Building Applications
Network applications involve communication of two or more hosts
Often, complex functions need to be realized
Can think of application communication in an abstract way

Common Services
Many applications may share common functionalities
Can you think of examples?

These functionalities need to be integrated on each application

Or be abstracted in common services
E.g. FTP vs. Video-on-Demand

Both follow the server-client model

Establish a request/reply channel, and message stream channel (one reliable, other unreliable)

Use smallest number of channel abstractions

FTP utilizes request/reply
Video-on-demand uses message stream channel
Both use a host-to-host communication protocol

A simple layering example

RRP: Request/reply protocol
MSP: Message streaming protocol
HHP: Host-to-host protocol

Layering Abstraction

Layer: A set of functionalities encapsulated in an object that can be used by other network components
Example: The network layer implements the end-to-end packet delivery
Why layering? Think complexity and common services
Layers consist of protocols
Looking into layers a bit closer

Protocols in each layer have
  Service interface with upper layer/lower layer
  Peer-to-peer interface with host on same layer

Hierarchical Layer Structure

Layering implies the use of a layer hierarchy

Encapsulation

The process of embedding a header or trailer
The OSI Network Architecture

OSI: Open Systems Interconnection

- 7 layers x. protocol specifications for each layer.
- Acts like a reference model rather than a real-world protocol graph.
- First three layers are implemented in all nodes.

Layers of the OSI model

- **Physical Layer**: Transmission/reception of raw bits.
- **Data Link Layer**: Maps bits into frames, dictates sharing of common medium, corrects/detects errors, re-orders frames.
- **Network Layer**: Routes packets to destination, may perform fragmentation and re-assembly.
- **Transport Layer**: Flow (congestion) control, error control, transparent transport to upper layers.
- **Session Layer**: Establishes connection among hosts, duplex, half-duplex, graceful connection termination, combination of streams.
- **Presentation Layer**: Negotiation of format of data exchanged between hosts.
- **Application layer**: Application services such as FTP, X.400 (mail), HTTP.

The Internet Architecture

- **FTP**: File Transfer Protocol
- **HTTP**: Hypertext Transport Protocol
- **TFTP**: Trivial File Transfer Protocol
- **DNS**: Domain Name System
- **TCP**: Transmission Control Protocol
- **UDP**: User Datagram Protocol
- **IP**: Internet Protocol
Comparison of the two architectures

<table>
<thead>
<tr>
<th>OSI</th>
<th>TCP/IP</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 Application</td>
<td>Application</td>
</tr>
<tr>
<td>6 Presentation</td>
<td></td>
</tr>
<tr>
<td>5 Session</td>
<td></td>
</tr>
<tr>
<td>4 Transport</td>
<td>Transport</td>
</tr>
<tr>
<td>3 Network</td>
<td>Internet</td>
</tr>
<tr>
<td>2 Data link</td>
<td>Host-to-network</td>
</tr>
<tr>
<td>1 Physical</td>
<td></td>
</tr>
</tbody>
</table>

Motivation for IP Networks (In Order)

Communication should continue despite failures
- Survive equipment failure or physical attack
- Traffic between two hosts continue on another path

Support multiple types of communication services
- Differing requirements for speed, latency, & reliability
- Bidirectional reliable delivery vs. message service

Accommodate a variety of networks
- Both military and commercial facilities
- Minimize assumptions about the underlying network

Other Driving Goals, Somewhat Met

Permit distributed management of resources
- Nodes managed by different institutions
  - though this is still rather challenging

Cost-effectiveness
- Statistical multiplexing through packet switching
  - though packet headers and retransmissions wasteful

Ease of attaching new hosts
- Standard implementations of end-host protocols
  - though still need a fair amount of end-host software

Accountability for use of resources
- Monitoring functions in the nodes
  - though this is still fairly limited and immature