The Internet: Protocols and Security
Announcements

- Monday: PS 11
- Monday: Lab Exam 2
On Wednesday:

- Protocols
- history
Addressing
IP Addresses

• Each computer on the Internet is assigned an IP Address consisting of four numbers between 0 and 255 inclusive

  ____ · ____ · ____ · ____

Example: 128. 2. 13. 163

Data sent on the Internet must always be sent to some IP address

• How many bits per address? How many computers can be on the Internet at the same time?

IP stands for “Internet Protocol”
Where do IP addresses come from?

- An IP address isn’t part of a computer!
- Groups of addresses are allotted to various organizations by IANA (Internet Assigned Numbers Authority)

These organizations assign addresses to computers.

- Static versus dynamic assignments
  - static for important server machines
  - dynamic for others
What does an IP address “say”

- Identifies a particular machine at a particular time
- Identifies (somewhat vague) geographic location based on organization that “owns” it

What it doesn’t say
- who is using the machine to do what
- what kind of machine it is
packet switching

going from here to there: basic transportation mechanism
The path from “here” to “there”

- For now, think of sending a message (group of bits) from one machine to another through the Internet
- We attach the source and destination IP addresses to the message
- “The Internet” gets it from source to destination
  - but how? using packet switching
Design Decisions

- No limit on message size
- Flexible and robust delivery mechanism
Circuit Switching: the road not taken

- Two network nodes (e.g. phones) establish a **dedicated connection** via one or more switching stations.
Circuit switching

- Advantages
  - reliable
  - uninterruptible
  - simple to understand

- Disadvantages
  - costly
  - inflexible
  - wasteful
  - hard to expand
Packet Switching

- Two network nodes (e.g. computers) communicate by **breaking the message up into small packets**
  - each packet sent separately
  - with a serial number and a destination address.

- Routers forward packets toward destination
  - table stored in router tells it which neighbor to send packet to, based on IP address of destination

- Packets may be received at the destination in any order
  - may get lost (and retransmitted)
  - serial numbers used to put packets back into order at the destination
Packet Switching

Router

ISP

Router

Router

ISP

Router
Routing and Internet structure

- Core provides transport services to edges
  - routers and gateways forward packets
  - Internet Service Providers (ISPs) provide data transmission media (fiber optic etc.)
  - domain name servers (DNS) provide directory of host names (more on this next time)

- Edges provide the services we humans use
  - individual users, “hosts”
  - private networks (corporate, educational, government…)
  - business, government, nonprofit services
end-to-end principle

Internet article of faith
Core architectural guideline

- Idea: routers should stick to getting data quickly from its source to its destination!
  - they can be fast and stupid

- Everything else is responsibility of edges, e.g.
  - error detection and recovery
  - confidentiality via encryption
  - ...

Benefits of End-to-end

- Speed and flexibility

- Support for innovation: routers need know nothing about apps using their services

- Equality of uses: routers can’t discriminate based on type of communication (net neutrality)
Controversies

- End-to-end principle under pressure
  - because of technical developments
    - video streaming requires high-quality delivery service
  - because of social and economic developments
    - lack of trust because of bad actors on the Internet
    - profit opportunities for ISPs
    - corporate and government monitoring of communications
Internet Society: a range of partners from non-profit agencies, local and global NGOs, academia, technologists, local councils, federal policy and decision makers, business (www.isoc.org)

Internet Service Providers (ISPs) regulated in the USA by the Federal Communications Commission (FCC)
network neutrality

current issue
Net neutrality principle

- All communications are treated equally
  - regardless of source, destination, or type
Where is there net neutrality?

- In principle, most places

- But some governments already censor or otherwise control the Internet within their borders
Net neutrality and the FCC
(grossly oversimplified)

- Historically the FCC prohibited ISPs from violating net neutrality
  - 2014: Federal court ruled FCC had no authority for their then-current regulations because ISPs were not “common carriers”.

- Recently
  - February 2015: FCC voted (on party lines) to enforce net neutrality based on a different legal authority.
  - Verizon, Comcast, etc. unhappy
  - Facebook, Netflix, Google, etc. happy
The Internet and Python
# mail (run where there is a local mail server)

```python
import smtplib
from email.mime.text import MIMEText

def mail_demo()
    msg = MIMEText('Give me an A!')
    msg['Subject'] = 'My grade'
    msg['From'] = 'student@example.org'
    msg['To'] = 'jmfrye@andrew.cmu.edu'
    server = smtplib.SMTP('localhost')
    server.send_message(msg)
    server.quit()
```
# web (run this wherever)

from urllib.request import urlopen

def web_demo() :
    page = urlopen('http://www.cs.cmu.edu/~15110')
    print("Opened URL ", page.geturl())
    print("Contents:")
    for line in page :
        print(line.decode('ISO-8859-1'))
Higher Protocols
“Higher” and “lower” level protocols

- Network protocols are organized in layers

- IP packet delivery is the lowest *layer* of the Internet protocol *stack*

- “Higher” layers use services provided by “lower” layers

- Each layer is responsible for a type of service
Layers of the Internet ("higher" to "lower")

- **Application Layer** provides services to human beings
  - e.g. browser, email client, Skype

- **Transport Layer** provides services to applications
  - converts between application messages and IP packets
  - figures out which application to deliver a message to
  - possibly detects and corrects delivery errors

- **Internet Layer** provides services to transport layer
  - determines next “hop” for a packet and sends it there

- **Link Layer** provides services to internet layer
  - physically converts between signals and bits
Example: Layering the Web

CLIENT MACHINE
ask for a web page
request connection
best-effort packets
physical data transport

HTTP

TCP

IP

(Various)

SERVER MACHINE
send a web page
acknowledge request
best-effort packets
physical data transport
Transport Layer
from IP packets to application messages
Transport Layer

- Splits application messages into IP packets and maps applications to *port number*
  - IP address identifies machine, but port number identifies an application operating on that machine (web, email, etc.)

- Transport Control Protocol (TCP)
  - Creates a *reliable* bi-directional stream (source address/port and destination address/port)

- User Datagram Protocol (UDP)
  - Creates a single one-way message to a remote application (destination address/port)
  - used for voice, video, DNS lookup, ...
Transport Layer

Reliable TCP connection

virtual circuit

unreliable network delivery

actual packet-switched delivery
Suppose A and B are the TCP programs of two computers.

- An application asks A to send a message to an application at B.
- A breaks the message into several packets.
  - Each packet includes parity information, so B can check it for accuracy.
  - Packets are sent via IP.
- B receives the packets.
  - If B is missing a packet or receives a corrupt packet, it can request retransmission.
  - If the packet is OK, B sends an acknowledgement.
- If A doesn’t get an acknowledgement, it will retransmit.
- B assembles the incoming packets in order and provides the message to the appropriate application.
Network Address Translation (NAT)
Network Address Translation (NAT)

• Used to accommodate more users on the Internet, security, and administration.

• The gateway assigns an additional code called a port for each user. Packets are tagged with the port.

• The gateway knows where to route the messages on the private network, but all messages from that private network share the same single IP address.
Domain names from 98.139.183.24 to yahoo.com
URL: http://www.andrew.cmu.edu/user/nbier/15110/index.html

Email address: nbier@andrew.cmu.edu

We don’t want IP addresses in our URLs or email addresses—why not?

Domain Name Service (DNS) translates names to addresses
Problem: so many names! How to make lookup fast?

Solution: hierarchy of name servers

- Each machine knows a name server, which knows how to find a root name server
- root name servers know DNS servers for each top-level domain (e.g., "edu", "com", "net", "uk", "ru")
- top-level domain servers know DNS servers for each second-level domain (e.g., "cmu.edu", "co.uk")
- second-level domain servers know each host directly in their domain (e.g., "www.cmu.edu") and DNS servers for each third-level domain (e.g., "andrew.cmu.edu")
DNS Hierarchy (fragment)
DNS Lookup

"Where's www.wikipedia.org?"

1. "Try 204.74.112.1"
2. "Try 207.142.131.234"
3. "It's at xxx.xxx.xxx.xxx"

root nameserver
198.41.0.4

org. nameserver
204.74.112.1

wikipedia.org. nameserver
207.142.131.234

DNS Recurser

? !

Client-server architectures

web, mail, streaming video, and more
Client-server Architectures

CLIENTS

SERVER (e.g. www.google.com)
Client-server Architectures

- Architecture: an organizing principle for a computing system
- Most common architecture for Internet applications: client-server
- Server is always on, waiting for requests
  - server software (e.g. Apache) tells TCP (transport layer software) on its own machine “please listen for messages with port number 80”
  - client software (e.g. Chrome) tells TCP “please send this message to machine xxx.xxx.xxx.xxx with port number 80”
  - TCP gives message to IP, which sends it through internet to server machine; IP at server machine delivers to TCP at server machine
  - TCP at the server machine delivers the message to Apache
World Wide Web = html + http

html = HyperText Markup Language, an encoding
- tells what a page should look like and
- what other pages it links to

http = HyperText Transfer Protocol
- agreement on how client and server interact
HTML: an encoding

Example: using your favorite plain-text editor create the following text file:

```html
<html>
<head>
<title>15110, Summer '16, Example web page</title>
</head>
<body>
<h1>Hello World!</h1>
</body>
</html>
```

In a browser type its name in the address bar, e.g. file:///Users/pennyanderson/CMU/110/week11/example1.html
Now add


save as example2.html

and load

Code for getting information across the Internet
HTTP: hypertext transfer protocol

- Protocol for communication between web client application (e.g. Chrome, Safari, IE, Firefox) and web server application (e.g. Apache)

- Agreement on how to ask for a web page, how to send data entered into a form, how to report errors (codes like 404 not found), etc.
Uniform Resource Locators

• A Web page is identified by a Uniform Resource Locator (URL)

  protocol://host address/page

• A URL

  http://www.cs.cmu.edu/~15110/index.html

  Protocol to use
Overview of web page delivery

1. Web browser (client) translates name of the server to an IP address (e.g. 128.2.217.13) (using DNS)
2. Establishes a TCP connection to 128.2.217.13 port 80
3. Constructs a message
   
   GET /~15110/index.html HTTP/1.1
4. Sends the message using TCP/IP
5. Web server locates the page and sends it using services of TCP/IP
6. The connection is terminated
Layers and Encapsulation

- Message: “GET /~15110/index.html HTTP/1.1”

- TCP segment:
  control information including sequence number, so-called port number for web server;
  + message

- IP packet:
  control info including source address, destination address, fragment sequencing information + TCP segment

Separate responsibilities

Request/get web page

Connect client and server reliably

Best-effort packet switching
Applications communicate on the Internet via application protocols like
- HTTP for the web
- SMTP for email
- RTSP for streaming media

Application protocols rely on
- Domain Name Servers for name translation, and
- transport protocols like
  - TCP for reliable two-way connections
  - UDP for one-way “datagrams”

Transport protocols rely on IP for packet delivery
Next Time:
Be prepared to discuss readings

image: http://xkcd.com/538/