Internetworking

Outline

Internet Architecture
Best Effort Service Model
Layering revisited (cause it’s real important)

• Recall pros and cons of packet switched networks
  – Pros: High utilization, low startup overhead
  – Cons: No guarantees on delay and loss
• But files are transmitted without “holes”?
• Reliable transfer is a function of a specific protocol layer (TCP)
• Distinct functions are separated into layers
  – lower layers as black boxes (like C library)
• Layering simplifies description of functions and enables interoperability in heterogeneous environment
ISO Architecture

One or more nodes within the network
Internet Architecture

- Defined by Internet Engineering Task Force (IETF)
  1. Application: interacts with user to initiate data transfers (browser, media player, command line)
  2. Transport: reliable, in-order delivery of data (TCP and UDP)
  3. Network: addressing and routing (IP)
  4. Data Link: defines how hosts access physical media (Ethernet)
  5. Physical: defines how bits are represented on wire (Manchester)

- Information is passed between layers via encapsulation
  - Header information is attached to data passed down layers
- Multiplexing between layers
- Layers access other layers via API’s (eg. sockets)
- Communication at a specific layer is enabled by a protocol
Hourglass Design

• Single protocol at network level insures packets will get from source to destination while allowing for flexibility
IP Internet

• Concatenation of Networks

• Protocol Stack
IP Service Model

• Connectionless (datagram/packet-based)
• Best-effort delivery (unreliable service)
  – packets are lost
  – packets are delivered out of order
  – duplicate copies of a packet are delivered
  – packets can be delayed for a long time
• Datagram format

![Datagram Format Diagram]

<table>
<thead>
<tr>
<th>Field</th>
<th>Position</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Version</td>
<td>0-3</td>
<td>Version</td>
</tr>
<tr>
<td>HLen</td>
<td>4-7</td>
<td>Header Length</td>
</tr>
<tr>
<td>TOS</td>
<td>8-11</td>
<td>Type of Service</td>
</tr>
<tr>
<td>Length</td>
<td>12-15</td>
<td>Length</td>
</tr>
<tr>
<td>Ident</td>
<td>16-19</td>
<td>Identification</td>
</tr>
<tr>
<td>Flags</td>
<td>20-23</td>
<td>Flags</td>
</tr>
<tr>
<td>Offset</td>
<td>24-27</td>
<td>Offset</td>
</tr>
<tr>
<td>TTL</td>
<td>28-31</td>
<td>Time to Live</td>
</tr>
<tr>
<td>Protocol</td>
<td>32-35</td>
<td>Protocol</td>
</tr>
<tr>
<td>Checksum</td>
<td>36-40</td>
<td>Checksum</td>
</tr>
<tr>
<td>SourceAddr</td>
<td>41-55</td>
<td>Source Address</td>
</tr>
<tr>
<td>DestinationAddr</td>
<td>56-70</td>
<td>Destination Address</td>
</tr>
<tr>
<td>Options (variable)</td>
<td>71-85</td>
<td>Options (variable)</td>
</tr>
<tr>
<td>Pad (variable)</td>
<td>86-90</td>
<td>Pad (variable)</td>
</tr>
<tr>
<td>Data</td>
<td>91-105</td>
<td>Data</td>
</tr>
</tbody>
</table>

CS 640 7
Datagram Forwarding

• Strategy
  – every datagram contains destination’s address
  – if directly connected to destination network, then forward to host
  – if not directly connected to destination network, then forward to some router
  – forwarding table maps network number into next hop
  – each host has a default router
  – each router maintains a forwarding table

• Example

<table>
<thead>
<tr>
<th>Network Number</th>
<th>Next Hop</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>R3</td>
</tr>
<tr>
<td>2</td>
<td>R1</td>
</tr>
<tr>
<td>3</td>
<td>interface 1</td>
</tr>
<tr>
<td>4</td>
<td>interface 0</td>
</tr>
</tbody>
</table>
Forwarding Tables

• Suppose there are $n$ possible destinations, how many bits are needed to represent addresses in a routing table?
  – $\log_2 n$

• So, we need to store and search $n \times \log_2 n$ bits in routing tables?
  – We’re smarter than that!