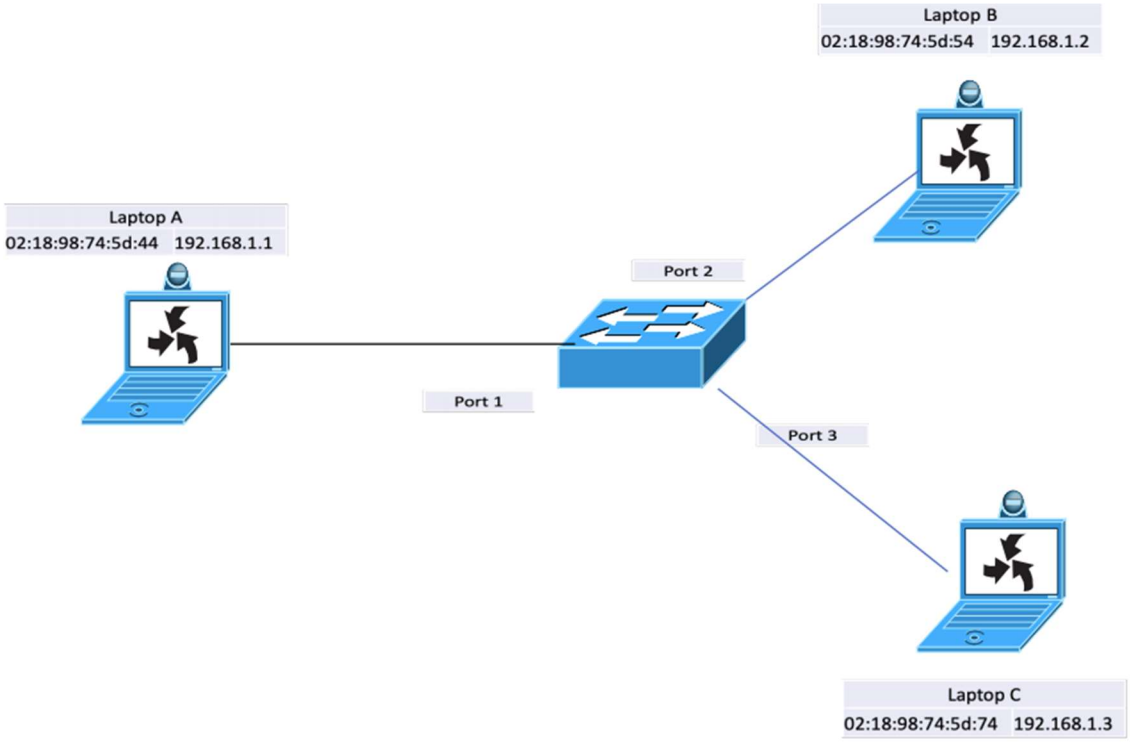
CS-GY 6843 Spring 2021 Midterm

Last Name: Vo First Name: Brandon _Net ID:_bdv9527

1) Refer to the following diagram with 4 elements:



a) (5 pts) Identify the broadcast domain(s) by listing the networking elements that make up the broadcast domain

These three laptops appear to be on the same subnet and share the same ethernet line connected by one switch. If one laptop sends a broadcast message, it would get sent to the other two laptops as the switch can flood the messages to all ports.

They all share one broadcast domain.

b) (5 pts) Identify the collision domain(s) by listing the networking elements that belong to each collision domain.

There are three collision domains where each collision domain of A, B, and C are all separated by the switch in the middle. A has its own collision domain, B has its own collision domain, and C has its own collision domain.

This is because the switch separates collision domains into different ports. A, B, and C have their own different ports to separate the collision domains. If the switch has already recorded A, B, and C's MAC Address in the CAM table, then the switch can forward messages directly to their respective ports, reducing the chances of a collision happening.

c) (10 pts) If laptop A issues an ARP Request for 192.168.1.2 what do each of the elements learn, if anything? Identify the protocol steps (layer2 & layer3) and addresses. What is stored in the ARP tables of each host and the CAM table of the switch?

Laptop A is issuing a layer 2 ARP Request to laptop B's IP Address. This ARP Request will ask for the Mac Address associated with the IP address of 192.168.1.2 as well as provide the details of Laptop A's MAC address and IP address. This ARP Request is broadcasted to the ethernet's entire broadcast domain.

Inside the ARP Request is the destination IP Address of 192.168.1.2, destination MAC Address of FF:FF:FF:FF:FF:FF to signify that it's a broadcast, the source IP Address of 192.168.1.1 and A's MAC Address of 02:18:98:74:5d:44.

When the switch receives A's ARP Request, it won't know where B is and the switch will notice that the message is a broadcast message, so the switch will flood the message to every port, meaning laptops B and C will receive A's ARP Request. The switch's CAM table also associates port 1 with Laptop A's MAC address, so the switch will learn that Port 1 holds Laptop A.

Laptop C will read A's ARP request and record Laptop A's MAC address and IP Address on Laptop C's ARP table, but it will also drop A's request because the requested IP address does not match C's IP address.

Laptop B will receive A's ARP request. Laptop B will record A's MAC address and IP Address onto its ARP table and check if the requested IP address matches B's IP address. Once it confirms the match, B will create an ARP Reply message containing B's IP address and MAC address. B will send this message via layer 2 as a unicast message.

The Switch's CAM table

MAC Address	Port
02:18:98:74:5d:44 (Laptop A)	Port 1

The CAM table record's A's MAC address before flooding all ports. None of the other laptops have sent a message to the Switch, so only A is known.

A's ARP table is empty because at this stage, Laptop B hasn't replied yet. Nothing has responded to Laptop A at this point.

B's ARP table:

MAC Address	Associated IP Address
02:18:98:74:5d:44	192.168.1.1 (Laptop A)

B receives A's ARP request and writes A's IP address and MAC address.

C's ARP table:

MAC Address	Associated IP Address
02:18:98:74:5d:44	192.158.1.1 (Laptop A)

C receives A's ARP request because the message was broadcasted into the entire domain, so C records A's MAC address and IP address.

d) (10 pts) If laptop B sends an ARP Reply to laptop A what do each of the elements learn, if anything? Identify the protocol steps (layer2 & layer3) and addresses. What is stored in the ARP tables of each host and the CAM table of the switch?

B sends an ARP Reply containing the source IP Address of 192.168.1.2 and source MAC Address of 02:18:98:74:5d:54. This message will be sent to the destination IP Address of 192.168.1.1 and the destination MAC Address of 02:18:98:74:5d:44.

The switch receives B's ARP reply and associates B's MAC address to Port 2. The switch already knows that laptop A's MAC address is on Port 1, so it will forward the message directly to Laptop A. Nothing is sent to port 3 because the ARP reply is unicast. Laptop A receives B's ARP reply and IP address. Laptop A records B's IP address and MAC address into its ARP table.

CAM Table of the switch

MAC Address	Port Number
02:18:98:74:5d:44 (Laptop A)	Port 1
02:18:98:74:5d:54 (Laptop B)	Port 2

The CAM table doesn't hold anything for port 3 or laptop C because it never got a message from C.

A's ARP Table

MAC Address	IP Address
02:18:98:74:5d:54	192.168.1.2 (Laptop B)

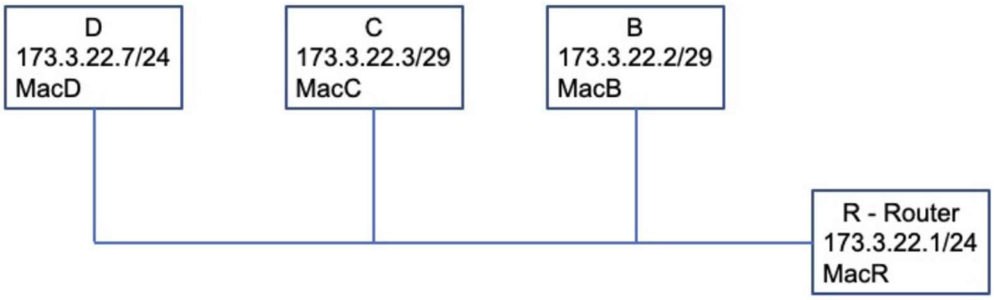
B's ARP table

MAC Address	IP Address
02:18:98:74:5d:44	192.168.1.1 (Laptop A)

C's ARP table

MAC Address	IP Address
02:18:98:74:5d:44	192.168.1.1 (Laptop A)

2) (10 pts) In the following network diagram, all computers are on the same shared Ethernet segment. If B sends an ICMP Request to 173.3.22.7 what does each of the hosts learn and how do they respond?



D's /24 subnet is includes the usable IP address range of 1-254 for the subnet mask (not counting the broadcast and network address). While B and C share the same subnet mask of size /29, they are inside the same subnet and are in the same IP address range as D's subnet. B and C shares the same subnet, and they are both overlapping with D's subnet. D's IP address also happens to end in a 7 which, on a /29 subnet, would be the broadcast address. This address, however, is outside of the available IP range of C and B's /29 subnet (1-6), meaning that D is outside of C and B's broadcast domain. The router R is also on 173.3.22.1 which is within range of all 3 hosts, meaning that all 3 hosts can reach the router.

When B attempts an ICMP request to 173.3.22.7 (Mac D), B will first check its ARP table to see if there is an associated MAC Address with that IP address. If B doesn't see one, it will create an ARP Request and broadcast it throughout B's subnet. This ARP Request will reach C where C will record B's MAC Address and IP address, but C will ultimately drop B's ARP Request as C's IP Address will not match the requested IP address.

Router R will receive B's ARP request and broadcast it on every line. This allows D to receive B's ARP request. D will record and associate B's MAC address and IP address received from the ARP request. D will check and confirm that it holds the destination IP Address that B was looking for. D will then send a unicast ARP reply to B, but because B is within D's subnet range, D can check its subnet mask to see B is within range and send its ARP reply directly to B.

B will then receive D's ARP Reply and record the MAC address of D into B's ARP table. B will then have to send the ICMP request itself and checks D's subnet mask and notice that it doesn't match. This is because while B is inside D's subnet, D is outside of B's subnet.

B will need the assistance of the router R. B will check Router R's subnet mask and see that it matches with B. B can check if R is in B's subnet range which it is, so B encapsulates the ICMP request within an Ethernet frame and send it to Router R.

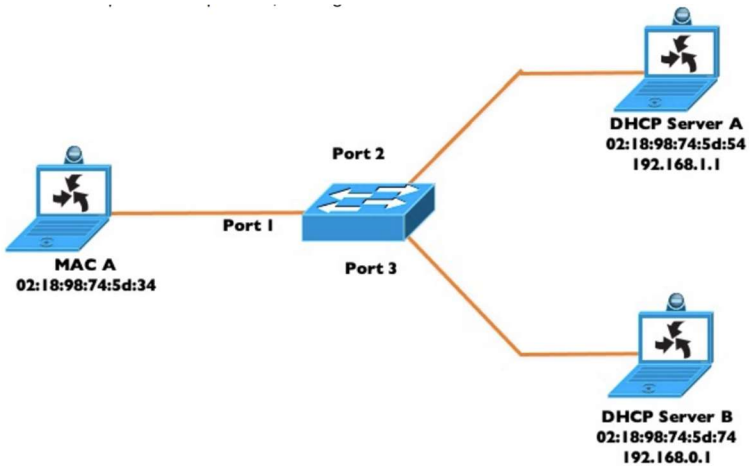
Router R will receive the message and check for the location of D. It will check the subnet mask and find that D is within R's range. R will check the IP address to see that D is indeed within range and send B's ICMP message to D.

Once D receives B's message, it will unravel the message and read B's ICMP request. It will create its own ICMP reply message, encapsulate it within an Ethernet frame and check the ARP table for B's MAC address and IP address.

Because D already recorded B's MAC address, it will send the ethernet frame directly to B instead.

- C learns about B's MAC Address and IP Address from B's ARP request but drops the message.
- B learns about D's MAC address and IP Address from D's ARP Reply sent directly to B.
- D learns about B's MAC address and IP address from B's ARP request from the Router R.

3) (20 pts) Describe the DHCP operations that occur when MAC A wants to obtain an IP address. Be specific as to protocol, messages and addresses.



DHCP Discover

Source IP Address	0.0.0.0 (Local Host)
Source MAC Address	02:18:98:74:5d:34 (MAC A)
Destination IP Address	255.255.255.255 (Broadcast)
Destination MAC Address	FF:FF:FF:FF:FF:FF (Broadcast)

Mac A starts with no IP address, so it needs to receive one from a DHCP server. It doesn't know what is on the network, so it has to broadcast a DHCP Discover message via UDP on port 67 to the ethernet network. The DHCP Discover contains a source IP address of 0.0.0.0 sent to the broadcast address of 255.255.255.255 with the Source MAC Address of 02:18:98:74:5d:34 attached to the message. The destination MAC Address will use FF:FF:FF:FF:FF:FF because MAC A doesn't know what's on the network nor what MAC Addresses any of the other hosts have. The 0.0.0.0 IP address is used because MAC A has no IP address, so it has to use an agreed upon address designated as the localhost. The switch will receive this message and flood all ports with this message. It will also associate port 1 with the MAC address of A.

DHCP Offer

Source IP Address	192.168.0.1 for Server B 192.168.1.1 for server A
Source MAC Address	02:18:98:74:5d:54 for server A 02:18:98:74:5d:74 for server B
Destination IP Address	The leased IP Address given by server A or server B
Destination MAC Address	02:18:98:74:5d:34 (MAC A)

DHCP Server A and DHCP Server B will receive A's DHCP request message. They will respond by sending their own DHCP offer messages. This message contains an IPv4 Address

within their possible range along with a lease timer. The IPv4 Address being leased out along with other settings are granted by what configurations the DHCP servers had set up prior to providing IP Address leases.

DHCP Server A will possibly send a DHCP Offer message to lease the IPv4 Address 192.168.1.2 or higher. This is sent on UDP Port 67 and contains the leased IP Address of 192.168.1.2 as the Destination IP Address and A's MAC address as the Destination MAC address. It also attaches server A's IP Address of 192.168.1.1 as the Source IP Address and server A's MAC address of 02:18:98:74:5d:54 as the Source MAC Address.

DHCP Server B will possible send a DHCP Offer message leasing the IP Address of 192.168.0.2 or higher. This will be sent on UDP Port 67 with the Source IP Address of 192.1.168.0.1 (DHCP Server B's IP Address) and a Source MAC Address of 02:18:98:74:5d:74 (Server B's MAC Address). This message holds the Destination IP Address of the leased IP Address 192.168.0.2 and the Destination MAC Address that Mac A holds, 02:18:98:74:5d:34.

Both servers will send their offers as a DHCP Request packet on UDP port 67. The switch will receive both of these messages and forward them to port 1 towards MAC A. This also allows the switch to associate Port 2 with DHCP Server A and Port 3 with DHCP Server B as the switch will read the MAC addresses provided by the servers. From this point on, the switch will be able to forward any messages to their respective ports because the switch's CAM Table has now recognized Mac A, DHCP server A, and DHCP server B. Because it recognizes the ports based on MAC Addresses, the IP Addresses will not change the switch's behavior.

After sending their messages, a timer begins where if there is no response by the time the clock finishes, the lease provided by the DHCP Request will automatically be rescinded.

DHCP Request

Source IP Address	0.0.0.0 (Local Host)
Source MAC Address	02:18:98:74:5d:34 (MAC A)
Destination IP Address	255.255.255.255 (Broadcast)
Destination MAC Address	FF:FF:FF:FF:FF:FF

Mac A will receive both DHCP Offers and chooses which one to accept. Usually, the first DHCP offer received is the one chosen. Mac A responds by echoing back the DHCP offer message. Mac A writes a DHCP Reply which contains the IP Address configurations of the offer that Mac A received. Mac A broadcasts the DHCP reply message to all ports to ensure that any other outstanding offers are dropped.

This gets sent to the switch which sends it out to all ports due to the message being a broadcast. DHCP A and DHCP B gets this reply and check to see if the configurations match the offer they sent out. If one does not see a match, then they rescind their original offer and reclaim the IP address they offered. The server that does see a match will respond with a DHCP Acknowledgement message back to Mac A to confirm that Mac A can use the address.

The DHCP Request message contains the destination address of 255.255.255.255 and the destination MAC address of FF:FF:FF:FF:FF:FF. This is because the DHCP Request message is broadcasted to the network. The source IP address will still be 0.0.0.0 because the client still needs an acknowledgement from the server before using the IP Address, but it will provide its own MAC address as the source MAC address.

DHCP ACK

Source IP Address	The IP Address of server A or B, depending on which offer MAC A accepted.
Source MAC Address	The MAC Address of server A or B, depending on whomever MAC A accepted
Destination IP Address	The leased IP Address accepted by A.
Destination MAC Address	02:18:98:74:5d:34 (MAC A)

When the DHCP server receives the DHCP Request message that matches their lease configurations, that particular DHCP server will return a DHCP Acknowledgement message back to the MAC A. The DHCP ACK informs MAC A that the IP address has now been granted to the client and that MAC A is free to use the IP Address now.

The DHCP ACK is sent on UDP Port 67 towards a Destination IP Address based on what server B or server A provided with A's MAC address being the Destination MAC Address. DHCP Server A or DHCP Server B will provide their own MAC Address and IP Address as the Source IP Address of this ACK message depending on which lease MAC A accepted.

The switch picks up the DHCP ACK message and forwards it directly to port 1 towards Mac A. Mac A receives the DHCP ACK message and can now use the IP Address provided until the lease expires or is released.

4) (10 pts) traceroute is a program that shows all the routers (IP address) an IP packet goes through to reach its destination address. Knowing the IP packet data structure and the IP protocol explain how traceroute works.

Traceroute is a program that takes advantage of ICMP's TTL (Time to Live) header. The client sends an IP packet to the intended destination IP address, but it starts by sending a packet with a TTL of 1. In ICMP, the TTL is decreased every time the packet reaches a router. Normally, this is used as a safety measure for packets stuck in infinite loops. When the TTL reaches 0, the router that sees the 0 TTL will drop the packet and reply with an ICMP Time Exceeded error message to the source IP address.

This error message is sent to the source IP address, but it will also include the IP Address of the router that dropped the packet. When the source client of the traceroute receives this message, it will learn the IP address of the router that the packet stopped at.

The goal of traceroute is to identify every router used to reach its destination, so during the traceroute program, the source will send a packet with a TTL of 1, receive info about the first router that receives the packet. It sends a packet with a TTL of 2 and sees the second router that

receives the packet. The client will keep sending packets with one more TTL than the previous ICMP packet until one packet reaches the intended destination. Every successive router that received a packet with TTL of 0 will return an error message and provide its own IP address.

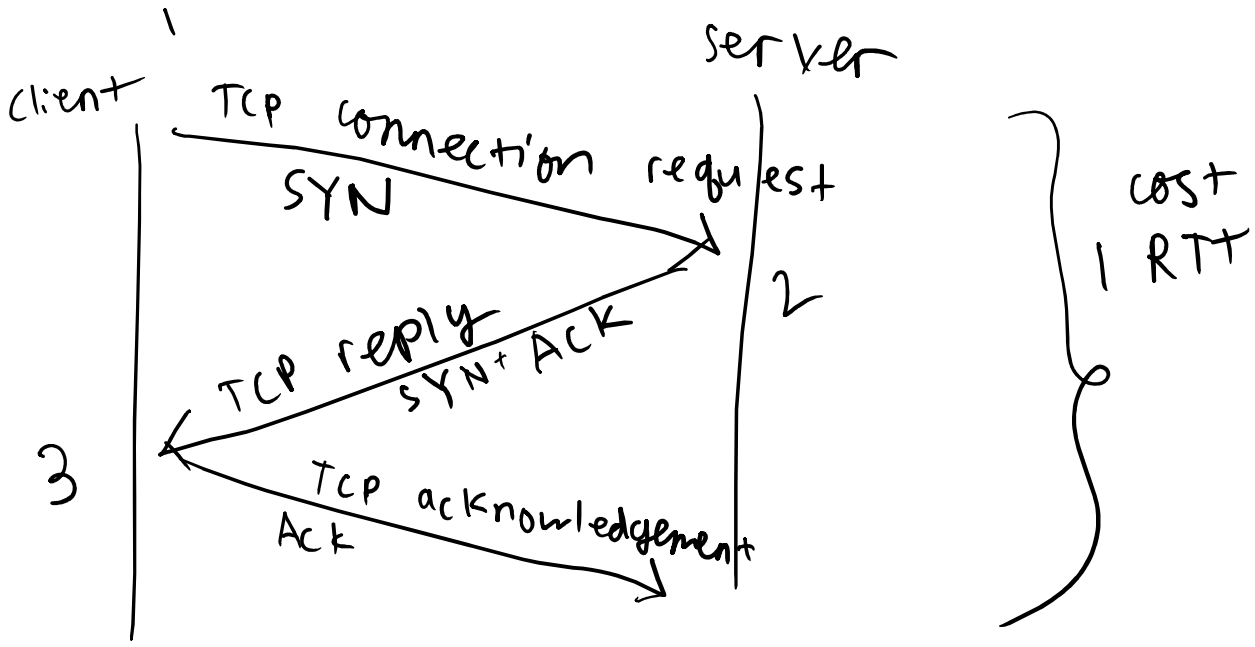
This provides the client with a list of every router that every traceroute message jumped to on its way to the destination, effectively providing a path of every router it crosses on the way to the destination IP address along with the IP Address of every router found along the way.

5) (30 pts) Explain how TCP works; sequence numbers, ack numbers, receive window, sliding window, etc. You may use one or more diagrams to help with your explanation.

When a client needs to send a message to a server, they both need open sockets for open ports. Both sides open their own sender buffers and receiver buffers. The server will open a TCP socket and bind it to an open port while the client socket will use a random port assigned by its OS. The server will open its socket and remain on standby until the client attempts to reach the server.

The client and server will initiate a 3-way handshake where the client will send a TCP message with a special flag to the server. TCP segments come with a flag representing 6 possible options. In this case, the client will send a TCP segment with the SYN flag to initiate a TCP connection with the server. The server will see this request and reply with a TCP message containing the SYN + ACK flag to inform the client that the server is willing to open a TCP connection. The client will receive the message and send a TCP Acknowledgement segment back to the server. Once the server receives the TCP Acknowledgement, the connection has been established and now data can be sent between them. The 3-Way Handshake process costs 1 RTT to establish. Each frame shared between the client and server must be acknowledged, costing 1 RTT for every segment.

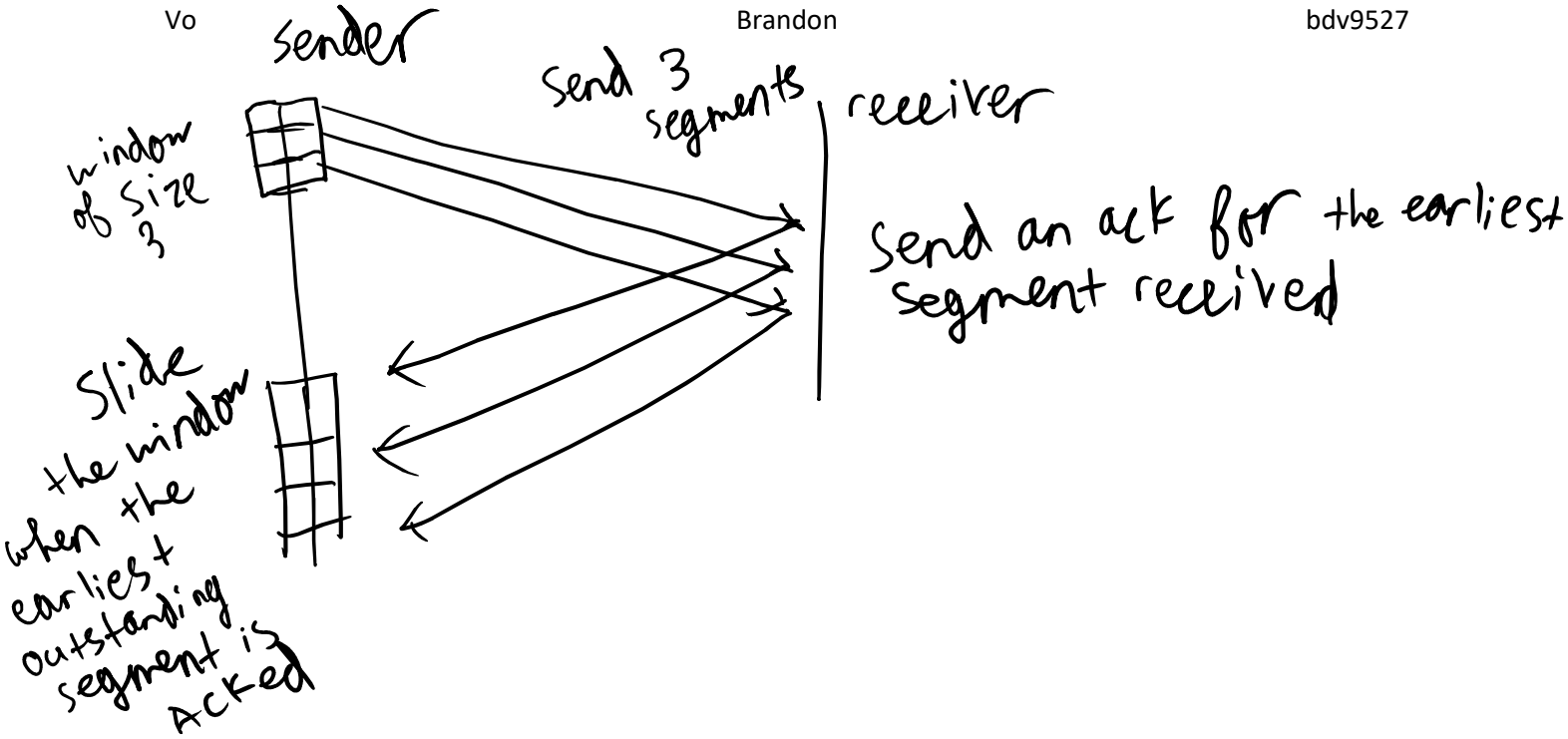
How many packets are to be placed in a frame is determined by other functions added on top of the TCP connection such as Nagle's algorithm. Under Nagle's algorithm, the sender will wait additional time to accumulate as many bytes as it can before sending out a frame. Once the frame is full or if an acknowledgement from a previous segment has been received, then the sender will transmit the frame. This reduces the payload by sending as many full frames of data as permitted.



The TCP 3-way handshake

Whenever the client or server wants to send something to the other machine, the sender will fill their send buffer with as many bytes as possible and transmits if the buffer is near max capacity or if the sender was prompted to immediately send; this buffer gets sent over the TCP connection where the receiver can read the packet through the receiver's own receive buffer. If the data being sent is considered urgent, then the segment will be sent with an URG flag to signify the importance to the receiver.

Because TCP is a connection-oriented protocol, it requires a method to ensure that every packet sent out can be received and rearranged to be in-order. Every TCP segment contains the ports of the source and the destination, its own sequence number, and a checksum calculated by the sender on top of the data being sent. In order to determine the order of packets being sent, each TCP segment is given a sequence number based on the first byte of a packet along with the size of the bytes from the previous TCP packets sent out. This identifies packets based on byte size so that out of order packets can be identified if there is a jump in sequence numbers of frames being received.



TCP Sliding Window Protocol

What the sliding window protocol does is provide a method of transmitting multiple packets of data while also keeping track of what has and hasn't been received.

The Sliding Window Protocol monitors what frames have been sent and what have been acknowledged. The sender sends TCP segments with a sequence number which must have a corresponding acknowledgement number. The receiver will send these acknowledgement numbers in various ways depending on what method specified such as delayed ACK, but the receiver must acknowledge the earliest segment received that still fits its receiver window. The server must respond with a TCP ACK message to inform the sender that the segment has been received and needs to slide its window to send the next set of segments.

Sliding window begins when the sender needs to send out a collection of frames to a receiving host. It will construct a window that determines the amount of frames that can be sent out within the window's range. The size of this window will vary as long as the window is smaller than the sequence number range of the frames being sent out; this prevents overlapping. The sender will transmit every frame within its window and begin a timer for each frame. These segments have a sequence number attached to them and require an acknowledgement that matches their number before the sender's window can slide right.

The receiver will have its own window determining what frames need to be received and acknowledged before moving. Once it receives those segments, it checks the sequence number where it will send an acknowledgement message based on the earliest frame received. It will also slide its own receiver window only when the earliest packet has been received. This system makes it so that the sender and receiver can detect gaps in their window buffer, allowing the client or receiver to guess what particular segment was lost in transmission.

If the corresponding acknowledgement number for the segment has not received (identified with a timer for that particular frame expiring), then there is a sign that the segment was lost in transmission, prompting the sender to retransmit the frame until an acknowledgement that matches the retransmitted segment has been received. When receiving an acknowledgement, the sender's window buffer will slide right only if the earliest outstanding frame has been acknowledged.

What this allows for is that even if the segments are out of order, the receiver can check the sequence number of the segments and essentially rearrange their order and see if there is a missing frame.

On top of making sure every segment is in-order, TCP also requires that every packet was sent correctly without any lost. When the receiver gets a segment, it will check the checksum of that packet. A checksum is the sum of the bits of all relevant information added up, so if the checksum doesn't match the calculated sum of the TCP header, then it means the segment suffered from interference and lost some data along the way. This may warrant a retransmission. Otherwise, a matching checksum identifies that nothing has gone wrong with the data.

Once all the data has been transmitted, the client will send a request to terminate the connection with the server. The client will send a TCP segment with the flag FIN. The server will see the request and return an acknowledgement message to the client. The client returns its own acknowledgement to tell the server that it will now terminate the TCP connection. The server and client will terminate the sockets that correspond to the ports being used for this TCP connection, ending the TCP communication process.

Alternatively, if there was an issue with the TCP connection on one side, one of the machines can send a segment with the RESET flag as an emergency shut down for the connection.