Identifies higher layer protocol
  - E.g., TCP, UDP
- De-mux’ing key at destination host
IPv4 Header Fields ...

- **Header checksum**
  - Doesn’t cover data

- **Recalculated by routers (TTL drops)**

- **Disappears for IPv6**

<table>
<thead>
<tr>
<th></th>
<th>Version</th>
<th>HLen</th>
<th>TOS</th>
<th>Length</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Identifier for Fragments</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TTL</td>
<td>Protocol</td>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Source Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Destination Address</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Options (variable)</td>
<td>Pad (variable)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Data

2/1/2010
IP Addresses and Datagram Forwarding

- IP addresses have hierarchy
  - MAC addresses are basically random

- How the source gets the packet to the destination:
  - if source is on same network (LAN) as destination, source sends packet directly to destination host, using MAC address
  - else source sends data to a router on the same network as the source (using router’s MAC address)
  - router will forward packet to a router on the next network over (by sending out through a different one of its interfaces, and MAC address on that network for next router)
  - and so on...
  - until packet arrives at router on same network as destination; then, router sends packet directly to destination host (MAC address)

- Requirements
  - every host needs to know address of a router on its LAN
  - every router needs a routing table to tell it which neighboring network to forward a given packet on
  - Need some kind of support for mapping IP address → MAC address
IP vs. MAC addresses

• All 128.208 addresses are this way
• ...

Routing table

128.208.1.142
128.208.1.140
140.142.13.107
140.142.13.107
142.150.210.12
ICMP

• What happens when things go wrong?
  – Need a way to test/debug a large, widely distributed system

• ICMP = Internet Control Message Protocol (RFC792)
  – Companion to IP – required functionality

• Used for error and information reporting:
  – Errors that occur during IP forwarding
  – Queries about the status of the network
ICMP Generation

Error during forwarding!

source

dest

IP packet

ICMP packet

IP hdr  ICMP hdr  IP packet

Type  Code  Checksum
Common ICMP Messages

• Destination unreachable
  – “Destination” can be host, network, port or protocol
• Packet needs fragmenting but DF (don’t fragment) flag is set
• Redirect
  – To shortcut circuitous routing
• TTL Expired
  – Used by the “traceroute” program
• Echo request/reply
  – Used by the “ping” program
• Cannot Fragment
• BustedChecksum

• ICMP messages include portion of IP packet that triggered the error (if applicable) in their payload
ICMP Restrictions

- The generation of error messages is limited to avoid cascades ... error causes error that causes error!

- Don’t generate ICMP error in response to:
  - An ICMP error
  - Broadcast/multicast messages (link or IP level)
  - IP header that is corrupt or has bogus source address
  - Fragments, except the first

- ICMP messages are often rate-limited too.
Key Concepts

• Network layer provides end-to-end data delivery across an internetwork, not just a LAN

• Routing decisions are sequence of “which hop next?”
  – Packet routing isn't picking a full path

• Next: More detailed look at routing and addressing