PROTOCOLS AND INTERNETWORKING

Ken Wong
Washington University in St. Louis

kenw@arl.wustl.edu
www.arl.wustl.edu/~kenw

POSTAL EXAMPLE

- The Post Office
  - **Service**: Mail Delivery
  - **Service Access Point**: Post Office
    - The interface between the lower and upper protocol layer
  - **Service Data Unit**: A Letter
    - The packet handed down to a lower protocol layer
- **Protocol Data Unit**:
  - A mailbag containing letters to a common destination
  - Packets exchanged between peer entities
- **Customer**
  - **Protocol Data Unit**: A Letter
  - **Protocol header**:
    - Destination address, source address, stamp

A COMMUNICATION (NETWORK) PROTOCOL

- **A Communication (Network) Protocol**
  - **Rules**: that govern the exchange of messages between network processes
  - **Message Formats**
- **Protocol Specification**
  - **Semantics**: A statement of relations between a set of communicating processes
  - **Syntax**: The notation used in the specification
- **The Ideal Specification**
  - Easy to understand; Unambiguous; Scalable; Supports abstraction; Easy to prove properties
- **Specification Syntax**
  - **Natural Language**
  - **Programming Language**
  - **Processes as FSMs**
REQUEST-REPLY PROTOCOL EXAMPLE (1)

- **System**
  - A *transmitter* process sends integer messages to a *receiver* process
  - Messages go over an error-free link
  - The transmitter always has another message to send after sending a message
  - The receiver has a single message buffer which takes a finite time to process
  - Want lossless (reliable) message delivery

- **Message Formats**
  - Transmitter Process: Each message is an arbitrary integer
  - Receiver Process: Each message is a 1 indicating that a message was consumed

- **Protocol Rules**
  - Process States: 1 (message was sent); 0: (message was received)
  - A process in state 0 can send one message to the other process; it then enters state 1
  - A process in state 1 waits for one message from the other process and then enters state 0
  - Is the protocol reliable?

REQUEST-REPLY PROTOCOL EXAMPLE (2)

Transmitter

```
0 1
```

Receiver

```
0 1
```

- A condition/event causes a state transition
- Actions are associated with each state

THE INTERNET

- **The Internet**
  - A collection of networks that are reachable with IP
  - Organized into a multilevel hierarchy

- **An Internet-Capable Host**
  - Has a 32-bit *IP address* (e.g., 128.192.64.10, 0x80C0400A)
  - Knows how to *route packets* to their destination
  - Formats data into *IP packets*

- **Internet Growth**
  - Addressing, routing, and IP were designed to scale.

BASIC INTERNET TECHNOLOGY

- **Packets**
  - Packets carry information
  - A packet has 2 parts:
    - A *payload* (information content)
    - A *header* (metadata (information about the payload))

- **Store-And-Forward Technology**
  - The metadata allows a packet to be stored at a router for eventual delivery
  - The *packet can be released when convenient*
  - Direct analogy with the post office
  - Less expensive to operate than the telephone network
INTERNET PROTOCOL LAYERS

- Services at one level usually depend on services at lower levels (leads to protocol layers)
- The layers form a protocol stack

INTERNET PROTOCOL LAYERS

- Physical
  - Interface between a data transmission device and transmission medium
- Network
  - Accessing and routing across the same network
  - Exchange data between endsystem and network
  - Endsystem addressing
- Internet (IP)
  - Routing between different networks
  - Endsystem addressing that hides network heterogeneity
- Transport (UDP, TCP)
  - Process addressing (Port number)
  - Reliable, ordered delivery
- Application

IP PROTOCOL LAYERING EXAMPLE

- Telnet Protocol
- TCP Protocol
- IP Protocol
- Ethernet Protocol

IPv4 ADDRESSING

- One unique IP address required for each active host-interface card
  - A central authority allocates blocks of IP Addresses to organizations
- 32-bit IP Addresses (4 octets)
  - Class C: 1 2 8 . 2 5 2 . 1 5 3 . 1 0
  - Class B: 10 - - - - - - - - - - - - - - - -
  - Class A: 0 - - - - - - - - - - - - - - - -
- Router: A host with an interface on more than one network
  - Default Route: "Near-by" versus distant networks
  - Computing consistent routing tables
IP ADDRESSING

- Suppose that you have 3 networks (0, 1, 2) each with 2 hosts (0 and 1).
- How do you get a packet from host 2.0 to 1.1?

TCP HEADER

<table>
<thead>
<tr>
<th>Bits</th>
<th>0</th>
<th>4</th>
<th>10</th>
<th>16</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Port</td>
<td>Destination Port</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Acknowledgement Number</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header Length</td>
<td>Unused</td>
<td>6 Flags</td>
<td>Window</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Checksum</td>
<td>Urgent Pointer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Options + Padding</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

IPv4 PACKET

<table>
<thead>
<tr>
<th></th>
<th>0</th>
<th>4</th>
<th>8</th>
<th>16</th>
<th>19</th>
<th>24</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Vers.</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hdr. Lengh</td>
<td>ToS</td>
<td>Total Length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Identification</td>
<td>Flags</td>
<td>Fragment Offset</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Time to Live</td>
<td>Protocol</td>
<td>Header Checksum</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source IP Address

Destination IP Address

IP Options (if any)

Padding

Data

...

OPERATION OF TCP

Host A

App

TCP

IP

NAP

Physical

Router

Network 1

NAP 1

NAP 2

Network 2

Host B

App

TCP

IP

NAP

Physical

Logical TCP Connection

Port (or SAP)
**THE INTERNET PHILOSOPHY**

- **The End-To-End Argument**
  - Functions that can be best handled by endsystems should not be duplicated in the network
  - Ends systems should handle delivering data reliably, NOT the network

- **Decentralized Control**
  - Ends systems should NOT depend on the network for reliable/timely data

- **Endsystem Control**
  - Allows the network to be built with few assumptions (the network can be dumb)
  - Easy to add new network types (e.g., satellite) into the Internet
  - However, upgrading the network may be hard!