Midterm Review

1. Chapter 1
   1. What is the internet:
      * Millions of connected computing devices running network apps
      * Communication links
        + Fiber optic, radio, satellite
        + Transmission rate = bandwidth
      * Packet switches: routers and switches
      * Internet standards
        + RFC: **R**equest **F**or **C**omments
        + IETF: **I**nternet **E**ngineering **T**ask **F**orce
   2. Network edge
      * Network types(end systems):
        + Mobile network
        + Home network
        + Institutional network
      * DSL(**D**igital **S**ubscriber **L**ine)
      * DSLAM(**D**igital **S**ubscriber **L**ine **A**ccess **M**ultiplexer)
      * CMTS(**C**able **M**odem **T**ermination **S**ystem)
      * HFC(**H**yper **F**iber **C**oax)
      * DSL vs. Cable
        + DSL uses the telephone line to transport voice and data.
        + Cable has dedicated channels frequencies for each, first 6 for video, 7-8 data, 9 controller.
        + DSL needs a DSLAM in the central office.
        + Cable needs a CMTS on the head end.
        + Cable uses a HFC.
      * Wireless
        + WLANs(**W**ireless **L**ocal **A**rea **N**etwork)
          - Wi-Fi
        + WWAN(**W**ireless **W**ide **A**rea **N**etwork)
          - 3G
          - 4G
          - LTE
      * Packets
        + L = length of the packet
        + R = rate of the transmission link, link capacity, link bandwidth
        + Transmission delay = L/R = bits/(bits/second) = second
      * Physical Media
        + TP(**T**wisted **P**air)(copper)
          - Category 5: 100 Mbps
          - Category 5e: 1 Gbps
          - Category 6: 10 Gbps
        + HFC(copper)
        + Fiber Optic Cable(fiber glass)
          - Every light pulse carries 1 bit
          - 10-100 Gbps
          - Low error rate, immune to electromagnetic noise.
        + Radio
          - Terrestrial microwaves
          - WLAN
          - WWAN
          - Satellite
   3. Network core
      * Has no idea what type of data it is transmitting
      * FDM(**F**requency **D**ivision **M**ultiplexing, circuit switching) vs TDM(**T**ime **D**ivision **M**ultiplexing, packet switching)
      * Internet structure:
        + Access net points(ISP)
        + Regional points ISP
        + ISP(**I**nternet **S**ervice **P**rovider)
        + IXP(**I**nternet **E**xchange **P**oint)
        + Content provider network
   4. Sources of packet delay
      * Nodal processing: check but errors, determine output link
      * Queueing delay
        + R = bandwidth
        + L = packet length
        + a = average packet arrival time(speed)
        + La/R ~ 0, small to no delay
        + La/R => 1, large delay
        + La/R > 1, more work than the receiver can handle, infinite delay.
      * Transmission delay
      * Propagation delay
        + d = length of physical link
        + s = propagation speed
        + d/s
      * **Try traceroute**
      * Full queues will drop arriving packets
      * Throughput: transfer rate between sender and receiver.
        + Instantaneous: rate at point in time
        + Average: the average over a relatively longer time span
   5. Protocol Layers:
      * Internet Protocol Stack(TCP/IP)
        + Application
        + Transport: data transfer, UDP, TCP
        + Network: routing datagrams, IP, routing protocols
        + Link: data transfer to networking elements
        + Physical
      * ISO(**I**nternational **O**rganization for **S**tandardization)/OSI(**O**pen **S**ystem **I**nterconnection model)
        + Presentation: decrypt, encrypt, and compress data
        + Session: synchronization, check pointing, recovery??
      * Encapsulation
        + Switch only has the link and physical layer
        + Router only has the network, link, and physical layer
   6. Network Security
      * Malware
        + Virus: self-replicating infection by receiving/executing object
        + Worm: self-replicating infection by passively receiving object that gets itself executed
      * Spyware Malware
      * Infected hors can be enrolled in a botnet, DDoS (**D**istributed **D**enial **o**f **S**ervice)
      * DDoS using machines registered with botnet, overwhelm the targets servers with traffic
      * Packet sniffing:
        + Wireshark
      * IP spoofing: send packets with false source addresses
   7. Internet history
      * 1983 – deployment of TCP/IP
      * 1982 – smtp email protocol defined
      * 1983 – DNS defined for name-to-IP-address translation
      * 1988 – TCP congestion control
      * 1990 – HTML, HTTP, commercialization of the web
2. Chapter 2
   1. Principles of network applications
      * Server vs Client
        + Always-on host??
        + Permanent IP address
        + Data centers for scaling
        + May have a dynamic IP address
        + Communicates with the server
        + Does not communicate directly with another client
        + Maybe intermittently connected.
      * P2P
      * Sockets
        + Process sends/receives messages from its socket
        + Sockets are the bridge from the application layer to the transport layer
      * Process
        + Every process has an identifier which contains the IP Address, and the port number
      * App-layer defines: Type of message exchange, message syntax, message semantics, rules, open protocols, and proprietary protocols
      * What transport server does an app need?
        + Data integrity: no toleration for data loss
        + Throughout: some apps require a minimum throughput
        + Timing: require low delay
        + Security: encryption, decryption
      * TCP(**T**ransmission **C**ontrol **P**rotocol) vs UDP(**U**ser **D**atagram **P**rotocol)
        + TCP
          - Reliable transport
          - Flow control: sender wont overwhelm the receiver
          - Congestion control: does not overwhelm the network(internet)
          - Does not provide: timing and minimum throughput guarantee, security
          - Connection oriented: setup required between the client and server
        + UDP
          - Unreliable data transfer
          - Does not provide: reliability, flow control, congestion control, timing, throughput, security, and connection setup
      * Securing TCP
        + TCP & UDP do not have encryption
        + SSL(**S**ecure **S**ockets **L**ayer)
          - Encrypts the TCP connection
          - Data integrity
          - End-point authentication
   2. Web and HTTP
      * HTTP(**H**ypertext **T**ransfer **P**rotocol)
      * HTTP uses TCP
      * Uses port 80
      * HTTP messages are application layer messages
      * HTTP is stateless, doesn’t remember
      * HTTP connections
        + Non-persistent
          - Response time: 2RTT + file transmission time
          - Requires 2RTT for each object
          - OS overhead for each TCP connection
        + Persistent
          - Response time: RTT + all file transmissions
      * RTT(**R**ound **T**rip **T**ime)
      * HTTP messages
        + Request
          - Chapter 2 – page 27
        + Response
          - Chapter 2 – page 31
        + Uploading form input
          - Post: leaves the value in the header
          - URL: has a random URL
        + HTTP/1.0: GET, POST, HEAD
        + HTTP/1.1: GET, POST, HEAD, PUT, DELETE
        + Response status codes:
          - 200 OK
          - 301 moved permanently, provides new location
          - 400 Bad request
          - 404 Not found
          - 505 HTTP version not supported
      * Cookies
        + Keep state
        + What are cookies used for:
          - Authorization
          - Shopping carts
          - Recommendations
          - User session state
        + Proxy server: my cache server
        + Cache can be the client and the server, it provides us with information, server, requests information from original server
        + Why cache?
          - Reduce response time
          - Reduce traffic
          - Helps poor content providers, provide content
        + Conditional GET, used for caching. Uses the field, if-modified-since.
          - If NOT modified then the return is 304 Not Modified
   3. FTP
      * RFC 959, port 21
      * Uses TCP
      * Maintains state
      * Uses “out-of-band” control connection, establishes another data connection every time it needs to transfer a file.
      * Uses ASCII commands
        + USER
        + PASS
        + LIST
        + RETR
        + STOR
      * Return codes are similar to HTTP
        + 331 Username OK
        + 125 data connection already open
        + 425 cannot open data connection
        + 452 error writing file
   4. E-mail
      * Three major components
        + User agents: compose, read, edit messages, outlook
        + Mails servers: contains message queue
        + SMTP(**S**imple **M**ail **T**ransfer **P**rotocol)
      * Uses TCP, port 25, RFC: 2821
      * Direct transfer between servers
      * Three phases for transfer:
        + Handshake
        + Transfer
        + Closure
      * Commands are similar to HTTP and FTP
        + HELO, MAIL FROM, RCPT TO, DATA, and QUIT
        + 250 Message accepted for delivery
        + 221 closing connection
      * Message must be 7-bit ASCII
      * User persistent connection
   5. Mail message format
      * RFC: 822, standard for text message format
      * Header contains: to, from and subject fields
      * POP3(**P**ost **O**ffice **P**rotocol), RFC: 1939
        + Example: page 58
        + Stateless
      * IMAP(**I**nternet **M**ail **A**ccess **P**rotocol), RFC: 1730
        + Allows user to structure the email folder
        + Keeps state.
      * HTTP
   6. DNS(**D**omain **N**ame **S**ystem)
      * Distributed hierarchy of many name servers
      * Hosts communicate with name servers to resolve names.
      * DNS services
        + Hostname to IP address translation
        + Host aliasing
        + Load distribution
        + Mail server aliasing
      * Why not centralize DSN: single point failure, traffic, maintenance, distance
      * 13 DNS root name servers worldwide
      * DNS Servers
        + Root DNS server
        + TLD(**T**op **L**evel **D**omain) DNS server
        + Authoritative DNS severs
        + Local DNS name server(default name server)
          - Each ISP has one
          - Acts as a proxy
      * DNS name resolution
        + Iterative query
        + Recursive query
      * DNS servers cache mappings with a TTL (**T**ime **t**o **L**ive) timer.
      * IETF proposed to update cache mappings in RFC 2136?
      * DNS RR(**R**esource **R**ecord)
        + Format: (name, value, type, TTL)
        + Types:
          - Name: hostname, value: IP address
          - CNAME: name: alias, value canonical name
          - NS: name: domain, value: hostname of authoritative name server for this domain
          - MX: value: name of mail server, associated with name
      * DNS protocol messages
        + Types are query and replay
        + Page 71
        + 16 bit identification
      * Inserting records into DNS
        + Register name at DNS registrar
        + Provide IP ADDRESSES OF authoritative name server
        + Registrar inserts two RRs into TLD, NS and A
      * Attacking DNS
        + DDoS attacks
          - Bombard root servers with traffic, does not work
          - Bombard TLD servers with traffic, more dangerous
        + Redirect attacks
          - Man-in-middle: intercept queries
        + Exploit DNS for DDoS: send queries with spoofed source address
      * P2P Applications