Intro to Communication Networks

EE 122 Notes

by Angela Wang and Rahul Shah

Contents

1. Internet Basics

The internet is made up of hosts and routers attached through links that deliver information by arranging it into packets and transmitting the packets.

Packets
A string of bits arranged according to a specified format.

Hosts
Source or sink devices that host applications that generate or use the information exchanged. They have a distinct location-based 32-bit IP address.
ex. computers, printers, servers, webcams

Routers
Receives and forwards packets between hosts and other routers.

Links
Wired, wireless, or optical systems that transports bits between two routers or hosts

IP Address
A unique address for every host that is a bit string of a specific format.
ex. IPv4 is a 32-bit string of a format $a.b.c.d$ where a, d, c, d are the decimal value of the four bytes. IPv6 is newer protocol that is a 128-bit address.

Domain Name Service (DNS)
Hosts attached to the Internet have a name in addition to an IP address. DNS is a distributed directory service that translates names into IP addresses. The internet is divided into zones, each with a DNS server that maintains the addresses of hosts in each zone.

World Wide Web (www)
A collection of hyperlinked resources like web pages, video streams, music files. Each resource is identified by a URL.

Universal Resource Locator (URL)
Specifies a computer and file in that computer along with the protocol that should deliver that file.
ex: http://www.eecs.berkeley.edu/ wlr.html

Hyper Text Transfer Protocol (HTTP)
Protocol that specifies the request/ response rules for getting a file from a server to a client. So the protocol sets up a connection between the server and client, requests a specific files, and then closes the connection when the transfer is complete.

1.1 Switching

The section of the set of links that a packet follows from its source to destination.

Circuit Switching
The set of links is selected once and the data rate required on that set of links is reserved for a duration of time.
ex. telephone networks, where a set of links is chosen for a complete telephone conversation, and a data rate is required for the duration of the conversation

Packet Switching
A set of links is individually chosen for each packet, which allows for flexible routing. If a link goes down or a host or router is disabled, packet switching can reroute information on an alternate path. Datagrams is a version of packet switching where each router chooses the next link for each packet.

Virtual Circuits
The set of links is selected once, but the data rate required is not reserved on the links. Data is transported using packets and sends packets of a connection on the same set of links.
ex. Multi-Protocol Label Switching (MPLS) and Asynchronous Transfer Mode (ATM)

1.2 Routing

The process of selecting a path for packets to get from a source to a destination address.

Routing Table
A table of the shortest paths that a router regularly computes and updates. Specifies the next hop for each destination address. If there are N devices and each device has its own entry then the routing table will be N^2 in size. To reduce the size of the tables, we can split the devices into groups.

Classless Interdomain Routing (CIDR)
One was we can group devices is by taking advantage of location-based addressing. IP addresses $(a.b.c.d)$ are organized similar to telephone numbers, where a is the most general location is analogous to a phone number's country code, d is analogous to a house number. For CIDR the longest common prefix or initial string of bits in the addresses going to the same destination will be grouped together.
ex. If there are N groups of devices with M devices each. Lets say device 1.2 wants to send something to device $N.M$. Firth the packet will go to router 1, then router 1's routing table has the next hop to router N as router 5. Then, similarly at router 5 there is the next hop towards router N. After the packet reaches router N, the packet will then be forwarded to device $N.M$. So each routing table will be of size $N-1$+M.

1.3 Transmission

The bits of a packet are converted into electrical or optical signals and sent by a transmitting node. These signals are then received by another node and converted back into bits

Error Detection
During transmission, there may be interference to the signals, causing bits of a packet to get corrupted. Parity bits, or bits that indicate the xor of a set of the packet's bits, can be used to detect these errors. Header checksums can also be used. The transmitting and receiving node perform a calculation with the bits in a packet's header, and an error is detected if the results differ.
ex. Hamming codes

1.4 Congestion

Too many packets may be forwarded to a router at the same time, causing the routers run out of memory and drop packets.

Automatic Retransmission Request (ARQ)
Guarantees that a source retransmits the packets that do not reach the destination without errors. The destination acknowledges all correct packets it receives and the source retransmits packets if it does not receive an acknowledgement (ack) from the destination within a specific amount of time.

Congestion Control
Hosts want to have the largest throughput they can have. When hosts start losing an excessive number of packets, they assume it is due to congestion and the host decreases the number of packet sent when they miss acks. The hosts increase the packet transmission rate when acks are received.

Flow Control
Fast devices may send packets too rapidly to slower devices, causing the slower device to miss packets. To prevent this the receiving devices send the amount of free buffer space it has in each ack, and the source stops transmitting when the number of un-acked bits sent crosses the amount of free space.

2. More Internet Basics

We are going to touch on some essential features and metrics that characterize a network.

2.1 Sharing

Sharing is an essential part of networks, as we can reduce the amount of long links needed to connect multiple devices.

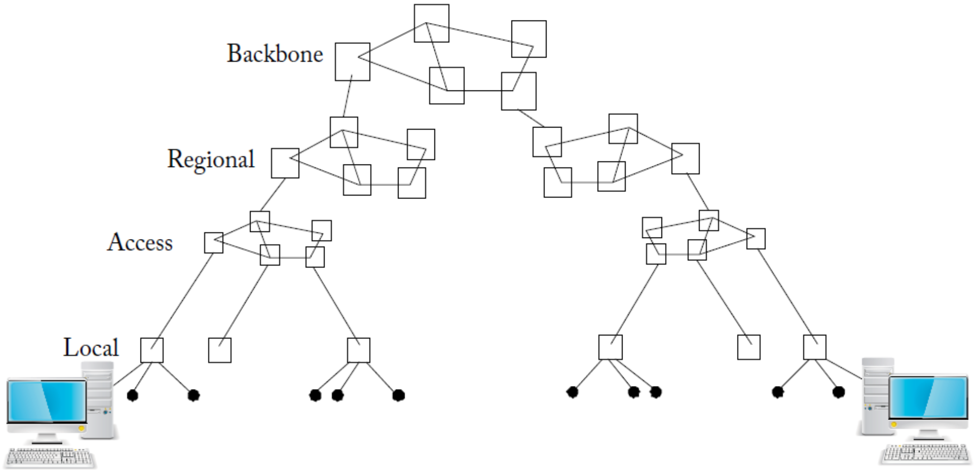

Figure 2.1: Hierarchy of networks where multiple devices are sharing a local network, multiple local networks are sharing an access networks, and so on

Multiplexing Gain
Sharing is possible because devices are not active all of the time. Multiplexing gain measures the benefit of sharing a link between many users and is the amount of active users sharing a link. If C is the link data rate, M is the multiplexing gain, and C_{user} is the data rate seen by the user.

$$C_{user} = C/M$$

ex. A thousand users are sharing a link, but only ten are active.

$$C_{user} = C/10bps$$

2.2 Link Metrics

Link Rate
Each link is characterized by data rates in bits per second. ex. Common link rates for cable modem uplink (device to Internet) are 131 Mbps and downlink rate (Internet to device) are 343 Mbps.
ex. Links are broadband if its rate exceeds 25 Mbps downlink and 4 Mbps uplink. If its rate does not exceed that amount it is considered narrowband.

Frequency
Cycles per second (Hz).
ex. $V(t) = A\sin(2\pi f_0 t)$, f_0 is the frequency

Bandwidth
Measures the width of the range of frequencies on a link.
ex. A telephone line can transmit signals over a range of 300 Hz to 1 MHz, so the bandwidth is

$$10^6 - 300 = 999700Hz \approx 1MHz$$

Signal to Noise Ration (SNR)
The ratio of the power of the signal at the receiver over the power of the noise at the receiver. Sometimes given in dB, but we would link the ratio.

$$dB = 10\log_{10}(ratio)$$

Shannon Capacity
The larger the bandwidth of a link the faster the link rate. The noisier the channel is, the smaller the SNR, and the more bandwidth needed to reliably transmit the signal. Shannon Capacity quantifies the relationship between the C the maximum reliable link rate, SNR the noise, and W the bandwidth. It is also the theoretical link rate limit that can be achieved.

$$C = W\log_2(1 + SNR)$$

2.3 End to End Metrics

Delay
The elapsed time for a packet to traverse between two points. Queuing time is the waiting time for a packet at a node before it can be transmitted. Transmission time refers to the time it takes for a packet to be transmitted over a link at the link rate. Propagation time is the time for the physical signal to get from the starting point to the ending point. Processing time is the time consumed to performing the required operations to process a packet at a node.

$$Delay_{A\_to\_B} = Q/R + P/R + T_{propagation}$$

$$Delay_{A\_to\_C} = Q/R + 2P/R + T_{processes} + 2T_{propagation}$$

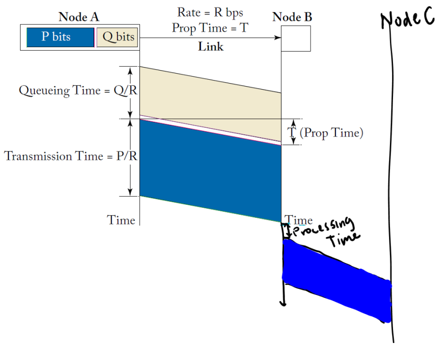

Figure 2.2: Delay from Node A to Node B to Node C

Outstanding Packet
Status of a packet is outstanding if the sender has transmitted the packet but has not received an acknowledgement.

Window Size
The maximum allowed number of outstanding bytes.

Round-Trip Time
The time between sending out a packet and recieving an ack.

Throughput
Data rate for a particular application, bits per second. Not the same as link rate, because particular applications may send a sequence of packets with gaps between each of the packet transmissions.
ex. A source S sending a packet to a destination D. Lets say there is a window size of three packets, so the sender can only sent three packets betfor waiting for an acknowledgement. The throughput of this application is less than the link rate, because of the time that the sender has to wair for an acknowledgement.
ex. If multiple devices A, C, and D are linked to a router B. If A wants to send information to C and D, the throughput of the connection from A to C is $R/2$ and from A to D is $R/2$, because the maximum rate from A to B is only R.

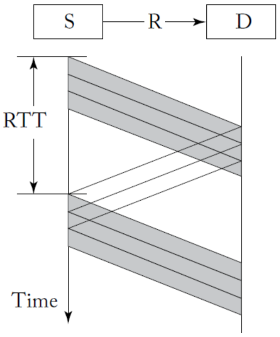

Figure 2.3: Throughput limited by window size

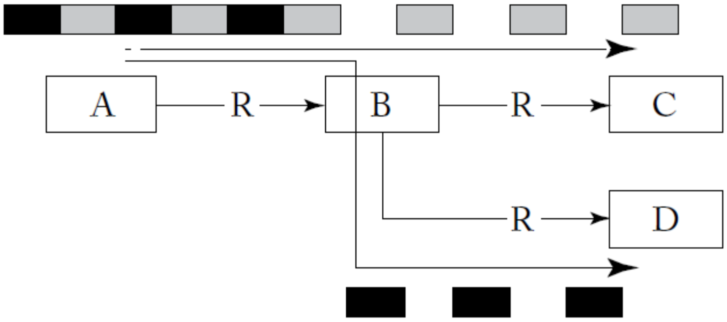

Figure 2.4: Throughput limited by bottleneck link

Delay Jitter
The difference between the longest and shorted delivery time among the packets of that connection. Successive packets sent may not face the same delay across a network due to congestion or other issues.
ex. For streaming applications, a constant stream of packets in necessary. Lets say that the delay jitter J is known. The destination can store the packets that arrive in a buffer. Then, the first packet should be stored for J seconds before playing it back, and the subsequent packets should be able to be streamed from the buffer.

M/M/1 Queue
Memoryless arrival, Memoryless service, 1 server model that allows us to estimate the delay and backlog at transmission lines. Let us discretize our example into 1 second intervals and assume that the probability a packet arrives each second is λ and the probability the server finishes servicing a packet each second is μ.
Average service time per packet is $\frac{1}{\mu}$
Utilization of the system is $\rho = \frac{\lambda}{\mu}$
Average time that a packet spends in a buffer or being served $T = \frac{1}{\mu - \lambda}$
Average queueing time $T - \frac{1}{\mu}$
Average number of packets stored in a link's transmitter queue or with the server $L = \frac{\lambda}{\mu - \lambda}$

Little's Result
Little's Result is a metric helps relate the average number of packets, L, with the average arrival rate, λ, and time, W, a packet spends in a system.

$$L = \lambda W$$

ex. Suppose on average there are 1 billion users sending 10 MBytes of data per day on the Internet and each packet takes 0.4 seconds on the Internet. Then, $\lambda = 10^9 \frac{8*10^7 bits}{24*3600 sec}$ and $W = 0.4s$, and the average number of bits in transit $L = \lambda W$.
ex. We can also use Little's Result in the following way. Suppose the average amount of data in transit per day on the Internet is 43 GBytes, and on average users send 10 MBytes of data per day with each packet taking 0.4 seconds. On average, how many users are using the Internet? So far we are given $W = 0.4sec$, $L = 43GBytes \approx 3.7*10^1 1bits$, and $\lambda = \#users \times \frac{8*10^7 bits}{24*3600 sec}$. So using Little's Result we can solve for $\lambda = \frac{L}{W} \approx 9.25*10^1 1bits$, then solve for $\#users = \frac{\lambda}{926bps} \approx 9.99*10^8 \approx 1\ billion\ users$

2.4 Layers

The Internet is arranged into five main layers for modularity and design simplicity.

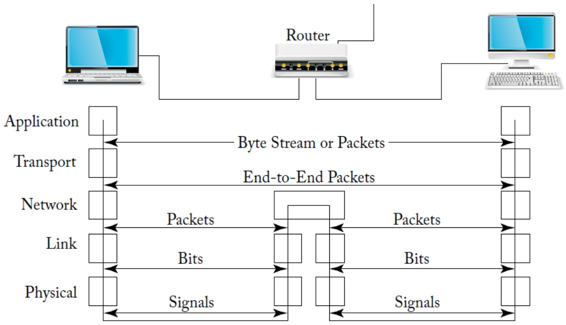

Figure 2.5: Layers of Internet

Physical Layer
The physical delivery of bits across a medium. For example, transmitting a signal across a radio network.

Link Layer
Directs and interprets the signals transmitted into bits to deliver packets across links.

Network Layer
Uses the link layer to deliver packets across multiple links from source to destination.

Transport Layer
Uses the end to end delivery of the network layer to transport packets and byte streams from a process in a source computer to one in a destination computer.

Application Layer
Applications that use the transport layer services.

2.5 Network Application Terminology

There are many different ways applications allow hosts to exchange information.

Client/Server
Web browsing uses this model. The user host is the client that connects to a server, like google.com. The client asks the server for files and the server transmits them.
Server farms host collections of servers and have a system for balancing the load of requests.

Peer-to-Peer P2P
Stores files in user hosts instead of specialized servers. A user can request a file from a list of user hosts and the hosts can deliver different parts of the file in parallel.

Cloud Computing
A user makes use of the computing service hosted by a collection of computers attached to a network. Users can lease services from a cloud computer provider instead of purchasing an application themselves. Some examples include Amazon AWS, Microsoft Azure, Salesforce, ...

Content Distribution System
A set of servers at various points in a network that improve the delivery of information to users. One example is if a user requests a file from one server, the server may redirect the request to a different server that is closer to user (may be based on IP addresses).

Multicast/Anycast
Delivers files or streams to a set of hosts that have subscribed to a multicast and the server sends them the information.
Anycast refers to the delivery of a file to any one of a set of hosts.

Push/Pull
Hosts pull information from servers. Hosts push information to servers.

3. Ethernet

Ethernet is a robust wired network that is widely used today to provide low latency and stable network connections.

3.1 Aloha Network

Wireless network where devices transmit packets to a central communication node with frequency f_1 and receives packet ack(nowledgement)s from the central communication nodes at frequency f_2. If a device does not receive an ack after a given time, it assumes the packet has been dropped and retransmits after a random time interval.

Randomized Multiple Access (RMA)
Device transmissions are not scheduled ahead of time, instead devices resolve conflicts with a distributed and randomized mechanism:

- If an no outstanding ack, transmit
- If no ack is received, wait for a random delay and repeat 1.

3.2 Ethernet

A wired version of the Aloha network, where all devices share a common cable. Uses a protocol similar to RMA, with two major differences

Carrier Sensing
A device can sense when a channel is idle or not.

Collision Detection
A transmitting device can sense collisions, or times when another device starts transmitting.

Truncated Binary Exponential Backoff
A scheme where a station is picking a time slot to transmit. A station picks X uniformly in $\{0, 1, ..., 2^{n-1}\}$ where $n = \min\{m, 10\}$ and m the number of collisions the station experienced with the same packet. When m reaches 16, the station gives up and declares an error.

- **Wait** For a new packet, $t_{\text{wait}} = 0$. For retransmission, t_{wait} is a random time from the backoff scheme.
- **Transmit** After waiting time, when channel is idle, transmit
- **Collision Detection** If a collision is detected, abort transmission and repeat (1) for retransmission. Else repeat (1) for a new packet.

The main idea of the scheme is that the selected waiting time gets more random after multiple collisions, thus reducing the chances of repeated collisions. The probability that the stations collide k times is $\frac{1}{2} \cdot \frac{1}{4} \cdot \frac{1}{2^{k-1}}$.

Capture/ Winner Takes All
An unlucky station that happens to collide tends to have to keep on waiting as other stations that did not collide transmit. Station A and B want to transmit. During the first attempt they collide, so A picks $X = 0$ and B picks $X = 1$. After A transmits there is immediately another packet to transmit and because B's backoff counter has decremented, both stations collide again. Then A picks X from $\{0, 1\}$, while B picks from $\{0, 1, 2, 3\}$. It's likely that A will pick a smaller X than B.

Ethernet/MAC Addresses
Every computer Ethernet attachment is identified by a globally unique 48-bit string. Because the addresses are not location-based, the Ethernet switches maintain tables that list the addresses of devices that can be reached via each of its ports.

3.3 Switched Ethernet

Switches sends a packet only to the output port toward the destination of the packet. Multiple packets can go through a switch at once. If two packets are sent towards the same output port, the switch can store the packets until it can transmit them.

Learning
When a computer with Ethernet address x sends a packet to a computer address y, it sends a packet [y—x—data] to the switch. Then, the switch reads the destination address y, looks in a table, called forwarding table, to find the port to which y is attached and sends the packet on these wires. If it gets a packet with a destination address that is not in the table, the switch sends a copy of the packet on all the ports, except the port on which the packet came. Whenever it gets a packet, it adds the corresponding source address to the table entry that corresponds to the port on which it came.

Spanning Tree Protocol
Protocol where switches find the tree of shortest paths rooted at the switch with the smallest ID.

- switches send packets with the info [myID—CurrentRootID—DistanceToCurrentRoot]
- Switch relays packets whose Current Root ID is the smallest the switch has seen so far and adds one to distance to current root
- Eventually switches only forward packets from the switch with the smallest ID with their distance to that switch.

3.4 Aloha Network

Time-Slotted Aloha Protocol
Time is divided into slots, where each slot is enough to send one packet. If there are n stations that transmit independently with probability p in each time slot.

$$R(p) = P(one\ station\ transmits) = np(1-p)^{n-1}$$

p is set by maximizing $R(p)$ and we see that $p = \frac{1}{n}$, plugging this into $R(p)$ we get

$$R(\frac{1}{n}) = (1 - \frac{1}{n})^{n-1} \approx \frac{1}{e}$$

Non-Slotted Aloha Protocol
Lets say the time slots are $\tau << 1$ small and it takes $\frac{1}{\tau}$ time slots to transmit one packet. Then there are $\frac{2}{\tau} - 1$ time slots, where another node can start transmitting creating a collision. Then the probability that station 1 is successful is $S(p) = p(1-p)^{(n-1)(\frac{2}{\tau}-1)}$. The average number of successful transmissions per unit time is then $\frac{nS(p)}{\tau}$, where we can solve for an optimal $p* = \frac{1}{(\frac{2}{\tau}-1)(n-1)+1}$ and the success rate is $\approx \frac{1}{2e}$

3.5 Hub Ethernet

Devices are attached to a hub with a point to point cable. When a packet arrives at the hub it repeats on all other ports. If two or more packets arrive at the hub at the same time, it sends a signal to all ports that a collision has been detected.

Maximum Collision Detection Time
If two nodes A and B try to transmit at the same time. The signal from A travels to the hub which repeats it. The signal from A then keeps travelling toward B by t_{PROP}, where t_{PROP} is the maximum propagation time between two devices in the network. Node B sends its signal right before t_{PROP}, then the signal from B will reach node A around $2t_{PROP}$, where shortly after node A will detect a collision. Thus, nodes should wait a random multiple X of $T = 2t_{PROP}$ to start transmitting.

Efficiency
The average time to send a packet is t_{TRANS}. The probability that exactly one station transmits is $R(\frac{1}{n}) = \frac{1}{e}$, thus the average time until first success is e time slots of duration T, so on average stations wait $(e-1) \cdot T = 2(e-1)t_{PROP}$ and the fraction of time during which stations transmit successfully is

$$\eta = \frac{t_{TRANS}}{2(e-1)t_{PROP} + t_{TRANS}}$$

For every transmission with duration t_{TRANS} there is a wasted time of $(e-1) \cdot 2t_{PROP} \approx 3.4t_{PROP}$, thus the fraction of time when the stations are using the network to transmit packets successfully is $\frac{t_{TRANS}}{t_{TRANS}+3.4t_{PROP}}$

4. WiFi

WiFi networks are a form of ”wireless Ethernet”. Similar to Ethernet, randomized access schemes are used.

4.1 Distributed Coordination Function (DCF)

An infrastructure mode where a set of devices are equipped with a radio that communicates with an access point (AP).

Basic Service Set (BSS)
The set of devices that communicate with a given access point.

4.2 MAC Layer Protocol

The MAC Sublayer uses binary exponential backoff with RTS/CTS.

Short Inter-frame Spacing (SIFS)
When a station (a device or the access point) receives a correct WiFi packet, it sends an ACK after an SIFS (for Short Inter-frame Spacing)

DCF Inter-frame Spacing (DIFS)
To send a packet, a station must wait for the channel to be idle for DIFS plus the random backoff delay. Because DIFS > SIFS, a station can send an ACK without colliding with a packet.

Backoff Delay
The backoff delay is chosen uniformly from $\{0, 1, ..., CW_n\}$ time slots where

$$CW_n = \min(CW_{\max}, 2^n(CW_{\min} + 1) - 1)$$

where n is the number of previous attempts of transmitting that packet (collisions or other issues). Unlike Ethernet, the backoff counter is only decremented by one every Slot Time during which the channel is idle. If another station transmits, the station has to wait for DIFS before continuing.

Extended Inter-Frame Spacing (EIFS)
Each station must decode every packet it receives before it can determine if the received packet is for itself. If a station receives an incorrect WiFi packet, it must wait for EIFS before attempting to transmit, to see if the intended receiver actually received the packet correctly. EIFS is calculated with respect to the intended receiver might send an ACK at the lowest data rate allowed by the standard.

Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA) Protocol

1. A station has a packet to transmit, pick a random backoff delay picked uniformly in 0, 1, ..., 31 Slot Times.
2. Wait for channel to be idle for DIFS time.
3. Decrement the backoff counter for every idle slot time.
    - If a corrupted packet is received from any other station, wait for EIFS time.
    - If another station transmits, return to 2.
    - If backoff counter decremented to 0, go to 4.
4. Transmit the packet.
5. After the receiver gets the packet, wait for SIFS time to send an ACK.
6. Wait for an ACK. If no ACK is received after ACK Time-Out (ACK-TO), generated a random backoff delay CW_n and return to 2.

Two problems are associated with this protocol

Exposed terminal problem
A device refrains from transmitting because it is falsely sensing that the medium is busy, but its intended destination is different from that of the ongoing transmission. In infrastructure mode WiFi networks, since all communication takes place via the AP, the reader can convince oneself that the exposed terminal problem is not really a serious issue.

Hidden terminal problem
Two sending devices cannot hear each other, so they falsely sense when the channel is idle. One of them can begin transmitting even if the other device is already transmitting. WiFi networks use RTS/CTS messages to overcome this problem.

1. A sender wants to transmit data, using the CSMA/CA protocol, it sends a RTS message to all devices near it with the address of the receiver and size of packet it wants to transmit.
2. The receiver sends a CTS (Clear to Send) message to all devices near it with the size of the packet that it will be receiving. (CSMA/CA still)
3. After receiving the CTS message and SIFS slot times, the sender begins data transmission.
4. Receiver sends an ACK after SIFS slot times.

When other stations hear the RTS and CTS, they refrain from transmitting until the packet P and the ACK have been sent. Consider the situation where two devices hidden from each other want to send data to the AP. If the first device sends out an RTS message and the AP responds with a CTS message, the other device should hear at least the CTS message.

Network Allocation Vector (NAV)
Other MAC sublayer indications of how long the channel is going to be busy is in the frame header, where in the duration field of frame headers, which is used to update the NAV variable at each device, which indicates how long the channel is going to be busy. Using NAV for keeping track of the channel status is Virtual Carrier Sensing. RTS/CTS is a way of distributing NAV information.

4.3 Efficiency of MAC Protocol

Efficiency of the WiFi MAC protocol is defined as the data throughput that the stations can achieve.

Single Device
A single device is continuously transmitting to or receiving from the AP without RTS/CTS messages. With a single device, there are no channel collisions. efficiency = Time to send packet/(Time to send packet + Physical layer overheads + MAC Layer Protocol time(backoff counter, DIFS, SIFS, ACKs...))

Multiple Devices

We can use Markov chains to analyze the efficiency of WiFi networks.

Markov Chains
at a given epoch k, each slot time after a station has a packet to transmit, each state is denoted by $(s(k); b(k))$ for a given device where $s(k)$ denotes the number of previous attempts for transmitting the pending packet at this device, and $b(k)$ denotes the backoff counter.

1. A station that gets a new packet to transmit. The state of that station is the node at the top of the diagram. After waiting for DIFS, the station computes a backoff value uniformly in 0, 1, ..., 31. The state of the station is then one of the states (0,0), (0,1), ..., (0, 31). The first component of the state (0) indicate that the station has made no previous attempt at transmitting the pending packet. The second component of the state is the value of the backoff counter from the set 0, 1, ...31.
2. The station then decrements its backoff counter at suitable epochs (slot times when channel is idle, DIFS time,...), and the state can then move from (0,5) to (0, 4).
3. Eventually, the backoff counter gets to 0, so that the state is (0, 0) and the station transmits.

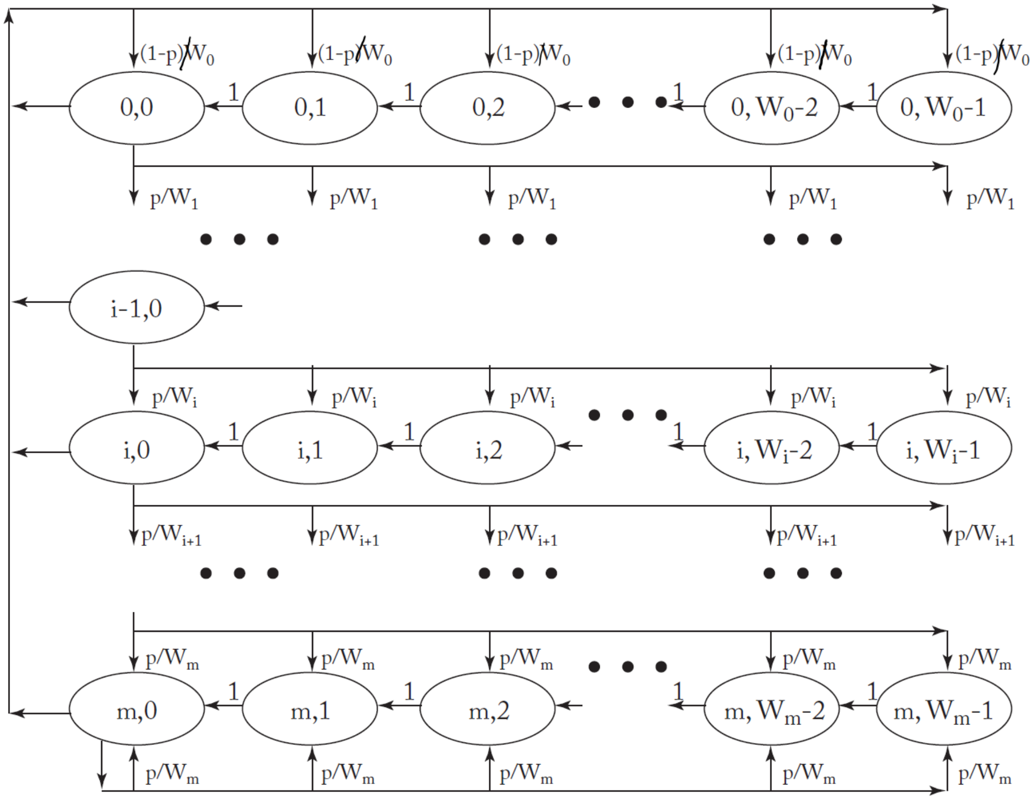

Figure 4.1: MAC Protocol Markov Chain

- If the transmission collides, then the station computes another backoff value X, now uniformly distributed in 0, 1, ...63. The state of the station then jumps to 1,x where the first component indicates that the station already made one attempt.
- If the transmission succeeds, the state of the station jumps to the top of the diagram.

The key simplifying assumption in this model is that, when the station transmits, it is successful with probability $1 - p$ and collides with probability p, independently of the state $(i, 0)$ of the station and of its previous sequence of states. α is the probability that at a given transmission opportunity (i.e., when its backoff counter reaches 0), this station attempts to transmit.

$$\pi(x)\sum_{y\neq x} P(x,y) = \sum_{y\neq x} \pi(y)P(x,y)$$

Using these balance equations and the fact that the probabilities must sum to 1 $\sum_{(i,j)} \pi(i,j) = 1$. We can solve for the $\pi(x,y)$ probabilities.

Efficiency analysis
Assume that all the n stations have the same average probability α of transmitting. The probability $1-p$ that this station succeeds (and does not collide) is then the probability that the other $n-1$ stations do not transmit.

$$1 - p = (1-\alpha)^{n-1}$$

. To find the steady state probability of being in each state of the model, we can use the balance equations for the Markov chain, where total flow in and out of a state are in equilibrium. Then we can solve for when $alpha$ because we know that a station transmits when it is at the states $\pi(i, 0)$, so $\alpha = \sum_i \pi(i, 0)$. Using α we calculate the network throughput (overall rate of data transmission) as follows. Time duration between two consecutive epochs of the Markov chain has a successful transmission with probability $\beta := n\alpha(1-\alpha)^{n-1}$ Let T be the average length of this duration. The network throughput is then given by βBT where B is the average number of data bits transmitted during a successful transmission. Indeed, during a typical duration, 0 data bits are transmitted with probability $1 - \beta$ and an average of B data bits are transmitted with probability β. Thus,βB is the average number of data bits

transmitted in an average duration of T.
T corresponds to either an idle Slot Time or a duration that contains one or more simultaneous transmissions. If the duration contains exactly one transmission, T corresponds to the transmission time of a single packet, an SIFS, the transmission time of the ACK, and a DIFS. In the case of more than one transmission in this duration, T corresponds to the longest of the transmission times of the colliding packets and a DIFS.

4.4 Physical Layer

WiFi networks transmit radio waves in unlicensed spectrum bands allocated by the Federal Communications Commission(FCC) centered around 2.4 or 5 GHz bands. Here the available spectrum around 2.4 GHz is divided into 11 channels with 5 MHz separation between the consecutive center frequencies. In order to minimize co-channel interference, channels 1, 6, and 11 are commonly used by adjacent BSSs to maximize channel separation (25Mhz). In a relatively large building or campus, Extended Service Set (ESS) can be created by linking BSSs. ESSs make it is possible for a device to move from one BSS to the next in a seamless fashion.

5. Routing

Internet routers use a two-level scheme: inter-domain and intra-domain routing. Intra-domain routing is within a domain or network of devices and inter-domain routing across different domains. Intra-domain algorithms find the shortest path to the destination. The inter-domain routing uses an algorithm where each domain selects a path to a destination domain based on preference policies.

Internet Service Providers (ISPs)
Providers that own networks and make agreements to carry each other's traffic.

Peering Agreement
ISPs provide free connectivity to each other's local customers. For the ISPs with these agreements, they exchange the routes to their customers.

Transit Agreement
One ISP provides or sells access to all destinations in its routing table. They annouce their customers, but they do not provide free connectivity

5.1 Inter-Domain Routing

Inter-domain routing focuses on choosing a path across those nodes to go from one source domain to a destination domain. The paths that domains choose may be influenced by factors such as distance, price, reliability, etc...

Path Vector Algorithm
Routers advertise their preferred path to the destination, and the results propagate until the source knows all paths from the source to the destination.

Border Gateway Protocol (BGP)
Based on the path vector algorithm, the source router obtains all paths it can take to the destination and then select a path based on its preferences. With the path vector algorithm, we can easily find the shortest path, but if the preference rules are not chosen in a methodical order, the algorithm may not be able to converge.

Multi-Exit Discriminators
If an ISP has two or more connections to another ISP, the destination ISP may attach a discriminator to its local destinations. The discriminator represents a metric from the router attached to the connection to the destination, so that the source ISP can better choose between which connection is better.

5.2 Intra-Domain Shortest Path Routing

Inside a single domain, routers want to find the shortest paths. Two algorithms that we can use for this is Dijkstra and Bellman-Ford.

Link State Algorithm
Routers exchange link state messages containing a list of items of the form [*neighbor*, *distance to neighbor*]. The *distance to neighbor* can be another metric that takes into account preferences. After exchanging those messages, routers will have a complete view of the network.

Dijkstra's Algorithm
Dijkstra's is built on top of the link state algorithm, and recursively computes the set $P(k)$ of the k-closest nodes to node A, and breaking ties using a deterministic rule. Each node will put the next step along the shortest path to each node in its routing table and the number of steps in this algorithm is the number of nodes.

Bellman-Ford
Bellman-Ford is based on the the distance vector algorithm. In the distance vector algorithm, routers regularly send their current estimate of the shortest distance to the other routers to their neighbors. Bellman-Ford algorithm uses these distance vector to continuously update its shortest distances to the destination, first finding the paths that are one hop away, two hops away, and so on. One problem with Bellman-Ford is that it may take a long time to converge to new shortest paths when links fail.

1. Initialize nodes to $d_0(i) = \text{inf}$ for $i \neq i_0$ and $d_0(i_0) = 0$.

2. For every step $n + 1$, node i gets an estimate $d_n(j)$ from one of its neighbors j, and updates its estimate by $d_{n+1}(i) = mind_n(i), d(i, j) + d_n(j)$
3. Eventually $d_n(i) = d(i)$

If $d(i)$ changes if we start at a different i_0. One approach to address the fact that some links might break, if a node i gets an estimate from neighbor j has a larger than previous estimate, node i can reset its estimate to inf and restart the Bellman-Ford algorithm.

5.3 Message types

A unicast message is one where a source sends a message to a single destination. An anycast message is one where a message needs to be received at any member of a set of nodes. A multicast message is when all members of a group must get a message. A broadcast message is sent to all other nodes.

Anycast
The Bellman-Ford algorithm can be used for unicast routing. For a node i, let $d(i)$ be the minimum distance from that node to the anycast set. Then, we see that $d(i) = \min\Big(d(i, j) + d(j)\Big)$ where j are the neighbors of i. To account for the anycast case, $d(k) = 0$ for any node k in the anycast set.
Dijkstra's shortest path algorithm can also be used, where the algorithm terminates when a node of the anycast set it reached.

Multicast
The goal for a multicast message is to have the sum of link lengths to be minimized while reaching all destination nodes.

Steiner Tree
A tree with all destination nodes shaded and with the minimum sum of link lengths that reaches all the destination nodes. The most efficient approach is to use one Steiner multicast tree for each multicast source, but this is hard to do in practice.

Forward Error Correction
Using retransmissions for multicasting would quickly get out of hand. So it is simpler to add additional information to packets to make transmission easier. Packet erasure codes is a scheme to recover from erasures by sending extra and combination of packets. For example, if one needs to sent two packets $P1$ and $P2$, you can compute the XOR of the two packets to get $C = P1 \bigoplus P2$. Then the user only needs to get any two of the packets $P1, P2, C$ to be able to decode $P1$ and $P2$. We can extend this scheme to multiple packets.

6. Internetworking

The internet is a collection of networks, the goal of internetworking is to connect networks. We will explore how Internet connects Ethernet networks.

6.1 Goals

Connectivity
All device on a given network can send packets to one another, where a device encapsulates a packet with the appropriate format for the network and suitable addresses.

Broadcast-Capability
Each network is capable of broadcasting packets.

6.2 Internetworking Compoenents

Each device has a IP address and MAC address, and are organized into subnets.

Subnets
Within a network, devices may be divided into smaller groups called subnets. A subnet mask is the number of leading bits in the IP addresses that are shared by all devices in that network.

Gateway Router
A router that connects the Ethernet network to the rest of the Internet.

Domain Name System (DNS)
A server that translates the name of a computer into its address. Each zone is maintained by some independent administrative entity, and it corresponds to some directory server that stores the addresses of the computers with the corresponding names. A modification of a zone only affects the directory server of that zone and not the others.

Address Resolution Protocol (ARP)
Enables devices to find the MAC address of interfaces on the same Ethernet that corresponds to a given IP address.

1. Host H1 wants to find the MAC address that corresponds to the a given IP address (IPx), sends Broadcast ARP packet of the form with a destination address, specifies that its a ARP request, and the ARQ request itself of them form [all—e1—who is IPx?]. Essentially this request is saying ”to all devices, from e1, who is IPx?”, where e1 is the MAC address of the host.

2. Ethernet switch that receives a broadcast packet repeats the packet on all its output ports

3. Device gets the ARP request packet, it compares its own IP address with the address IPx, if the IP addresses do not match, the request is ignored. If they match the device answers the request with a packet [e1—ex—I am IPx].

6.3 Internetworking Examples

We will look at two examples of how devices send packets to other devices.

Same Subnet
Based on location-based addressing in the IP addresses the host knows that the other device is within the same subnet, but it still needs to know the MAC address of the device it is sending too. Once it find out ex, the MAC address of the second device, the host H1 forms a packet of the form [ex—e1—IP1—IPx—Data], where ex is the destination MAC address, e1 is the source MAC address, IP1 is the source IP address, IPx is the destination IP address, and DATA is the rest of the packet. Note that [IP1—IPx—Data] is encapsulated in the Ethernet packet.

Different Subnets
Based on location-based addressing in the IP addresses the host knows that the other device is not within the same subnet.

1. The host H1 first sends a packet to the gateway.

2. Using the IP address of the gateway, the H1 uses ARP to find the MAC address of the destination.
3. H1 then sends the packet [eg—e1—IP1—IPx—Data] to the gateway router, where eg is the MAC address of the gateway router.
4. The gateway decapsulates the packet to recover [IP1—IPx—Data], consults its routing table to find the output port that corresponds to the destination IPx.
5. The host gateway router then sends the packet to the destination's gateway router.
6. The destination gateway router decapsulated the packet, consults its routing table and sees that the packet is within its subnet.
7. The packet is sent from R2 using the same procedure as Same Subnet.

Finding IP addresses
What happens when the host only knows the name of the other host but not the IP address? Then the source host will use DNS to find the other IP address.

Fragmentation
If a link can only transmit a certain amount of data, the network protocol IP fragments the packet and uses the header of the packet to specify ID and what part of the packet it is.

6.4 Dynamic Host Configuration Protocol (DHCP)

When you attach a device to a network, the network assigns an IP address to the device from a pool of addresses with DHCP.

1. Attach device to the network and send a DHCP request asking for an IP address.
2. Router sends the request to a DHCP server, which maintains a list of available IP addresses.
3. The DHCP server allocates an address to the device.
4. Devices periodically send requests to renew the lease on the IP address
5. If the lease is not renewed, the server puts the address back into the pool of available addresses.

DHCP is commonly used by ISPs to assign addresses to devices of their customers, reducing the number of addresses that the ISP must reserve.

6.5 Network Address Translation (NAT)

Internet engineers realized that IP addresses could run out, thus a new addressing scheme that used 128 bits instead of 32 was created. DHCP is one mechanism that reduces the number of IP addresses needed by allocating them temporarily instead of permanently. The Network Address Translation (NAT) is another scheme that enables us to reuse IP addresses. A home network router may have a set of devices, with a set of private IP addresses using port numbers. At the transport layer, the packet has a structure [source IP — destination IP — source port — destination port — ... — data]. If the private addresses of devices in the home network are IPa, IPb, and IPc, where the NAT has an address IPa.

1. IPb sends a packet [IPb — IPx — TCPm — TCPn — ...] to a device with IP address IPx, where TCPm is the source port number and TCPn the destination port number
2. The NAT converts the packet into [IPa — IPx — TCPb — TCPn — ...], where TCPb is the port number chosen by NAT device that notes that TCPb corresponds to (IPb, TCPm)
3. When the destination with address IPx and port TCPn replies to this packet with [IPx — IPa — TCPn — TCPb — ...]
4. When the NAT gets this packet, it maps back the port number TCPb into the pair (IPb, TCPm), and it sends the packet [IPx — IPb — TCPn — TCPm — ...] to device IPb.

7. Transport

The network layer of the Internet provides packet delivery from one host to another. The transport layer is in charge of end-to-end delivery across the Internet between a process in the source to one in the destination device, along with error, congestion, and flow control, and multiplexing.

7.1 Transport Services

Ports
Logical ports that distinguishes information flow attached to application processes. There are three types of ports: well-known ports for fixed processes, registered ports for company specific applications, and dynamic/ private ports that can be assigned dynamically. Enables multiplexing of packets of different applications on the host.

7.2 User Datagram Protocol (UDP)

Delivers individual packets, unreliable delivery. A UDP packet has the source port, destination port, length of the packet, packet checksum, then the payload.

7.3 Transmission Control Protocol (TCP)

Delivers a byte stream, reliable delivery as two hosts arrange for retransmission of failed packets. The source does congestion control in the routers and the destination device does flow control. The header of a TCP packet contains contains packet control information like sequence number, ack, window size, and other flags.

1. Client sends a SYN packet with random number X
2. Server responds with a SYN.ACK with random number Y and ACK sequence number $X + 1$
3. Client sends first data packet with sequence number $X + 1$ and an ACK (got the SYN.ACK) with $Y + 1$
4. Server sends ACK.
5. Repeat 3-4 until done with sending data.
6. Source sends FIN packet
7. Destination sends FIN.ACK
8. Repeat FIN, and FIN.ACK from destination to source

Error control

With TCP, hosts control errors through retransmissions that are not acknowledge before the ACK timeout. Here are two error control schemes.

Stop-and-Wait
Source sends a packet, waits for up to T seconds for an acknowledgment, retransmits the packet if no acknowledgment. Even if there are no errors, the source can send only one packet every T sec, and T has to be as large as a round-trip time across the network.

Go Back N
At any time, the source can have sent up to N packets that have not yet been acknowledged. When destination gets a packet, it sends an ACK with the sequence number of the next packet it expects to receive, in order. If no ACK for a specific packet, the source retransmits that packet and the subsequent packets. The source slides its window of size N so that it starts with the last packet that has not been acknowledged in sequence. This window, referred to as Transmit Window in the figure, specifies the packets that the source can transmit. With no errors, Go Back N sends N packets every round trip time (assuming that the N transmission times take less than one round trip time). Thus, the throughput of this protocol is N times larger than that of Stop-and-Wait.

Retransmitting the unACKed packet along with all the subsequent packets are unnecessary if the subsequent packets are received correctly. Selective acknowledgements can be used. Receiver sends a positive ACK for all the packets it received correctly.

Timers

Round trip time varies greatly from one connection to another, so the timeout value must be dynamically adapted.

- Source measures round trip times T_n, time between the transmission of the n^{th} packet and the corresponding ack.
- Source calculates the average $A_{n+1} = (1-b)A_n + bT_{n+1}$ of the round trip times $\{T_1, ..., T_n\}$
- Source calculates the average deviation $D_{n+1} = (1-b)D_n + b|T_{n+1} - A_{n+1}|$ of the round trip times $\{T_1, ..., T_n\}$
- Timeout value is $A_n + 4D_n$

Congestion Control

Multiple flows share link, and devices do not know the network topology, bandwidth of links, or number of flows that share a link.

Additive Increase-Multiplicative Decrease (AIMD)
Consider two devices A and B with flow rates x and y, respectively, sharing a link with bandwidth C.

1. Sources increase x and y additively as long as they receive acknowledgements.
2. When no ack, assume the router had to drop the packets, which is seen as $x + y > C$, Sources divide their rate by 2 and repeat 1.

Fast Retransmit Scheme
Source starts retransmitting a specific packet after three duplicate ACKs to avoid having to wait for a timeout and dividing the window size by 2. Then the source waits a round trip time to see if it gets an ACK, if not then divide window size by 2.

Fast Recovery Scheme
After source receives three duplicate ACK of packet x (x is the packet number in the current window of packets being sent counting from 1), source sends a copy of the missing packet and updates its window size as $W \xrightarrow{W} /2 + 3$, the increases its window by one whenever it gets another duplicate ACK. The source will receive $W - X - 1 - 3$ duplicate ACKs, so the total change in window size will be $W \xrightarrow{W} /2 + 3 \xrightarrow{W} /2 + 3 + W - X - 4$.

Adjusting the Rate
If no packets are lost, the sliding window protocol sends N packets every round trip time. TCP adjusts the rate by adjusting the window size $W = N$, by $1/N$ for every packet received, so after one round trip time the window size becomes $W \xrightarrow{N} +N * (1/N) = W + 1$. When the source misses an ACK, it divides the window size with the schemes above. This becomes unfair, because connections with shorter round trip times increase their rate faster.

Slow Start Scheme
The scheme might take a long time to reach an acceptable rate, so to speed up the initial phase, the connection starts by doubling its window size every RTT. To double the window size in a round-trip time, the source increases the window size by 1 every time it gets an ACK. Thus, if the window side was N, the source should get N ACKs in the next round-trip time and increase the window by N, to 2N. When the source misses an ACK, after a timeout, it restarts the slow start phase with the window size of 1.

Combined TCP Window Size

1. Starting with the slow start phase, the window increases exponentially quickly, doubling every roundtrip time.
2. When a timeout occurs when the window size is W_0, the source remembers the value of $W_0/2$ and restarts with a window size equal to 1 and doubles the window size every roundtrip time until the window size reaches $W_0/2$.
3. The protocol enters a congestion avoidance phase. During the CA phase, the source increase the window size by one packet every round-trip time. If the source sees three duplicate ACKs, it performs a fast retransmit and fast recovery. When a timeout occurs, the source restarts a slow start phase.

7.3.1 Flow Control

Congestion control is the mechanism to prevent saturating routers. Flow control prevents saturating the destination of an information flow.

- The end host of a TCP connection sets aside some buffer space to store the packets it receives until the application reads them. When it sends a TCP packet, that host indicates its Receiver Advertised Window (RAW), the amount of free buffer space it currently has for that TCP connection.
- The sender of that connection then calculates RAW-OUT where OUT is the amount of outstanding bytes in transmit. RAW-OUT is the number of bytes that the sender can safely send to the receiver without overflowing the receiver buffer.
- The sender then calculates the minimum of RAW-OUT and its current congestion window and uses that minimum to determine the packets it can transmit.

8. Models

Models are an important way to analyze different network protocols. ”Backpressure” protocals optimize the schedule, routing, and congestion control.

8.1 Graphs

Imagine that the vertices of a graph correspond to wireless nodes and that two vertices are connected if the nodes cannot transmit without interfering with one another. Another application of coloring is frequency allocation to access points, where each color represents a WiFi access point.

Directed graph
A set of vertices, nodes, and arcs (pair of vertices).

Capacity
Each arc (i, j) has a capacity $C(i, j)$ which is a positive number related to the max rate at which a flow can go through that arc.

Cut
A subset of notes. The capacity of a cut S is the sum of the capacities of the arcs form S to the complement S^c of S

Max-Flow, Min-Cut
The maximum flow from some vertex v to another vertex w is equal to the minimum capacity of the cuts S where $v \in S$ and $w \in S^c$. The rate of flow from v to w cannot exceed the value of any cut S with $v \in S$ and $w \in S^c$.

Connected
All vertices can be reached from every vertex.

Complete
Any two nodes are connected by a link.

Degree
The number of neighbors a vertex has

Coloring
An assignment of colors to each vertex so that no two neighboring vertices have the same color.

Independent Set
All the nodes of the same color can transmit without interfering with one another. It is considered a maximum independent set if every other node is attached to one of the nodes in the set.

Chromatic number
The minimum number of colors required to color a graph.

Clique
A set of vertices that are pairwise connected, then all these nodes must have a different color. The chromatic number is at least as large as the largest number of nodes in a clique.

Brook’s Theorem
The chromatic number of a connected graph with maximum degree Δ is at most Δ, except if it is an odd cycle or a complete graph, in which case it is at most $\Delta + 1$.

8.2 Queueing

Queueing theory is a way to leverage models to quantify the delays and backlogs of packets, as the fluctuations in arrivals and packet lengths are causes of delays and backlog.

M/M/1 Queues
(From Pranav Srinivasa’s discussions).
We had a brief intro into M/M/1 queues in chapter 2. For this queueing model, arrivals and service completions are independent, memoryless, and stationary, and customers arrive one at a time. The arrival process is modeled as a poisson process and the service times are modeled by exponential distributions. The Poisson distribution is defined over $\mathbb{N}$ and is often used to model the number of times an event occurs in an interval of time. It is defined by a single

parameter μ, where $\mu = \lambda * T$ for a rate λ and time interval T.
If $X \sim Poisson(\mu)$, then random variable X has the following probability mass function (pmf):

$$P(X = k) = e^{-\mu}\ \frac{\mu^k}{k!}$$

The distribution's expecation is μ and its variance is also μ.
In the communication context, packets of data arriving at a server can be modeled as a Poisson process. The Poisson distribution is commonly used because it only requires knowing the average of what we want to model. Also, the time between 2 arrivals in a Poisson process is exponentially distributed.
The exponential distribution is defined over the domain $[0, \infty)$ and is characterized by a single parameter (rate): λ.
If $T \sim exp(\lambda)$, then random variable T has the following probability density function (pdf):

$$f_T(t) = \lambda e^{-\lambda t}$$

Consequently T's expectation (μ_T) is $\frac{1}{\lambda}$ and variance (σ_T^2) is $\frac{1}{\lambda^2}$ (exercise: check these!).

In the context of communication, the exponential distribution is often used for modeling the time interval between the arrival of 2 packets or "chunks" of data. The exponential distribution's memoryless property is particularly useful and this problem will involve rederiving this property.
Then we can use an infinite DTMC (Discrete Time Markov Chain) with the length of an M/M/1 queue as states and a very small time period ϵ as one epoch (time between transitions) to gain insights about delays and backlogs. Let λ and μ be packet rates. We can write the balance equations and find the invariant distribution to calculate the proportion of time that a queue spends in each state (assuming $\lambda < \mu$).

$$\pi(0) = (1 - \lambda\epsilon)\pi(0) + (\mu\epsilon)\pi(1)$$

We see that we can rearrange to get

$$\pi(1) = \frac{\lambda}{\mu}\pi(0)$$

where we have "cancelled" the ϵ. (To be completely rigorous, we would need to use limits as ϵ tends to 0, but we omit this level of rigor and instead rely on the vague notion of canceling or possibly using L'hospital's rule).
The remaining equations are of the form:

$$\pi(i) = (1 - \lambda\epsilon - \mu\epsilon)\pi(i) + (\lambda\epsilon)\pi(i-1) + (\mu\epsilon)\pi(i+1)$$

For i =1, we rearrange to get:

$$(\lambda + \mu)\pi(1) = \lambda\pi(0) + \mu\pi(2)$$

Then substitute to get $\pi(2)$ in terms of $\pi(0)$:

$$(\lambda + \mu)\frac{\lambda}{\mu}\pi(0) = \lambda\pi(0) + \mu\pi(2)$$

$$((\lambda + \mu)\frac{\lambda}{\mu} - \lambda)\pi(0) = \mu\pi(2)$$

After simplifying fractions, we get:

$$\pi(2) = (\frac{\lambda}{\mu})^2\pi(0)$$

Now we notice a pattern (which can be proved via induction if we choose):

$$\pi(i) = (\frac{\lambda}{\mu})^i\pi(0)$$

Next, we know that the sum of all $\pi(i)$ must equal 1 since they represent probabilities. We use the geometric series formula to simplify since $\lambda < \mu$:

$$\sum_{i=0}^{\infty}(\frac{\lambda}{\mu})^i\pi(0) = \pi(0) * \frac{1}{1 - \frac{\lambda}{\mu}} = 1$$

This implies that:

$$\pi(0) = 1 - \frac{\lambda}{\mu}$$

where we previously learned that $\rho = \frac{\lambda}{\mu}$ is the utilization of the M/M/1 queue. It makes sense that 1-utilization gives us the proportion of time that the queue is empty (not being used). Similarly,

$$\pi(i) = (\frac{\lambda}{\mu})^i(1 - \frac{\lambda}{\mu}) = \rho^i(1-\rho)$$

Then to find the average length of the queue, we know the proportion of time that the queue spends in each state, where the state is the length of the queue. We take the expectation of the length to find the average length:

$$E[L] = \sum_{i=0}^{\infty} i * \rho^i(1-\rho)$$

$$E[L] = (1-\rho)\sum_{i=0}^{\infty} i * \rho^i$$

Now we use another series formula to simplify since $\lambda < \mu$:

$$E[L] = (1-\rho)\frac{\rho}{(1-\rho)^2}$$

$$E[L] = \frac{\rho}{1-\rho}$$

$$E[L] = \frac{\lambda}{\mu - \lambda}$$

Then, using Little's Law we can find the average length of time each packet spends in the queue. Little's Law is $L = \lambda * T$ where L is the average number of things in a system, λ is the incoming rate and T is the time that 1 thing spends in a system.

$$T = \frac{L}{\lambda} = \frac{1}{\mu - \lambda}$$

Jackson Networks

A network of interconnected M/M/1 queues. There are J queues. Customers of classes $c = 1, 2, 3, ...C$ arrive at the network independently in a memoryless way, i.e., as Poisson processes with respective rates λ_c. A customer of class c goes through a set $S(c) \subset \{1, ...J\}$ of queues in some specific order. For simplicity, we assume that no customer visits a queue more than once. The service times in queue j are independent of other service times and of the arrivals and are exponentially distributed with mean μ_j , the same for every class of customers.

Total arrival rate of customers into the network is $\lambda = \sum_{c=1}^{C} \lambda_c$.

Total rate of customers that go through some queue j is $\gamma_j = \sum_{c=1}^{C} \lambda_c * \{j \in S(c)\}$.

Average delay is $W_c = \sum_{j \in S(c)} \frac{1}{\mu_j - \gamma_j}$

Average number of customers is $L_j = \frac{\gamma_j}{\mu_j - \gamma_j}$

Jackson networks can be used to estimate the average delay customers get along with balancing traffic.

It is important to note that if two queues are in tandem (one after another), the arrival times of the second queue are not independent of the services times of the first queue. Thus, system of two queues in tandem is not a Jackson network and the average delay of packets in this system is not given by the formula for a delay in a Jackson network, but from simulation we can see that the discrepancies between the Jackson network's predictions and the actual model are small.

9. LTE

Long-Term Evolution (LTE) is a tech for implementing cellular networks. It then evolved to LTE-Advanced but LTE has been updated overtime – as of 2010, it now includes 4G when before it was just up to 3G.

9.1 QoS

LTE emits high Quality of Service, even in poor channel conditions. This is known as *transmit diversity*, and is often done via MIMO.

Spatial Multiplexing
However, instead of MIMO we can instead use *Spatial Multiplexing* when favorable conditions arise. Spatial Multiplexing is when each antenna is used to transmit different data to achieve a higher overall data rate via switching between antennas.

Note: This requires you to have another algorithm at the receiver to combine the signals from the different receiving antennas.

GPS
A *Generalized Processor Sharing* system is one where the packets are classified into K classes and wait in corresponding first-in-first-out queues until the router can transmit them. Each class k has a weight w_k. The scheduler serves the head of line packets at rates proportional to weight of their class. That is, the instantaneous service rate of class k is $w_k C/W$ where C is the line rate out of the router and W is the sum of the weights of the queues that are backlogged at time t.

Note: This is actually a fictitious model – it cannot actually be implemeneted since the scheduler mixes bits from different packets and does not respect packet boundaries.

WFQ
Weighted Fair Queuing is an approximation of GPS: The packets are classified and queued as in GPS. The scheduler transmits one packet at a time, at the line rate. Whenever it completes a packet transmission, the scheduler starts transmitting the packet that GPS would complete transmitting first among the remaining packets.

9.2 5G

This is an initiative for the next-generation low-latency, high-reliability communication network, though it comes with the downside of only being useful when in close proximity to specific communication towers which are still emerging.

Time Domain
Functions that are a function of time, like $f(t)$.

Frequency Domain
Functions that are a function of frequency, like $F(\omega)$.

Note: We can convert between time and frequency domain via the Fourier Transform, and multiplication in one domain is a convolution in another.

Convolution
The convolution is a commutative operation: $(v*u)(n) = \sum_{k=-\infty}^{\infty} v(n)u(n-k) = \sum_{m=-\infty}^{\infty} v(n-m)u(m) = (u*v)(n)$.