Networks

Protocols
TCP/IP and applications
WWW
Networking with Ruby
Reading: CSO 4.1 -- 4.4

Terminology

- A computer network is a set of interconnected computers
- Goals:
  - communication (send information to others, download from a company website, ...)
  - shared resources (backups, parallel computing, “the global grid”)
- A local area network (LAN) is a small group of systems, typically in one building
- A wide area network (WAN) can be very large, e.g. cover the whole US
- An internetwork (internet) is a collection of two or more networks with interfaces that transfer traffic between the subnets
- The Internet is a global network of systems communicating via TCP/IP

Protocols

- For a set of machines to exchange information they need to agree in advance on a communication protocol
- A protocol may define
  - the message format (size, encoding, structure, ...) 
  - address information
  - a sequence of operations
    - “hello, are you ready to accept a message?”
    - “yes, go ahead”
    - “here is page 1 of 13”
    - “got it”
    - “here is page 2 of 13”
    - ...
    - “sorry, I didn’t get that”
    - “here is page 5 again”
    - ...
    - “that’s the last page of my message”

Protocol Stacks

- Communication involves a layer of cooperating protocols
- operations at one layer are implemented by operations at a lower layer
- An example from the textbook:
Internet Protocols

The Internet is based on a four-layer set of protocols commonly known as **TCP/IP**
- **TCP** = transmission control protocol
- **IP** = internet protocol

The application layer defines interactions between user programs
- HTTP (hypertext transfer protocol)
  - “get x”
  - “200 OK”
  - “404 not found”
- SMTP (simple mail transfer protocol)
- FTP (file transfer protocol)
- many others

For more info: “internet protocol suite” at Wikipedia

Transport layer protocols prepare messages for transmission
- **TCP**
  - breaks messages into **packets**
  - attaches address information
  - adds checksums and other descriptions
- The TCP software on the receiver
  - checks for errors, and requests a retry if necessary
  - makes sure all packets are delivered (e.g. request a retry if a packet is lost)
  - reassembles the message from packets and passes it to the application
- Other transport protocols deal with streaming media, voice over IP (VOIP), ...

Network layer protocols deal with routing between nodes in a network
- breaks transport packets into smaller fixed size packets
- nodes maintain tables to manage network traffic
- may send several packets at once, over different routes

Current standard: **IPv4**
- 32-bit addresses
- almost used up!

Newer standard: **IPv6**

Link layer protocols pass data between two systems in a local network
- also defines connection between a local system and the gateway to another network

Examples:
- ethernet
- wireless (IEEE 802.11)
- PPP (point-to-point protocol), used for DSL and dial-up connections

Sometimes the physical connection is viewed as another layer (e.g. some sources say there are five layers in the Internet protocol suite)
- e.g. ethernet over twisted pair or coax cable
Local Area Network Connections

- In a small local area network machines talk directly to each other
  - use ethernet cables to connect each machine to a **switch**
  - switches are often placed in a closet, with cables running to offices

![Switch (aka Hub) Diagram]

Local Area Network Connections

- A wireless base station is another type of hub
  - most base stations also have ethernet connections

![Wireless Base Station Diagram]

Media Access

- At the physical level the communication medium is often shared by all the systems on a local network
  - a machine broadcasts its message on its local network segment
- When a machine wants to send a message it checks to make sure no other system is broadcasting ("carrier sense media access", or CSMA)
- A **collision** is possible if two machines start broadcasting at the same time
  - with ethernet machines can tell if their message is corrupted (CSMA/CD)

![Media Access Diagram]

Media Access

- With WIFI a system can’t tell when its message collides with another one
  - the protocol tries to avoid collisions (CSMA/CA)
  - system asks base station if it can transmit, waits for OK
- The “hidden terminal” is another source of collisions
  - in this picture A can’t tell if B is already broadcasting
Routers

- To make a larger internet use routers to pass information from one local subnetwork to another.
  - A router is like a switch, except it has an additional WAN (wide area network) connection.
  - Most wireless base stations are also routers.

Addresses

- TCP/IP requires each system on the Internet to have a unique address.
  - An IP address is a 32-bit number.
  - Usually written using “dotted decimal” notation.
  - Break the 32 bits into 4 8-bit pieces, write the equivalent decimal number for each segment.
  - 8 bits => 0..255.

Names

- To domain name system (DNS) assigns names to systems.
  - Top level domains have names like .edu, .gov, .com.
  - An organization can register a new name through ICANN (e.g., faber.edu).
  - The organization assigns local names (cs.faber.edu, rotc.faber.edu, ...).
  - Network administrators assign names within their local networks (www.cs.faber.edu).

Name Servers

- There are a lot of computers on the Internet.
  - Not likely that each system knows the names of all the others.
  - When a new system is installed, it registers its name with a name server.
  - Name servers share their information with other servers.
  - A system will contact its local name server when it needs to look up a new name.
Dynamic IP Addresses

- The dynamic host configuration protocol (DHCP) can assign IP addresses dynamically
  - some machines (e.g. teleost, fugu, wolferine) have permanent IP addresses
  - when fintan (my laptop) is in my office I connect via ethernet and get a dynamic address from a CIS department router
  - when fintan is at home, it gets a new IP address from a wireless router, which in turn gets its address from Comcast

Address Translation

- An important part of a router’s job is to translate local addresses as it forwards a packet

Ports

- Operating systems use an abstraction known as a port to manage network communication
  - ports are implemented in software, inside the OS
  - when a message arrives at a host it includes a port number to identify the application layer protocol that will handle the message
  - the OS passes the message on to the application that is “listening” on the port

Clients and Servers

- Most internet communication is organized around clients and servers
  - a client initiates a conversation by sending a message to a server
  - the client knows the server’s IP address
  - example: the web server at teleost.cs.uoregon.edu (128.223.8.115) is listening on port 80

The server is a program that is always running

- many servers are started by the system administrator when the OS starts (e.g. mail servers, web servers)
- the standard application level protocols that are part of every TCP/IP implementation (smtp, ftp, etc) have predefined port numbers
- users can start their own server software (e.g. database server) on other ports
- may need to check first to see a port isn’t already being used
Daemons

- The OS/X Activity Monitor application shows O/S jobs as well as user jobs.
- In Unix an O/S job that listens for traffic on a port is a daemon.
  - `httpd` is the web server (responds to incoming HTTP requests).
  - `ntpd` is the “network time protocol daemon.”

The World Wide Web

- The term **hypertext** refers to a piece of text that includes links to other text.
  - Ideas for cross-referencing text have been around for a long time.
  - Adding links between text files on computers was an active research topic in the 1970s.
- Apple released an application called HyperCard in 1987.
  - Documents were known as “stacks.”
  - A “card” was a single page in a stack.
  - Sound, images, or pieces of text could be links to other cards.
- Example: a stack on Beethoven’s Ninth Symphony.

The World Wide Web

- In the early 1990s Tim Berners-Lee developed the idea of combining hypertext with TCP/IP.
  - His application layer protocol was HTTP (hypertext transfer protocol).
- The **world wide web** was his term for the set of computers that communicated via HTTP.

URL

- Hypertext links on the world wide web are **uniform resource locators (URLs)**:
  - The first part identifies a scheme for locating the resource.
  - The remaining text depends on the scheme.
  - The scheme is not always a protocol (e.g., `file:` means the resource is a local file).

```
http://example.com/authors/Shakespeare/Julius_Caesar.html
```

The scheme is the name of the protocol. The DNS name is the server holding the document.

- Directory path indicating the location of the document within the file system.
URL

- Some other examples of URLs:
  - sftp://ncbi.nih.gov/genomes/Bacteria
  - telnet://teleost.cs.uoregon.edu
  - mailto:conery@cs.uoregon.edu

- Note that strings following the scheme name depend on the scheme:
  - an e-mail address for mailto
  - a file path for file
  - a domain name and file path for sftp and telnet

Web Servers

- HTTP interactions use a client-server model:
  - a user runs a client program (a browser)
  - the client sends requests to a server
  - The server is an application running on the host specified by the DNS portion of the URL
  - By default the request is sent to port 80 on the server
  - If a user knows a server is listening on another port the port number can be included with the URL
    - if I’m running my own special purpose server I can tell it to listen on port 8000
    - to get information from this server I would use a URL with the port number:
      - http://fugu.cs.uoregon.edu:8000/index.html

Web Browsers

- The first web browsers had a command line interface:
  - text-only browsers are still in use (e.g. lynx, curl)
  - A program named Mosaic (1993) was the first browser to use a graphical user interface:
    - images linked with a document were displayed along with the document
    - was a free download from NCSA (Univ of Illinois)
  - Mosaic sparked the growth of WWW and the Internet

HTML

- The messages passed between HTTP clients and servers are text files:
  - How does a browser figure out how to display a page?
    - how does it know which parts are plain text and which are links?
    - how does it know which pieces of text are titles (to be displayed in large bold letters)?
    - where do the images come from?
    - how does the browser know to put some images on the upper left and embed others in the text?
HTML

- A browser expects the documents sent by the server to be formatted in HTML
  - HTML = hypertext markup language
- A "markup language" has special symbols to indicate formatting commands
  - example: I use a language named LaTeX when I write technical papers
  - the $ symbol marks the beginning and end of an equation
  - a backslash marks the beginning of a markup command

output produced by pdflatex

1 Introduction

It takes $k = \lceil \log_2 n \rceil$ bits to represent items.

Anchors

- Markup commands can have arguments
- Inside an anchor \texttt{href} = "http://xxx" means
  - "when this text is clicked have the browser send a request for the URL http://xxx"

Networks and Ruby

- Ruby has a library named Net with methods that help Ruby programs interact with other programs on the Internet
- Example: a program that fetches a set of basketball scores from the NCAA

```ruby
require 'net/http'
true
server = "www.ncaa.com"
"www.ncaa.com"
"/basketball-mens/default.aspx?id=114&date=20080403"
http = Net::HTTP.new(server)
#<Net::HTTP 200 OK readbody=true>
```
Networks and Ruby

You can also write Ruby program that acts as a web server
Start your program and connect it to a free port on your system
   require 'webrick'
   server =
       WEBrick::HTTPServer.new(:DocumentRoot => '.', :Port => 9000)
   server.start

Now you can give your web browser a URL that connects to your new server:
   http://localhost:9000/index.html

We're going use these Ruby libraries in a lab
   we'll experiment with HTML code and page layout commands to develop our own
   web-based electronic voting machine....

Review

The important concepts introduced in these slides:
   - computer systems exchange information using protocols
   - we often talk about a “protocol stack” - layers of software where programs at one
     layer trust lower levels to implement major operations
   - the Internet protocols are known as TCP/IP
   - the four layers in TCP/IP are application, transport, network, and data link
   - most modern local area networks use ethernet or WIFI for the physical connections
   - systems broadcast data to others in the local net

Some terms to understand:
   - client, server
   - IP address, domain name, name server, DHCP
   - hypertext, markup language, HTML