ICS 451: Course Review

- Network Programming
- Application Layer: HTTP, DNS, email
  - client/server organization, data encodings
- Transport Layer: TCP, UDP
  - connections, reliable transmission, adaptive timers, flow and congestion control
- Network Layer: IP and routing
- Data Link Layer: Ethernet and 802.11, framing, Medium Access Control and MAC addresses
- Principles of networking

Network Programming

- sockets, IP, addresses, ports, address families
  - SOCK_STREAM and SOCK_DGRAM
    - a stream has no message boundaries
  - sockaddr, sockaddr_in, sockaddr_in6
    - sockaddr * usually points to sockaddr_in/6
- in C, the OS doesn't know the length of your buffer, so it must be given explicitly

  content length <= buffer length

    - content length specified on send/sendto
    - content length returned by recv/recvfrom
- connect, or listen, bind, accept

networking concepts: layers

- each layer provides services to the layer above
  - using the services of the layer below
- logically, each layer communicates with the same layer on a different machine
  - actually, each layer sends and receives using the layer below
    - except at the physical layer!
- layering simplifies protocol definition and implementation
  - layering almost requires multithreading

standard layers

- application layer
- (presentation layer)
- (session layer)
- transport layer
- network layer
- data link layer
- physical layer

Application Layer

- end-to-end communication
  - usually requires TCP and DNS
    - DNS too is in the application layer
- does everything not provided by lower layers
  - often includes security and encoding of different types of data
- headers are often human-readable
  - e.g. http, email, ftp

HTTP

- client-server system over TCP
- DNS in URL also used to select virtual server
- header is in plain text
  - all but first line have field: value
  - request has METHOD request HTTP/version
  - reply has HTTP/version code explanation
  - header ends with emtpy line
- many different content types, including html

email

- complex system evolved over the years
  - always uses TCP
  - in addition, also has application-level retransmissions every few hours
- servers and clients not clearly separate
  - email agent on host may be independent of email program used to write and read email
  - webmail server is an email client
- text-based protocol

DNS

- client-server architecture over UDP (port 53)
  - recursive queries: server returns answer
  - iterative queries: server may return another server
- binary encoding of packet
  - length+content encoding of domain names
  - pointers help compress packets
- essential for the modern internet
  - including www and email

Transport Layer

- end-to-end (not hop-by-hop) service
- typically TCP or UDP
- UDP provides ports and datagrams
- TCP provides ports and reliable byte streams

TCP

- connection setup (3-way handshake)
  - header bits: SYN, FIN, ACK, RST
- sequence and ack numbers
  - ack number is seq+length of payload
    - i.e. “next expected sequence number”
  - retransmission after timeout
  - out-of-order packets result in duplicate acks
  - Nagle algorithm
- flow control window
- TCB

TCP congestion control

- congestion “window” on sender
- AIMD – additive increase, multiplicative decrease
- adaptive timer
  - Binary Exponential Backoff on timer expiration
- slow start
- limits on achievable performance
  - at most, ¾ of the bottleneck bandwidth

Network Layer

- end-to-end
  - over hop-by-hop data-link layer
- almost only IP: IPv4, IPv6
- main problem solved: how to get data across various networks
- accomplished by having routing table:
  - for each destination, interface and next hop

Internet Protocol

- best effort:
  - packets can be lost, corrupted, duplicated, reordered, etc.
- addresses interpreted with respect to netmask
  - network part of the address is used in routing
  - the entire address is used for final delivery
  - network mask has initial "1" bits identifying network part of the address
- Time To Live or Hop Limit limit packet lifetime
  - both measure hops (usually) or seconds

IPv4 details

- 32-bit addresses and netmasks
- fragmentation supported in routers
  - Don't Fragment (DF) bit set in TCP segments
  - More Fragments (MF) bit set on all but the last fragment
- hop-by-hop header checksum
  - changes at each hop as TTL changes
- 8-bit protocol number, 16-bit packet length
- 20-byte basic header, may have options

IPv6 details

- 128-bit addresses and netmasks
- end-to-end fragmentation only
- no header checksum
  - upper layers now required to use checksum
  - lower layers normally use CRC
- 8-bit next header, 16-bit payload length
- extension headers may follow 40-byte header

IP routing overview

- automatic way of setting up routing tables
  - uses interface IP addresses (configured)
  - uses local broadcasts
- IGPs: find optimal routes
  - for some idea of optimal, e.g. least hops
- EGP (BGP): find routes that satisfy policy
  - among the best routes for policy, choose routes that cross the fewest Autonomous Systems

IP IGPs

- RIP: distance vector
  - with infinity of 16
  - split horizon with poisoned reverse
  - RIPv2 supports netmasks
- OSPF: link state
  - flooding to deliver link state to all routers
  - Dijkstra's shortest path algorithm to compute routes
  - areas allow partitioning of information
  - backbone area must connect to all other areas

related protocols and systems

- ICMP
  - ping: echo request and reply
  - ICMP error messages
  - traceroute: send with low (but increasing) TTL, listen for ICMP error messages
- DHCP
- NAT
- Firewall

Data Link Layer

- Framing problem: how to tell where frame starts and ends
  - reserve special symbols for frame start/end
  - if symbols can occur in the data, they must be escaped by bit-stuffing or byte-stuffing
- Medium Access Control problem: how to tell when a transmission will not collide with others
  - Aloha: just send when ready, retransmit in case of collision
  - but peak performance for Aloha is low, ~18%

Ethernet

- Carrier Sense Multiple Access with Collision Detection: CSMA/CD
  - carrier sense: do not send if someone else is sending
  - collision detection: if a collision occurs, everyone knows it and discards a packet
    - limits the size of the network
    - not needed in a full duplex network where both sides send simultaneously on point-to-pint links
- high throughput (but not 100%), low latency
- requires cables

Ethernet systems

- hubs: forward each bit, or jamming signal
  - broadcast only, low latency
- learning switches: forward each packet
  - break up the collision domain
  - only broadcast when needed
- learning switches with STP
  - distributed root election using uniqueness of MAC addresses
  - shortest path to root is enabled, all else blocked
  - allows redundant links

802.11/WiFi

- Carrier Detection Multiple Access with Collision Avoidance: CDMA/CA
  - carrier detection: send only if nobody else is
  - collision avoidance: do not send if RTS or CTS was heard
- ack required on wireless medium
- ad-hoc mode or infrastructure mode
- infrastructure mode based on Wireless Access Points (WAPs)
  - 802.11 supports exchanging packets with 802.3

some principles of networking

- what can go wrong, will go wrong sometimes
- good models (such as client/server) are useful
- layering
- can have reliability or real-time, not both
- window must be greater than bandwidth-delay product, or throughput will be limited by window
- performance matters
- security is hard but not impossible
  - security by obscurity only OK if really secure