

Week.11.txt

- March 22nd, 2021
 - IEEE 802.11: Multiple Access
 - Once a station associates with an access point, essentially, it has established connectivity with the Wi-Fi network
 - The next step is for the station/host to transmit or receive data
 - The shared medium nature of Wi-Fi networks, similar to Ethernet's shared medium nature, requires a mechanism to arbitrate which station/host gets to transmit data frames at what time
 - This is the job of the media access control (MAC) layer
 - MAC is a sub-layer of the link layer
 - Ethernet utilizes CSMA/CD for medium access
 - For Wi-Fi, the mechanism is a little different
 - CSMA stands for carrier sense multiple access
 - This means that before transmission can occur, the station needs to listen to the medium to decide whether there are other ongoing transmissions
 - Wi-Fi transceivers are capable of implementing their own CSMA protocol. They can sense the medium, and make sure that its transmission will not interfere with ongoing transmissions
 - However, unlike Ethernet, there are some key differences
 - In Ethernet, the station/host can continue to monitor the power level in the medium to infer whether there are other transmissions that interfere or collide with its own transmissions
 - This is collision detection
 - Ethernet is able to infer, most of the time, whether a frame has been successfully delivered, because if there is no collision, then the frame was, most likely, correctly received by the receiver
 - This is the reason that Ethernet does not rely on any other mechanism for the purpose of triggering retransmission of frames
 - Typically, Wi-Fi stations cannot transmit and receive at the same time. If a Wi-Fi station is transmitting, then it has no way to determine if there are other ongoing transmissions happening around it, at the same time
 - Imagine a situation where you are in a shouting game with a friend. Both of you shout and speak very loudly to the point that you cannot hear each other, or anyone else. The volume is at a point where neither of you can detect if other people are talking at the same time
 - This is the situation with wireless devices. The proximity of the receiver's and transmitter's Wi-Fi interface card is the reason why they cannot detect

- Wi-Fi transceivers cannot detect ongoing transmissions, while it is transmitting
- Since Wi-Fi interface cards have a maximum distance that their waves can travel, it is possible that stations cannot detect if other station's are transmitting, due to the physical distance between them
 - This is referred to as the hidden terminal problem
 - Even if a particular Wi-Fi station is not transmitting, it cannot detect if a hidden terminal is transmitting, due to distance
- To summarize:
 - Ethernet uses CSMA/CD to detect collisions
 - It is random access
 - CSMA means that future transmissions don't collide with ongoing transmissions, because the device will listen to the medium before transmitting
 - Wi-Fi stations use CSMA/CA; it's different from Ethernet
 - It has no collision detection
 - All frames are transmitted to completion
 - Collision detection is not implemented because:
 - It is difficult to receive, or sense collisions, when transmitting due to weak received signals (fading)
 - Some collisions cannot be sensed/detected
 - i.e. Hidden terminals
 - Acknowledgements are sent for frames that are successfully transmitted
 - ACKs are needed, because without collision detection, you don't know if the transmission collided or not
 - The goal of CSMA/CA is to avoid collisions
 - 'CA' stands for collision avoidance

- i.e. Figure of Hidden Terminals

[illegible]

```

#                                     %          *                               #
#                                     %          *                               #
#      * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * #
#                                     %                               #
#                                     %                               #
#                                     %                               #
#                                     % % % % % % % % % % % % % % % % % % % % % #
#                                     % % % % % % % % % % % % % % % % % % % % % #
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

```

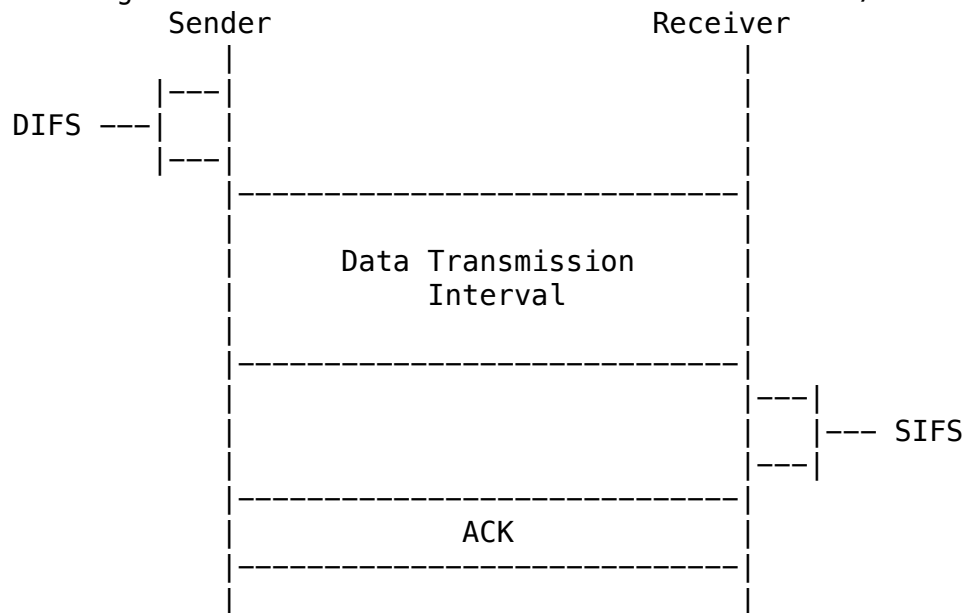
- The hidden terminal problem is illustrated by this figure
- In this figure, there are 3 stations:
 - 'A', 'B', & 'C'
 - 'A' & 'C' are transmitters
 - 'B' is a receiver
- The transmission radius of each station/terminal is shown in the figure
 - '*' represent the transmission radius of 'A'
 - '#' represent the transmission radius of 'B'
 - '%' represent the transmission radius of 'C'
- The transmission radius is the area that another station needs to be in, in order to receive the transmission
 - Anything outside of the transmission radius will not be able to detect the transmission from the corresponding station
 - i.e. If 'C' sends a transmission to 'B', then 'A' will not be able to detect it
- Assume that 'A' & 'C' have something to transmit to 'B'
 - Since 'A' & 'C' are on either side of 'B', their physical distance is large enough to the point that they cannot hear each other. This is because the signal propagation in the medium is so weak that by the time it is received by 'B', the other station cannot detect it. Hence, 'A' is a hidden terminal to 'C', and 'C' is a hidden terminal to 'A', because they cannot detect each other's signal. Thus, CSMA/CD cannot be used in a Wi-Fi setting; it can only be used in Ethernet
 - Instead of detecting collision, like Ethernet does, Wi-Fi uses a different mechanism, that is not as effective as CSMA/CD, called collision avoidance (CA)
- The media access control (MAC) protocol in Wi-Fi is called CSMA/CA
 - CSMA stands for carrier sense multiple access
 - CA stands for collision avoidance
- Medium Access Control Logic
 - From the perspective of an 802.11 sender, medium access control is as follows:
 - If the sender has a new frame it needs to transmit, the

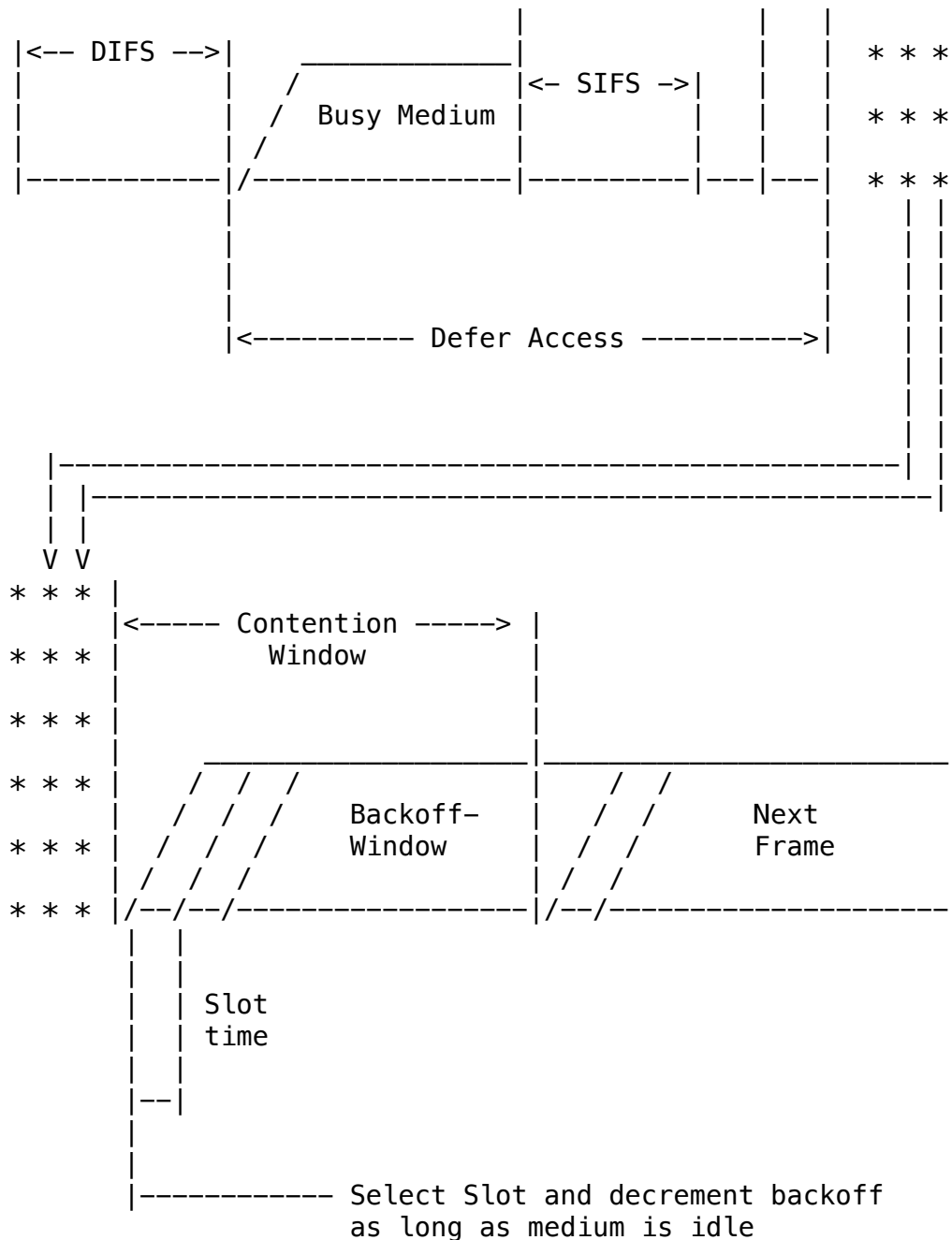
first thing it will do is perform carrier sensing. It will sense the channel to see if it is idle or not. The amount of time that the sender listens to the medium is a specific value called 'DIFS'; it is a very short time interval.

- If the sender does not detect any transmission within 'DIFS' time, then it will send the entire frame. During transmission, the sender cannot detect if there are other concurrent transmissions
- If the sender does detect transmission or collision in the medium, then the sender will try to retransmit the frame. Naturally, this means that the previous transmission experienced collision due to ongoing transmissions in the channel. In this situation, the sender will yield its transmission, and give the other stations an opportunity to send their frame(s). This is similar to Ethernet. Now, the sender chooses a value and starts a backoff timer. Meaning, the sender will wait for a period of time before it tries to transmit its data. During the waiting period, and while the medium is idle, the sender decrements the timer. If the medium is not idle, and there are ongoing transmissions, then the timer is suspended
 - If a timer is decremented as long as there are no other transmissions in the medium. However, if a station suddenly starts transmitting, then the timer is suspended/frozen. The timer will continue its countdown after the channel becomes idle again after 'DIFS' time. Once the timer reaches 0, the sender will transmit its frame
- After data has been successfully transferred from the sender to the receiver, the sender still has no idea whether the frame was successfully received by the receiver, because the sender or receiver cannot detect collision. Thus, the sender needs to rely on the receiver to send an acknowledgement if the transmission is successful
 - In the case that no acknowledgement is received, then the transmission has, most likely, failed. As a result, there may be contention in the medium. Thus, the sender needs to increase its random backoff interval in a similar manner to exponential backoff in Ethernet. Finally, the sender will try to retransmit the frame
- There is a maximum number of retries the sender makes before it gives up transmitting the current frame, and tries to transmit the next frame. The next frame might already be in the sender's queue
- The behaviour of a receiver in IEEE 802.11 protocol is

relatively straight forward:

- The receiver will simply listen to the medium for any incoming frames destined to it. The receiver uses the destination MAC address in the frame to determine if a particular frame is meant for it
- Once a frame is successfully received, a CRC check is performed to detect errors. If the frame passes the CRC check, then the receiver sends an acknowledgement to the sender
 - Acknowledgements are sent after a small time interval called 'SIFS'
 - 'SIFS' is slightly shorter than 'DIFS', and gives a higher priority to the transmission of acknowledgements
- Ongoing transmissions indicate that the medium/channel is busy
 - A station needs to stop transmitting before the medium can become available again
- An attacker can jam the medium/channel through constant transmission, and by completely ignoring the media access control (MAC) layer
 - However, typical user behaviour will not jam the medium, because users typically use Wi-Fi in intervals, which does not constantly transmit data
 - i.e. Surf the web, read a webpage, click link, read again, etc.
- In Wi-Fi media access, control is completely decentralized
 - A central coordinator is not required to assign schedules
 - Medium access control (MAC) in Wi-Fi is a distributed approach to solve medium contention
- i.e. Diagram of 802.11 Communication Between Sender/Receiver






- This diagram illustrates where interframes are utilized
- 'DIFS' is always utilized before the transmission of data frames
 - It is used by the station/sender to determine how long it needs to listen to the medium, before it can start transmitting
 - The amount of time determines whether the medium is idle or not
 - It is possible that after listening to the medium for 'DIFS' time, the station/sender determines that the medium is busy

- The station/sender will continue listening to the medium
- The station/sender needs to wait at least 'DIFS' time before it can start the counting down the backoff timer
 - Thus, the medium has to be idle for at least 'DIFS' time
- Once the station/sender successfully transmits a frame, the receiver will send an acknowledgement to the sender
 - The receiver (only) needs to wait a shorter period of time ('SIFS') to respond with an acknowledgement
 - SIFS is shorter than DIFS
- Suppose that at the end of a 'DIFS' for one station, another station wants to transmit a data frame. The other station is forced to listen, and wait until the medium becomes idle for at least 'DIFS' time. If its backoff timer reaches 0 before the end of 'DIFS' time, then the station's transmission will have a lower priority compared to the receiver sending an acknowledgement to the initial station
- Suppose that at the beginning of 'DIFS', a particular station wants to transmit a datagram, and its backoff timer is 0. Also, at this time, the receiver has just (successfully) received a packet/frame sent from a previous transmission. So, in this situation, because the acknowledgement utilizes a shorter interframe space ('SIFS'), it gets priority over the particular station's transmission. Thus, the acknowledgement will be sent first, and then the particular station gets to transmit its data frame to the receiver
 - Stations that have a data frame to send will now see a busy medium, due to the acknowledgement, and they will have to defer their transmission until the medium becomes idle again
 - This is the reason for different interframe spaces for different types of frames. In this example, the acknowledgement ('SIFS') gets (higher) priority over the data frame ('DIFS')
 - Note: This lecture does not cover other types of data frames such as 'EIFS'
- Note: 'SIFS' > 'PIFS' > 'DIFS' > 'EIFS'
- Different interframe spaces can be associated, or used with, different types of traffic in a Wi-Fi network
 - i.e. For streaming connections, it is better to use a shorter interframe spacing. However, for best effort service such as data transfer, regular 'DIFS' is sufficient
- This is one way to manipulate the priority of access for the transmission medium in 802.11
 - However, this class will stick to 'DIFS' and 'SIFS'

- Know the differences between them!
- Types of different interframe spacing:
 - Short IFS (SIFS)
 - Shortest IFS
 - Used for ACK, CTS, poll response, etc.
 - Utilized for immediate response actions
 - Point coordination function IFS (PIFS)
 - Midlength IFS
 - Used by centralized controller in PCF scheme when using polls
 - Distributed coordination function IFS (DIFS)
 - Longest IFS (data, RTS)
 - Used as minimum delay of asynchronous frames contending for access
 - Extended Interframe space (EIFS)
 - Used when received frame contains errors
- The relationship between the interframe spaces is:
SIFS > PIFS > DIFS > EIFS
- More On Medium Access Control Logic
 - The backoff timer is chosen uniformly between 0 and the maximum value
 - The maximum value is called the 'contention window size'
 - Typically, it is multiples of 'slots'
 - A 'slot' is a very short time interval
 - Typically, it is several microseconds, but it mostly depends on the version of 802.11
 - i.e. 802.11ac has a slot interval of 9 microseconds; earlier versions have a longer 'slot' interval
 - The contention window, and backoff timers, are measured in units of slots
 - There is some similarity between the congestion window and the contention window
 - The contention window controls the average backoff time a station needs to wait before it attempts its transmission
 - In a way, it regulates how fast a station can pump/inject data into the network
 - In 802.11, the contention window starts from 7
 - This is the minimum value
 - Initially, a station will select a backoff value between 0 and 7. As the station starts to experience contention, the contention window increases exponentially, utilizing the following formula: $CW' = (((CW + 1) * 2) - 1)$
 - The next contention window size is 15, followed by 31, and so on
 - The maximum CW window size is denoted as 'CW_max'
 - The station will try a maximum of 7 times, so 7 retries/retransmissions, before it gives up
 - i.e. Diagram of Congestion Window Increasing Exponentially



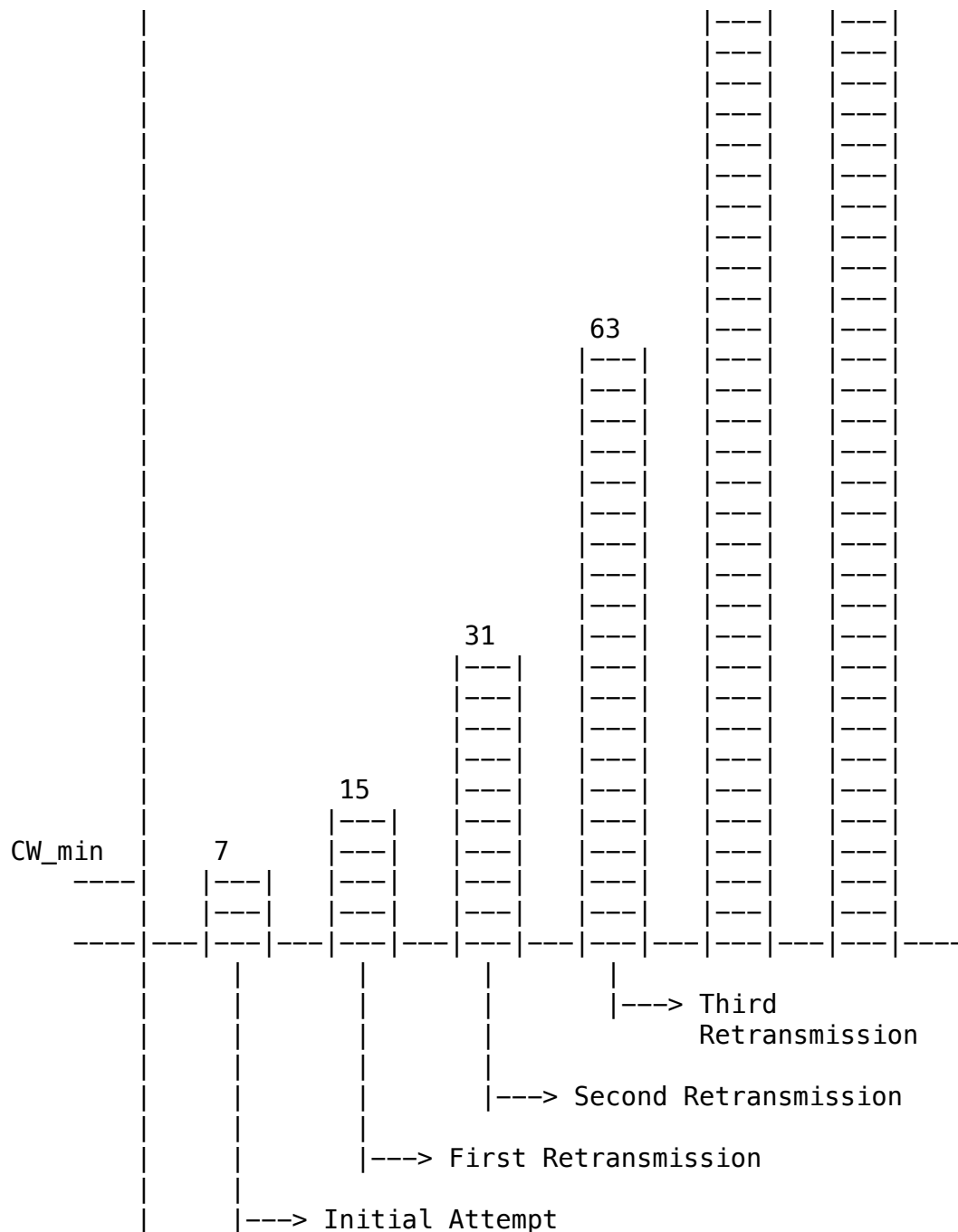
CW_max

255



127





- To summarize:

- The contention window is measured in slots
 - Each station maintains a contention window (CW)
 - Initially it is set to ' CW_{min} '
- Upon collision, $CW' = ((CW + 1) * 2 - 1)$ until it CW reaches ' CW_{max} '
 - The equation represents exponential backoff
- CW is reset to ' CW_{min} ' upon successful delivery

- Example

- i.e. Diagram of Wireless Stations Sending Packets

sides of receiver 'C'. Both stations want to send their frames to 'C'. This situation is the hidden terminal problem. The receiver, 'C', can hear both nodes, but node 'A' is not aware of node 'B', and node 'B' is not aware of 'A'. Node 'A' & 'B' cannot hear each other's transmission

- The backoff interval is given. It is randomly chosen between 0 and the contention window. Node 'A' has a backoff timer of 2,4. Node 'B' has a backoff timer of 2,3.
 - Note: Backoff timers will be provided on quizzes/ tests/exams
- The entire process starts at time 0, denoted by t_0 . Prior transmissions are not considered, and both nodes start their backoff timer at t_0 . At this point, the backoff timer is 2
- According to the CSMA/CA protocol, both stations will countdown their backoff timer. Assuming that there are no other transmissions in the medium, both stations will attempt to transmit their data/frames
 - Since 2 stations are transmitting at the same time, their signals will collide with one another at the receiver. Hence, receiver 'C' will not be able to receive either packets from node 'A' or node 'B'
- In a normal case, assuming receiver 'C' successfully receives a frame from either node without collision, 'C' will wait for a short interframe ('SIFS') time, and then 'C' will send an acknowledgement. In other words, for every frame that is successfully received by 'C', it will wait 'SIFS' time before sending an ACK for the corresponding frame
 - From the point-of-view of 'A' or 'B', they have no idea whether the frame was successfully received or not. Hence, they have to wait for an acknowledgement to come back until they can make further transmissions with 'C'
- Next, both stations, 'A' & 'B', will wait for 'SIFS' time, with the intention of receiving an acknowledgement from the receiver, 'C'. However, neither station gets an acknowledgement from 'C', because 'C' did not receive anything. Hence, it does not send an acknowledgement.
 - Note: 802.11 does not use negative acknowledgements; it only uses positive acknowledgements, when frames are successfully received
- Once station 'A' & 'B' realize that their previous transmission was not successful, they will try to retransmit. Assuming both stations have not exhausted their maximum number of retries, they will choose their next backoff counter based on exponential backoff, listen to the medium for 'DIFS' time, and then start the countdown for the backoff timer. The backoff timer for

node 'A' is 4, and the backoff timer for 'B' is 3

- The countdown of the backoff timer only starts after the stations detect the medium to be idle for at least 'DIFS' time
- Since 'B' has a smaller backoff timer than 'A', 'B' will finish its backoff counter before 'A'. When this happens, 'B' will transmit its data to 'C'
 - Since only 'B' is transmitting, its data will be successfully received by 'C'
- Once 'C' successfully receives data from 'B', 'C' will wait for 'SIFS' time before it responds with an ACK
 - During the transmission between 'B' & 'C', station 'A' will know that the medium is busy. Station 'A' will pause its backoff timer. Its timer starts at 4, gets decremented by 3, so the remaining counter is 1
- After 'C' sends an acknowledgement to 'B', station 'A' will sense the medium for 'DIFS' time, and conclude that the medium is idle. Now, 'A' will continue counting down its backoff timer. So, after 1 slot the backoff timer reaches 0, and now 'A' can successfully transmit its data to 'C'
 - Once 'C' successfully receives the packet/frame, 'C' will wait for 'SIFS' time, and then send an acknowledgement to 'A'
 - Note: Since the 'SIFS' interval is shorter than the 'DIFS' interval, the receiver can send an acknowledgement, and avoid collisions with other frames/packets
- Note: The access point does not serve as a coordinator that directs transmissions. Also, an algorithm like 'Round Robin' isn't used to ensure that everyone gets a turn to transmit. Instead, the stations/senders figure it out by utilizing a combination of:
 - Exponential backoff
 - Listening to the medium
 - Retransmitting data

Since the backoff timer is randomly selected from an interval, the stations/senders will select different backoff timers after a collision. Therefore, at least one station will be successful in transmitting a frame

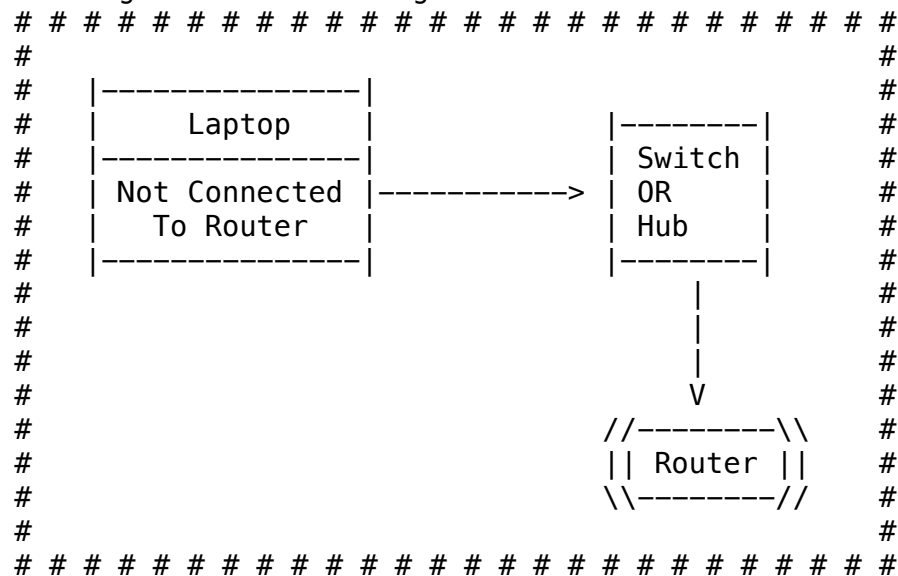
- Summary

- This concludes the discussion of link layer protocols. The covered content includes:
 - Ethernet
 - WLAN
 - DS, BSS, IBSS, ESS
 - Media access control (MAC) in Ethernet & Wi-Fi
 - CSMA/CA
 - Physical and virtual carrier sensing
 - Defer transmission after a busy period

[illegible]

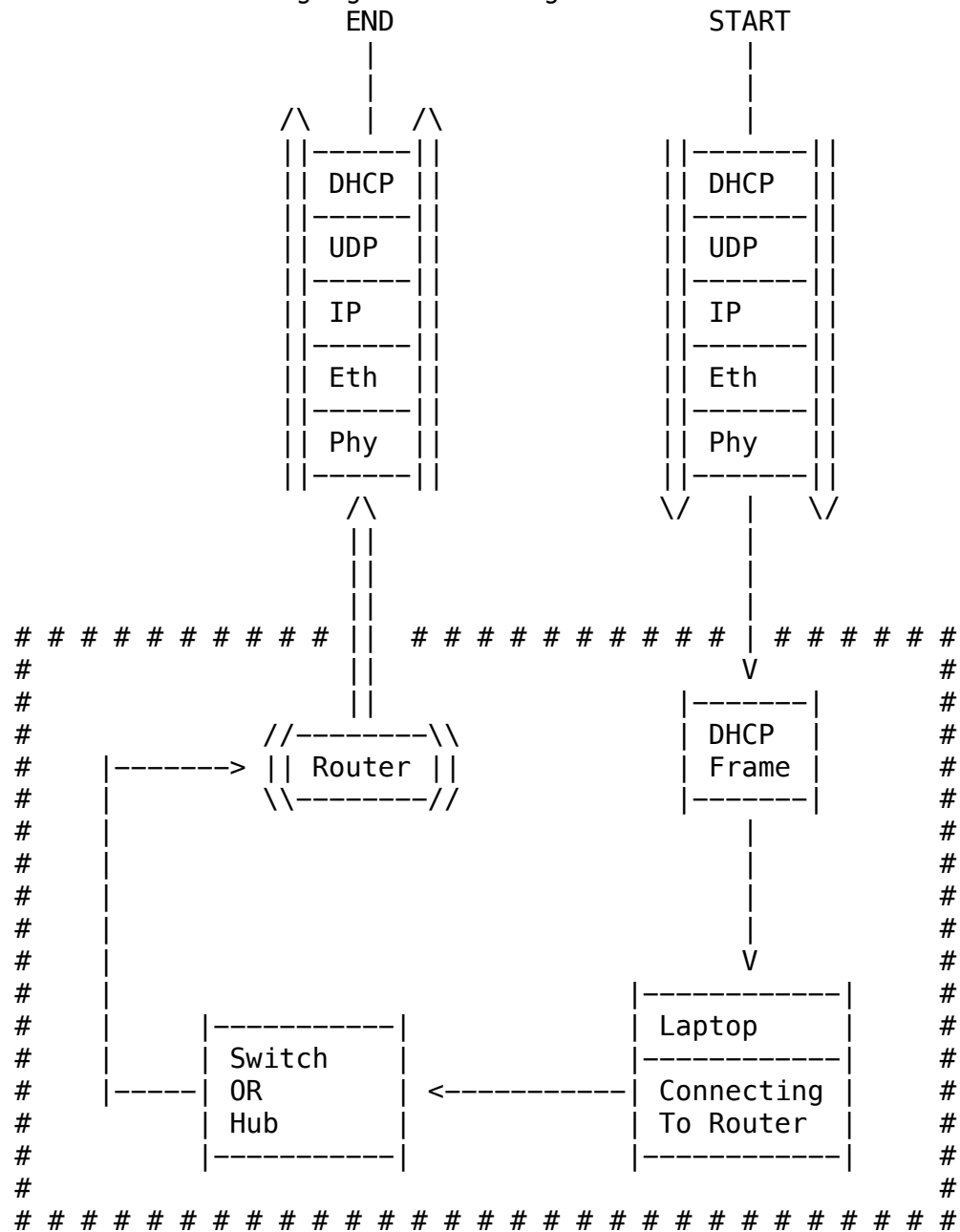
- device first connects to an access point
 - Can also be used to re-allocate an IP address
 - Used to configure the first hop router
 - ARP
 - Used to obtain the MAC address of a device connected to the same local area network
 - i.e. "What is the MAC address of X.X.X?"
 - Stands for address resolution protocol
 - Link Layer:
 - Wi-Fi
 - This is the connection between the laptop and the access point
 - Ethernet
 - This is the connection between the routers in the network
 - The protocols in the TCP/IP stack are either directly involved in the process of connecting, or they support it
- A Day In The Life: Connecting To The Internet (1)

- i.e. Diagram of Connecting to an Access Point



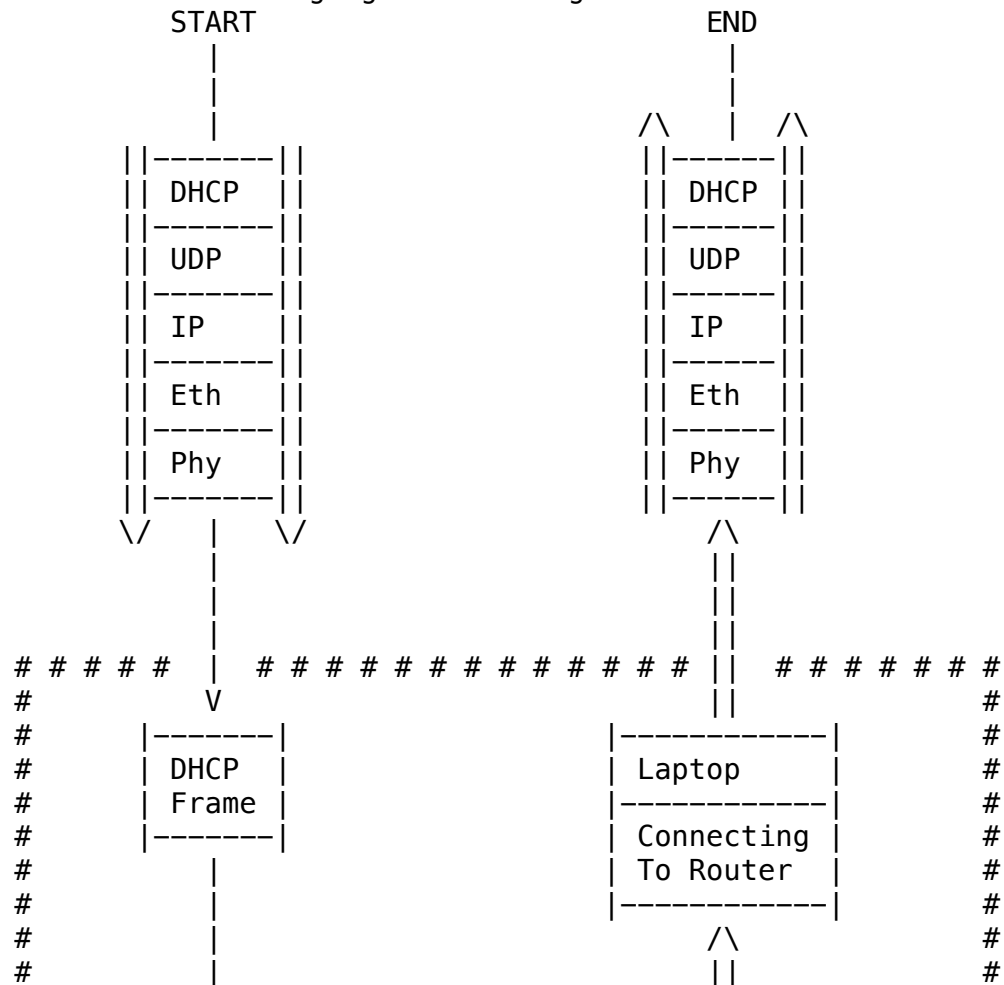
- Before a device can access the Internet, it needs to be associated with an access point; also known as router
- If a device (i.e. Laptop) connects to an access point via Wi-Fi, then the 802.11 protocol is used to facilitate the association process
 - First, the access point sends out beacon frames with information that allows the connecting device to create and send an association request message
 - Upon receiving an association request message, the access points replies with an association response message
- If a device is connected via Ethernet, then there is no exchange of association request/response messages

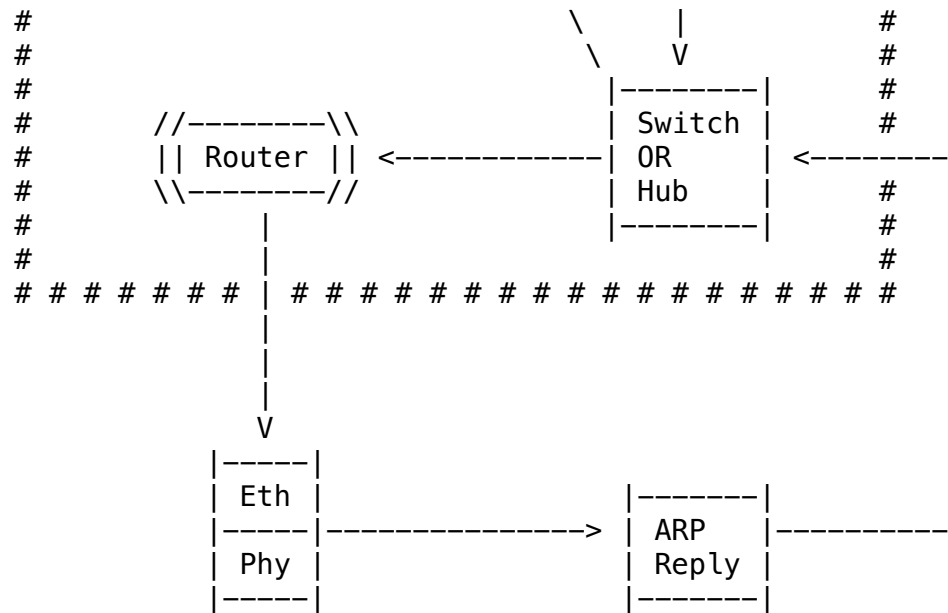
- Note: The router runs DHCP
- A Day In The Life: Connecting To The Internet (2)
 - i.e. Diagram of Setting Up Connection Between Access Point & Device: Exchanging DHCP Messages



- A device, such as a laptop, connecting to a network needs to get its own IP address, the address of the first-hop router, and the address of the DNS server
 - To accomplish this, DHCP is utilized
 - The connecting device uses a broadcast message to communicate with the router
 - The destination IP address of the broadcast message is FF-FF-FF-FF-FF-FF

- The switch forwards the broadcast message to all devices that are connected to it, including the router, which is running the DHCP server
- After a successful exchange of association messages, the DHCP protocol is utilized to allocate an IP address to the newly connected device, and it is used to obtain information about the first-hop router
 - DHCP requests are encapsulated in UDP, which are then encapsulated in IP, and finally the entire UDP/IP frame is encapsulated in 802.3 Ethernet, before it is sent throughout the Ethernet backbone
 - When devices receive a corresponding DHCP frame, then Ethernet is demuxed to IP, and then demuxed to UDP, and finally it is demuxed to DHCP
 - If the device is wirelessly connected to the access point (AP), then the access point will convert the 802.11 frame to an 802.3 frame
 - DHCP is a good example of where UDP is useful
- A Day In The Life: Connecting To The Internet (3)
 - i.e. Diagram of Setting Up Connection Between Access Point & Device: Exchanging DHCP Messages



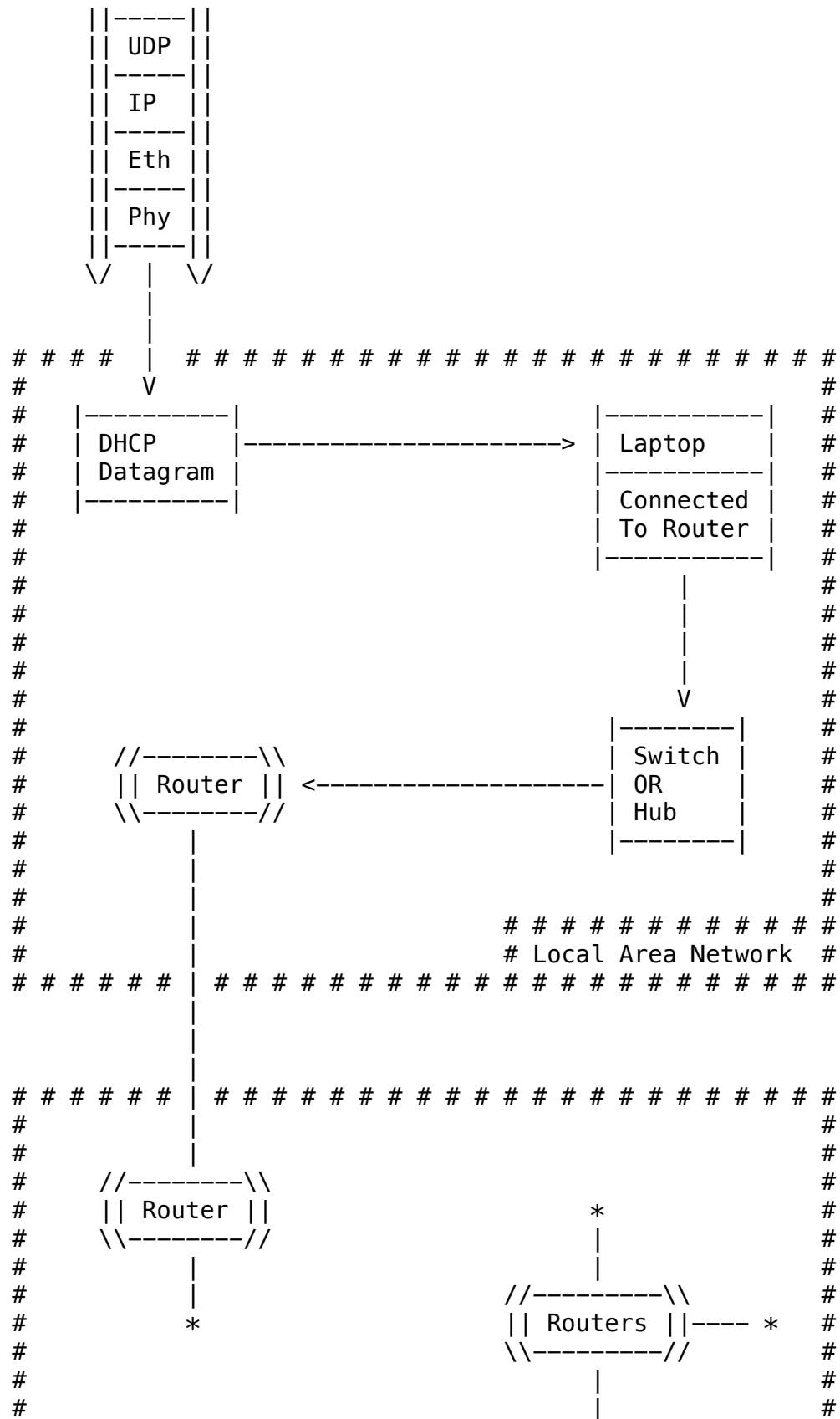


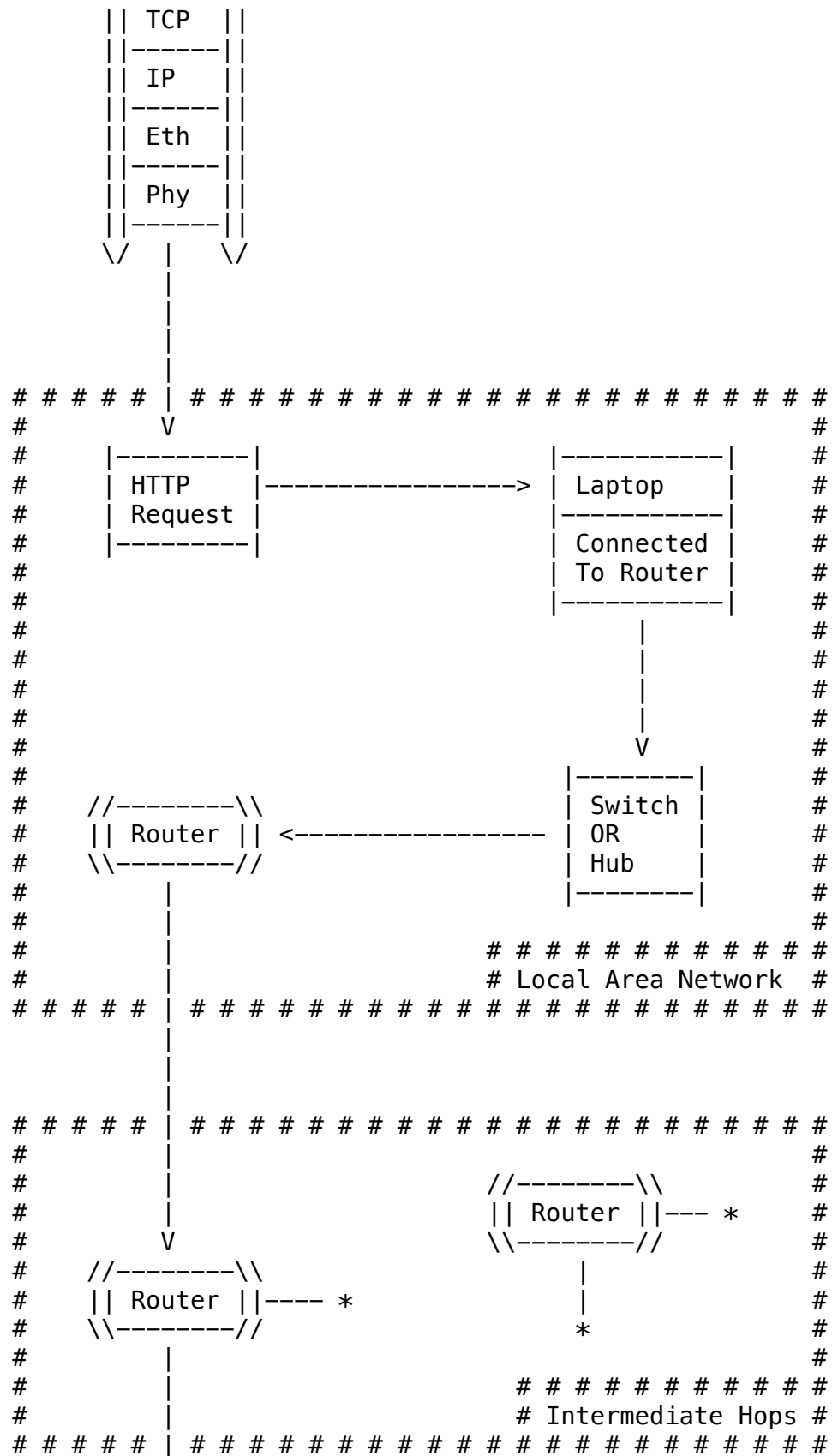
- Once an IP address is allocated to the connected device, the browser on the device can send HTTP requests. However, it needs to know the IP address of the web server
 - This is accomplished via DNS queries
- Typically, a DNS server, or servers, do not reside on the same local area network
- When a DNS query is created, it is encapsulated in UDP, and then it is encapsulated in IP, and finally the UDP/IP frame is encapsulated in Ethernet
 - Before this frame can be sent to the router, the connected device needs the MAC address of the router
 - The MAC address is obtained via ARP
- The connected device learns the IP address of the first hop router through DHCP. Now, it needs the MAC address of the first hop router
 - To obtain the MAC address of the first hop router, the connected device sends an ARP query broadcast message throughout the LAN, via link layer frame, to obtain the IP to MAC address mapping
 - When the router receives an ARP query, it responds with an ARP reply, which contains the MAC address of the router's interface
 - When the connected device receives the ARP reply, it locally caches the IP to MAC address mapping
- Once the newly connected device learns the MAC address of the first hop router, it can now send a frame containing a DNS query
 - UDP is used to send the DNS query
- A Day In The Life: Using DNS
 - i.e. Diagram of DNS Lookup

```

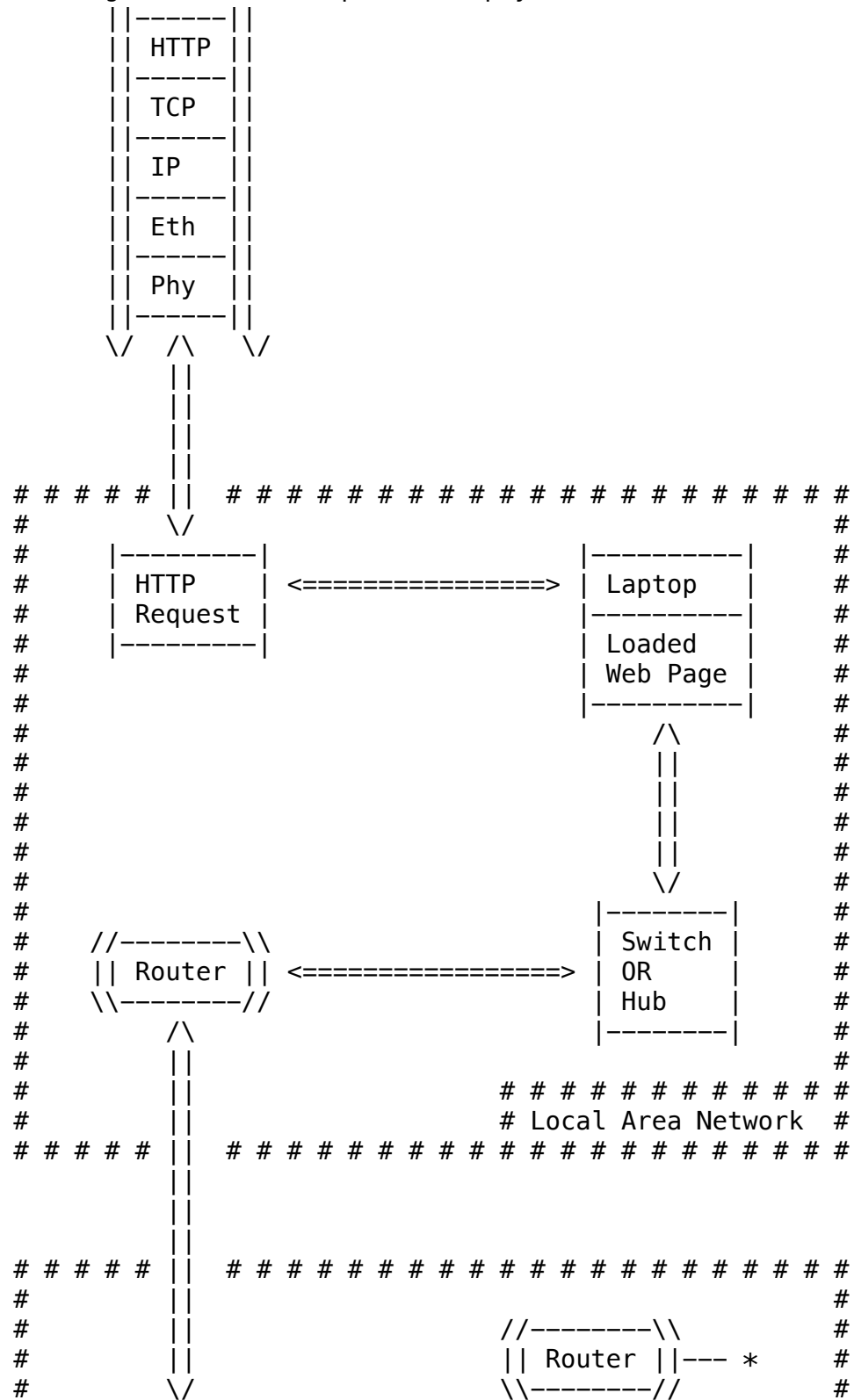
||-----||
||  DNS  ||

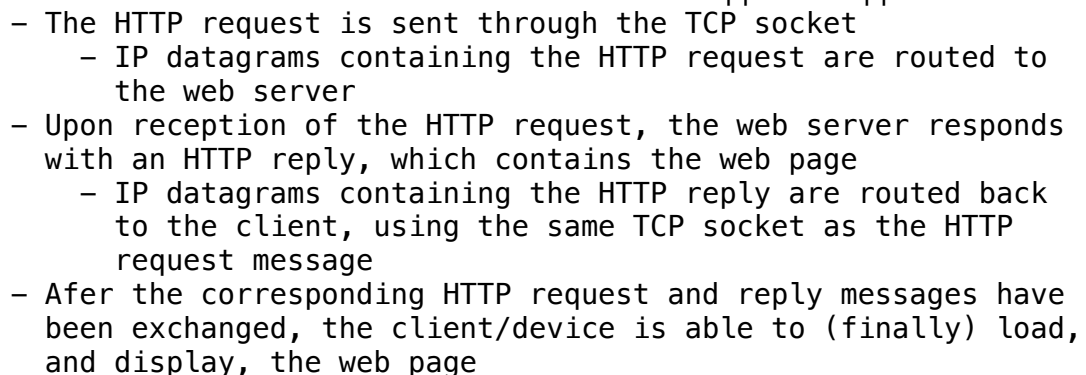
```



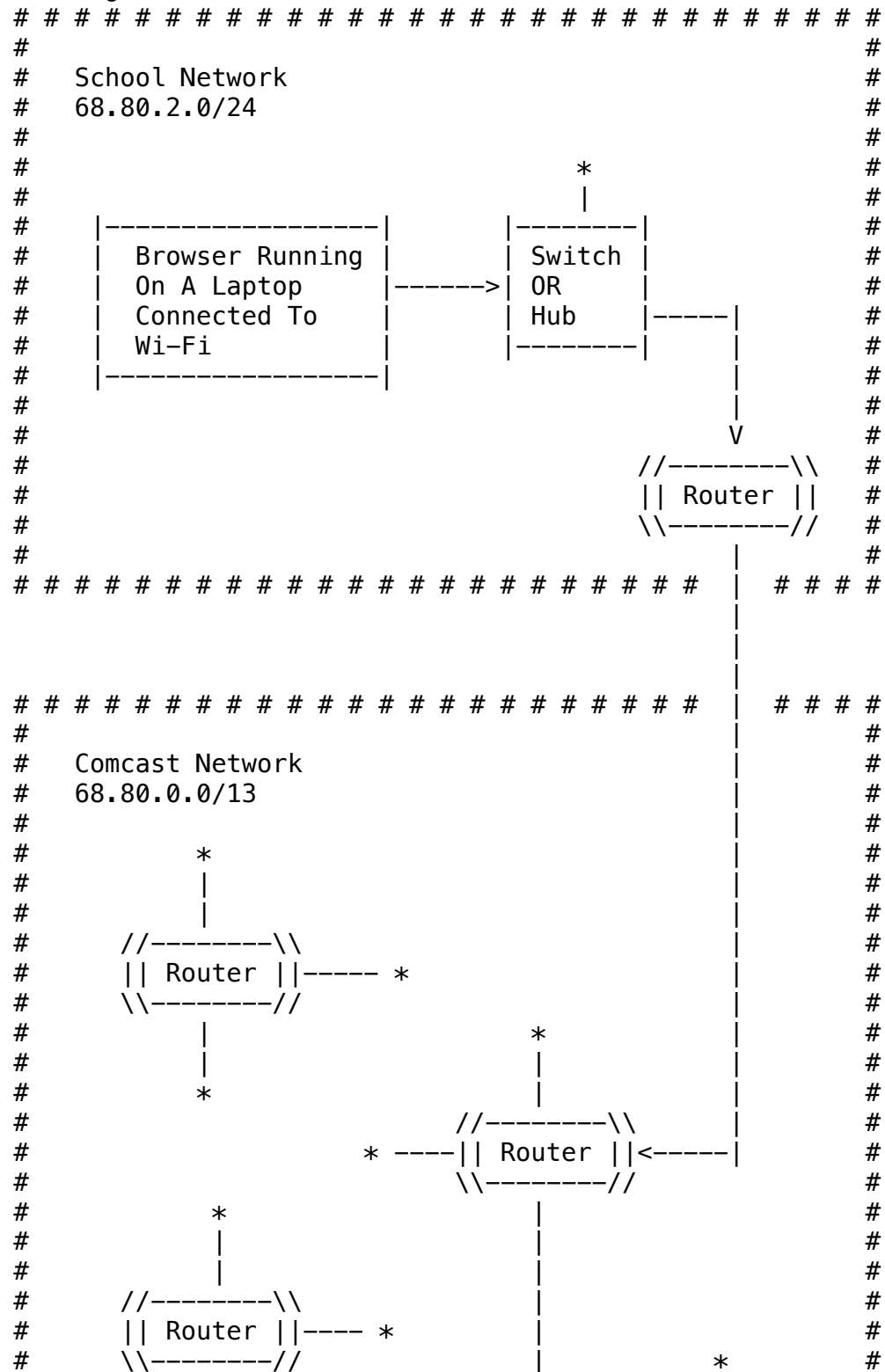


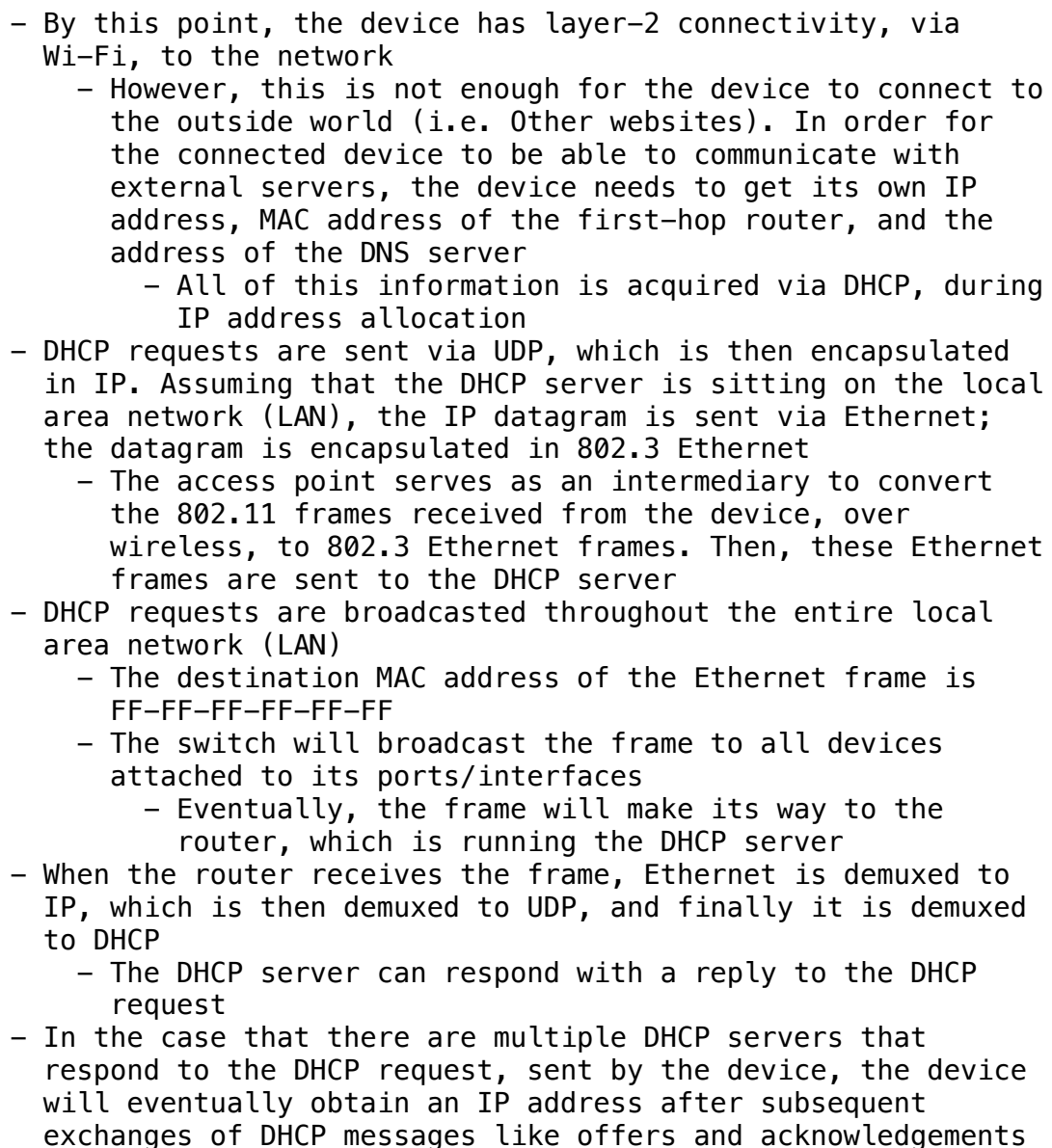
- routing protocols to populate the IP forwarding table
- A Day In The Life: HTTP Request/Reply
 - i.e. Diagram of HTTP Request & Reply



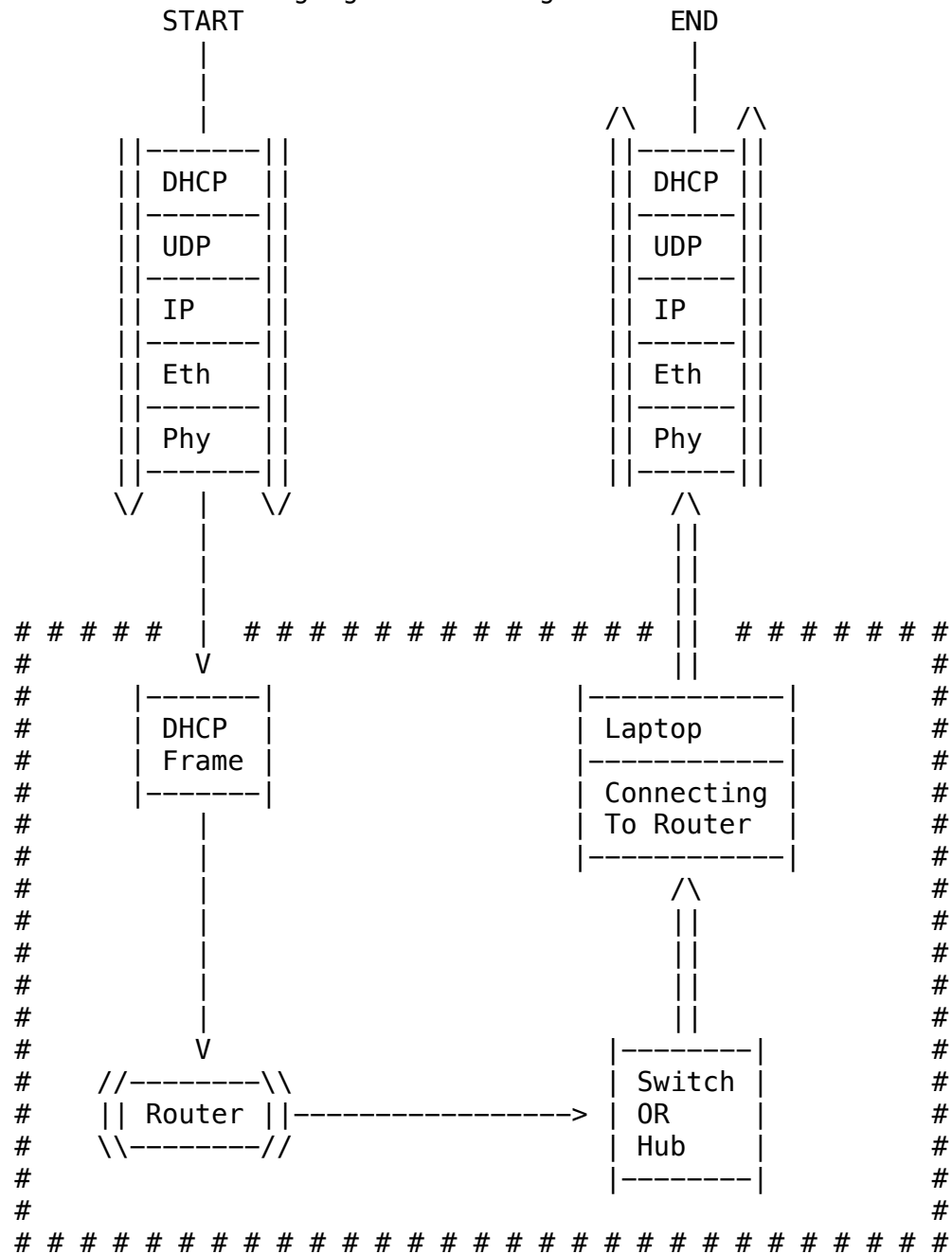


- March 24th, 2021
 - A Day In The Life: Scenario
 - i.e. Diagram of Scenario





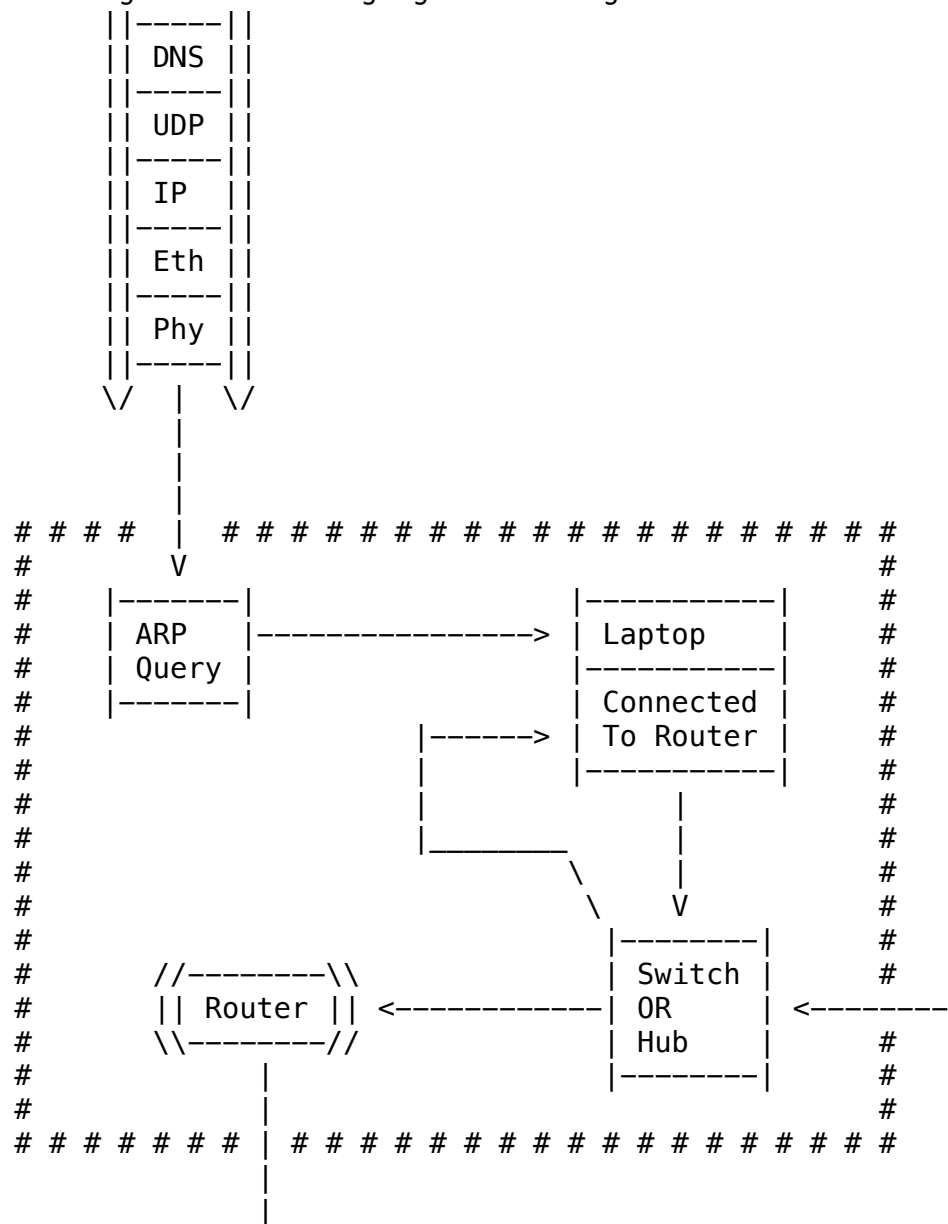
- A Day In The Life: Connecting To The Internet (3)
 - i.e. Diagram of Setting Up Connection Between Access Point & Device: Exchanging DHCP Messages

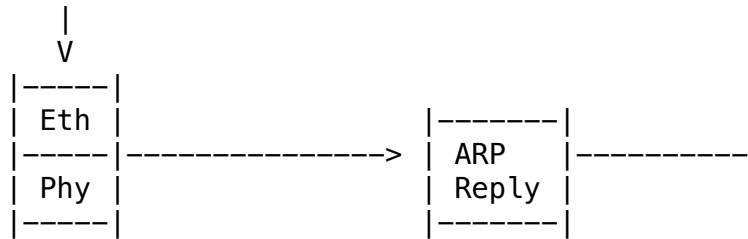


- All DHCP messages, or messages associated with DHCP, are encapsulated in UDP/IP
- The DHCP ACK, or offer, message to the client/device provides information such as:
 - Client's (allocated) IP address
 - IP address of the first-hop router (for the client)
 - Name & IP address of the DNS server
- Before the DHCP server sends a DHCP ACK for the client's

DHCP request, the ACK is encapsulated and then forwarded to the switch

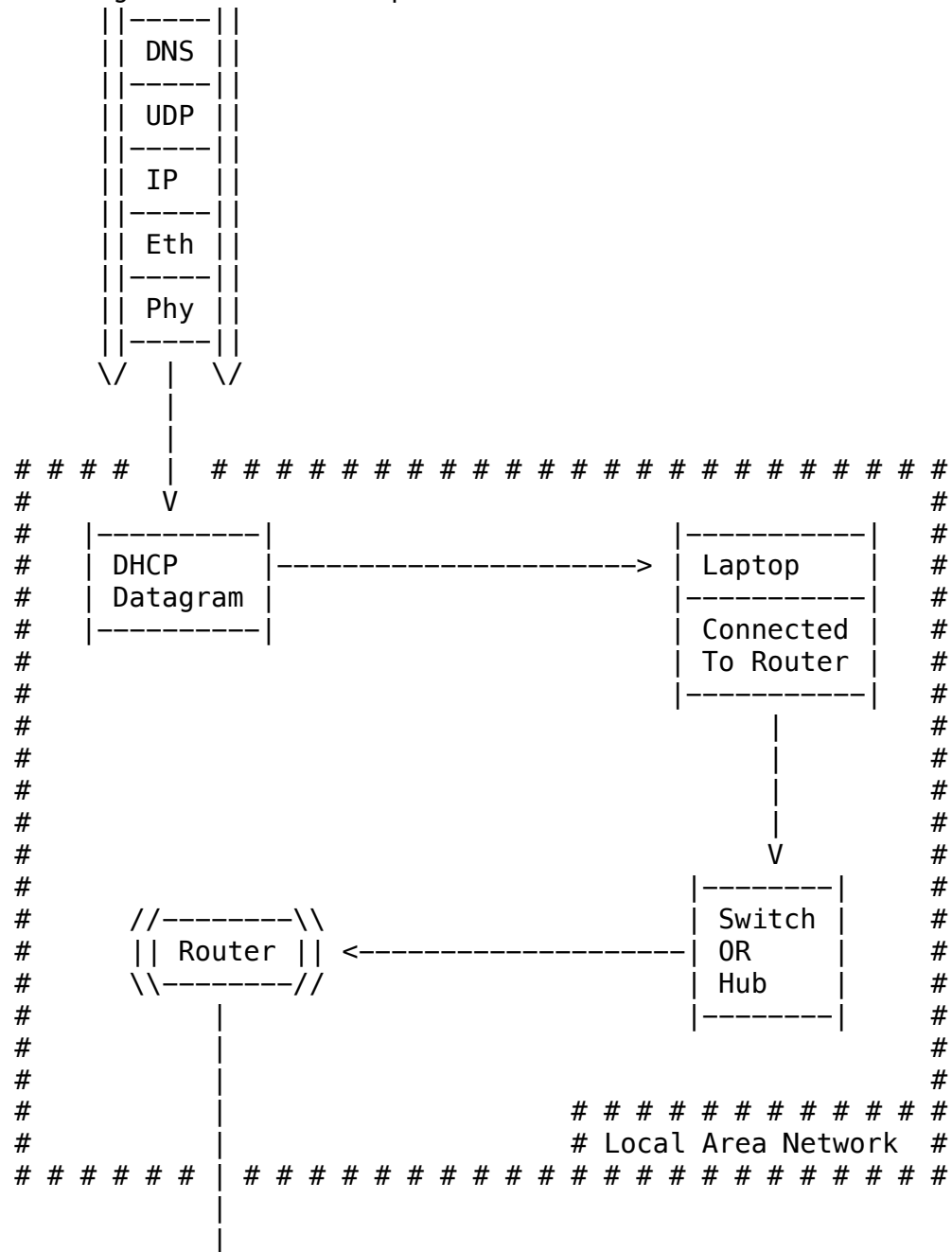
- The switch's self learning mechanism allows it to forward the DHCP ACK to the correct client
 - Once the client receives the frame, it is demultiplexed
 - Once the (DHCP) client receives its corresponding DHCP ACK reply, it now has all the necessary information it needs to connect to the outside world
 - Once all DHCP messages are exchanged, the client has a local IP address, the IP address of its first hop router, and the name/address of the DNS server
- A Day In The Life: ARP (Before DNS, Before HTTP)
- i.e. Diagram of Exchanging ARP Messages



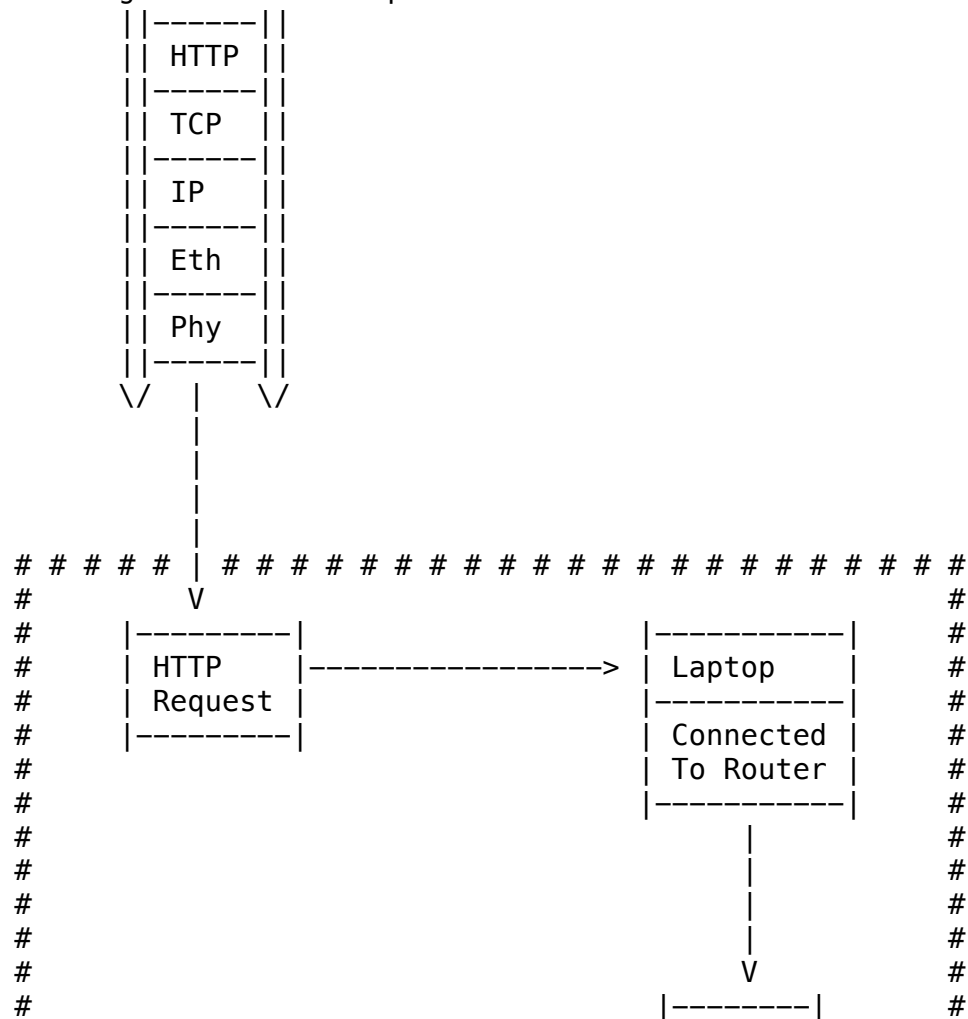


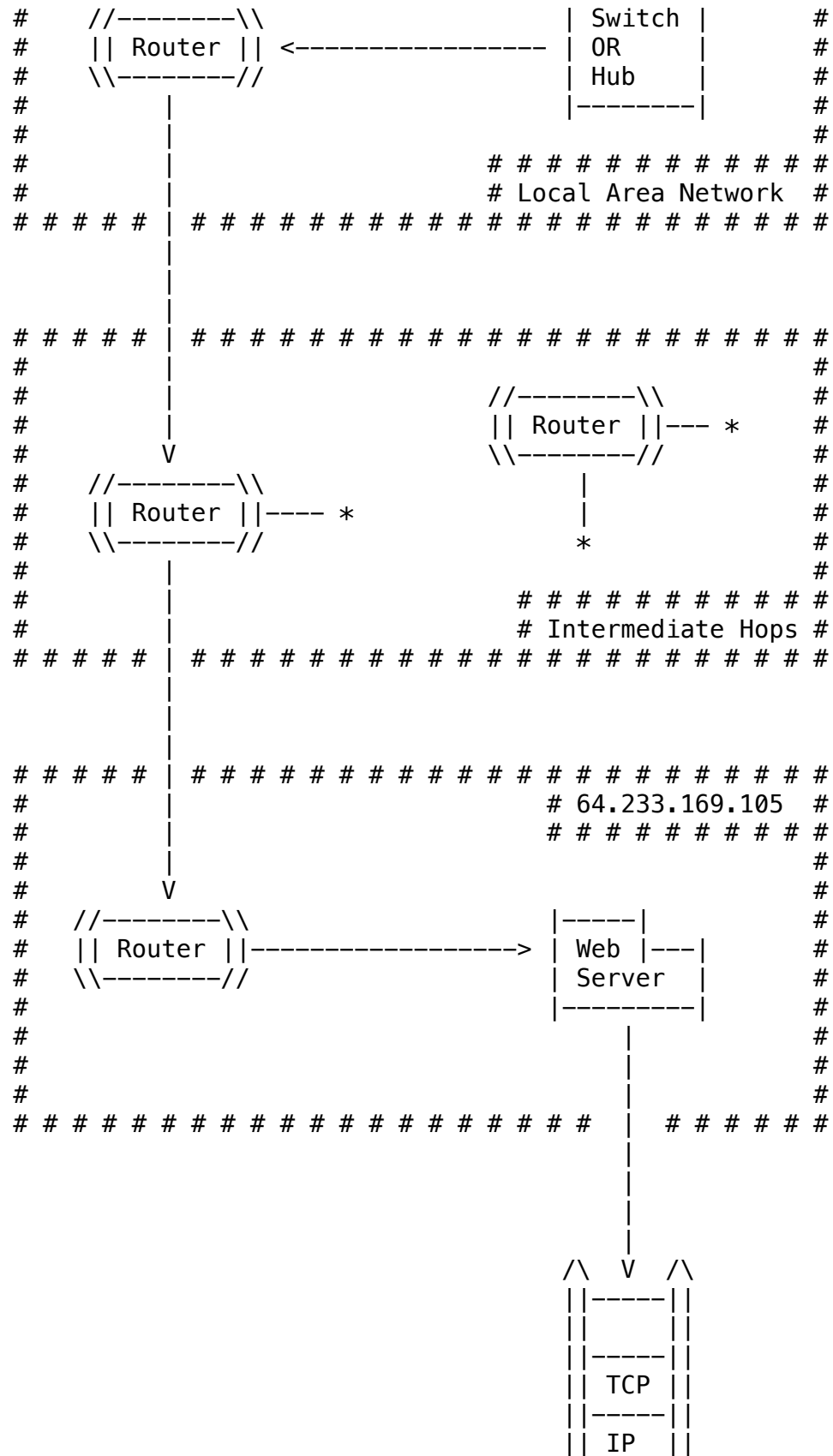
- Now, the device/client is almost ready to open up the browser, send HTTP requests, and connect to remote servers, such as 'www.google.com'. However, the device needs more than just the domain name, it also needs the IP address of the server
 - The IP address of the remote server is needed for the purpose of IP forwarding
 - The domain name is used in conjunction with the domain name service (DNS) server to determine the IP address of the remote server
- DNS requests/queries are created, and sent by the client
 - Each query is encapsulated in UDP, then IP, and finally it is encapsulated in Ethernet
 - DNS queries can go through recursive or iterative resolution. They are resolved by one of:
 - Cached information on the local DNS
 - A DNS server along the hierarchy
 - The authority DNS corresponding to the 'www.google.com' domain
- Typically, the local DNS server is not on the same local area network (LAN) as the client/device. In this situation, the host needs to forward the DNS query to the first hop router, which will forward the query to the local DNS server
 - In this case, the host/client needs more information than just the IP address of the router, it also needs the MAC address of the router, in order to put together a frame that contains the DNS query message
 - UDP/IP is used to send the DNS query message to the first hop router
 - The MAC address of the router's interface is obtained via ARP
- The mapping between the IP address, and the MAC address, of the first hop router is determined using the address resolution protocol, ARP
 - The host/client generates an ARP query broadcast message, and sends it throughout the network. Eventually, the router will receive it, and respond with an ARP reply message
 - The ARP reply message, sent by the router, contains the MAC address of the router's interface
 - Note: ARP is a link layer protocol that exchanges messages among hosts on the same local area network (LAN)

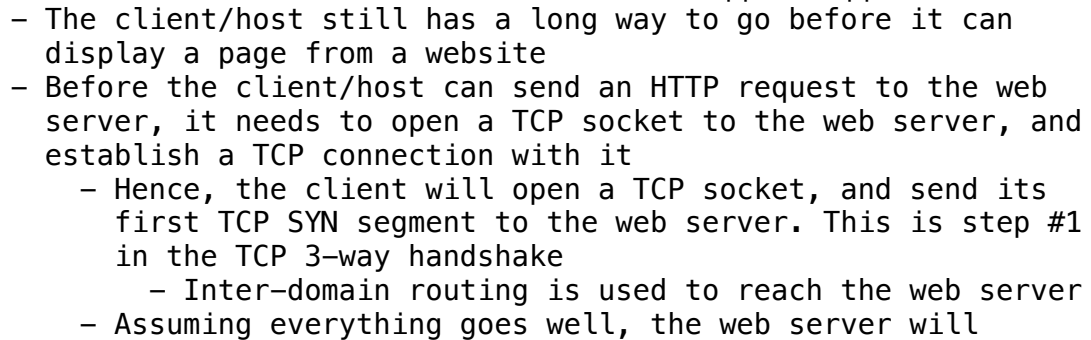
- Once the host/client acquires the MAC address of the first hop router, via ARP, the host/client sends the DNS query to the first hop router
 - The destination address of the frame is set to the MAC address of the first hop router
 - Upon reception of the DNS query, the router observes the payload in the MAC layer frame, and determines that the frame is destined to a DNS server. Hence, it forwards the frame, and its data
- A Day In The Life: Using DNS
 - i.e. Diagram of DNS Lookup



- The first hop router, and the intermediate routers, use routing protocols to determine which port an IP datagram needs to be forwarded to
 - The forwarding tables on each router is populated via a combination of inter-domain and intra-domain routing protocols, to determine the (least cost) path to the DNS server
 - Intra-domain routing protocols include:
 - 'RIP'
 - 'OSPF'
 - 'IS-IS'
 - Cisco's proprietary routing protocols
 - The only inter-domain routing protocol is 'BGP'
- Once the IP datagram reaches the DNS server, it is demultiplexed
 - In response to the DNS query, the DNS server replies to the client with corresponding IP address of the domain name, such as 'www.google.com'
- A Day In The Life: TCP Connection Carrying HTTP
 - i.e. Diagram of HTTP Request

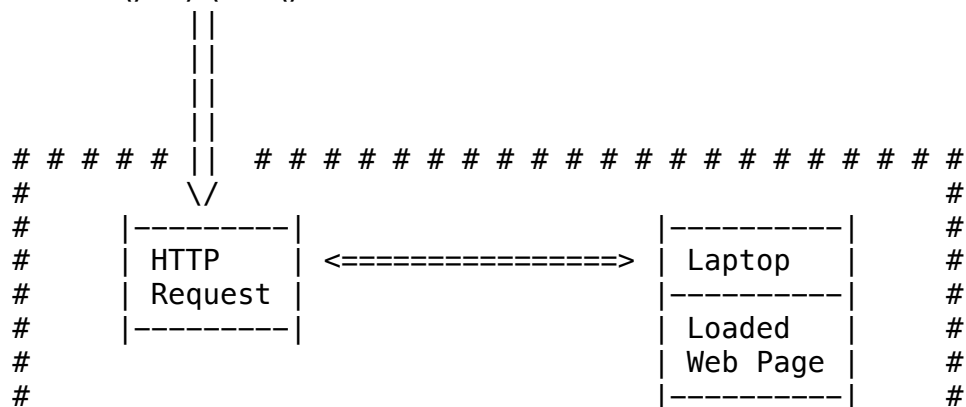


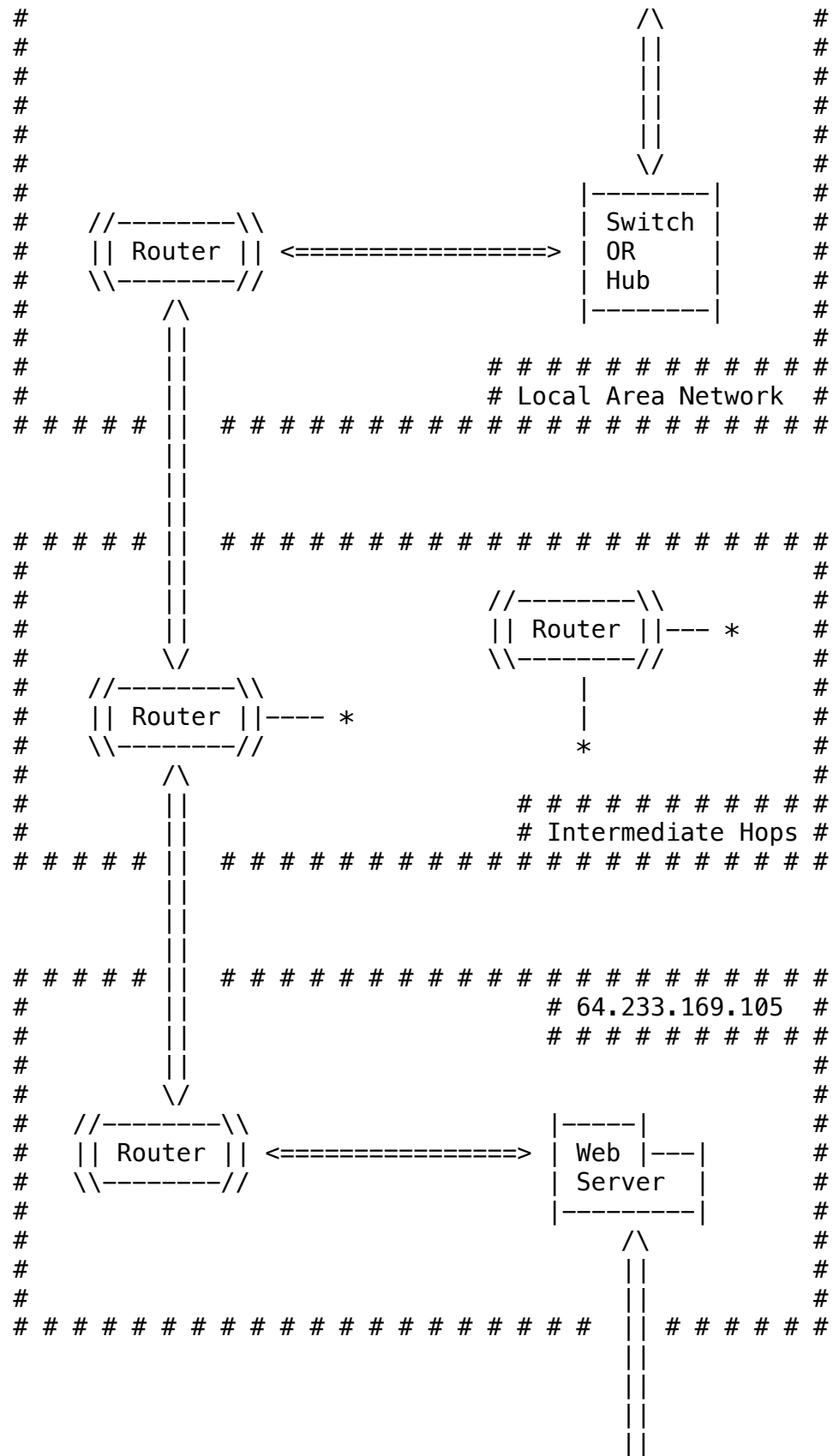


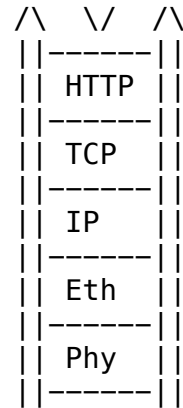


with a TCP SYNACK segment. This is the 2nd step in the TCP 3-way handshake

-
- A diagram of a protocol stack. It consists of five rectangular boxes stacked vertically, each containing a protocol name. From top to bottom, the boxes are labeled: HTTP, TCP, IP, Eth, and Phy. The stack is bounded by two vertical dashed lines. Below the stack, there are three symbols: a 'V' on the left, a '\wedge' in the center, and a 'V' on the right.







- The payload in the messages/segments sent correspond to the HTTP request
 - HTTP messages are application layer data
 - The destination IP address in the messages corresponds to the web server that the host/client is trying to reach
 - Note: This IP address was learned via DNS
 - HTTP request messages are sent into the TCP socket
- Upon reception of the HTTP request message, the server responds with an HTTP reply message
 - The HTTP reply message contains the web page
- The messages exchanged between the client and web server are routed based on the forwarding table on intermediate routers
- A Day In The Life: Conclusion
 - These are all of the protocols that are involved in loading a web page from a server
 - A lot of things need to work together for this very simple operation to be successful
 - From connecting to a local area network, to requesting and receiving the web page, a lot of protocols are used
- Network Security
 - This is the final chapter in this course
 - Covers basic concepts in security and network security
 - The goal of this chapter is to understand principles of network security, such as:
 - Cryptography and its many uses beyond "confidentiality"
 - Authentication
 - Message integrity
 - Key distribution
 - This chapter covers security in networks, such as:
 - Security in application, transport, network, and link layers
- What Is Network Security
 - Questions that arise when discussing network security:
 - What is network security?
 - What are the requirements of protecting communication or connection in the IP network?
 - There are many facets to security, such as:

- Confidentiality
 - First requirement of network security
 - Means that only the sender and receiver can have information about the content that is exchanged
 - The content/information should not be available to third parties, nor should they be able to receive the message
 - The typical means to achieve confidentiality is to encrypt the messages that are exchanged between the sender and receiver
 - There are some nuances to realizing confidentiality
 - i.e. How will sender/receiver communicate what details about the encryption, such as keys
 - There needs to be a way for the sender and receiver to correctly decode the message
 - In addition to encrypting the content of the message, other information such as identity, timing, and frequency of messages also needs to be kept a secret; this information should not be disclosed
 - Often times network traffic or network applications may have a certain signature in terms of the traffic pattern they generate
 - An attacker may be able to obtain some information about the nature of the message exchange by looking at the timing of inter-arrival time between sent and received messages, or inter-arrival time of transmitted messages
 - However, confidentiality ensures that an attacker cannot figure out the content of the messages exchanged
 - Timing information is useful to the sender and receiver; it is also useful to attackers
- Authentication
 - Confidentiality ensures that only the intended receiver can understand the content of the message.
 - However, how can someone be sure that the receiver that is receiving the message is the intended receiver?
 - Also, from the receiver's point-of-view, how can he be sure that the message originated from the intended sender?
 - The job of authentication is to confirm the identity of both communicating parties
- Message Integrity
 - Confidentiality ensures that the content of the message cannot be understood by third parties

- Message integrity ensures that the information being transmitted is not altered
 - An attacker can alter the contents of a message without knowing exactly what is being transmitted
 - i.e. If an attacker is able to perform a man-in-the-middle attack, he can insert himself between the sender and the receiver, and brute force change a portion of the bits that are being transmitted. Without any knowledge of what is being transmitted, an attacker can cause disruption by simply changing the contents of the message
 - The purpose of message integrity is similar to error detection. It ensures that the sender and receiver are able to tell if a message has been altered or not
- Access/Availability
 - This requirement is unique to network security
 - It is also a prominent problem
 - Attackers launch denial of service (DOS) attacks to hinder the accessibility and availability of (web) services to users
 - i.e. TCP SYN flood attacks can prevent users from accessing a service on the web. TCP SYN flood attacks are very problematic and damaging; they are very prevalent in today's Internet
- To summarize:
 - Confidentiality: Only the sender and intended receiver should be able to "understand" the content of the message
 - Authentication: The sender and receiver should be able to confirm each other's identity
 - Message Integrity: Messages sent by the sender and receiver should not be altered, in transit or afterwards, without detection
 - Access & Availability: Services must be accessible and available to users
- Outline
 - Discuss existing vulnerabilities in the TCP/IP protocol stack
 - Only the common attacks, and their countermeasures, will be discussed
 - Topics:
 - Attacks and counter measures
 - This is the focus of this lecture
 - Security primer
 - i.e. Some primitives from a cryptography point-of-

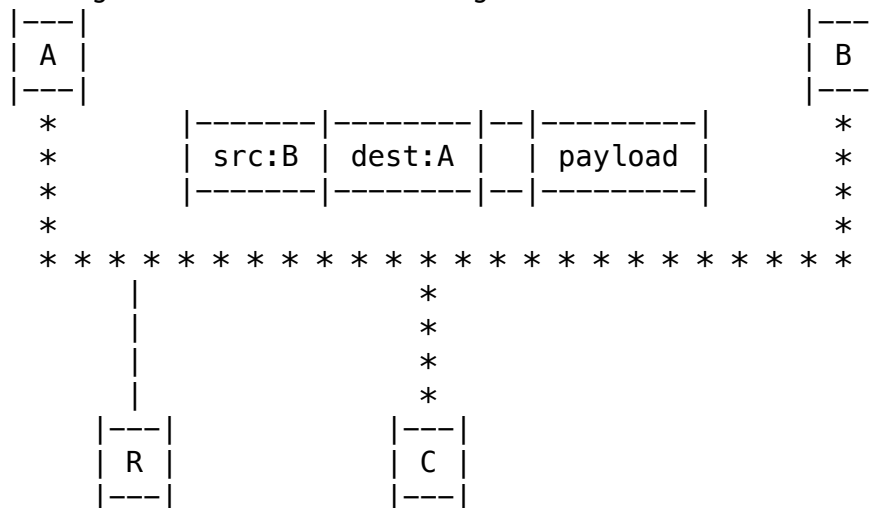
view

- Security in different layers
 - i.e. Security protocols
- Internet Security Threats: Mapping (1)
 - Mapping isn't really an attack; it may not have a direct consequence, but it is a precursor attack that can help an attacker launch other types of attacks
 - i.e. When robbers plan a bank robbery, they don't just barge in on a whim with guns blazing, and demand money. Instead, they send people to check the bank's daily routines, observe when is the bank busy, when is it slow, what path does the security guard take, when is he on break, etc.
 - Network attacks start off similar to this scenario; the attacker needs to find out what kind of services are available to exploit
 - Attacks are not blindly launched, without prior investigation
- The goal of mapping is to "case the joint", and find out what services are available, or implemented in the network
 - It is performed before the attack
 - There are many different tools attackers can use to determine the type of services implemented on a network
- The most basic/primitive mapping tool is 'ping'
 - 'ping' is used to determine what hosts have addresses on the network
 - 'ping' is primitive because all it can do is check whether a host is ON or not
 - It can be used to get some information on how long it takes to reach a host, but that's about it. It cannot be used to obtain more detailed information about what services are implemented in the network
 - Note: 'ping' is also used in the IP protocol, for ICMP messages. It is used to detect whether a host is active or not by sending a ping to a particular IP address
- 'Port scanning' is a sophisticated mapping tool
 - It tries to establish a TCP connection to every single port, in sequence, to see what happens
 - However, this is very easy to detect. A network administrator can simply view which devices are initiating TCP connection requests. An excess amount of TCP connections to every single port in succession is a dead giveaway that an attacker is port scanning
 - Tools such as 'nmap' are used for port scanning
 - It uses raw IP sockets to send IP datagrams to a particular set of ports, TCP or UDP ports, on the network, to determine/detect what kind of services are available

- Interestingly, 'nmap' is used by both attackers and network administrators. In fact, the creators of 'nmap' claim that the purpose for developing and releasing 'nmap' is to help network administrators perform security audits and network exploration
 - i.e. If an attacker compromises a computer, then the network administrator might want to see what kind of services are running on the computer, and in the local area network
 - Official Website: <https://www.insecure.org/nmap/>
- Question: Considering what mapping does, and how it works, are there any countermeasures for mapping? In other words, is there a way to prevent an attacker from being able to "case-the-joint", and find out what services are available?
 - Answer: No; there is no way to prevent an attacker from mapping a network. There are ways to detect whether an attacker is trying to map a network. For instance, an excessive number of 'ping' requests could be an attacker trying to map services on a network. However, stuff like this is not 100% accurate. Other sophisticated ways such as using a honeypot may mislead the attacker into thinking there is something valuable, but it won't prevent the overall issue.
- Internet Security Threats: Mapping (2)
 - There are no tools that can accurately detect 100% of the time if a malicious attacker is trying to map services that are implemented on a network
 - The best that can be done is record all traffic that enters and exits the network, and look for suspicious activity
 - i.e. Examine IP address, look for ports being scanned sequentially, etc.
 - If a device generates a lot of 'ping' messages to the local host, then this can be considered suspicious activity
 - An attacker may perform a naive port scan, which tries to establish a connection to TCP ports in numerical sequence
 - This kind of activity can be easily detected by utilizing a packet filter gateway, or even running some kind of Wireshark on the local area network (LAN)
 - By observing suspicious activities, a network administrator may be able to detect an intrusion
 - Frequently, network administrators respond to attacks or suspicious activity after an attacker has made an attempt
 - Administrators analyze network logs to determine if an attacker attempted to intrude/compromise the network
 - A large commercial website may run software in the background to continually perform traffic analysis in

real time

- Internet Security Threats: Packet Sniffing
 - Packet sniffing is not a real attack, but it can be used to facilitate other types of attacks
 - Using Wireshark, a device is able to put its network interface card into monitor mode, or promiscuous mode, and be able to capture all traffic in its local area network. This works for both Ethernet or Wi-Fi based local area networks
 - For WLAN, as long as a device is in the same local area segment, the broadcast medium ensures that transmission from other hosts can reach the device, and all other devices
 - For Ethernet, a device is only able to sniff packets within its own local segment, and not necessarily the entire local area network (LAN), because the device is isolated through the Ethernet switch
 - Sniffing allows attackers to parse messages, like TCP segments, received by a device, and extract important information such as sequence number. Information like this can be used to inject malicious traffic into the network
 - Traffic that is not encrypted can reveal sensitive information such as passwords, the website you are connecting to, the content you are viewing, etc.
 - i.e. Diagram of Packet Sniffing



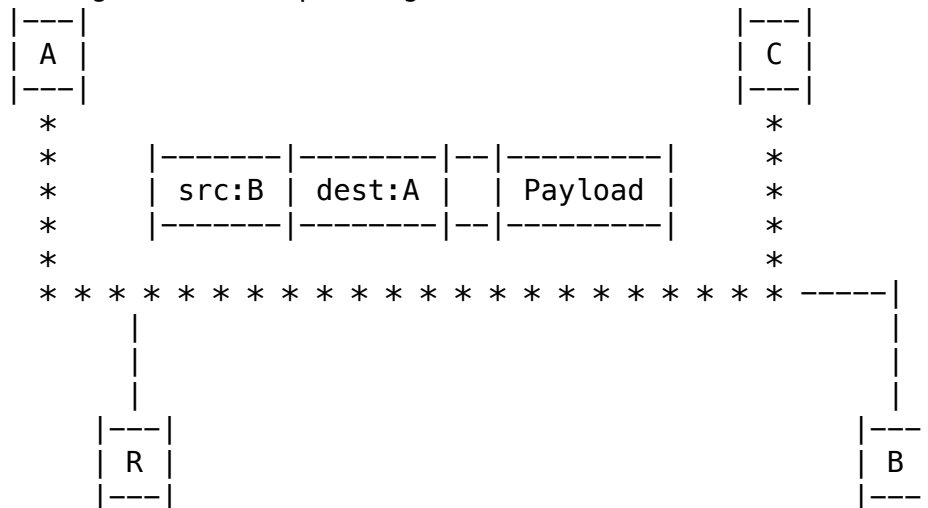
- In this diagram, host 'C' is sniffing packets coming from host 'B', that are destined to server 'A'
- The asterisks ('*') show where the packet is headed
- Question: Is there anything a user can do to detect packet sniffing? Can you detect if someone is sniffing traffic in your local area network?
 - Answer: No, you cannot detect packet sniffers in your local area network. It is difficult to detect packet sniffers because sniffing is entirely passive. An attacker is (simply) listening, and that's it. He is not

doing anything beyond that, like generating data and injecting it into the network. In contrast, mapping requires an attacker to send data to devices connected to the LAN, which can be detected via network logs. Detection is only possible if a pattern is generated; but sniffing is entirely passive (listening)

- Packet sniffing is analogous to a person sitting in a restaurant, and eavesdropping on other people's conversation. It's almost impossible to tell if a person sitting nearby is reading the newspaper or listening to somebody else's conversation
- However, there are some situations where a network administrator may be able to detect packet sniffing
 - When a device places its network interface card (NIC) into promiscuous mode, then the NIC will allow all link layer frames to be passed on to the operating system for further processing. Typically, only broadcast frames, or frames destined to the local host are allowed to make it past the NIC, and all other frames are dropped. So, some tools are able to generate specific frames, or specially crafted messages, with the intent of getting dropped by the NIC, but since the NIC is in promiscuous mode it will pass the frame to the operating system where a response may be generated. Hence, the sniffer, or attacker, is no longer silent and is no longer passively listening. However, this technique does not always work. It is a start, but it heavily depends on the actual implementation, and whether the operating system will elicit a behaviour to a specially crafted message
 - The bottom line is: Packet sniffing is very difficult to detect, even with a solid understanding of how it works
- The best way to maintain confidentiality of data, or network traffic, is to encrypt it. Hence, even if an attacker is sniffing your packets, the attacker will not be able to decode any information
 - When connecting to wireless networks, it is never a good idea to associate with an open Wi-Fi network. Ideally, you should go with a close network that utilizes something like WPA-2. This provides link layer security by encrypting the frames that are exchanged over the air
- Internet Security Threats: IP Spoofing (1)
 - IP spoofing allows an attacker to change the IP address of a packet before sending it to the destination
 - The receiver cannot tell if the source IP address is

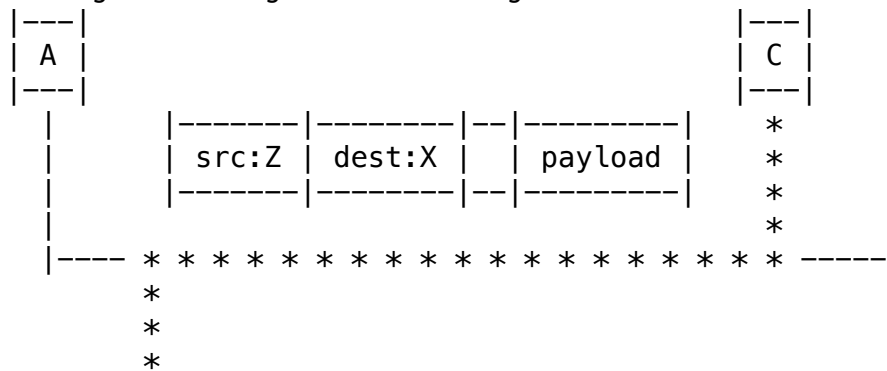
spoofed or not

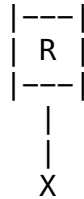
- Typically, attackers spoof IP addresses for 2 reasons:
 1. To ensure that network administrators cannot trace the packet back to the source, attackers use a spoofed IP address instead of their own
 2. If an attacker manages to compromise a lot of zombies for launching large scale attacks, he would want to protect his minions/zombies. Remaining undetected and anonymous benefits attackers. Hence, the compromised hosts may spoof their IP address to ensure that they cannot be easily detected or traced back
- i.e. Diagram of IP Spoofing



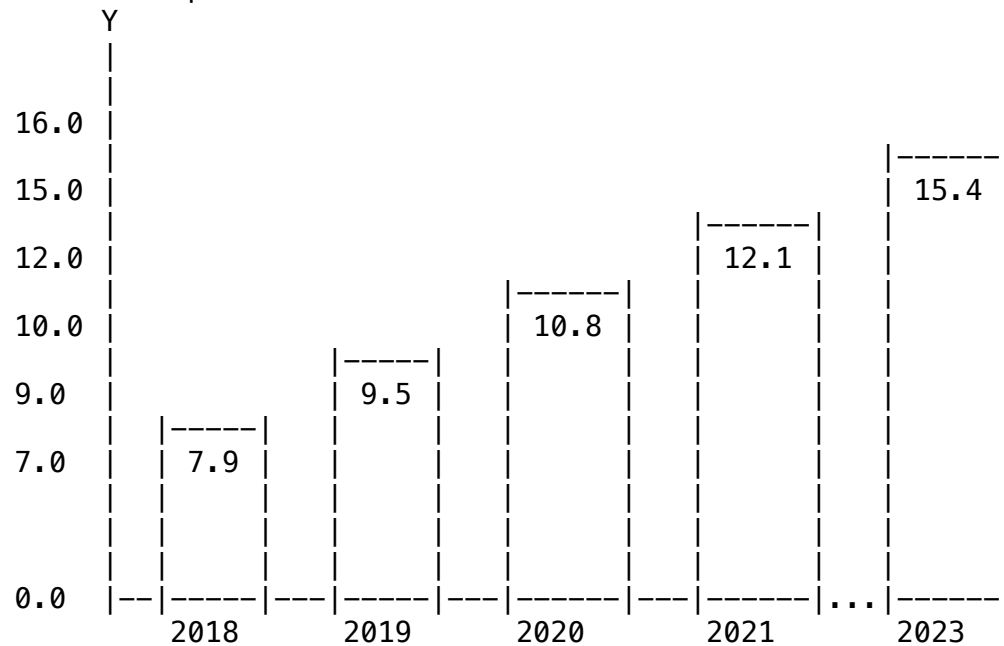
- In this example, host/node 'C' is pretending to be host/node 'B'. Node 'C' creates an IP packet, and in the source IP address column, it puts the IP address of node 'B' instead of its own IP address
- Once server 'A' receives the packet, server 'A' has no idea whether the datagram originates from host/node 'C' or host/node 'B'
- Through IP spoofing, not only can 'C' attack 'A', but it can also attack 'B' by generating a fake response to 'B' by pretending to be 'A'
- Question: Is it possible to detect if hosts in a local area network are spoofing their IP address and pretending to be someone else?
 - Answer: In the network layer, there is no way for a router to detect whether an IP packet with a particular source address originates from a legitimate host, or an imposter host. Without additional data it is not possible to uniquely distinguish different hosts. This is because all network layer data can be spoofed before the datagram is sent to the router. When exchanging IP datagrams, hosts cannot rely on a higher layer protocol for something like a uniquely security token. Thus, it

- Internet Security Threats: IP Spoofing (2)
 - Attackers use IP spoofing to protect themselves or a compromised host, or zombie
 - It is very difficult to detect if a host is masquerading as another host
 - For downstream routers it is even more difficult, maybe even impossible, because there is no way for them to tell the difference between real information and spoofed information
 - Typically, if the attacker is closer to the source, then more information is available for assessment
 - Ingress filtering is when routers do not forward outgoing packets with invalid source addresses
 - i.e. If an attacker tries to send an IP datagram whose source address is not in the network, then the router will not forward/send the packet
 - The term 'ingress' implies that the origin of the IP datagram is known
 - Typically, attackers use spoofed IP addresses that do not exist in their LAN
 - Hence, network administrators can monitor all traffic that leaves the network, and if a packet has a source IP address that does not belong to the LAN, then the administrator can conclude that IP spoofing has happened
 - If every network administrator implements ingress filtering on their local area network, then the Internet will be a safer place, because IP spoofing will be much harder to do
 - However, the reality is that everyone needs to implement ingress filtering to prevent IP spoofing. Even if one single person does not implement it, then an attacker can potentially utilize an a different IP subnet in its spoofed IP datagrams
 - In other words, ingress filtering is great, but it cannot be mandated for all networks
 - i.e. Diagram of Ingress Filtering





- The spoofed packet sent from host 'C' is dropped by router 'R' once it sees that the source IP address of the packet is 'Z', which does not correspond to any IP address in the local area network. Hence, the destination network, 'X', does not receive the packet
- For every hop a packet makes, it becomes harder and harder to detect IP spoofing; even though IP addresses have some kind of locality information associated to them, like geographical information
 - Thus, IP spoofing is a powerful tool/technique used by attackers to launch (severe) anonymous attacks
- Denial Of Service (DOS)
 - The unique thing about network security is its requirement for accessibility, connectivity, and availability of services
 - Currently, the most prevalent type of attack conducted on the Internet is denial of service (DOS) attack
 - i.e. Graph of Total Number of DDoS Attacks

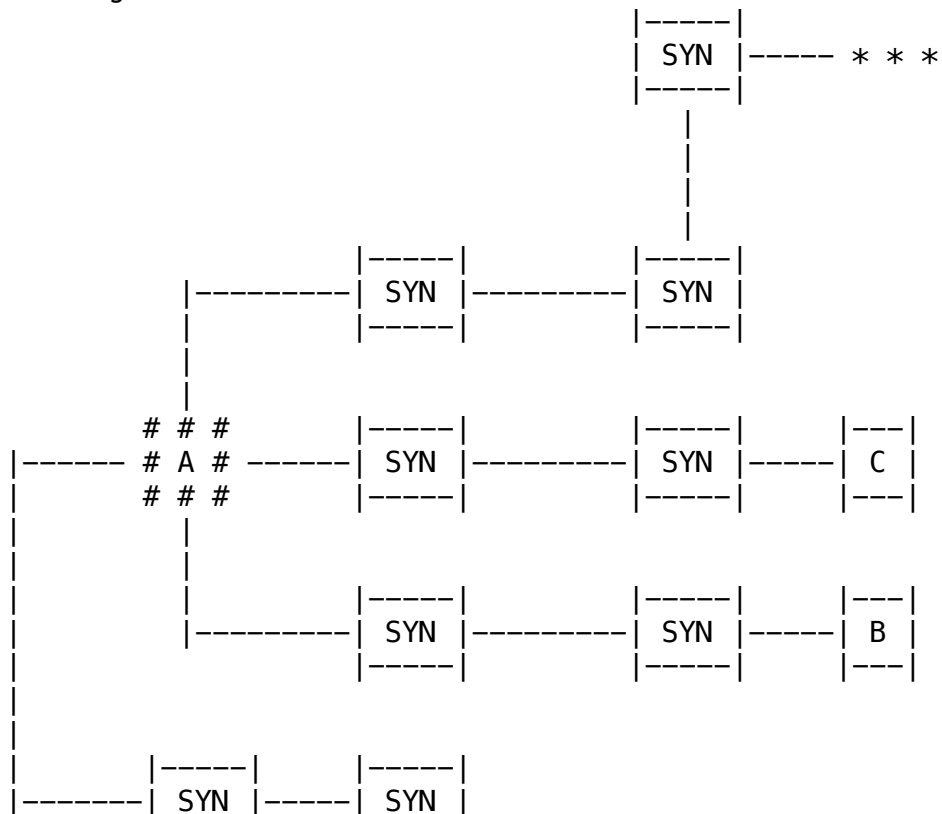


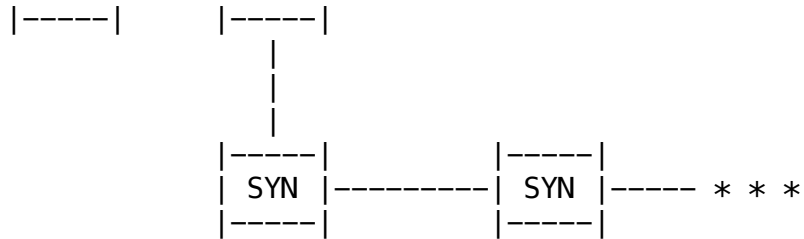
- The 'Y' axis is the number of DDoS attacks in the millions
- The 'X' axis is the year
- This graph/data is from Cisco's Annual Internet Report
 - It covers the range: 2018 – 2023
- Note: This graph does not show how much traffic is

generated for a single attack. Instead, it is about the instances of denial of service (DoS) attacks

- In 2018, 7.9 million DoS attacks were carried out
- Cisco projects that DoS attacks will exceed 15 million in 2023
 - This is very concerning
- Note: A separate report from another company found that DoS attacks increased during COVID
- In 2020, a large scale attack towards Amazon was conducted by attackers. Fortunately, Amazon did a good job in thwarting/throttling most of the attack, and prevented the attack from disrupting a significant portion of their services
- The basic definition of a denial of service (DoS) attack is disrupting the connectivity or access of a service, and preventing a victim, or victims, from accessing the service
- There are many different ways to launch a denial of service attack. For a comprehensive list of DoS attacks, refer to Wikipedia
- Question: Why do you think that DoS attacks have become so prevalent, and are continuing to occur more and more frequently?
 - Answer:
 - Monetary gain
 - Attackers can take down services provided by small businesses, and then force them to pay a ransom for restoration of their services
 - Boredom
 - Because of COVID, more people are locked indoors with nothing better to do
 - Rivalry
 - Company 'XYZ' wants to bring down Company 'ABC', in order to attract more customers
 - This is commercial espionage
 - Governments maybe behind attacks due to political differences or political gain
 - Annoyance
 - "Some men just wanna watch the world burn"
 - Bragging
 - Hacker groups brag on online forums about taking down big companies
- DOS: Sync Attack
 - Assignment 4 showcases one form of a denial of service (DOS) attack called TCP SYN attack
 - In the early days of the Internet, some operating systems had buggy implementations of TCP, which allowed unfinished, or half-open, connections to consume the device's memory, which can lead to a crash
 - An unfinished, or half-open, TCP connection is when the

- server receives the initial SYN segment, but the client does not complete the 3-way TCP handshake connection
- A flood of SYN segments can cause hardware resources to be depleted very quickly
- Modern operating systems, and embedded devices, have improved (software) implementations that limit the number of unfinished, or half-open, connections
 - Typically, a queue/buffer is utilized to counter SYN attacks
 - If the buffer is full of unfinished connections, then no further (incoming) connections will be accepted
 - However, this approach isn't too effective, because this is precisely what attackers want. A full buffer will drop new SYN segments from attackers and legitimate users. Hence, no one can establish a connection to the server/service
- Another way to minimize SYN attacks is to minimize the amount of resources that need to be allocated to unfinished, or half-open, connections
 - i.e. Instead of allocating a TCP control block in its entirety, it is better to maintain a minimal amount of information. This allows the system to have a bigger buffer that can tolerate more SYN segments. Thus, the system can handle SYN flood attacks
- i.e. Diagram of SYN Flood Attack

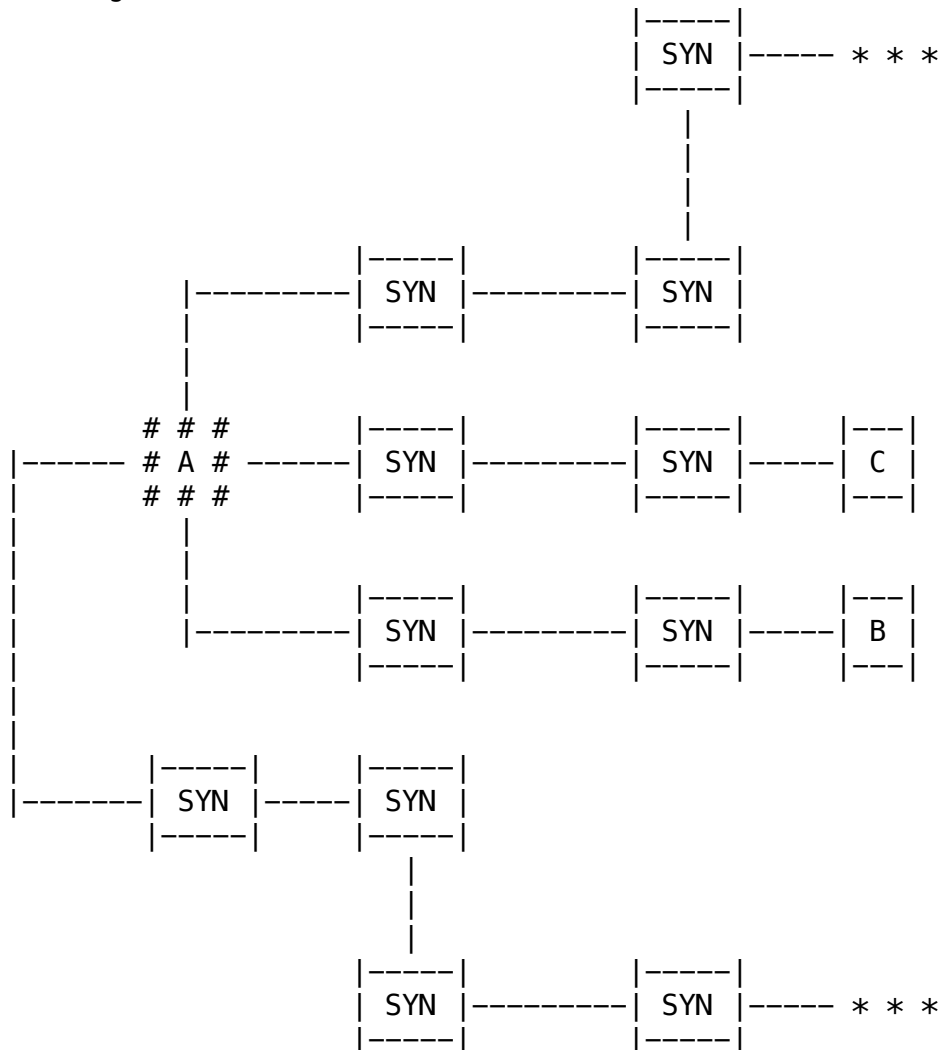




- In this diagram, multiple SYN segments are sent to server 'A', from multiple hosts/clients such as 'B' & 'C'
 - Note: The source IP address in SYN attacks are always spoofed. Hence, an IP address that corresponds to a particular host may not be carrying out the attack
- Denial Of Service (DOS): Countermeasures
 - Performing some kind of filtering is an effective strategy in countering denial of service (DOS) attacks
 - i.e. If a lot of SYN segments originate from the same limited number of hosts/zombies, then their segments/traffic can be filtered out, and they won't reach the service
 - Tracebacks help network administrators learn the origins of denial of service (DOS) attacks
 - This allows administrators to perform ingress filtering or shut down the affected services
 - Note: Attackers mostly utilize innocent hosts, whose machines they have compromised, when performing large scale attacks
 - Question: How does traceback work?
 - Traceback requires the support of intermediate routers. These routers can perform some kind of digest of flows and connections that traverse them
 - Put simply, it is kind of like leaving a trail of tiny bread crumbs. Small bits of information are dropped here and there, along the routers. Then, when a service experiences an attack, the network administrators can backtrace from the bits of information that are temporarily stored in the intermediate routers' memory or local buffer. Once all information has been collected, the admins can piece together the entire situation, and find out where the attack originated from
 - Traceback is not specifically used to counter SYN flood attacks. It is also used to combat IP spoofing, and other types of attacks
 - Note: Traceback requires additional information to be maintained by intermediate routers, and it requires the cooperation of routers
 - Currently, SYN attacks are not effective because they are well understood, and lots of countermeasures can be deployed

to mitigate the damage caused by SYN attacks

- However, from an attacker's point-of-view, a SYN flood attack is very useful, and can serve as a distraction
- To summarize, some countermeasures against SYN flood attacks that are deployed by network administrators are:
 - Reducing the amount of resources a half-opened, or unfinished, connection can utilize
 - Increasing the buffer size for half-opened, or unfinished, connections
 - Performing some kind of packet filtering to prevent malicious packets from reaching the host
- i.e. Diagram of SYN Flood Attack



- In this diagram, multiple SYN segments are sent to server 'A', from multiple hosts/clients such as 'B' & 'C'