Networking & the Internet

- No computer is an island — it needs to talk to other computers, as well as other devices
- This ability to communicate, thus forming a *network* of machines, is the basis for another important subspecialty in computer science
- Since computer science in general is all about "telling a machine how to do things," the specialty of *networking* is all about "telling a machine how to talk to other machines" — efficiently, accurately, and conveniently

Network Classifications

- Classification by size refers to the physical distances between connected machines: *local area (LAN)*, *metropolitan area (MAN)*, or *wide area (WAN)*
- Classification by "ownership" leads to *open* vs. *closed* or *proprietary* networks
- Classification by *topology* involves the way that devices are connected: *ring*, *bus*, *star*, to name a few
- The Internet is *a* network…which happens to be *very* wide area, based on *open* technologies, and is capable of handling different network topologies

Network Protocols

- In a theme that should be growing familiar now, a key ingredient toward successful networking is the existence of an established, standardized way or set of rules by which devices are to communicate with each other — these are called *protocols*
- Protocols occur at many levels, ranging from just getting data from one place to another (a *data link* level) all the way to how to get explicit tasks done (e.g., sending/receiving e-mail, instant messaging, Web browsing: these are *application*-level protocols)

Combining Networks

- Physical barriers and sheer distance require technologies that can *connect* different networks
  - *Repeaters* extend bus networks, like amplifiers
  - *Bridges* ensure that only signals that need to move across networks are relayed
  - *Switches* can connect > 2 networks
  - Wireless networks can have *relays* that extend the wireless range
- The above devices *merge* physical networks so they appear as one network; other technologies maintain the distinct "identity" of individual networks
  - This "network of networks" is called an *internet*; *the* Internet is "just another" internet, albeit a very large, wide-ranging one
  - *Routers* serve the function of connecting two (or more) distinct networks; in this context, each network is viewed as a *subnet* of the larger one

Communication Models

- Moving beyond the problem of simply getting devices connected to each other, there are also variations in the *roles* these devices play when talking to each other
- A *client/server* model designates one device as a *server* — a provider of facilities such as file sharing, printer sharing, e-mail, databases, Web pages to the other devices on the network, called *clients*
- In a *peer-to-peer* model, the devices are "equals" — they mutually give and receive services; in this model, each device is a *servent*: a "server-client"

Distributed Systems

- The big dream of many domains is the creation of effective *distributed systems* — something that feels like a unified, integrated application, yet is somehow divided or shared by multiple *hosts* on a network
- The Web can be viewed as a distributed *information retrieval* system; multiplayer network games count too
- Frequently, new distributed systems need to be built from the ground up; technologies like *Enterprise Java Beans* and *.NET* seek to establish *frameworks* that make it easier to build new types of distributed systems

The Internet

- As mentioned, the ***I***nternet is sort of "just another" ***i***nternet…though of course it is indeed quite special, as it is the largest internet around on virtually all counts: geographic distribution, number of hosts, etc. (at least as far as we know)
- The "secret sauce" behind the Internet, a technology called *packet switching*, was invented in the 1960s as a collaboration among universities and companies working under DARPA — the Defense Advanced Research Projects Agency

Internet Structure

- The Internet, as a "network of networks," reflects this structure specifically through *domains* — separate subnetworks (which can themselves also be internets) with a distinct identifier
- Each domain connects to the Internet through a specially designated router called a *gateway*; it is literally the portal through which network traffic can enter or leave a domain's internal network
- Entities that let people into the Internet via their domain are known as *Internet service providers* (ISPs)

Internet Addresses

- The core protocol that drives the Internet (and many other internets) is aptly called *IP* for *internet protocol*
- As part of this protocol, every device that is reachable on the Internet is given an *IP address* — this is a 32-bit value frequently written in *dotted decimal notation*: the four bytes written separately in decimal, separated by periods (e.g., 157.242.56.68, 17.254.0.91)
- The first 2-3 bytes of this address identify the network or domain to which the host belongs, while the remainder identifies the specific host in that network

- Domain identifiers are distributed, managed, and controlled by ICANN, the *Internet Corporation for Assigned Names and Numbers* (headquartered in… Marina del Rey, California!); to get a domain, an entity contacts one of the official *registrars* that have been designated by ICANN for this purpose
- Of course, numbers aren't the most convenient thing for us humans, so each numeric domain is also associated with a *domain* or *host name* — a dot-separated, human-readable identifier such as "lmu.edu"
  - While IP addresses go from generic to specific, domain names go from specific to generic; the last "dot term" in a domain name is a *top-level domain* or TLD, again managed by ICANN (e.g., *.com*, *.edu*, *.tv*, etc.)
  - No single computer can hold every name of every host and domain on the Internet — thus, designated *domain name servers* exist that act as "dictionaries" for converting names into addresses (not surprisingly, this is called a "DNS lookup"); of course, communicating with this *domain name system* requires adherence to the *DNS protocol*

Application-Level Protocols

- So we've said that one can't really get anything done on a network unless we establish a protocol
- There are tons of them…here's a sampling of the ones associated with common Internet applications:
  - *e-mail* involves the electronic transmission of messages (but you knew that already); more than one protocol is actually associated with e-mail
    - *POP3* and *IMAP* are two protocols by which e-mail can be accessed from a *mail server*
    - *SMTP* is a protocol for transferring ("sending") e-mail from one machine to another, ultimately finding its way to the e-mail addressee
  - *File transfer* involves the transmission of files from one host to another; again, many protocols are associated with this activity
    - *FTP* is one of the oldest file transfer protocols, and actually forms the basis of the *HTTP* protocol used by the Web
    - FTP isn't secure, so new alternatives such as *SFTP* and *SCP* exist
    - Many operating systems have their own file transfer protocols: Windows had *SMB*, which eventually evolved into *CIFS*; Mac OS has *AFP*
  - *Remote login* protocols enable users to "go to" a different host and perform activities there; this is like having a *remote shell* to the OS of another machine
    - Text-based protocols (thus providing text-based shells) include *telnet* and *SSH*; SSH is preferred today because it is more secure
    - Graphical remote login protocols are also available (thus providing a GUI shell to a remote user), such as *VNC* and OS-specific protocols such as Microsoft's *RDC* and Apple's *Remote Desktop*

The Worldwide Web

- Of course, the "BPON" — big protocols on the net — are the ones associated with the Worldwide Web: primarily *HTTP* and its secure cousin, *HTTPS*
- The Web needs no introduction; however some of the technical terms associated with it might:
  - The Web is essentially a *content delivery system*; its key idea was the use of *hyperlinks* to allow one *Web page* to refer to, and possibly retrieve, any another page on the Web
  - Content is identified using a human-readable (though not always understandable!) piece of text called a *uniform resource locator* (URL); hyperlinks are URLs that are embedded in a Web page, and clicking on them retrieves the page (or content) specified by that URL
  - Hosts that provide Web pages are *Web servers*; the related set of pages delivered by one or more particular Web servers comprise a *Web site*

The Hypertext Transfer Protocol (HTTP)

- HTTP is conceptually simple to understand: first, the Web client (frequently, but not always, a *Web browser*) is given the URL of the content to be retrieved
- Part of the URL specifies the *host name* of the Web server that provides this content; the client contacts this host and passes the URL to its Web server
- The Web server locates or constructs the content specified by the URL, then sends it back to the client
  - Of course, things can go wrong — for example, the Web server might not have any content that corresponds to the given URL, resulting in the notorious "404 error"

Web Content

- The information delivered by a Web server can really be anything: images, movies, audio, text, and anything else that can take digital form
- However, the Web draws most of its power from a very particular type of content — the *Web page*
- A Web page is a piece of text that conforms to a language (*HTML* or *hypertext markup language*); specifically, today, this language is *XHTML*, where the "X" stands for *XML*, which is a general-purpose language for representing all kinds of information

- XHTML uses *tags* to indicate the role or meaning of a particular sequence of text in a Web page; for example, the *title* tag indicates the text to display as the title of the Web browser's window
- *Anchor* tags (simply written as *a*) indicate hyperlinks; other tags run the gamut of specifying images (*img*) to describing structured data (lists via *ul* and *ol*; *table*) to facilitating user input (*form*, *input*, *textarea*)
- A fully rendered Web page is almost never a single XHTML document; before the page can be displayed in all of its glory, the XHTML document typically refers to other content (also via URL) that specify how the document is to be displayed (*CSS*) as well as how it should behave…and this is where JavaScript comes in!

JavaScript and Web Pages

- A JavaScript program running inside a Web browser has access to a special built-in object: the *document*
- The *document* object represents the *Web page* that contains or invokes the JavaScript program; using *document*, JavaScript programs can *look at and change* the content of that Web page!
- This is the secret to JavaScript input and output (beyond the simplistic *prompt* and *alert*) — JavaScript programs communicate with users *through the Web page that contains them*

Dynamic Web Pages

- Notice how we have said nothing about *how* a Web server determines the content that it is to deliver when presented with a URL
- One way to do this is to make every URL correspond to some file on the Web server; when the server receives a URL, it simply delivers the file for that URL
- However, the power of the Web lies in the fact that URLs don't *have* to always be files — frequently a URL simply provides information that a Web server uses to *dynamically construct* the Web page to be delivered

Details, Details, Details

- As with all of the topics in this course, this handout barely scratches the surface of computer science's networking subspecialty
- A deeper study of networking in general involves *queuing theory* and *graph theory*
- The complete structure of modern networks is based on the seven layers of the *OSI reference model*
- Secure communications require the development of *encryption* and *authentication* algorithms

- Network applications beyond e-mail, file transfer, remote login, and the Web include instant messaging, audio or video streaming, database connectivity, computer-supported collaborative work (CSCW), blogging, and many, many more
- Web technologies include the full collection of tags available in XHTML, presentation via *cascading style sheets* (CSS), interactivity using JavaScript and XML (AJAX), and *server-side technologies* such as *PHP*, *Java Server Pages*, *Active Server Pages*, and others
- The list — and the learning — goes on and on, and while this is true of all disciplines, networking in particular has a unique sense or urgency and impact, just by the degree to which we depend on it today