Last Time …

- Networks are used to share distributed resources
  - Key problems revolve around effective resource sharing
- Statistical multiplexing
  - It’s well-suited to data communications
This Lecture

1. A top-down look at the Internet
2. Mechanics of protocols and layering
3. The OSI/Internet models

1. A Brief Tour of the Internet

- What happens when you “click” on a web link?

- This is the view from 10,000 ft …
9,000 ft: Scalability

- Caching improves scalability
  - Check cache (local or proxy) for a copy
  - Check with server for a new version

8,000 ft: Naming (DNS)

- Map domain names to IP network addresses
  - “What’s the IP address for www.netscape.com?”
  - “It’s 207.200.75.200”

- All messages are sent using IP addresses
  - So we have to translate names to addresses first
  - But we cache translations to avoid next time
7,000 ft: Sessions (HTTP)

- A single web page can be multiple “objects”
  - Fetch each “object”
    - either sequentially or in parallel

GET index.html

GET ad.gif

GET logo.gif

6,000 ft: Reliability (TCP)

- Messages can get lost

(lost)

retransmission

acknowledgment

- We acknowledge successful receipt and detect and retransmit lost messages (e.g., timeouts)
5,000 ft: Congestion (TCP)

- Need to allocate bandwidth between users

  How fast can I send?

- Senders balance available and required bandwidths by probing network path and observing the response

4,000 ft: Packets (TCP/IP)

- Long messages are broken into packets
  - Maximum Ethernet packet is 1.5 Kbytes
  - Typical web page is 10 Kbytes

- Number the segments for reassembly

GET index.html
3,000 ft: Routing (IP)

- Packets are directed through many routers

R: Routers
H: Hosts

Routers

1-10Mb/s < 1Gb/s 92Tb/s
2,000 ft: Multi-access (e.g., Cable)

- May need to share links with other senders

- Poll headend to receive a timeslot to send upstream
  - Headend controls all downstream transmissions
  - A lower level of addressing is used …
1,000 ft: Framing/Modulation

• Protect, delimit and modulate payload as a signal

Sync / Unique | Header | Payload w/ error correcting code

• E.g., for cable, take payload, add error protection (Reed-Solomon), header and framing, then turn into a signal
  – Modulate data to assigned channel and time (upstream)
  – Downstream, 6 MHz (~30 Mbps), Upstream ~2 MHz (~3 Mbps)

2. Protocols and Layering

• We need abstractions to handle all this system complexity

• A protocol is an agreement dictating the form and function of data exchanged between parties to effect communication

• Two parts:
  – Syntax: Words,
    • where the bits go
  – Semantics: Meaning
    • what the words mean, what to do with them

• Examples:
  – Ordering pizza
  – IP, the Internet protocol
  – TCP and HTTP, for the Web
Protocol Standards

- Different functions require different protocols
- Thus there are many protocol standards
  - E.g., IP, TCP, UDP, HTTP, DNS, FTP, SMTP, NNTP, ARP, Ethernet/802.3, 802.11, RIP, OSPF, 802.1D, NFS, ICMP, IGMP, DVMRP, IPSEC, PIM-SM, BGP, …
- Organizations: IETF, IEEE, ITU
- IETF (www.ietf.org) specifies Internet-related protocols
  - RFCs (Requests for Comments)
  - “We reject kings, presidents and voting. We believe in rough consensus and running code.” – Dave Clark.

Layering and Protocol Stacks

- Layering is how we combine protocols
  - Higher level protocols build on services provided by lower levels
  - Peer layers communicate with each other
Example – Layering at work

- We can connect different systems

Layering Mechanics

- Encapsulation and deencapsulation
A Packet on the Wire

- Starts looking like an onion!

\[
\text{EthernetHdr} \rightarrow \text{IPHdr} \rightarrow \text{TCPHdr} \rightarrow \text{HTTPHdr} \rightarrow \text{Payload (Web object)}
\]

- Start of packet
- End of packet

- This isn’t entirely accurate
  - ignores segmentation and reassembly, Ethernet trailers, etc.
- But you can see that layering adds overhead

More Layering Mechanics

- Multiplexing and demultiplexing in a protocol graph
3. OSI/Internet Protocol Stacks

Key Question: What functionality goes in which protocol?

The “End to End Argument” (Reed, Saltzer, Clark, 1984):

- Functionality should be implemented at a lower layer only if it can be correctly and completely implemented. (Sometimes an incomplete implementation can be useful as a performance optimization.)
- Tends to push functions to the endpoints, which has aided the transparency and extensibility of the Internet.

### Internet Protocol Framework

<table>
<thead>
<tr>
<th>Model</th>
<th>Protocols</th>
</tr>
</thead>
<tbody>
<tr>
<td>Link</td>
<td>Many (Ethernet, …)</td>
</tr>
<tr>
<td>Network</td>
<td>IP</td>
</tr>
<tr>
<td>Transport</td>
<td>TCP / UDP</td>
</tr>
<tr>
<td>Application</td>
<td>Many (HTTP, SMTP)</td>
</tr>
</tbody>
</table>
A Protocol is an ENCAPSULATION. One protocol’s payload may be another encapsulated protocol.
Key Concepts

- Protocol layers are the modularity that is used in networks to handle complexity
- The Internet/OSI models give us a roadmap of what kind of function belongs at what layer