Introduction to Networks and the Internet

CMPE 80N
Winter 2004
Lecture 22

Announcements

- “Internet behind the Web” video on March 10th.
- 5th. Quiz Friday, March 12th.
- Project 2 due on Friday, March 12th.
  - By midnight.
  - E-mail page URL to BOTH Kiran and Debasree.
- Final exam, Thu March 18th.

Announcements (cont’d)

- Discussion session for quiz 5 and final:
  - Debasree on Thu, 03.11 from 3-4.
  - Kiran on Friday, 03.12 right after class.
- Quiz 4 grades are out.
- Quiz 4 statistics:
  - Total points: 60.
  - Maximum score: 60.
  - Minimum 18.
  - Average: 44.
  - Number of papers: 82.

TCP Congestion Control
**Queuing**

Packets are queued inside router, waiting to be transmitted on outgoing link.

---

**Queuing (cont’d)**

- What happens if packets are coming in faster than router is able to forward them?
  - Needs to “queue” the packets as they are waiting to be transmitted.
- What happens when a queue becomes too long?
  - Packets are dropped!
- How can the receiver detected that a packet was dropped?
  - Look at the packet sequence number!

---

**Congestion**

- Network with 1 Mb/s lines and 1000 computers, half of which are trying to transfer files at 100 Kb/s to the other half.
  - The total offered traffic exceeds what the network can handle (congestion).
- **Congestion collapse:**
  - When congestion occurs, packets get dropped.
  - Due to packet loss, packets get retransmitted.
  - Congestions gets worse and worse!

---

**Congestion Control**

- Why do it at the transport layer?
  - Real fix to congestion is to slow down sender.
- Use law of “conservation of packets”.
  - Keep number of packets in the network constant.
  - Don’t inject new packet until old one leaves.
- Congestion indicator: packet loss.
**TCP and Congestion Control**

- Interprets packet loss as an indicator of congestion
  - When it senses packet loss, it slows down the rate of packet transmission
  - When packets are received correctly, sends packets faster
    - Still within the limits of the sliding window

**TCP Congestion Control**

- Like, flow control, also window based.
  - Sender keeps congestion window (cwin).
  - Each sender keeps 2 windows: receiver's advertised window and congestion window.
  - Number of bytes that may be sent is \( \min(\text{advertised window}, \text{cwin}) \).

**TCP Segment Header**

<table>
<thead>
<tr>
<th>Source port</th>
<th>Destination port</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Sequence number</td>
<td>Acknowledgment number</td>
</tr>
<tr>
<td>Window size</td>
<td></td>
</tr>
<tr>
<td>Options (0 or more 32-bit words)</td>
<td>Data</td>
</tr>
</tbody>
</table>

**UDP**

- Provides connection-less, unreliable service.
  - No delivery guarantees.
  - No ordering guarantees.
  - No duplicate detection.
- Low overhead.
  - No connection establishment/teardown.
- Suitable for short-lived connections.
  - Example: client-server applications.
**UDP Segment Format**

<table>
<thead>
<tr>
<th>0</th>
<th>15</th>
<th>31</th>
</tr>
</thead>
<tbody>
<tr>
<td>Source port</td>
<td>Destination port</td>
<td></td>
</tr>
<tr>
<td>Length</td>
<td>Checksum</td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source and destination ports: identify the end points.
Length: 8-byte header + data.
Checksum: optional; if not used, set to zero.

**TCP and UDP**

- **TCP** provides end-to-end communication. It takes care of **reliable, error-free transfer** of data, and **in-sequence delivery**
- **UDP** has **less overhead** compared to TCP, but **does not guarantee transfers**
  - TCP is preferred to transfer files
  - UDP is preferred to transfer audio/video streams
    - In real-time streaming, we cannot afford the delay consequent to packet retransmission
- Both protocols support **multiplexing**, i.e. they allow several distinct streams of data between two hosts

**Internet Applications**

**Client-Server Architecture**

Client

request

response

Server
**Client-Server Architecture**

Clients → Server → Clients

---

**The Domain Name System (DNS)**

- IP addresses are not easy to remember.
- The **Domain Name System (DNS)** maps IP addresses to host names.
- Host name is formed by machine name followed by domain name.
  - `Host_name.domain_name`
DNS (cont’d)

• The domain name is formed by the institutional site name and the Top-Level Domain name (TLD).
  – So the host name is of the form: machine_name.Ist_site_name.TLD_name

• Examples:
  – sundance.ucsc.edu
  – soe.ucsc.edu (alias for sundance.ucsc.edu)
  – italias.jpl.nasa.gov
  – www.cnn.com

TLD

• TLD names identify organization types or country codes.

• Examples:

  .com Commercial org.
  .edu Educational site in US
  .gov Government site in US
  .mil Military organization in US
  .net Network site
  .org Nonprofit organization
  .au Australia
  .ca Canada
  .fr France
  .de Germany
  .uk Great Britain
  .it Italy
  .es Spain

Countries define their own internal hierarchy (e.g., .ac.uk, .edu.au)

DNS (cont’d)

• Organizations can create any internal DNS hierarchy.

• Authority for creating new subdomains within a domain name is delegated to each domain.
  – Administration of ucsc.edu has authority to create cse.ucsc.edu and need not contact any central naming authority.
**DNS Name Space**

- DNS names are managed by a hierarchy of DNS servers.
  - Hierarchy is related to DNS domain hierarchy
- Root server at top of tree knows about next level servers.
- Next level servers, in turn, know about lower level servers.

**Example of DNS Hierarchy**

**Choosing DNS Server Architecture**

- Small organizations can use a single server.
  - Easy to administer.
  - Inexpensive.
- Large organizations often use multiple servers.
  - Reliability through redundancy.
  - Improved response time through load sharing.
Name Resolution

- “Resolving a name” means mapping the host name to the IP address.
  - Reverse mapping is also possible.
- A client computer calls a DNS server for name resolution
  – DNS request contains name to be resolved.
  – DNS reply contains IP address for name in request.

Using DNS Servers

- Each DNS server is the authoritative server for the names it manages.
  – If request contains name managed by receiving server, that server replies directly.
  – Otherwise, request is forwarded to the appropriate authoritative server.
- DNS request is originally sent to root server, which points at next server to use
  – Eventually, the authoritative server for the DNS name in the request is located and IP address is returned.

Internet Applications (cont’d)

- File Transfer (FTP).
- E-mail.
- Remote login.
- WWW.

The Web

- WWW, or the world-wide web is a resource discovery service.
  – Resource space is organized hierarchically, and resources are linked to one another according to some relation.
  – Hypertext organization: link “granularity”; allows links within documents.
  – Graphical user interface.
Hypertext/Hypermedia

- **Hypermedia** system allows interactive access to collections of documents
- Document can hold:
  - Text (hypertext)
  - Graphics
  - Sound
  - Animations
  - Video
- Documents linked together
  - Non-distributed (all stored locally - like CD-ROM)
  - Distributed (stored on remote servers)

Some History 1

- Started in 1989 at CERN, European center for nuclear research, in Switzerland.
- Original motivation: need for scientists around the world to collaborate and share multi-media information.
- Tim Berners-Lee came up with initial proposal of a web of linked documents

Some History 2

- First text-based prototype demo in 12.91.
- Release of first graphical interface, Mosaic, in 02.93 at NCSA by M. Andreessen.
- In 1994, Andreessen creates Netscape.
- In 1994, CERM and MIT set up the WWW Consortium to further develop the Web.
  - www.w3.org for more information.

The Client Side

- Users perceive the Web as a vast collection of information.
  - Page is the Web's information transfer unit.
  - Each page may contain links to other pages.
  - Users follow links by clicking on them which takes them to the corresponding page.
  - This process can go on indefinitely, traversing several pages located in different places.
The Browser

- Program running on client that retrieves and displays pages.
  - Interacts with server of page.
  - Interprets commands and displays page.
- Examples: Mosaic, Netscape’s Navigator and Communicator, Microsoft Internet Explorer.
- Other features: back, forward, bookmark, caching, handle multimedia objects.

The Server Side

- Web site has Web server running that answers requests for pages locally served.
  - Web server listens to port 80 for requests.
  - When request from client arrives, connection is set up.
  - Server replies.
  - Connection released.

Example

- User clicked on www.w3.org/hypertext/WWW/TheProject.html.
  - Browser asks DNS to resolve www.w3.org.
  - DNS replies with 18.23.0.23.
  - Browser sets up connection to 18.23.0.23 port 80.
  - Browser sends GET /hypertext/WWW/TheProject.html.
  - www.w3.org server sends TheProject.html file.
  - Connection released.
  - Browser displays TheProject.html, fetching and displaying all embedded objects (images, etc).

Observations

- Many browsers display status information at bottom of the screen.
- For each embedded object (in-line image like icon, picture, etc), browser establishes new connection.
  - Performance hit.
  - Revisions to protocol (HTTP) address this.
- Since HTTP is ASCII, easy for user to talk to Web servers directly (e.g., telnet to port 80).
More Observations

• Server’s response specifies object type (using MIME) followed by object body.
• For example:
  – Content-Type: Image/GIF
  – Content-Type: Text/html

HTTP

• HyperText Transfer Protocol.
• Each interaction: client’s ASCII request followed by MIME-like response.
• Use TCP as underlying transport protocol (although not required by standard).
• Several co-existing versions of HTTP.

HTTP Operations

• Commands (method) to be executed on object (Web page).

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GET</td>
<td>Request to read Web page</td>
</tr>
<tr>
<td>HEAD</td>
<td>Request to read Web page’s header</td>
</tr>
<tr>
<td>PUT</td>
<td>Request to store Web page</td>
</tr>
<tr>
<td>POST</td>
<td>Append to specified object (e.g., Web page)</td>
</tr>
<tr>
<td>DELETE</td>
<td>Deletes Web page</td>
</tr>
<tr>
<td>LINK</td>
<td>Connects 2 objects</td>
</tr>
<tr>
<td>UNLINK</td>
<td>Disconnects 2 objects</td>
</tr>
</tbody>
</table>

HTTP Responses

• Every request gets response with status information.
  – Status code 200: OK.
  – Status code 400: bad request.
  – Status code 304: not modified.
**Uniform Resource Locator (URL)**

- Way to identify objects (pages).
  - What is page called?
  - Where is it located?
  - How to access page?
- URL has 3 parts:
  - Protocol (or scheme).
  - Machine’s name/address.
  - Local name (file name).

**HyperText Markup Language (HTML)**

- Allows users to produce Web pages including text, graphics, pointers, etc.
- Markup language: describe how objects are to be formatted.
  - Contains explicit commands for formatting.
  - Example: `<B>` and `</B>`.
  - Advantages: easy to parse.

**URL**

- Ability to handle other protocols.
  - HTTP, FTP, news, gopher, mail, telnet.
- Universal Resource Identifier (URI).
  - Location transparency.
  - Replication.

**Types of Documents**

- Documents on the web can be of three types:
  - **Static** (those we have seen so far)
    - Defined in text file by page author
    - Remains unchanged unless edited by author
  - **Dynamic**
    - Generated on demand by HTTP server
  - **Active**
    - Execute code on the WWW browser in the host computer
Dynamic Pages

- A dynamic document is generated by the server at each new connection.
  - That’s why sometimes, when downloading from the same URL, we obtain different pages.

Common Gateway Interface (cgi) standard defines server-application interaction.
  - CGI programs can be as simple as adding the time or date to the page.

Browser may supply parameters to CGI program.
  - Browser extends URL with additional parameters separated by ?

“Personalizing” Web Content

- If the server has personal information about the user, CGI can be used to “personalize” the page content.
  - Based on a current set of preferences.
    • Stock quotes.
  - Advertising based on customer personal info, or past preferences.

Forms

- Forms permit a web page to have blank areas in which the user must enter information.
  - Makes it possible to enter data directly.
    • Name, address, credit card info…
  - Allows information to be sent to the server directly.

What is a Cookie?

- A server invokes a CGI program each time a request arrives for the associated URL.
  - The server does not maintain any history of requests.
    • But a history is useful to allow CGI program to participate in dialog (e.g., to avoid having a user answer questions repeatedly).
  - Information saved between invocations is called state information.
  - State information is kept at the client’s side!
What is a Cookie (cont’d)

- State information is passed by browser in the form of a **cookie**.
  - A cookie is just a piece of state information that can be a few hundreds bytes long.
    - Cookies mostly contain values assigned by the server; additional information is stored on the server itself.
  - The cookie is kept in the client's computer.
  - When it contacts the Web site again, the browser inserts the cookie in the request.
    - From the server’s perspective, it appears that the browser can store and return state information!

Plugins

- A plugin is a small program that knows how to interpret one specific data format.
  - Extends browser
  - E.g.: Adobe Acrobat plugin (to read pdf files), Quicktime plugin, Real plugin...

Summary

- What’s a network?
  - Why to network?
  - Telephone network.
  - Evolution of the PSTN.
  - Data networks.
    - Components.
    - Types of data networks.
    - Topology.
  - Protocols, layering, network architecture.
    - ISO-OSI and TCP/IP.
    - Encapsulation/decapsulation.

Summary (cont’d)

- Physical layer.
**Physical Layer: Summary**

- Different types of signal:
  - Analog and digital.

- Analog communication infrastructure:
  - Need to convert digital to analog before transmitting: ADC.
  - DAC before entering computer.

- Digitization:
  - Sampling.
  - Sampling period and frequency (samples/sec or Hertz).
  - Sample representation (quantization).
  - Bit rate.

- Modems.
- Input/output connections.
  - RS 232.
  - USB.
  - Firewire.

- Broadband.
  - ISDN.
  - ADSL.
  - Cable modem.
  - Satellite.

**Data Link Layer: Summary**

- Layer 2: Data Link Layer.
  - Functions.
  - Layer 2 sub-layers.

- MAC.
  - Why MAC?
    - Multiplexing techniques.
  - Different types of MAC protocols.
    - Centralized versus distributed.
    - Contention-based, round-robin, and scheduled access.

- Ethernet:
  - Contention-based.
  - Carrier sensing + collision detection + exponential backoff.
  - Performance.
  - Fairness.
DLL Summary (cont’d)

- Token passing (e.g., token ring):
  - Round-robin.
  - Decentralized (versus polling which is centralized).
  - Special frame called token.
  - Performance.
  - Fairness.

- Local Area Networks (LANs).
- Topologies.
- Existing protocol stacks: Ethernet, token ring.
- Network Interface Cards (NIC).
  - Protocol layers it executes.
  - Handling data transmission/reception.
- LAN connections:
  - Network cables, adapters.
- LAN interconnection:
  - Why?
  - Hubs and switches.

DLL Summary (cont’d)

- Wireless LANs.
  - 802.11.
  - Collision avoidance.
  - Wireless channel contention.
  - Hidden and exposed terminals.
- Other wireless networks.
  - Cellular networks.
    - Cell, frequency reuse.
  - Mobile ad hoc networks (MANETs).

- Data link layer.
  - Framing.
  - Error control.
    - Error detection versus error correction.
    - Parity.
Network Layer: Summary

- Main functions of the network layer:
  - Routing.
  - Forwarding.
- What is routing?
  - What does a routing algorithm do?
    - Different routes may exist.
- Forwarding.
- Switches and routers.
  - Switched network versus a network connecting a host to every other host directly.

Network Layer: Summary (cont’d)

- Switch internals.
  - Incoming and outgoing interfaces.
- Store and forward.
  - Queuing.
- Next-hop forwarding.
- Routing table.
  - What it is and what it is used for.
  - How to build one based on a given topology.
  - Role of hierarchical addresses.

Internetworking: Summary

- What is internetworking?
- Gateways, routers, bridges and hubs/repeaters.
- Different approaches to internetworking.
  - Translation.
  - IP-style.
- IP addresses.
  - Network and host numbers.
  - Address classes.
  - Static addresses and DHCP.
  - Routing tables revisited.
Internetworking: Summary (cont’d)
- Internetwork routing.
- IP.
  - Best effort.
  - Datagram (versus virtual circuit).
  - Encapsulation revisited.
  - MTU.
  - Fragmentation.

Transport Layer: Summary
- End-to-end.
- TPDU.
- Connection-oriented versus connectionless service.
- Encapsulation revisited.
- Transport-layer addressing.
- TCP and UDP.

Transport Layer: Summary (cont’d)
- TCP.
- TCP functions.
  - Connection establishment.
  - Reliability, ordering, duplicate detection.
    - ACKs, sequence numbers, timers, timeouts.
  - Flow control.
    - Receiver’s advertised window.
    - Sliding window.
  - Congestion control.
- UDP.

Application Layer: Summary
- DNS.
- Web.