4C03 Review

Networking

- The future of *Cloud Computing*
- Internet Network Layer: routing protocols (path selection) -> forwarding table
- IPv4 vs. IPv6 (IP Datagram header is 20 bytes overhead, TCP/UDP is another 20 bytes)
    - IPv4 uses 32 bit format, Datagram contains type of data, datagram length, TTL (time-to-live), 32 bit source IP and 32 bit destination IP, and fragmentation offset, options and data (TCP or UDP segments).
    - Large IP datagrams are fragmented into several datagrams and reassembled at final destination using header bits and frag offset (IPv4).
        - MTU is max transfer size (for one datagram)
    - IP-address: 32 bit identifier for host, router interface (IPv4)
        - <8 bits>.<8 bits>.<8 bits>.<8 bits>
- *Subnet* is how a device interfaces with other devices with the same subnet part of IP address, allows physical reach without intervening router.
    - Subnet mask 223.1.3.0/24 means, 24 bit subnet mask (all devices in that subnet will have 223.1.3.x)
    - CIDR classless inter-domain routing
        - Network gets subnet from IP address using ISP address
- Host gets IP address either from hard-coded file or DHCP (Dynamic Host Configuration Protocol)
    - DHCP allows host to obtain IP dynamically from network server when it joins network
    - Host (DHCP discover) -> DHCP server (DHCP offer) -> host requests IP (DHCP request) -> DHCP server sends address (DHCP ACK) [client now knows its IP, name, IP address of DNS, IP of first hop router]
- NAT (Network address translation)
    - All datagrams leaving local network (into the rest of the internet) have the same single source NAT IP address (each connected host has a different source port number), all devices in the local network belong to the same subnet.
    - Purpose: local network uses 1 IP address rather than multiple IPs for each device, changing ISPs does not change address locally (also hidden internal IPs are security bonus)
    - Implementation:
        - Outgoing datagrams need to have the (source ip, port #) replaced with (NAT IP, new port #) as destination addr
        - Each pairing of (source ip, port #) with (NAT IP, new port #) needs to be remembered in NAT translation table
        - Incoming datagrams need to have the (NAT IP, new port #) replaced with the (source IP, port #) stored in NAT
    - 16-bit port number allows for 60000 simultaneous connections with one LAN-side address

  - NAT violates end-to-end argument (should only process up to layer 3 as a router)
  - IPv6 will handle address shortage
  - 3 Solutions to NAT traversal:
    - Statically configure NAT to forward connection at port to server
    - UPnP IGD (Internet Gateway Device) Protocol
      - Automate static NAT port map configuration
    - Relay (NAT client establishes connection to relay, then external client connects to relay, relay bridges packets between connections)
- ICMP (Internet Control Message Protocol)
  - Carried in IP datagrams
  - Used by hosts & routers to communicate network-level info
    - Error reporting (unreachable host)
    - Echo request/reply (ping)
  - ICMP message type, code plus first 8 bytes of IP datagram causing error
    - 11-0 TTL expired
    - 0-0 echo reply
    - 3-0 dest. Network unreachable
    - 8-0 echo request
  - Traceroute
    - Source sends series of UDP segments to dest (TTL = 1, TTL = 2, until destination)
- IPv6 motivation
  - 32-bit address space soon to be completely allocated
  - 40 byte header (instead of 20 byte IPv4)
  - No fragmentation allowed
  - Contains priority among datagrams in flow, identify datagrams in same flow, still 32-bits per line, source address and destination address are both 128-bits.
  - Transitioning from IPv4 to IPv6 requires that on IPv4 routers, IPv6 datagrams are carried as payload in IPv4 datagrams (tunnelling)
- Routing and Forwarding
  - Routing algorithms determine end-to-end paths through network
  - Forwarding tables determine local forwarding at specific router
  - Routing Algorithms
    - Link-state algorithm
      - Dijkstra’s algorithm
        - Initialize: N’ = {u}, for all nodes v, if v adjacent to u, D(v) = c(u, v) or else D(v) = infinity
        - Loop: find w not in N’ such that D(w) is min -> add w to N’, update D(v) for all v adjacent to w and not in N’:
D(v) = min(D(v), D(w) + c(w, v))
      - Link costs known to all nodes
      - Compute least cost paths from one “node” to all other nodes

      - Creates forwarding table for that “node”
    - After k iterations, know least cost path to k destinations, need to check all nodes, w, not in N
    - Oscillations are possible if traffic changes
  - Distance Vector Algorithm (dynamic)
    - Bellman-Ford Equation
    - D_x(y) = min {c(x,v) + D_v(y)}, minimum taken over all neighbours v of x
    - Only knows the cost to each neighbour, maintaining their distance vectors
    - Key Idea:
      - From time to time, each node sends its own distance vector estimates to neighbors
      - When new DV estimate received, neighbour updates its own DV using B-F equation
      - Asynchronous, distributed (each node notifies neighbours only when DV changes)
      - “Good news travels fast” but “Bad news travels slow”
      - Count-to-infinity problem, a node can’t tell if it is part of the path that another node is sending a value to get to a path by. The updates will continue to grow if there is a disconnect in a chain until the error reaches infinity (which then corrects itself due to the relaxation property of B-F)
      - A fix is poisoned reverse; a node will tell another node that it is an infinite distance away from the third node, preventing the second node from routing through the first node to get to the third node.
      - Each node’s table is used by other nodes unlike LS where each node computes its own table.
- Hierarchical Routing
  - Internet is a network of networks, and each network admin may want to control routing their own way (administrative autonomy)
  - Collect routers into regions called “Autonomous Systems”
    - Routers in same AS run same routing protocol (intra-AS routing)
    - Gateway router is at the edge of its own AS but links to other AS through another router
  - Intra-AS sets entries for internal destinations
  - Inter-AS & Intra-AS sets entries for external destinations
  - *Hot potato routing:* send packets towards closest of two routers
- Intra-AS Routing (Interior Gateway Protocols or IGP)
  - RIP (routing information protocol)

- - - - DV algorithm (max hops is 16, each link is 1), DVs exchanged with neighbours every 30 sec in response message.
      - Poison reverse used to prevent ping-pong loops (distance set to 16 hops)
    - OSPF (open shortest path first)
      - Link-state algorithm (advertisements broadcast across whole AS)
      - All OSPF messages are authenticated (security)
      - Multiple same-cost paths allowed whereas only one is allowed in RIP
    - Hierarchical OSPF
      - Boundary Router (connects to other AS's) -> Backbone Router -> Area-Border Routers -> Area (contains internal routers)
      - 2 level hierarchy: local area, backbone.
        - LS advertisements only in area
    - Internet inter-AS routing: BGP (Border Gateway Protocol)
      - BGP provides each AS a means to obtain subnet reachability information from neighboring AS's: eBGP
      - Propagate reachability information to all AS-internal routers:iBGP
      - Advertised prefix includes BGP attributes:
        - Prefix + attributes = "route"
        - 2 important attributes:
          - AS-PATH: contains ASs through which prefix advertisement has passed
          - NEXT-HOP: the IP address of the router interface that begins the AS PATH
        - BGP route selection is based on policy decision, shortest AS-PATH and closest NEXT-HOP router
      - To get into forwarding table:
        - Router becomes aware of prefix, router determines output port for prefix, router enters prefix-port in forwarding table
- Link Layer
  - Hosts and routers are nodes.
  - Communication channels that connect adjacent nodes along communication path are links.
  - Link Layer Services
    - Framing, link access:
      - Encapsulate datagram into frame, adding header, trailer
        - MAC addresses used in frame headers to identify source, destination
          - Different from IP address
    - Reliable delivery between adjacent nodes
    - Flow control

-
  -
    - Error Detection
    - Error Correction
  - Link layer implemented in network interface card
- Error Detection
  - Not 100% reliable, may miss some errors, larger Error Detection & Correction bits (EDC) yields better detection and correction
- Parity Checking
  - Single bit parity, detect single bit errors
  - Two-dimensional bit parity, detect and correct single bit errors
- Internet checksum
  - Detect errors (flipped bits) in transmitted packet (only at transport layer)
    - Sender treat segment contents as sequence of 16-bit integers
      - Checksum addition (1's complement sum) of segment contents
      - Sender puts checksum value into UDP checksum field
    - Receiver computes checksum of received segment
      - If checksum equals checksum field, no error detected (but could still have errors)
- Cyclic redundancy check (better error detection)
  - Choose r+l bit pattern (G), choose r CRC bits such that <Data Bits, R> divisible by G (modulo 2)
  - Receiver knows G, if non-zero remainder, error
  - Detects all burst errors less than r+l bits
  - D * 2^r XOR R, r is number of bits of CRC
- Multiple access protocols
  - Collision if node receives two or more signals at the same time
  - Ideally: when M nodes want to transmit each can send at average rate R/M (channel rate R bps)
  - 3 classes:
    - Channel partitioning (divide channel for each node)
    - Random access (allow collisions, recover from collisions)
    - Taking turns (node takes turns, but nodes with more to send take longer turns)
- Channel Partioning
  - TDMA (time division multiple access)
    - Access to channel in "rounds", each station gets fixed length slot (length = packet trans time)
    - Unusued slots go idle
  - FDMA (frequency division multiple access)
    - Channel spectrum divided into frequency bands
      - Each station assigned fixed frequency
      - Unused go idle
- Random access protocols

  - When node has packet to send, transmit at full data rate R (no prior coordination among nodes)
  - Two or more transmitting nodes causes collision
  - Random access MAC protocol specifies:
    - How to detect collisions
    - How to recover from them
  - Slotted ALOHA
    - Assumptions: All frames same size, time divided into equal size slots, if 2 or more nodes transmit in slot, all nodes detect collision, synchronization
    - Fresh frame transmits in next slot, if no collision, node can send new frame in next slot, if collision node retransmits frame in each subsequent slot with prob. P until success
    - Pros: single active node can continuously transmit at full rate of channel, simple
    - Cons: collisions, wasting slots, idle slots
    - 37% of the time, channel used for USEFUL transmissions
  - Unslotted ALOHA
    - No synchronization, simpler
    - When frame arrives, transmit immediately
    - Collision probability increases
    - Only 18% useful transmissions (WORSE)
    - ALOHA is good for high load, inefficient at low load, 1/N bandwidth allocated even if only 1 active node.
  - CSMA (carrier sense multiple access)
    - Listen before transmit
    - If channel idle: transmit frame
    - If busy: defer transmit
    - Collisions can still occur, propagation delay means two nodes may not hear each other's transmissions
  - CSMA/CD (w collision detection)
    - Collision detected within short time, colliding transmissions aborted
    - Efficiency, as time to transmit (max size frame goes to infinity) and propagation time goes to 0, efficiency goes to 1 (better than ALOHA).
    - Efficient at low load, high collision overhead
  - Taking Turns
    - Polling: master node "invites" slave nodes to transmit in turn
      - Concern is the single point of failure (master)
    - Token passing: control token passed from one node to the next sequentially
      - Concern single point of failure (token)
- MAC addresses and ARP
  - 32-bit IP address used for network-layer

- MAC address used locally to get frame from one interface to another physically connected interface (same network in IP-addressing sense)
  - 48-bit MAC address
  - Each adapter on LAN has unique LAN address
- ARP (address resolution protocol) table:
  - Used to determine interface's MAC address, knowing its IP address
  - each IP node (host, router) on LAN has a table
  - IP/MAC address mappings for some LAN nodes
  - TTL after which address mapping will be forgotten (typically 20 mins)
  - A wants to send datagram to B, B is not in A's ARP table.
    - A broadcasts ARP query packet, containing B's IP
      - All nodes on LAN receive query
    - B receives ARP packet, replies with MAC address
    - A caches IP-to-MAC pair in ARP table
    - ARP is PnP, no net administrator intervention
- Routing to another LAN
  - Send datagram from A to B via R
    - A knows B IP
    - A knows IP of first hop router R
    - A knows R's MAC address
      - A creates IP datagram to B, creates link-layer frame with R's MAC address as destination, frame contains A-to-B IP datagram
      - When frame received at R, datagram removed and passed up to IP
      - R forwards datagram with IP source A to destination B
      - R creates the link-layer frame with B's MAC address as dest, frame contains A-to-B IP datagram
- Ethernet
  - Star prevails today, active switch in center, each spoke runs a separate Ethernet protocol (nodes do not collide)
  - Sending adapter encapsulates IP datagram in Ethernet frame
  - Connectionless – no handshaking between sending and receiving Network Interface Controllers
  - Unreliable – receiving NIC doesn't send ACKS or NACKs to sending NIC
    - Data dropped only recovered if sender uses higher layer rdt (reliable data transfer protocol found in TCP)
- Ethernet Switch
  - Store, forward Ethernet frames
  - Uses CSMA/CD to access segment
  - Transparent (hosts are unaware of switches)
  - PnP (switches do no need to be configured)

    - Each switch has a switch table to know reachability of hosts
      - Looks like routing table
    - Switches learn which hosts can be reached by recording sender/location pair in switch table, if a host (MAC address) is not found, then the frame is flooded to all interfaces except arriving interface, else it is just sent to the address.
  - Switches vs. Routers
    - Both are store-and-forward
      - Routers are network layer devices
        - Routers using routing algorithms and IPs
      - Switches are link-layer devices
        - Switches use forwarding tables, flooding and learning MAC addresses
  - VLANs allow people to use the same switch, but connect to different switch networks through the use of software which splits the ports of the switch.
    - Connects to another VLAN is the trunk port, passes frames to another switch.
- Cryptography
  - Encryption, authentication and integrity checking
  - Fundamental theorem of cryptography – secure shared-private-key cryptosystems exist
  - Fundamental tenet of cryptography – lots of smart people have tried to figure out how to break X, so far they have not been able to, therefore X is secure.
  - Fundamental assumption of cryptography – everyone knows how it works, the secret is the key.
  - Plaintext –(encryption)-> Ciphertext –(decryption)-> Plaintext
  - Caesar cipher – substitute for each letter of the message, a letter which is 3 letters later in the alphabet (shift).
    - Variant – pick a secret number n between 1-25 instead of always using 3, substitute each letter of the message with the letter which is n higher. (still easy to break).
  - Monoalphabetic cipher – arbitrary mapping of one letter to another.
  - 3 attacks to break encryption schemes:
    - Ciphertext only – have a ciphertext, brute force
    - Known plaintext – have ciphertext and a bit of plaintext from a previous ciphertext
    - Chosen plaintext – get text to be encrypted
- Secret Key Cryptography
  - Encrypt plaintext using key
  - For authentication, keys should be taken from a large enough space so there is no significant chance of using the same number twice.
  - Integrity checking, secret checksum algorithm to compute the right checksum for the message to be accepted as authentic.
- Public Key Cryptography

  - Asymmetric cryptography
  - Private key and a public key
    - Digital signature can only be generated by someone knowing the private key.
  - Public key cryptography is generally used to establish a temporary shared secret key used to encrypt the remainder of conversations.
- Generic Block Encryption
  - A cryptographic algorithm converts a plaintext block into an encrypted block.
  - Small key lengths are too easy to brute force, unnecessarily complex keys may have performance penalties.
  - 64-bits is reasonable length (2^64 input values and map it to a unique one of the 2^64 output values)
  - Substitutions – for each of the 2^k possible values of the input, specifies the k-bit output. Not practical for 64-bit blocks, but would be possible with blocks of about length 8-bits.
  - Permutations – specifies the output position for each of the k input bits.
  - Block encryption example:
    - Divide input into eight, 8-bit pieces
      - Do eight 8-bit substitutions using function
      - Permute the bits based on the key
      - Repeat for a number of rounds so that each input bit affects all the output bits.
- DES (Data Encryption Standard)
  - DES uses 56-bit key and maps a 64-bit input block into a 64-bit output block.
  - The key looks like a 64-bit quantity but one bit in each of the 8 octets is used for odd parity on each octet (rendering only 7 bits in each octet meaningful as a key), this parity checking actually makes the DES less secure.
  - The advantage to the 8 bits of parity is that it allows for sanity checking if the key could actually be a key, however there is a 1 in 256 chance that 64-bits of garbage could still have the same parity key and therefore look like a key. It is possible that 56-bits were used so that the security of DES was weak enough that NSA could break it.
  - DES – from the 56 bit key you generate 16 per-round keys (different 48-bit subset of the 56-bit key), take the 64-bit input and permute it. Each round then takes the 64-bit output of the previous round, and the 48-bit per round key, and produces a 64-bit output, after the 16th round, the 64-bit output swaps halves and is then subjected to another permutation. To decrypt you do it backwards using Key_16 first. Each round splits the 64-bit input into 2 32-bit inputs, with the right side input passed through the mangler function with the key for that round, and then it is XOR with the left half to form the new right side 32-bits and the previous right side 32-bits becoming the left side 32-bits.
  - We can decrypt by running DES backwards, because given R_n+1 and L_n+1 we know that R_n is L_n+1, therefore you have R_n, L_n+1, R_n+1 and K_n. We also know that

R_n+1 is L_n XOR mangler(R_n, K_n). Since you know R_n and have K_n, you will retrieve L_n.

  - DES is actually quite simple, as is IDEA. Algorithm is very mysterious, S-box swapping can result in magnitude less secure.
  - IDEA (International Data Encryption Algorithm)
    - Encrypts using a 128-bit key
    - Addition in IDEA is done by throwing away carries (mod2^16)
    - Multiplication done in IDEA using mod2^16 + 1
    - The number 0 which can be expressed in 16 bits would not have an inverse, so 2^16 which is in the range for mod2^16+1 but cannot be expressed in 16 bits is set to be 0.
    - Operations are reversible
    - To generate keys, divide into 8, 16-bit keys and then start over offsetting by 25-bits
- RC4
  - Long random string used to encrypt a message with a XOR operation is known as a one-time pad. A stream cipher generates a one-time pad and applies it to a stream of plaintext with XOR.
- Diffie-Hellman
  - Used for key establishment
  - There is no authentication
  - P is a large prime, g is a number less than p, P and g are known publicly, each user chooses a random 512-bit number and keeps it a secret.
    - Each then raises g to the power of their secret number (g^S_1, g^S_2)
    - Ts = g^S mod p
    - They exchange Ts, and then raise the received T to their secret number.
      - (g^S_1)^S_2
    - Then they both take that value and apply mod p, that gives them the same number.
      - That establishes their key.
      - There is no easy way to compute discrete logarithms, so it is secure even if a third person saw both g^S_1 and S_1
- Man-in-the-Middle attack
  - There is no authentication in Diffie-Hellman
  - Someone can establish a connection with both communicators, giving them both a false g^S_x, allowing the one in the middle to read all communications.
  - Diffie-Hellman is therefore only secure against passive attack where the intruder only watches the messages.
- Defenses against main-in-the-middle
  - Published Diffie-Hellman numbers

- Post public key in some PKI (public key infrastructure) that cannot be affected by a third party, makes DH immune to active attacks. (also eliminates first message of protocol)
- AUTHENTICATE DH EXCHANGE using a secret
    - Encrypt diffie-hellman exchange with the pre-shared secret
    - Sign the diffie-hellman value with your private key
    - Exchange a hash of the agreed upon shared dh value, name and preshared secret post dh exchange.

- ElGamal
    - Requires each individual to have a long-term public/private key pair (similar to DH)
    - Requires an individual to generate a new and different public/private key pair for each item that needs to be signed.
        - For message m, choose random number S_m and use permanent g and p, compute T_m
        - Then use digest function, then calculate X = S_m + d_mSmod(p-1) which is the signature
        - X, T_m and m are transmitted, to verify the signature you compute d_m, check that g^X = T_m(T^d_m) mod p
    - The point is that if the message is modified after being signed, the inputs change, changing the signature.
- Testing generator:
    - For public p, g such that p is prime and g is a generator:
        - Ex. P = 7, g = 3
            - 3^1 = 3, 3^2 mod 7 = 2, 3^3 mod 7 = 6, etc.
            - G is a generator if it generates all numbers in Z_p.
- Search Engines:
    - Search engines have scaled dramatically to keep up with the growth of the web.
    - Fast crawling technology is needed to gather the web documents and keep them up to date.
    - Google uses link structure of the web to calculate a quality ranking for each page.
        - PageRank
            - Maps are created of the hyperlinks, these allow rapid calculation of a web page’s PageRank (objective measure of citation)
            - Page A has pages T1..Tn which point to it, C(A) is defined as the number of links going out of a page A.
            - Model for user behaviour, the PageRank is the probability a page is visited by a random surfer
            - The damping factor d, is the probability a random surfer will request another random page from the current page.
            - If a page has many pages that point to it, it has a high PageRank
                - Or some from high PageRanked pages.

            - Anchor text helps search non-text information, and expands the search coverage with fewer downloaded documents.
            - Many crawlers download web pages, every web page has a docID, each doc is parsed and converted into a set of word occurrences (hits), the hits are sorted into barrels, the barrels are resorted by wordID, searcher uses the lexicon and pagerank to answer queries.
            - Indexing the web -> parsing, indexing, sorting
- Layers of the internet
    - Application, transport, network, link, physical
- Transport Layer
    - Segments
    - Communication between processes
    - PORTS
    - TCP – reliable, in order, congestion/flow control
    - UDP – unreliable, unordered, connectionless (no handshaking)
    - Use header info to deliver received segments to correct socket (sockets assigned to applications)
    - IP address & port numbers
    - RDT (reliable data transfer)
        - Detects packet loss through time out (no ACK), detects duplicates if ACK twice, premature timeout/delayed ACK
        - Go-back-N
            - Cumulative ack, timer for oldest unacked packet (if expires, retransmit)
            - Window shows packets sent but not acked and usable but not sent.
        - Selective Repeat:
            - Individual ack for each packet
            - Only retransmit individual packets
    - TCP
        - Cumulative ACK
        - TCP timeout based on RTT (round trip time)
        - Uses rdt
        - TCP fast retransmit
            - If sender receives 3 ACKs for same data, resend unacked segment with smallest seq#
        - Flow control
            - (limits number of unacked responses) Guarantees receive buffer will not overflow
        - 3 way handshake
            - Hello, Response, Connect
        - Congestion Control

    - Congestion – too many sources sending too much data too fast (lost packets, long delays) lowers goodput (useful bits) due to duplicates
    - ABR (available bit rate) – if underloaded, use bandwidth, if overloaded throttle to minimum guaranteed rate
    - Additive Increase/Multiplicative decrease
      - Increase cwnd by 1 MSS until loss occurs, then cut in half.
      - Sending rate = cwnd/RTT (bytes/sec)
      - TCP Slow Start (increase rate exponentially until first loss)
        - When loss detected, cwnd set back to 1 MSS
        - TCP RENO loss is detected by 3 duplicate ACKs (not timeout)
        - TCP Tahoe always sets cwnd to 1 (timeouts or 3 duplicate ACKS)
      - Slow Start to CA
        - Ssthresh = cwnd/2 right before the loss occurs
        - When exp reaches ssthresh, switch to linear
  - TCP Fairness
    - If K TCP sessions share the same bottleneck link of bandwidth R, each should have average rate of R/K
  - UDP for video, TCP for fairness and parallel connections
- App-layer Protocol
  - Defines rules on when and how processes send and respond to messages
  - SSL provides encrypted TCP connection (SSL is applayer)
  - HTTP (Hypertext transfer protocol)
    - Stateless
      - Maintains no information about past client requests
    - Client to server
    - Persistent vs. Non-Persistent (closes connection after object sent)
      - Non-persistent has overhead (requires 2 RTT instead of 1 because of establishing connection each time)
    - HTTP Status Responses
      - Status code appears in 1st line of server-to-client response
        - 200 OK, 404 Not Found, 505 HTTP Version Not Supported
      - GET request, GET response
    - Cookies allow the server to keep state, server sets cookie and stores id in db.
  - Web Cache allows client requests to be satisfied without involving origin server (like a buffer)
    - Purpose – they reduce response time for client requests, reduce traffic on main server
    - Fatter access link allows utilization to decrease, reducing total delay (but not cheap)

- Cheap, use local cache

- Conditional GET
  - Don’t send object if cache has up-to-date version (if-modified-since: date is the same)
- DNS (Domain Name System)
  - Internet hosts and routers use IP address (32-bit) for addressing datagrams
  - Name servers communicate to resolve names (address to name translation)
  - We do not centralize DNS because it can result in a single point of failure and traffic volume that cannot be managed.
  - Root DNS
    - Top-Level Domain (responsible for com, org, net domains)
      - Authoritative DNS servers (organization’s own DNS)
        - Local DNS server (cache with name/address pairings may be out of date)
  - Iterated query vs. Recursive query (puts burden of name resolution on contacted name server)
  - DNS resource records stored as (name, value, type, TTL)
    - When TTL expires, info is needs to be updated
    - Type A is name = hostname, value = IP
    - Type NS is name is domain, value is hostname of authoritative server
    - Type CNAME is alias, value is canonical name (the real name servereast.ibm.com)
    - Type MX is mailserver

- Packet Switching
  - Hosts break app-layer messages into packets
    - Forward packets from one router to the next, across links on path from source to destination
    - Store and forward: entire packet must arrive before it can be transmitted to next link (takes L/R seconds to transmit L-bit at R bps)
  - Queueing delay:
    - If arrival rate > transmission rate, queue will fill, packets can drop if buffer is full
    - Packets wait in queue to be transmitted on link
- Packet switching vs. Circuit Switching (reserves resources, but no sharing)
  - Packet switching allows more users to use the network (low probability everyone uses at same time)
  - Nodal Processing
    - Check bit errors, determine output link (port)
  - Queueing delay
    - Time waiting at output link for transmission while in buffer
  - Transmission delay
    - L/R, related to link bandwidth

    - Propagation delay
        - Length of physical link / speed in medium
    - D_nodalproc + D_queue + D_trans + D_prop = D_nodal (packet delay)
- Needham Shroeder vs. Kerberos (symmetric key cryptography)
    - Establish a session key between two parties on a network to protect further communication.
    - A to Server to B
    - User sends message to server identifying who they want to talk to, and their own nonce
    - Server encrypts data under K_bs also generating the K_ab, then A forwards message to B
    - B sends encrypted nonce under K_ab to A to prove he has the key.
    - A sends back (nonce_b – 1) encrypted under K_ab to verify that she is still alive
    - VULNERABLE TO REPLAY ATTACK
        - Take compromised K_ab value, replay message to B, no knowledge that key is no longer fresh
    - Flaw fixed in KERBEROS
        - Include timestamp or use of a nonce with message
            - B and A exchange N'_b with eachother in message initially
- Public Key Infrastructure
    - Set of hardware, software, people, policies and procedures to manage digital certificates
    - Binds public keys with respective user identities using a **certificate authority**
        - Used to verify that a particular public key belongs to a certain entity, securely stores the certificates in a central repository and can revoke if necessary
        - Trust in the user's public key relies on the trust in the validity of the CA's private key (because it is digitally signed using the CA's private key)