

Administrivia

- Memorial Day: next Monday
  - no class
- Hw 3
  - published Sunday morning
  - revised Sunday evening
    - thanks to "early adapters"
- Ass 4 lab: cancelled
  - more time to focus on Ass 4
  - distributed application

## Previously on CS553

- ✓ - kernel is used
- ✓ - processes & inter-process communication
- ✓ - concurrency
- ✓ - filesystems
- - networking / distributed systems

- networking: 2 or more computers sharing data

- problems created by networking: "fallacies of distributed computing"

- network topologies

network of networks: inter-network (internet)

- switches, routers, gateways

- ethernet packets (frames): metadata + payload  
↳ src, dest, ...

- OSI: 7 layers of protocol

- Ethernet / IP / TCP / UDP / application

↳ layers mapped roughly onto OSI framework

## Previously ... (cont.)

- each layer provides higher level of abstraction

- Data link - point-to-point communication
- Transport - routing
- Session - reliable connection (circuit)
- Application - high-level protocol (e.g. HTTP)

## Internet Protocol (IP) (cont.)

- "Best Effort" delivery of IP packets
  - unreliable network ("work around damage")
    - packets may be
      - lost
      - delayed
      - arrive out of order
      - duplicate

- TCP/UDP: protocols that sit on top of IP

- UDP: unreliable connection-less

- TCP: reliable connection-oriented

} receive out-of-order packets  
detects duplicates  
request retransmission of  
lost packets

protocols on top of TCP

FTP

HTTP

protocols on top of UDP

TFTP

audio/video streaming

## Hosts & Ports

IPv4 address:

{ 4 numbers 0..255  
separated by period

⇒ text representation

in protocols: 32-bit number (4 bytes)

\* running out of addresses

- like phone #s

- 640k ought to be enough for everyone

⇒ IPv6: 128-bit addresses (16 bytes)

"this time for sure"

- original idea: class A, B, C networks

⇒ netmask

CIDR - classless inter-domain routing

## Ports

- TCP/UDP
- 16-bit numbers
  - Service identifier
  - like global number for shared memory

/etc/services: text file containing port #s for well-known services

- 22 ssh
- 25 smtp
- 80 http

- use alt port #s for internal services
  - e.g. monitoring, REST
  - 8080, 8888, 8000 - 9000
- distributed lock service (eg Zookeeper) as "name service" for <<service ID>>

- 1st 1024 ports: "privileged"
  - unix: can only be accessed by root

\* did I mention root?

## DNS

IPv4 address: 32-bit number

- "convenient" text representation is not very convenient

⇒ database to map symbolic name to IP address  
/etc/hosts file

- not very scalable

⇒ Domain Name System

\* interview question (esp. Dev Ops):

how does DNS work?

in detail

- DNS: "registered" names (name registrar, e.g. Go Daddy)

- very large DB, frequent updates

- caching for repeated lookups (TTL: how long to cache)

- hierarchical lookup

foo.bar.com.

- o COM TLD: who is responsible for bar.com

- o name server returns foo.bar.com