CPSC 441 Notes

Brian Pho

January 2, 2018

Contents

Chapter 1

Introduction

- Internet can be view in two ways, hardware/software and as an infrastructure that provides services to applications
- A protocol defines the format and the order of messages exchanged between two or more communicating entities, as well as the actions taken on the transmission and/or receipt of a message or other event.
- Hosts = End System = Normal computers, phones, laptops, etc.
- DSL = Digital Subscriber Line
- Store-and-Forward Transmission
- Queuing Delays and Packet Loss
- Forwarding Tables and Routing Protocols
- Circuit Switching (reserve either time/frequency)
- FDM (frequency-division multiplexing)
- TDM (time-division multiplexing)
- Bandwidth = width of the frequency band
- PoP = Points of Presence
- IXP = Internet Exchange Point (where global ISPs connect to each other)
- Tier 1 ISP > Regional ISP > Access ISP
- nodal processing delay, queuing delay, transmission delay, and propagation delay; together, these delays accumulate to give a total nodal delay.
- The time required to examine the packet's header and determine where to direct the packet is part of the processing delay.
- At the queue, the packet experiences a queuing delay as it waits to be transmitted onto the link.

- The transmission delay is the amount of time required to push (that is, transmit) all of the packet's bits into the link.
- The time required to propagate from the beginning of the link to router B is the propagation delay.
- The transmission delay is the amount of time required for the router to push out the packet; it is a function of the packet's length and the transmission rate of the link, but has nothing to do with the distance between the two routers. The propagation delay, on the other hand, is the time it takes a bit to propagate from one router to the next; it is a function of the distance between the two routers, but has nothing to do with the packet's length or the transmission rate of the link.
- Packet Loss: With no place to store such a packet, a router will drop that packet; that is, the packet will be lost.
- Mbps = Megabits per second, small b = bit
- The Internet is a network of networks
- Edge routers are routers between the core network and access networks
- ISPs act as intermediate entities for users and the core
- Two ways to connect to internet (coaxial[Cable] vs telephone line [DSL])
- Coaxial is shared within a community, telephone line is not
- Protocol: defined format and order of messages sent and received
- Packet Switching: sending data in chunks
- Packet: a chunk of data that has a maximum size, has it's own identity, adds some data including sender and receiver
- Packets consist of a header and payload
- Store and forward: router must get the *entire* packet before it can forward it because the packet has information about where to send it next and check for errors
- Adding more routers increases the packet transmission delay
- Transmission delay: time from the first bit sent to the last bit received
- Routers use pipelining (can send and receive at the same time)
- To calculate max number of users supported by circuit switching, take max bandwidth divided by user bandwidth
- To calculate max number of users supported by packet switching, find average bandwidth used by a user (% time active * user bandwidth) and then use binomial distribution. $Pr\{number of active users > 10\}$

- Public vs private internet: private internets are by companies (content providers)
- ping [address], tracert [address]
- Queuing delay is variable
- Transmission delay is L/R and propagation delay is time to send
- Traffic intensity = λL/R where λ = average packet arrival rate and L = packet length (0 = small delay, 1 = large delay)
- Throughput: A connection is as fast as its slowest link (bottleneck)
- Internet protocol stack: Application (FTP, SMTP, HTTP) > Transport (TCP, UDP) > Network (IP) > Link (Ethernet, WiFi) > Physical
- Header for each layer: Message > Segment > Datagram > Frame
- Protocol Stack: Layers ONLY talk to the layer below or above it, not 2 below or 2 above
- Increasing number of layers in a protocol stack increases overhead and delay
- Called encapsulation because almost every layer adds its own header to the payload

Chapter 2

Application Layer

- Application architectures: Client-Server, Peer-To-Peer
- Client-Server: Server always on with permanent IP, client do not communicate directly with each other
- Peer-To-Peer: No always on server, self scalability, distributed
- Socket: connects the application layer to the transport layer
- We can broadly classify the possible services along four dimensions: reliability, throughput, timing, and security.
- SSL = Secure socket layer
- HTTP = Hyper Text Transfer Protocol
- HTTP uses TCP, first connect using TCP then exchange messages using HTTP
- Non-persistent HTTP - After sending at most one object, close TCP connection
- Persistent HTTP - Keep TCP connection open
- RTT = Round Trip Time (time between sending a receiving a packet)
- File transmission time = time to get packet
- With parallel TCP connection and non-persistent HTTP, it takes 4RTT
- For persistent HTTP, 3 RTT for establishing the TCP connection, sending a request for the base page, and then asking for the reference objects
- For HTTP request message, sp = space, cr = carriage return, and lf = line feed
- HTTP is state-less so use cookies to keep state
- Web caches (proxy server): if HTTP request is in cache, get from cache otherwise fetch
- Conditional GET: only get object if object has been modified
- SMTP = Simple Mail Transfer Protocol

- SMTP is a pushing protocol which is why the receiving user does not use SMTP to retrieve messages, must use a different protocol
- SMTP (send), POP IMAP HTTP (retrieve)
- POP3 cannot organize mail on the mail server
- IMAP keeps state of mail, POP3 doesn't retain state
- DNS: Domain Name System (distributed database)
- DNS is used to translate from hostname to IP address
- Root DNS: top DNS
- TLD DNS: responsible for com, org, net, edu, ca
- Local DNS: each ISP has one
- Authoritative DNS: organization's own DNS
- Root DNS > TLD (Top-level domain)
- Iterated Query for DNS is like breadth first search (heavier on local DNS)
- Recursive Query for DNS is like depth first search (heavier on upper DNS)
- 4 types of DNS, A (hostname, IP), NS (domain, hostname), CNAME (alias, true/canonical name), MX (name, mailserver name)
- Domain name is like an ID for a host
- MX = Mail Exchanger
- DNS requests are sent over UDP
- HTTP, SMTP are sent over TCP
- When a client downloads a video chunk, the connection measures the time to download and size of the chunk to calculate the approximate available bandwidth (R = B/T)
- DASH: Dyanmic Adaptive Streaming over HTTP
- DASH divides a video into multiple chunks
- DASH starts with the lowest quality chunk to test the connection
- DASH divides a video based on duration over size
- CDN: Content Distribution Network (Multiple copies of video at multiple locations)
- Manifest file: file with all chunks of video listed

Chapter 3

Transport Layer

- Transport layer (logical communication between processes)
- Network layer (logical communication between hosts)
- TCP: reliable, in-order delivery with congestion control
- UDP: unreliable, unordered delivery
- Multiplexing: data from multiple sockets
- Demultiplexing: host receives IP datagram with source IP, destination IP, source port number, destination port number
- Connection vs Connectionless oriented: connection uses IP as well as port number
- Choose a port greater than 1024
- A network datagram always has 1 segment
- On top of the transport segment is the header with source IP, destination IP, and additional header info
- Port numbers at 16 bits, IP addresses are 32 bits
- Packets at the network layer are called datagram
- Use TCP when the ordering of the packets matter
- Create new socket for every connection in TCP
- UDP is fast and more efficient in terms of bandwidth usage than TCP
- UDP is connectionless because there is no handshaking between the sender and receiver
- UDP segment header has 4 segments, source port number, destination port number, length, and checksum
- UDP checksum: detect errors in transmitted segment
- Add the header segments together to get sum, if carry-out of MSB is 1 then wrap around

- When calculating checksum, checksum = 0
- To turn a sum into a checksum, do two's complement on the sum
- XOR all of the end bits, if their is 1 at the end, maybe no error
- rdt = reliable data transfer
- udt = unreliable data transfer
    1. Reliable channel
    2. Channel with errors
    3. Channel with loss
- ARQ = Automatic Repeat reQuest
- rdt 1.0 - assume underlying channel is perfectly reliable
- rdt 2.0 - underlying channel my flip bits in packet
- ACK: acknowledgment that the packet receive has OK
- NAK: negative acknowledgment that the packet received has errors
- rdt 2.0 - ACK/NAK may be corrupted so use stop and wait (sender sends one packet then waits for receiver to respond)
- rdt 2.1 - handles corrupted ACK/NAK using sequential ordering of data
- Only need two sequence numbers since the sender will never send another packet without an ACK/NAK
- rdt 2.2 - Includes sequential numbers with ACK and replace NAK with ACK of the last successful packet received
- rat 3.0 - use timeout to retransmit the packet (retransmissions only happen with timeouts)
- Throughput of rdt 3.0 = $L/(L/R + RTT)$
- Time for just one packet to send is L/R + RTT
- We don't large packets because if they drop then it is a huge waste
- Increase throughput of rdt using pipelining

3.1 Pipelining

- Go-back-N: no buffering, cumulative ACK, one timer for oldest packet, needs N + 1 sequence numbers (N is the window size)
- Selective repeat: Each packet has its own ACK, timer, needs 2N sequence numbers
- Only retransmit when the timer times out

3.2 TCP

- If alpha is close to 0, then cares about old measurements. If alpha is close to 1, then cares about new measurements.
- Only retransmit the first packet in window if it times out. Don't retransmit the rest of the packets in the window
- Fast retransmit bypasses timeout if 4 ACKs are received
- To open a TCP connection needs 3 segments. To close needs 4 segments
- Packet loss detection: Timeout (severely congested W = 1), TripleDupacks(mildly congested W = W/2)
- Delay bandwidth = Size of window = R * RTT
- If throughput is equal to the bandwidth of the link, then there is packet loss. The best case with TCP is a throughput of 3/4 * R.
- Throughput max = W_{max}/RTT = C

Chapter 4

Network Layer

- Find out where to send packet by using the routing table. Routing tables are calculated using a routing algorithm.
- Delay Bandwidth Product = B = C * RTT where C = link capacity/bandwidth
- 6 subnets in example
- For an ISP to create sub-blocks, first 20 bits are fixed and then depending on number of blocks wanted, use some part of the 12 bits left
- NAT: Can map 2^{16} possible devices since port numbers are bound by 16 bits
- Private IP addresses are not unique, they're used for subnets in your home
- IPv6 Header: no checksums, hop limit = time to live
- IP Tunnels are connections using IP instead of link layer
- IP Tunnels are logical connections, not physical
- Network Operating System = Logically centralized routing controller
- Data plane is about what routers do locally
- For a subnet to connect to another subnet, both require at least one (edge) router
- Distance vector algorithm is better for distributed and doesn't require entire graph, Dijkstra's only work locally and requires knowledge of the topology of the network (very hard for internet)
- Distance vector algorithm will get rid of loops. Doesn't know how many iterations needed to run to determine optimal path
- Distance Vector Algorithm
    1. Initial tables with known cost values of its own node
    2. Each node sends its known values to its neighbouring nodes
    3. Recompute cost of each path with the new data using the Bellman Ford algorithm
    4. Send changes to neighbouring nodes

5. If there has been any changes, then recompute the costs

- If the costs of the nodes do not update, then the costs are said to have converged
- DV algorithm is dynamic (reacts to changes)
- DV convergence time depends on cost of each link
- DV algorithm can have loops if the cost changes, BGP cannot have loops, Dijkstra's cannot have loops
- Hierarchical Routing: Using Intra-AS and Inter-AS
- AS-Path = AS3, Next-Hop = 3a[1]
- AS-Path is used to check for loops and to stop retransmitting AS-Path
- Internet is single-path, not multi-path
- Hot Potato Routing: choose the path with the known least cost inside subnet (local minimum)

Chapter 5

Link Layer

- Need end-to-end reliability (TCP) because you can lose packets at the router (dropping packets due to full buffer)
- Two ways to lose packets: congestion at routers and loss during transmission
- Single bit parity check: make the parity bit = so that the total number of 1s is even. (Only works for odd number of errors)
- Slotted ALOHA: Collisions only happen within a time slot
- Unslotted ALOHA: (Trades synchronization for collisions) Collisions can only occur one time slot before, during the time slot, and one time slot after.
- CSMA: Collisions are still possible as unknown how many are listening
- CSMA/CD: Not easy to implement in wireless networks
- If propagation delay goes to 0, then probability of collision goes lower
- Polling vs TDMA: Polling doesn't have fixed time slots and has a master
- Subnet: A set of computers that can talk to each other **without a router** needed to send data
- MAC addresses have no hierarchy, IP addresses do
- Switches are routers that don't run the routing algorithms, DHCP
- CSMA: For first collision {0, 1}, the second {0, 1, 2, 3}, third {0, 1, 2, 3, 4, 5, 6, 7} choose randomly from this set. Exponentially increases with each subsequent collision.

Chapter 6

Physical Layer

- Problem with lower frequencies is that it requires a large antenna which isn't desirable for portable devices
- Problem with higher frequencies is that it water molecules will act like an antenna and absorb the signal
- For wired networks, the power of the receiver is almost the same as the power received by the sender ($P_R \propto P_S$)
- For wireless networks, the power of the receiver is proportional to the power received by the sender divided by the distance to the power of alpha, where alpha is a constant of the environment ($P_R \propto \frac{P_S}{d^\alpha}$)
- When the signal attenuation decreases. it's called path loss

6.1 SNR

- Signal to Noise Ratio: $(SNR)_{db} = 10log_{10}(\frac{signal\ power}{noise\ power}) = 10log_{10}SNR$
- db is not the unit but the converted SNR
- $SNR = \frac{signal\ power}{noise\ power}$
- If $SNR = 1$, then $SNR_{db} = 0\ db$
- If $SNR = 10$, then $SNR_{db} = 10\ db$
- If $SNR = 100$, then $SNR_{db} = 20\ db$
- If $SNR = 1000$, then $SNR_{db} = 30\ db$
- If $SNR = 2$, then $SNR_{db} = 3.01\ db$
- Ratio of the power in a signal to the power contained in the noise at the receiver
- A high SNR means a high-quality signal
- SNR sets upper bound on achievable data rate

6.2 Shannon Capacity

- $C = W\ log_2(1 + SNR)$
  - W = Frequency Bandwidth in Hertz
  - C = Capacity in bits per second
  - SNR = Signal-to-Noise Ratio (not in db)
- Represents theoretical maximum **error free rate** that can be achieved
- E.g. SNR = 0 db, W = 1 MHz. Answer = 1 Mbps (**MEMORIZE**)

6.3 CDMA

- E.g.
- A: 11101000 = [1,1,1,-1,1,-1,-1,-1]
- B: 10111011 = [1,-1,1,1,1,-1,1,1]
- Change all 0s to -1
- Dot product A and B = 0 (orthogonal)
- To Encode: Multiply the bit you want to send with your code (Change 0 to -1)
- To Decode: Dot product the data and divide by the length of the data
- Sending bit 0 flips the code
- SIFS: Short interface space

6.4 Comparing Ethernet and WiFi

- Ethernet → Collision detection → CSMA/CD → Cheap collisions
- WiFi → No Collision detection → CSMA/CD → Expensive collisions
- CSMA/CA
  - Transmit if idle for DIFS
  - If channel is busy, then back off (Ethernet only after collisions)
  - Feedback/ACK from receiver for WiFi
  - Ethernet → Unreliable → CSMA/CD → Cheap collisions
  - WiFi → Reliable → Stop-and-wait → Expensive collisions
  - RTS/CTS doesn't completely solve the hidden terminal problem
  - BER = Bit Error Rate

6.5 Mobility

- Handoff happens at the physical and link layer
- Handoff can lose packets when switching access points, TCP sees this as congestion due to lost packets and reduces window size
- Mobile IP is another IP address for a device. Two IP addresses for one device. Mobile IP isn't used in practice.
- Type of Mobility and it's Effect

Type of Mobility	**Effect**
Small-Scale (within the range of your AP)	Lower transmission rate
Medium-Scale (change AP, within subnet)	Delay of packet drop due to handoff (TCP tput drops)
Large-Scale (change network)	Restart connections

Chapter 7

Final Exam

- Slides + Self Study on multiple choice
- Exam Coverage
- Chapter 1 and 2 are not covered (no direct questions)
- Chapter 3: Only TCP (TCP congestion control (AIMD and slow start behaviour), TCP throughput and its relation to window size, TCP fairness) (No sequence numbers, stop and wait, etc.)
- Chapter 4-7: Everything
- (Unsure since he's still writing the exam)
- Network Layer (Longest prefix matching, router buffer requirements, IP addressing and subnets (Guaranteed), Dijkstra and Bellman-Ford algorithms (Guaranteed), BGP and Hot Potato routing (Guaranteed))
- Link Layer (Parity check and CRC (Guaranteed), ALOHA and its analysis (computing throughput and success probability), CSMA/CD and Binary Exponential Back-off, Taking turn protocols (computing throughput), ARP protocol (in a LAN and across LANs), Ethernet switches (self-learning and forwarding))
- Wireless Networks (Understanding SNR and Capacity, CDMA coding and decoding (orthogonality), CSMA/CA (the role of RTS/CTS mechanism), mobility (impact on TCP))
- Coverage for MC is just 4, 5, 6, 7. If prof decides to put chapter 3 on the final then it'll only show up on the written portion of the test.
- Know how to do CRC by hand