CPSC156a: The Internet
Co-Evolution of Technology and Society

Lecture 3: September 11, 2003
Internet Basics, continued

Acknowledgments: R. Wang and J. Rexford
Directly Connected Machines

(a) Point-to-point: e.g., ATM
(b) Multiple-access: e.g., Ethernet

Can’t build a network by requiring all nodes to be directly connected to each other; need scalability with respect to the number of wires or the number of nodes that can attach to a shared medium
Switched Network

- Circuit switching vs. packet routing
- Hosts vs. “the network,” which is made of routers
- Nice property: scalable aggregate throughput
Interconnection of Networks

Recursively build larger networks
Some Hard Questions

• How do hosts share links?
• How do you name and address hosts?
• Routing: Given a destination address, how do you get to it?
IP Addresses and Host Names

- Each machine is addressed by an integer, its **IP address**, written down in a “dot notation” for “ease” of reading, such as 128.36.229.231.
- IP addresses are the universal IDs that are used to name everything.
- For convenience, each host also has a human-friendly host name. For example, 128.36.229.231 is concave.cs.yale.edu.
- Question: How do you translate names into IP addresses?
Domain Hierarchy

- Initially, name-to-address mapping was a flat file mailed out to all the machines on the Internet.
- Now, we have a hierarchical name space, just like a UNIX file-system tree.
DNS Zones and Name Servers

• Divide up the name hierarchy into zones.
• Each zone corresponds to one or more name servers under the same administrative control.
Hierarchy of Name Servers

- Clients send queries to name servers.
- Name servers reply with answers or forward requests to other name servers.
- Most name servers perform “lookup caching.”
Application-Level Abstraction

- What you have: hop-to-hop links, multiple routes, packets, can be potentially lost, can be potentially delivered out-of-order
- What you may want: application-to-application (end-to-end) channel, communication stream, reliable, in-order delivery
Basic Architectural Principle: Layering

- HTTP (Web)
- Telnet
- Transmission Control Protocol
- Domain Name Service
- Simple Network Management
- User Datagram Protocol
- Internet Protocol
- SONET
- Ethernet
- ATM
The Physical Layer

- A network spans different hardware.

- Physical components can work however they want, as long as the interface between them is consistent.

- Then, different hardware can be connected.
The Role of the IP Layer

- Internet Protocol (IP): gives a standard way to “package” messages across different hardware types.

1. Message is put in IP packet.
2. Dial-up hardware gets packet to router (however it wants, but intact).
3. Routers look at destination, decide where to send it next.
4. Packet gets to destination network.
5. Original message extracted from packet.
IP Connectionless Paradigm

• No error detection or correction for packet data
  - Higher-level protocol can provide error checking
• Successive packets may not follow the same path
  - Not a problem as long as packets reach the destination
• Packets can be delivered out-of-order
  - Receiver can put packets back in order (if necessary)
• Packets may be lost or arbitrarily delayed
  - Sender can send the packets again (if desired)
• No network congestion control (beyond “drop”)
  - Send can slow down in response to loss or delay
# IP Packet Structure

<table>
<thead>
<tr>
<th>4-bit Version</th>
<th>4-bit Header Length</th>
<th>8-bit Type of Service (TOS)</th>
<th>16-bit Total Length (Bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>16-bit Identification</td>
<td>3-bit Flags</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>13-bit Fragment Offset</td>
</tr>
<tr>
<td>8-bit Time to Live (TTL)</td>
<td>8-bit Protocol</td>
<td>16-bit <strong>Header Checksum</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32-bit <strong>Source IP Address</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>32-bit <strong>Destination IP Address</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Options (if any)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Payload</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Main IP Header Fields

- **Version number** (e.g., version 4, version 6)
- **Header length** (number of 4-byte words)
- **Header checksum** (error check on header)
- **Source** and **destination** IP addresses
- Upper-level protocol (e.g., TCP, UDP)
- **Length** in bytes (up to 65,535 bytes)
- IP options (security, routing, timestamping, etc.)
- **TTL** (prevents messages from looping around forever; packets “die” if they “get lost”)

Getting from A to B: Summary

• Need IP addresses for:
  • Self (to use as source address)
  • DNS Server (to map names to addresses)
  • Default router to reach other hosts (e.g., gateway)

• Use DNS to get destination address

• Pass message through TCP/IP handler

• Send it off! **Routers** will do the work:
  • Physically connecting different networks
  • Deciding where to next send packets
Internet Architecture

dial-in access

ISP 1

interdomain protocols

ISP 2

private peering
gateway router
destination

NAP

ISP 3

destination

INSP 1

intradomain protocols

access router

commercial customer
Discussion Point

• Dial-up, intermittent access
  - Low-bandwidth, slow
  - Dynamic IP addressing more private?

• Cable, always-on access
  - High-bandwidth, fast
  - Static IP addressing less private?

Other examples of similar tradeoffs?