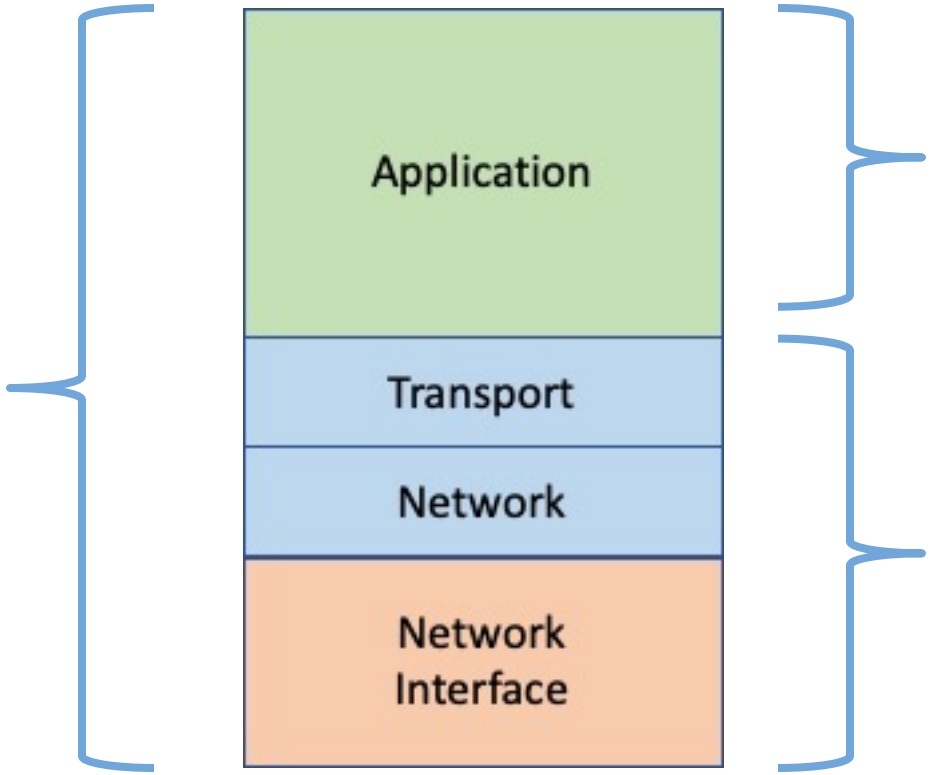
CS472 – COMPUTER NETWORKS

How Do TCP-Based Networks Work

Professor Brian Mitchell

How a TCP-based network works – lets look at the post office

Focus So Far:
A touch on
the entire
network stack



Now: Diving into the Layer Space (Chapter 2)

Future Lectures will work down the stack – Transport (Chapter 3), Network (Chapter 4/5), and Network Interface (Chapter 6)

How a TCP-based network works – lets look at delivering a package

Lets say I'm boxing up a package that I want to give to a friend

If my friend is local, I know how to get to his house, I'll probably put it in my car and drive it there my self

(DIRECT DELIVERY)

If my friend lives across the country, I'll probably drive it to the postoffice, and have them deliver it

(DIRECT DELIVERY TO POST OFFICE)

How a TCP-based network works – lets look at delivering a package

LOCAL

If my friend is local, I know how to get to his house, I'll probably put it in my car and drive it there my self

(DIRECT DELIVERY)

NOT LOCAL

If my friend lives across the country, I'll probably drive it to the postoffice, and have them deliver it

(DIRECT DELIVERY TO POST OFFICE)

In both of these cases, I am responsible for directly delivering the package locally.

I should be expected to know how to get to his house if he lives locally, or how to find my local postoffice

I should also not be expected to have much if any influence (or knowledge) over how the post office will route my package to my friend if he lives across the country

How a TCP-based network works – lets look at delivering a package

I have no ability to influence if the package is shipped by plane, truck, train. I also have no ability to influence the route to deliver the package

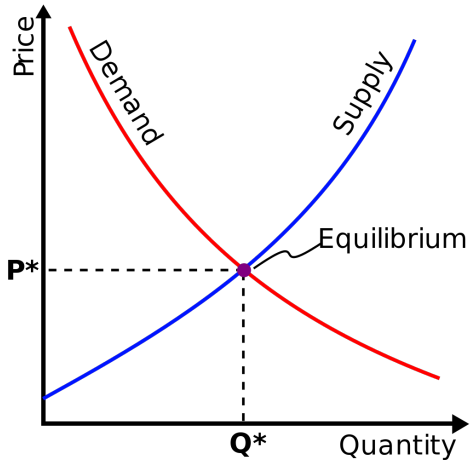
What Can I Influence?

- **The time to deliver the package (its importance)**
- **The security of the package (ensuring the contents were not tampered with)**
- **The value of the package (via insurance)**
- **The ability to guarantee the delivery of the package**

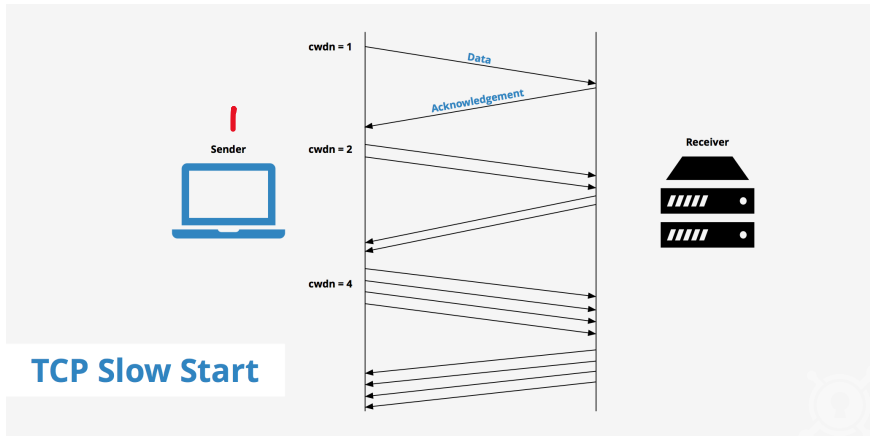
In other words, I can influence the Quality of Service (QoS) associated with the delivery of the package

How a TCP-based network works – congestion control

Similar to demand pricing used by companies like Uber and Lyft

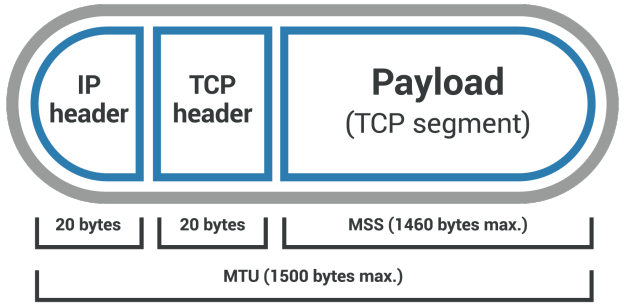


In networks, congestion control is the process of dynamically matching the speed of the sender with the receiver

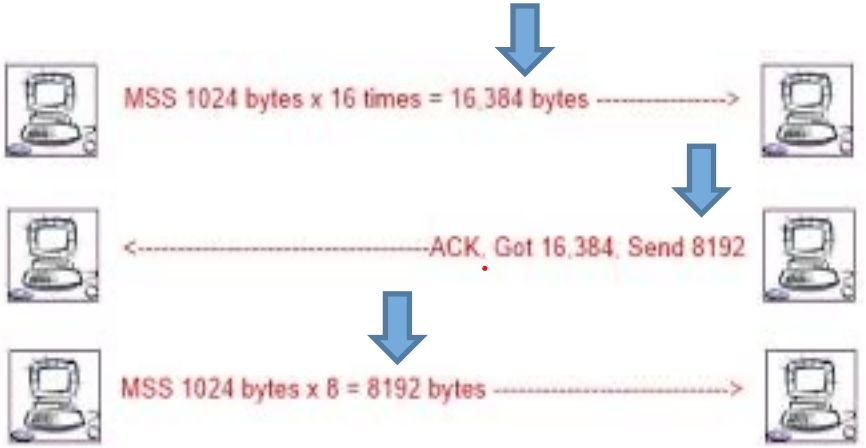


TCP Slow Start – RFC5681 incrementally increase what you send until the receiver lowers the window size or until packets loss is detected

Data packet

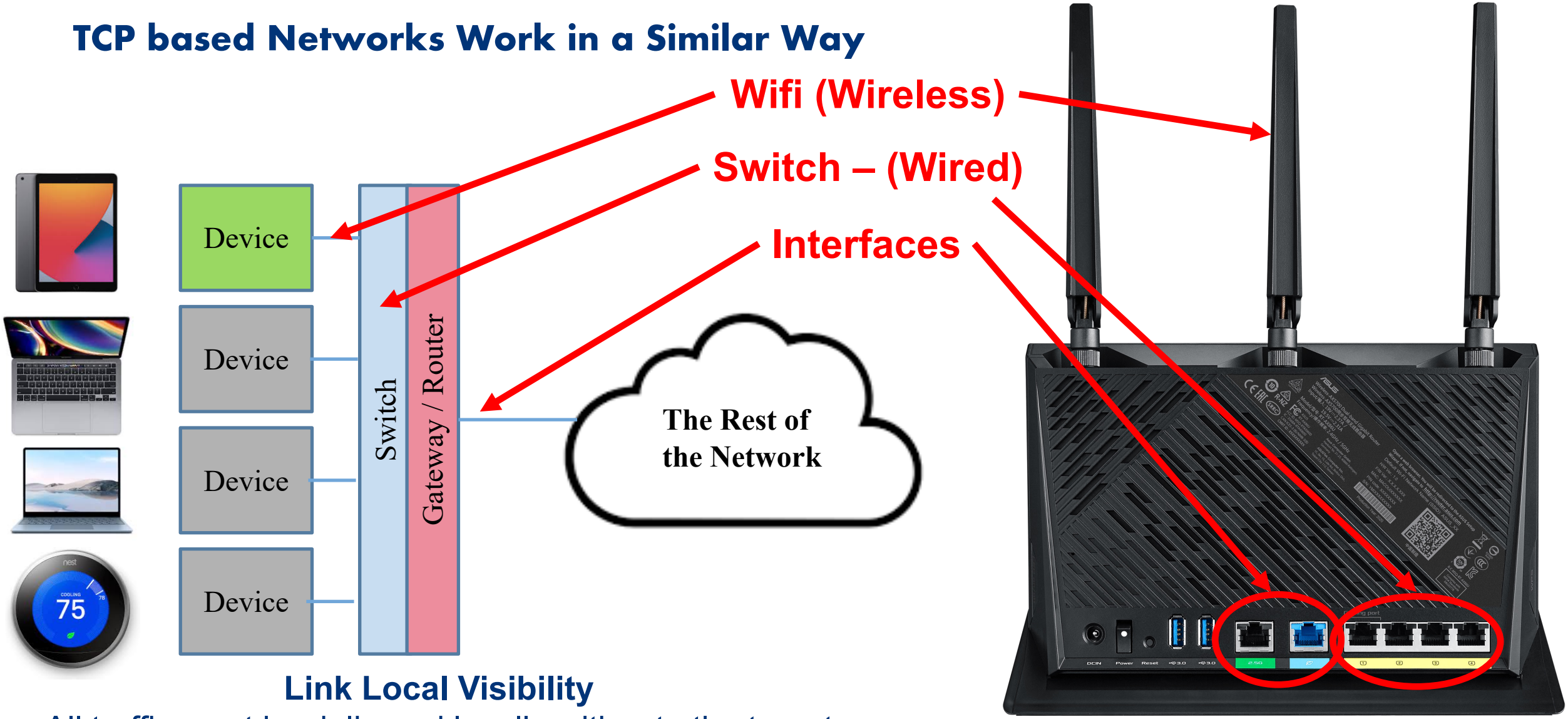


MTU=1500Max
MSS=1460 Max



TCP Window Size Adjustment – Receiver expands or contracts window size to speed up or slow down sender

TCP based Networks Work in a Similar Way



Link Local Visibility

All traffic must be delivered locally, either to the target device itself, or handed off to the Router/Gateway

Switches and Routers can be combined, but they don't have to be

Basic Concepts, we will cover more on this later

An IP Address

More specifically an IPv4 Address

32 bits

Generally referenced in dotted notation

Example:

192.168.45.10

0xC0A82D0A

A MAC Address

48 bits

First 24 indicate manufacturer of network interface

Last 24 are basically a serial number

Written as bytes in hex delimited by :

01:02:03:04:05:06

My mac address prefix: F0:18:98 – Apple Computer

A Network Interface

Hardware that connects to the network

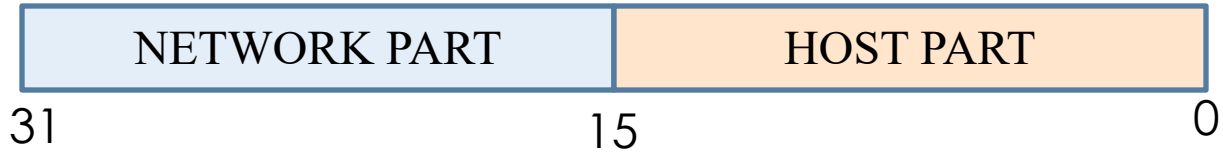
Can have 1 or more IP addresses

Has exactly one mac address

Many modern computers have more than one network interface – e.g., wired and wireless

Key Takeaways: A network enabled device must have one or more interfaces, each interface can have one or more IP addresses (that can change over time), each interface has one MAC address that does not change

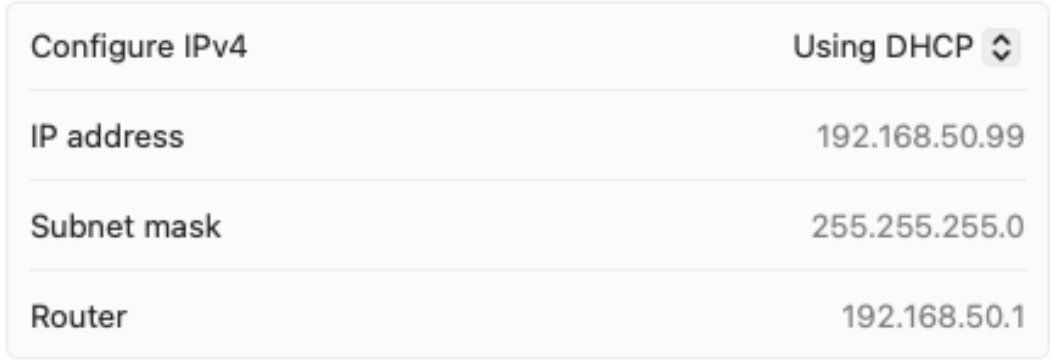
Some basics of IP addresses – we will cover a lot more on this later



- As mentioned, a network interface can have **one or more** IP addresses
- **Each IP address divides into 2 parts**, the network part and the host part, more on this later – for now, **that boundary can be anywhere in the 32 bits** using something **called CIDR -** (shown above, its in the middle -16 bits for the network part, 16 bits for the host part)
- There are a few **special IP addresses** to be aware of for now:
  - Local loopback address: 127.0.0.1 – does not generate any network traffic
  - All interfaces address: 0.0.0.0 – this address refers to all IP addresses on the local machine, including the local loopback address
  - Broadcast address – this address has all bits set to one in the host part. In the example above, since its 16 bits, the IP address will end with 0xFF.
  - Multicast address – in IPv4, these are special addresses in the range of 224.0.0.0 through 239.255.255.255 – more on this later, IPv6 only has multicast, not broadcast.

One Key Use of IP Addresses Is…

- One primary use of IP addresses is to ask the question – "Is the other device on the same network that I am on?"
- To do this we need a pair of IP addresses and network masks
- Remember, if the devices are on the same network, we send to them directly
- If the devices are not on the same network then we have to send to the gateway



So, to make the "on the same network determination" we need the IP address and mask for each device. We also need the IP address of the gateway/router

Some basics of IP addresses – we will cover a lot more on this later

IP: 10.250.40.97
Network Mask: 255.255.128.0 (must start with all 1's and then have all 0's)

```
                |----10-----| |--- 250----|  |-------40-------|  |---- 97 ----|
Ip:             0000 1010 1111 1010 0 | 010 1000 0110 0001
Net-Mask:       1111 1111 1111 1111 1 | 000 0000 0000 0000
Wildcard-Msk:   0000 0000 0000 0000 0 | 111 1111 1111 1111
Net:            0000 1010 1111 1010 0 | 000 0000 0000 0000
Host:           0000 0000 0000 0000 0 | 010 1000 0110 0001

Network:    10.250.0.0, Host: 0.0.40.97
Shortcut:   Host: 10.25040.97/17
```

Detecting addresses from same network

- Given IP addresses A and B, and netmask M.
  1. Compute logical AND (A & M).
  2. Compute logical AND (B & M).
  3. If (A & M) == (B & M) then A and B are on the same network.

- Ex: A = 165.230.82.52, B = 165.230.24.93, M = 255.255.128.0
  or better stated A=165.230.82.52/17 and B=165.230.24.93/17

- A and B are in the same network according to the netmask
- A & M == B & M == 165.230.0.0
- Key is to detect that the third octect has to be less than 127
  (or its most significant bit has to be a zero)

```
Netmask
1111 1111 1111 1111 1000 0000 0000 0000
  F    F    F    F    8    0    0    0

  165      230       82       52       IP ADDRESS
10100101 11100110 01010010 00110100
11111111 11111111 10000000 00000000   MASK /17
--------------------------------------
10100101 11100110 00000000 00000000   165.230.0.0 (Network part)

 165      230       24       93       IP ADDRESS
10100101 11100110 00011000 01011101
11111111 11111111 10000000 00000000   MASK /17
--------------------------------------
10100101 11100110 00000000 00000000   165.230.0.0 (Network part)

BOTH NETWORKS ARE THE SAME 165.230.0.0 SO BOTH DEVICES ARE ON THE SAME
NETWORK
```

Detecting addresses from same network

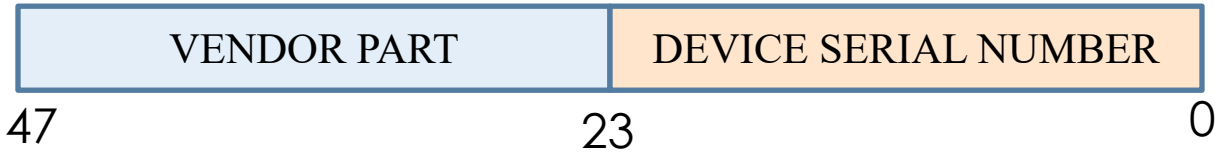
- Ex: A = 165.230.182.52, B = 165.230.24.93, M = 255.255.128.0

- A and B are NOT IN the same network according to the netmask
- A & M == 165.230.128.0 B & M == 165.230.0.0
- Key is to detect that the third octect must be less than 127

Routers using this matching approach to make decisions on if and where to forward datagrams

```
Netmask
1111 1111 1111 1111 1000 0000 0000 0000
  F    F    F    F    8    0    0    0

  165      230       182      52       IP ADDRESS
10100101 11100110 10110110 00110100
11111111 11111111 10000000 00000000   MASK /17
-----------------------------------
10100101 11100110 10000000 00000000   165.230.128.0 (Network part)

 165      230       24       93       IP ADDRESS
10100101 11100110 00011000 01011101
11111111 11111111 10000000 00000000   MASK /17
-----------------------------------
10100101 11100110 00000000 00000000   165.230.0.0 (Network part)

The 2 hosts are on different networks!
```

Some basics of MAC addresses – we will cover a lot more on this later



- As mentioned, a mac address is unique to an interface, but your computer likely has **multiple interfaces** (wired ethernet, wifi, etc).
- A MAC address is 48 bits that is split evenly – 3 bytes for vendor identifier, and 3 bytes for a unique serial number for the interface. Collectively, all MAC addresses should be globally unique. To decode your mac address you can go to - https://macaddress.io/
  - My wifi mac address starts with F0:18:98, which is registered to Apple Computer
- There are a few **special MAC addresses** to be aware of for now:
  - **Local loopback 00:00:00:00:00:00** – this is the mac address of 127.0.0.1 – and will not be sent out of the local device
  - **Broadcast FF:FF:FF:FF:FF:FF** – this mac address will indicate to the ethernet card
  - Advanced topic, ignore for now, Multicast MAC address – these start with **01:00:5E** and the device serial number part is created from the multicast IP address

Using MAC addresses

- We used IP address to determine if the destination is on the same network, or if not we send to the IP address of the gateway/router
- But we do not use IP addresses to send actual data, we use MAC addresses
- Now the question is – “How do we get the MAC address of the destination?”

How do you get the destination's MAC address? ARP for IPv4

For IPv4 we use the ARP (Address Resolution Protocol) to request the MAC address of the destination. https://www.rfc-editor.org/rfc/rfc826.html

For a request, the src and destination IP addresses are included; OPER=1, the src MAC address is provided, and the dest MAC address is the broadcast address

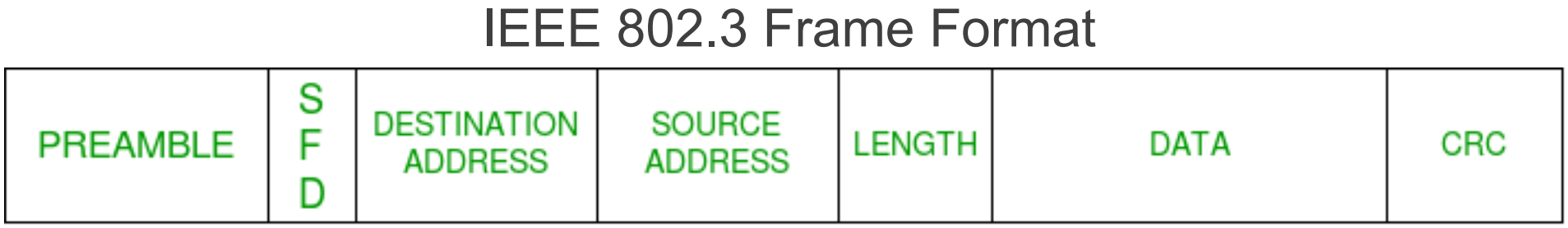
Every machine looks at the ARP request, and if a machine matching the dest IP address exists, it replies with OPER=2, and its MAC address in the SHA

Internet Protocol (IPv4) over Ethernet ARP packet

Octet offset	0	1
0	Hardware type (HTYPE)	
2	Protocol type (PTYPE)	
4	Hardware address length (HLEN)	Protocol address length (PLEN)
6	Operation (OPER)	
8	Sender hardware address (SHA) (first 2 bytes)	
10	(next 2 bytes)	
12	(last 2 bytes)	
14	Sender protocol address (SPA) (first 2 bytes)	
16	(last 2 bytes)	
18	Target hardware address (THA) (first 2 bytes)	
20	(next 2 bytes)	
22	(last 2 bytes)	
24	Target protocol address (TPA) (first 2 bytes)	
26	(last 2 bytes)	

```
Ethernet = 0x0001
IPv4 = 0x0800
HLEN = 6
PLEN = 4
1/req, 2/reply

6 byte sender
MAC Address

Senders IP address

FF:FF:FF:FF:FF:FF
(Broadcast for request)

Destination IP address
```

Putting Data on the Wire



A 7 byte pattern of alternating 1's and 0's – used for synchronization

Start of Frame Byte 10101011 – Indicates the beginning of a frame

Destination and Source MAC addresses e.g., 01:02:03:04:05:06

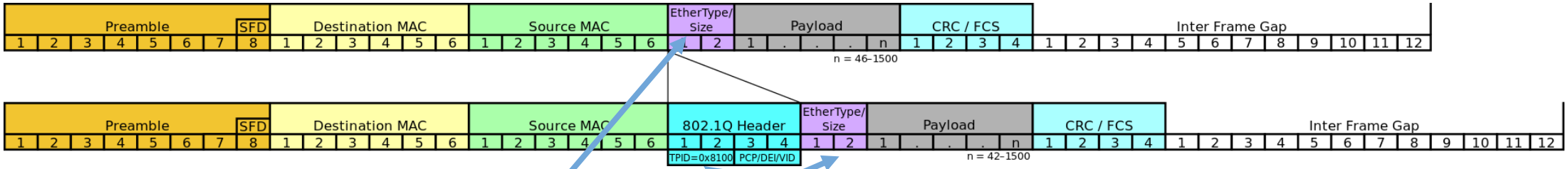
Length of the entire ethernet frame

Data (variable) must be at least 46 bytes, and zero padded if less

CRC error detection value to ensure data has not been corrupted

Putting Data on the Wire

Ethernet II Frame Format

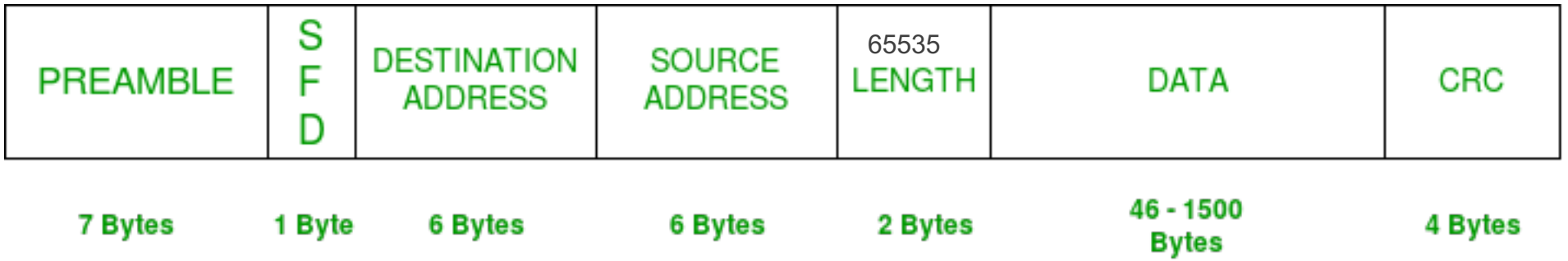


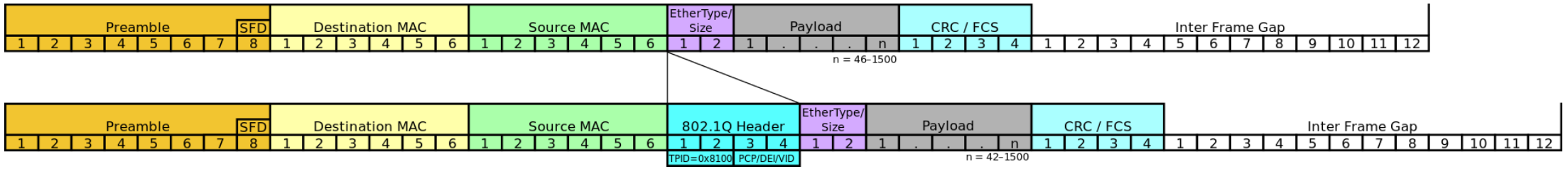
To preserve backwards compatibility Ethernet II reuses the Length field, replacing it with the Ethertype field, All values must start > 1536 0x600 as any value < 1500 is treated as a length in an 802.3 frame

New support for VLAN tagging

Note length is not needed as the end of the frame can be determined by the interframe gap – Carrier Signal without any modulation

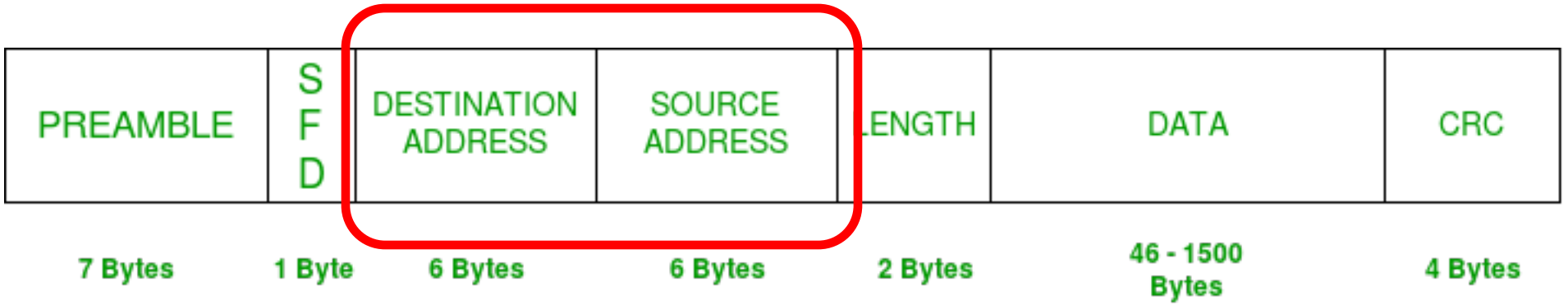
Putting Data on the Wire





Ethernet II Frame Format

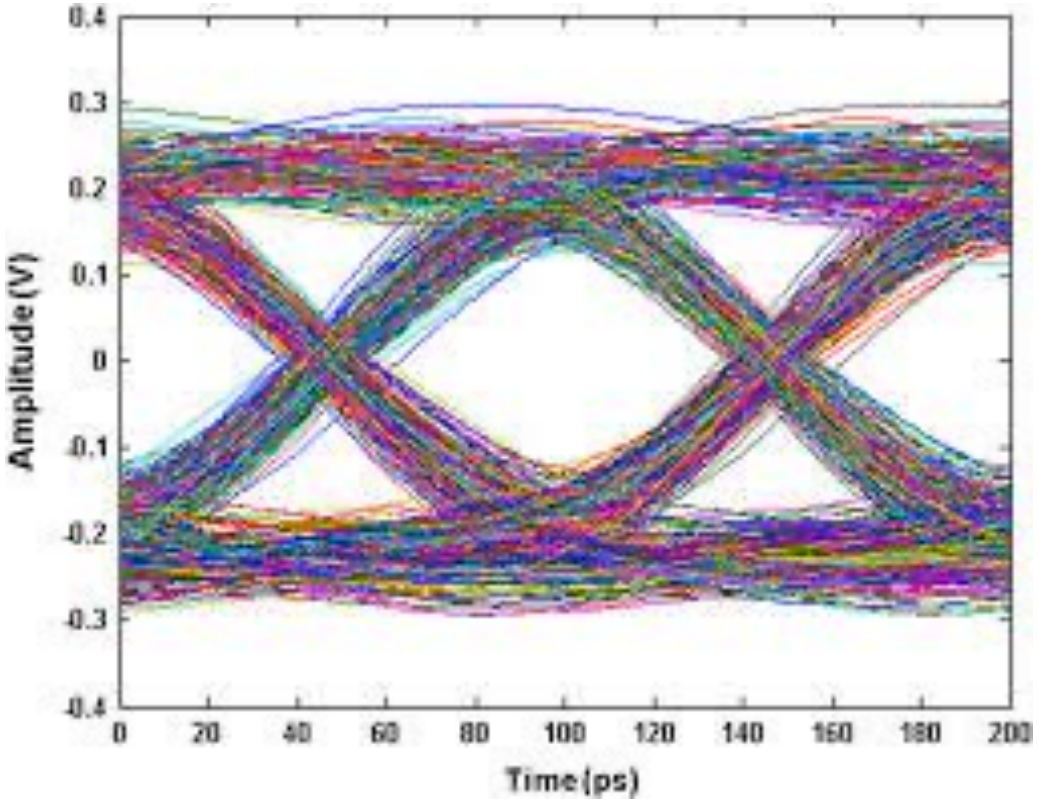
TCP based Networks Work in a Similar Way – Ethernet Frame 802.3



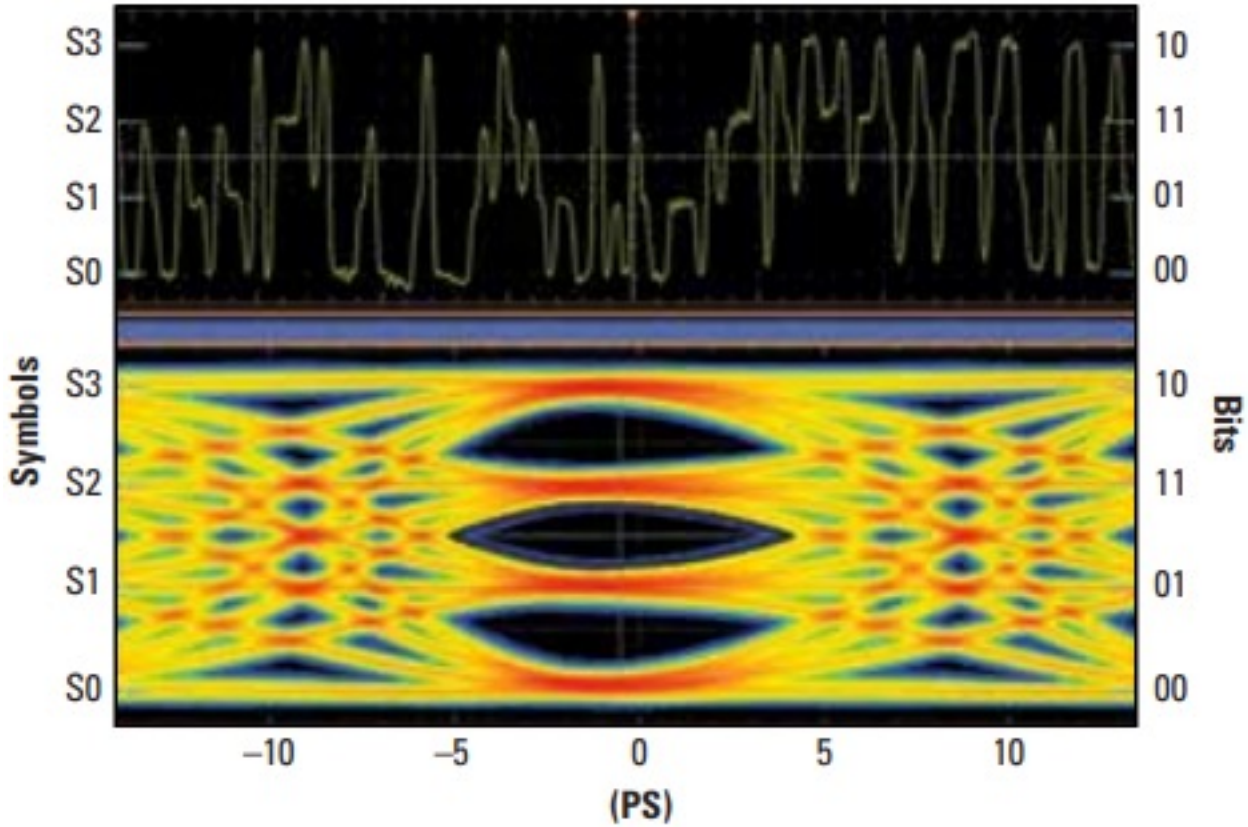
The key to understanding how things work is that all ”link local” traffic flows using MAC addresses. These are generally burned into hardware and do not change

THUS, you must have the destination MAC address to send data to another device on the network.

At the physical layer, that is either the target device itself, or the gateway/router interface (remember the post office example)

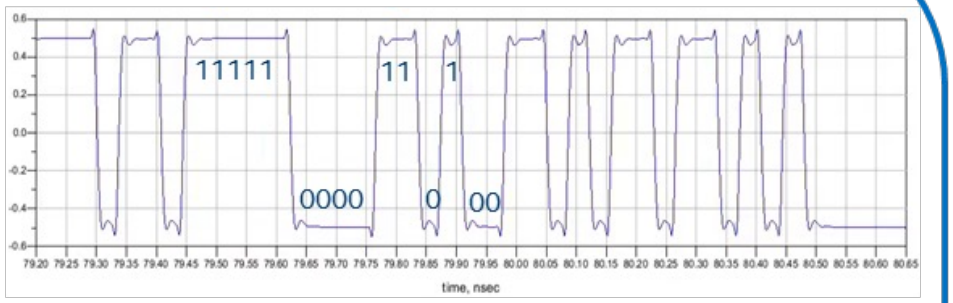


Example Pulse Amplitude Modulation (PAM-2) – Good up to 10Gbs



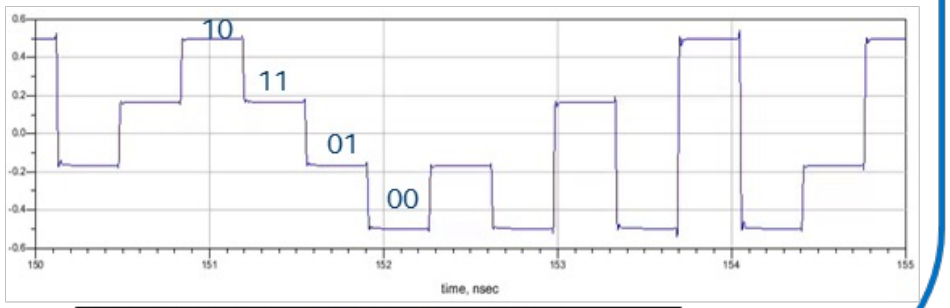
Example Pulse Amplitude Modulation (PAM-4) > 10Gbps

NRZ (or PAM-2) modulation

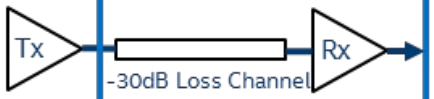


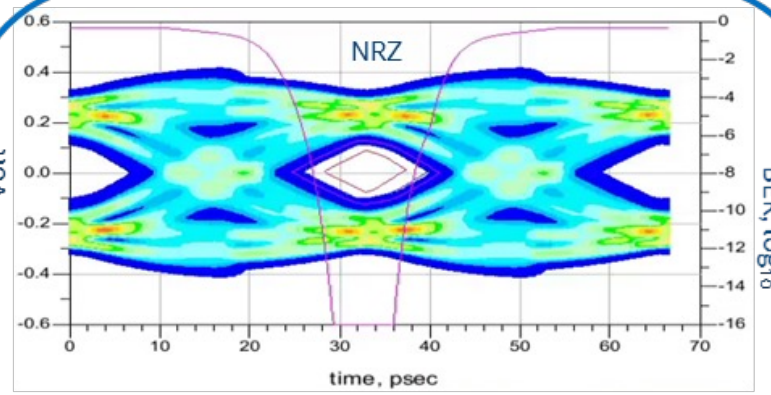
NRZ TX output with ideal termination

PAM-4 modulation in Gray code(mapping=0132)

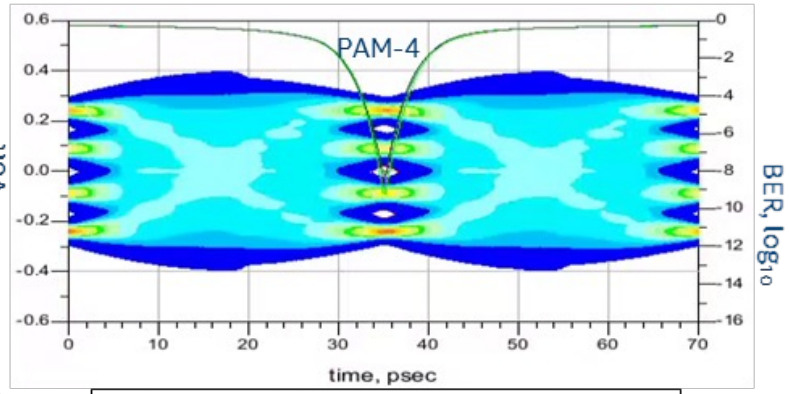


PAM-4 TX output with ideal termination
(For illustration, uniform 10-UI per level is used)



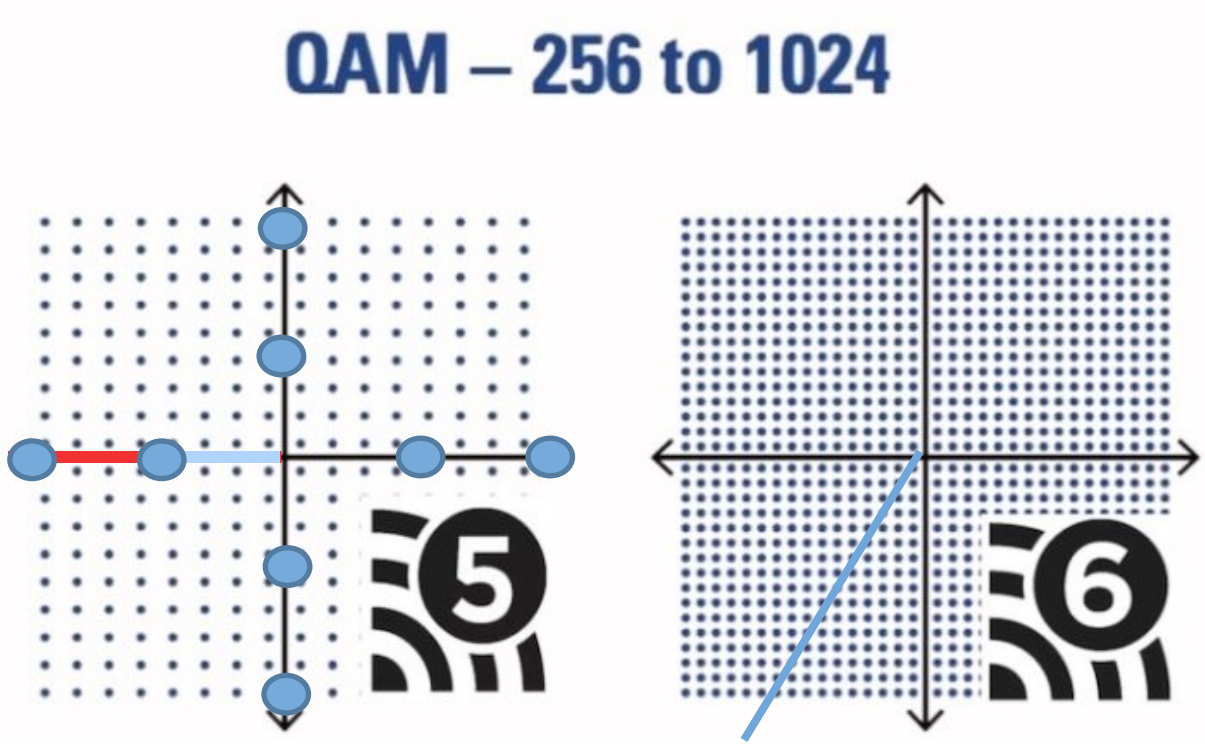


Eye Diagram and Bathtub curve at RX output



Eye Diagrams and Bathtub curves at RX output

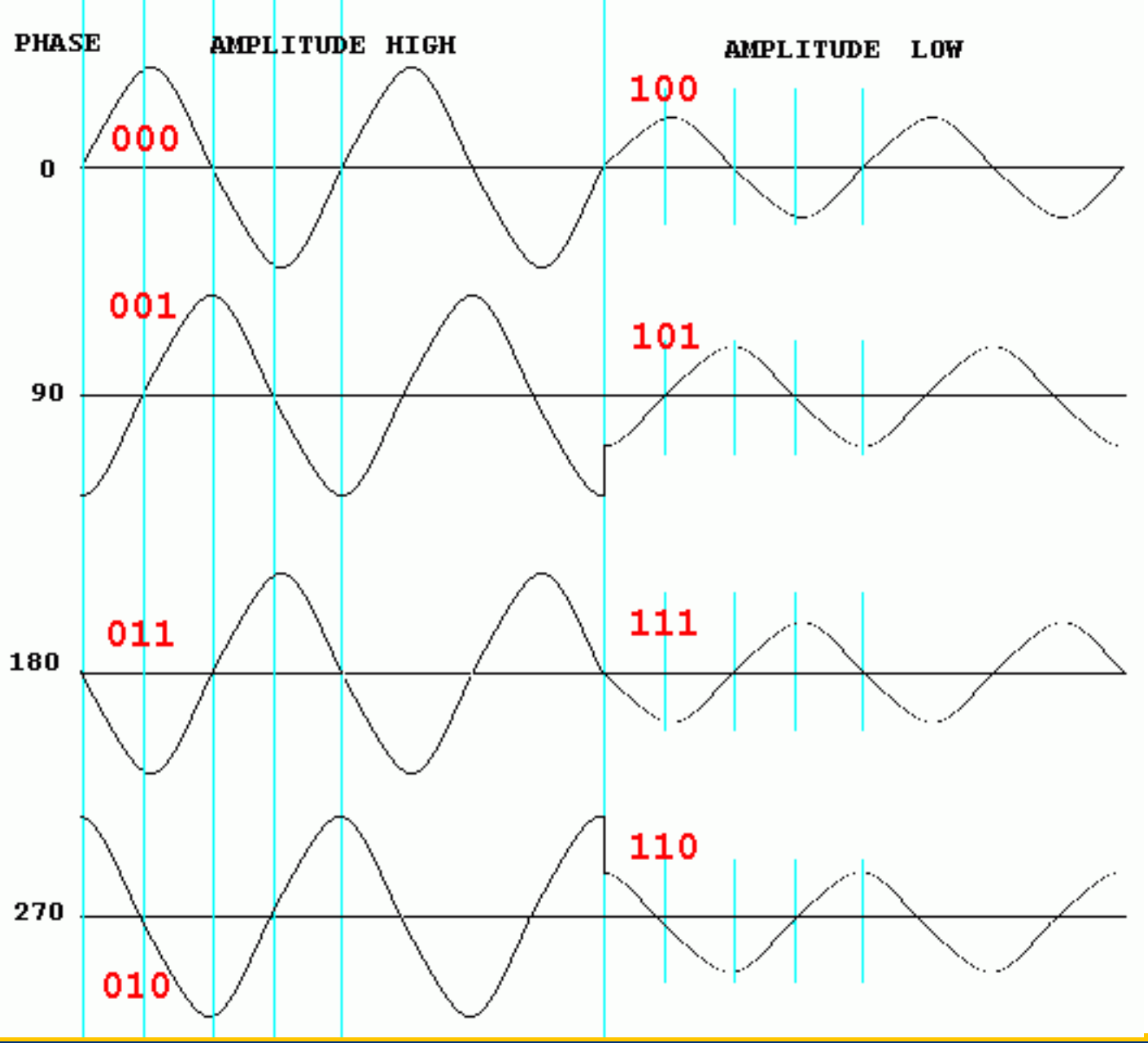
What about Wifi?



Quadrature amplitude modulation (QAM) for Wifi – 8 bits per cycle & 10 bits per cycle

QAM, blends amplitude and phase modulation

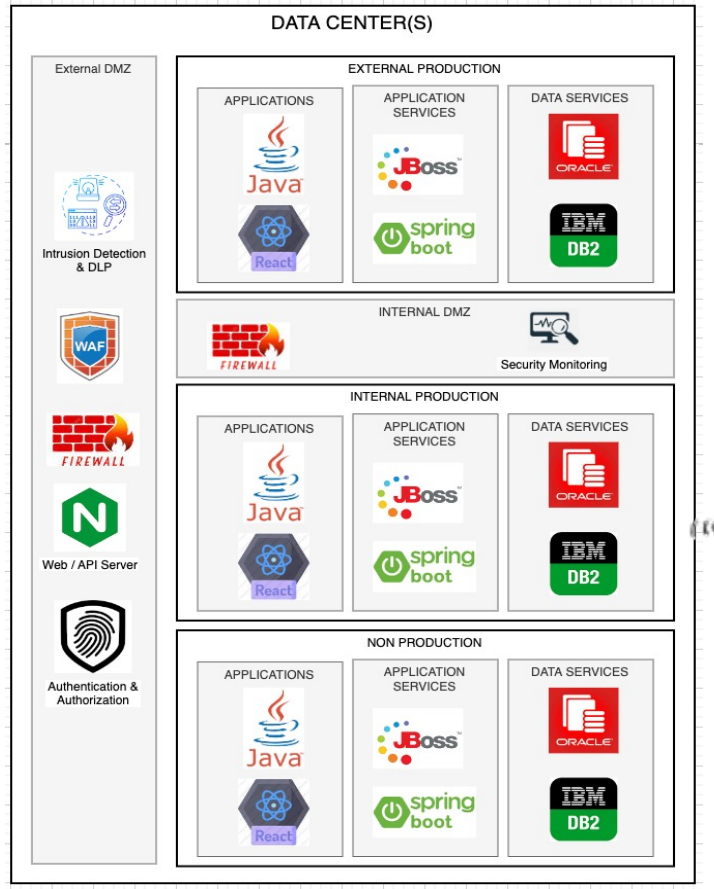
Example QAM8 (3bits per cycle)

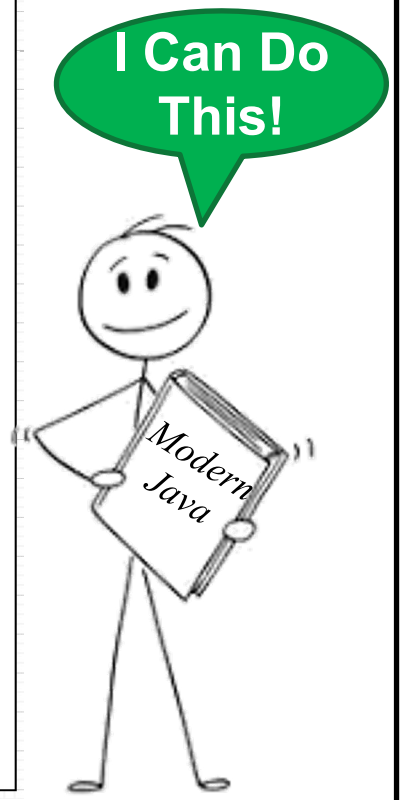


Back To Networking

This course is especially relevant these days

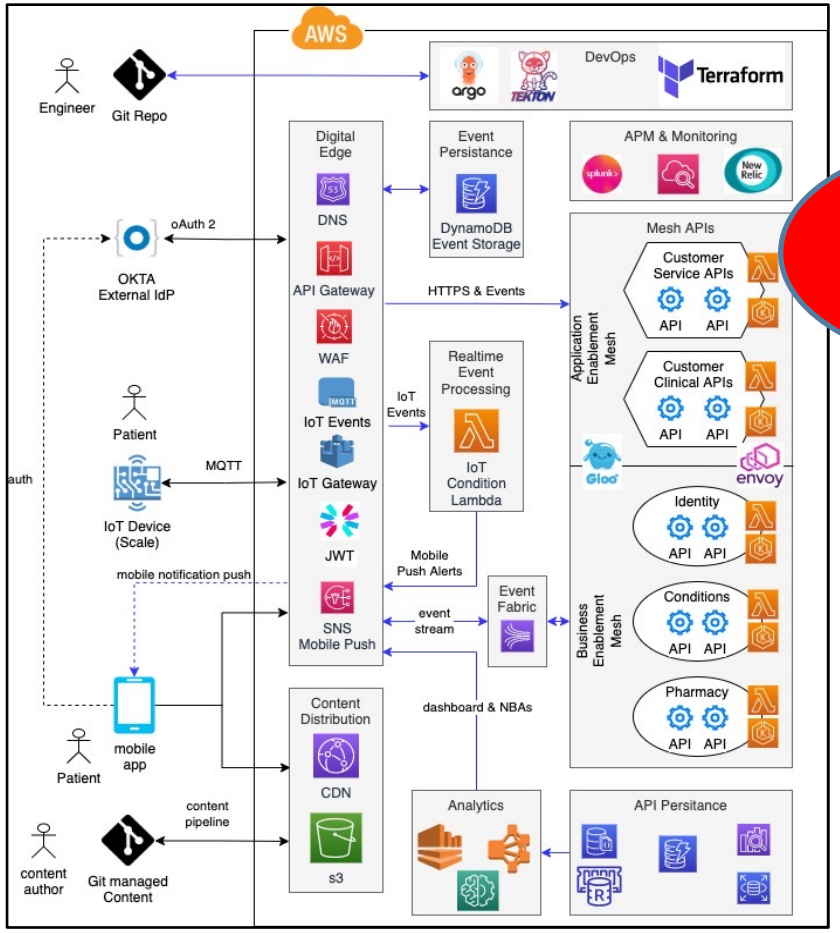
PRE CLOUD NATIVE





Please create a web service using Java with the Spring Boot Framework then deploy to the JBoss application server

CLOUD NATIVE

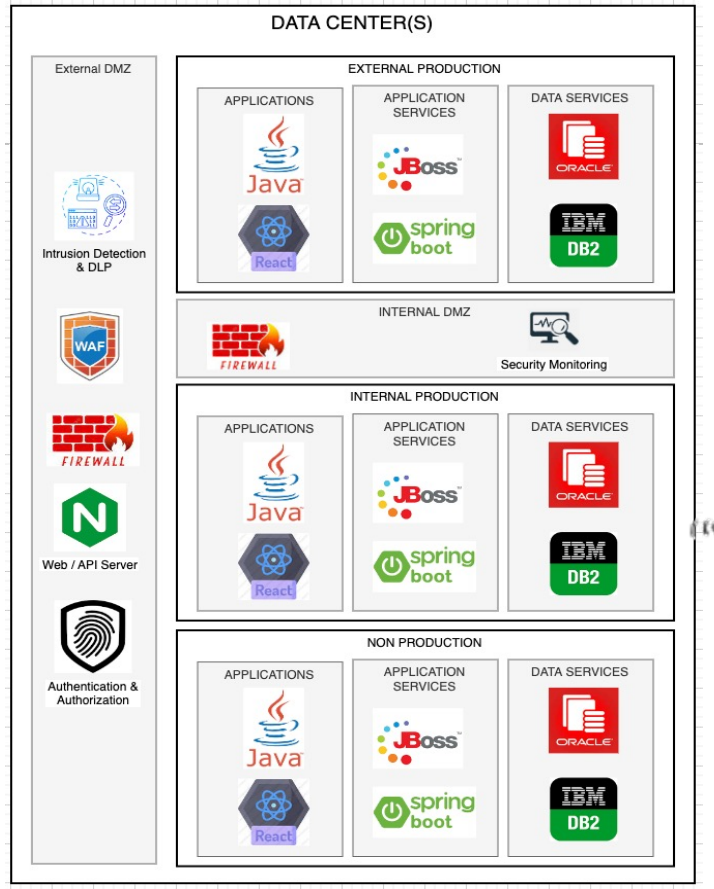


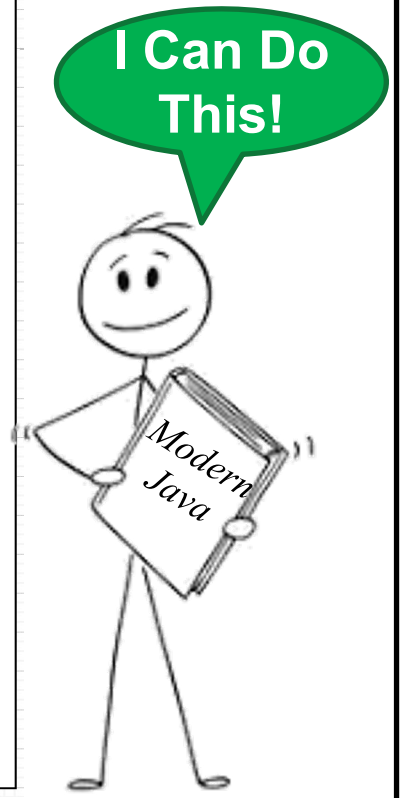
*Please define a software data center for your application given a **CIDR block of 10.0.0.0/20**, ensure it runs across at least 4 availability zones. Then build an API in GO deployed to Kubernetes*

Cloud Native Requires Software Engineers to Cover a Broad Space – It can be overwhelming

This course is especially relevant these days

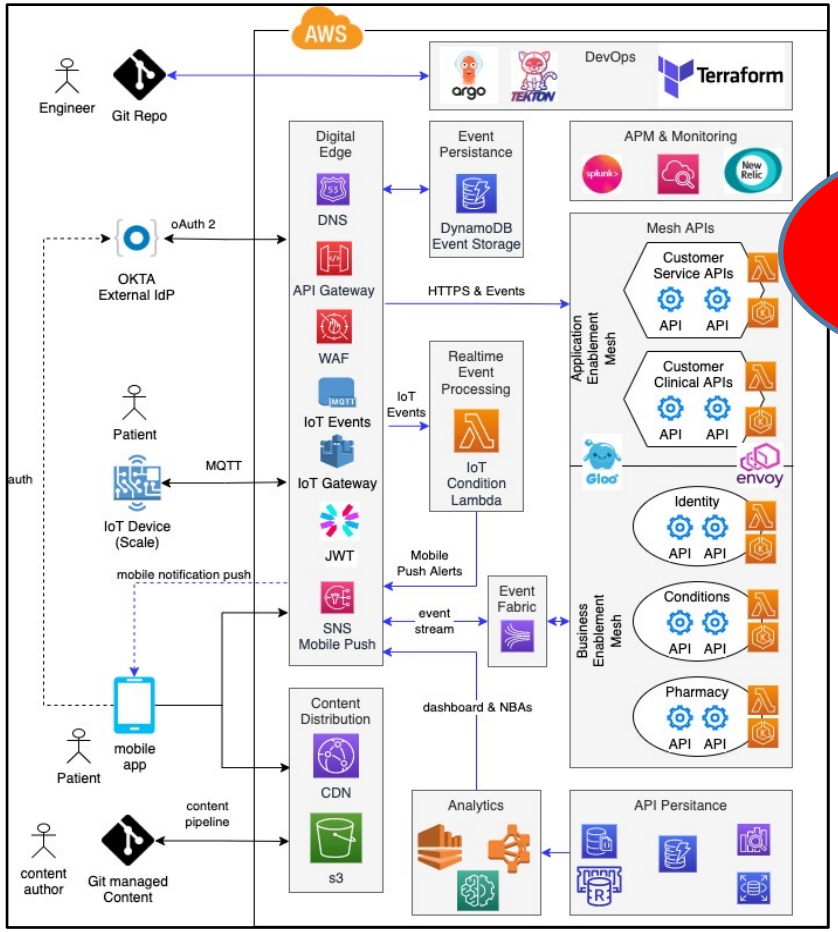
PRE-CLOUD NATIVE





In this world, unless im a network engineer, I need to know very little about networking

CLOUD NATIVE



In this world, the first step is to build a virtual data center from an IP block provided that can be optimized to your application, including network security. This will be the topic later in SDN.

Cloud Native Requires Software Engineers to Cover a Broad Space – It can be overwhelming

Conversation = Protocol

- How did you know what to say?
- Other conversations?
  - Ordering?
- What else did you assume?

- But it's really two conversations –
  - One on the "how are you doing?"
  - One on how the data gets from one side to the other
    - Speech
    - IM/Text
    - Sign Language
    - Smoke Signals
  - And they are independent – can break a conversation into interchangeable parts (we'll call these layers)

Conversation Reliability

- Conversation Reliability – Networks are inherently unreliable (especially as we get to the physical layer)
  - We don't know if:
    - The other party heard what we said
    - We heard what the other party said
    - If the other party is even still there, or if the other party is just slow to respond
- This is a big challenge in distributed computing
- Other issues:
  - Latency
  - HTTP/2 Head of Line Blocking
  - Error detection

The network (and associated protocols) is a critical piece of infrastructure that is required to mitigate many of these issues

Network connections between applications (think conversations)

- Connection-oriented
  - Two parties have a handshake before data
  - All data is in order
  - Like a telephone call

- Connection-less
  - No Handshake
  - No order
  - Like the postal system

- Can occur with any protocol / conversation
- Unique to that protocol.
- Other Key protocol classificagtion types: Statefull vs Stateless

Protocol

- The heart of the network is the protocol
- Similar to many things that you already use
  - Telephone conversation
  - Ordering food
  - What time is it?
  - Lots and lots of others

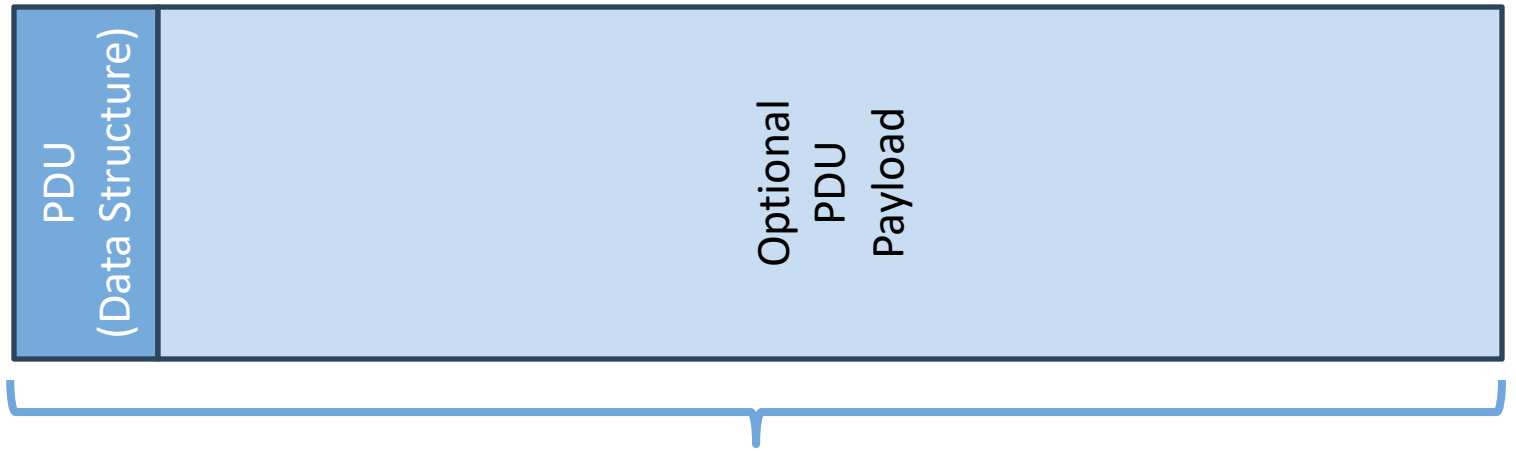
- Definition:

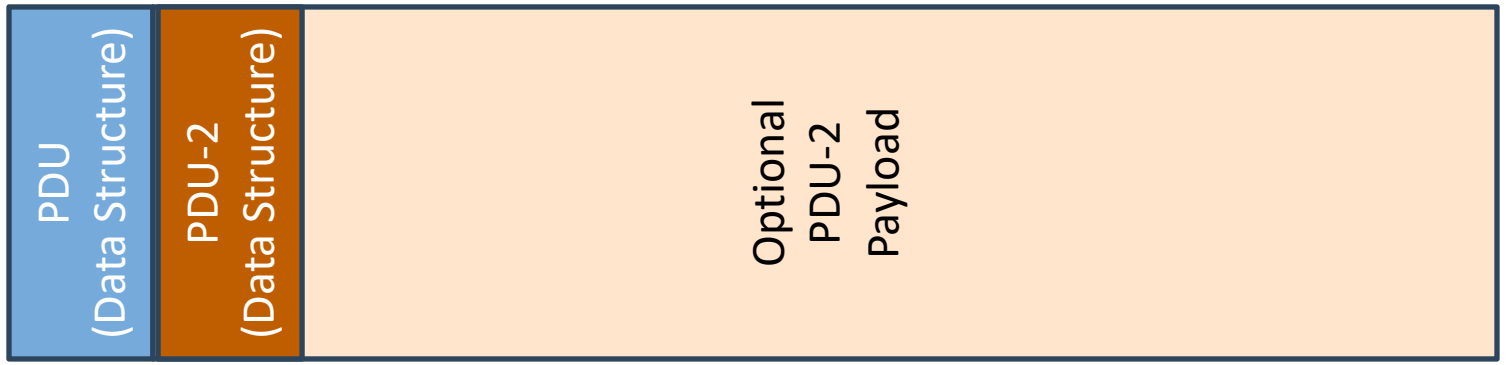
  Defines the messages sent and received and the actions that are performed when messages and network events are received.

Network Protocols

- **Deterministic** – only one possible action for each message in each state
  - For example, "How are you doing?" always gets "it depends"
    - Not random requests and responses
  - Deterministic Finite Automata (DFA) guides the conversation
    - State – different states of the conversation (ordering food example)
  - With network protocols the deterministic specification is governed by RFCs
- Time (some concept) – no response (aka timout) in a period of time is also an action
- Well defined messages (what makes them up) & how you know that they are done ("OVER")
- Who starts?
- What direction is the data?
- **TRUST!**

Key Concepts – PDU are Protocol Building Blocks



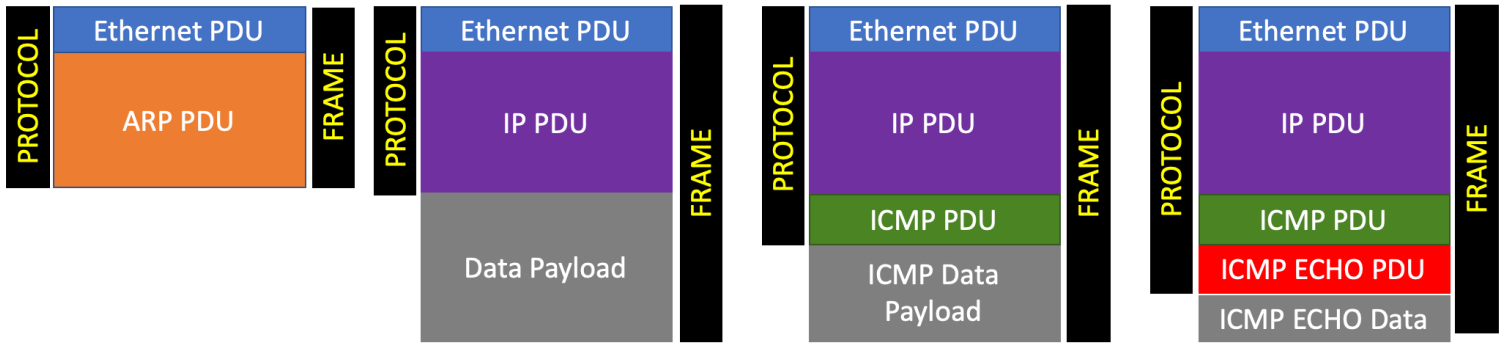


PDUs can be “Stacked” or Encapsulated – Eg, ARP, ICMP

Key Concepts – PDU and DFA

Remember this slide from before?

- PDU – Protocol Data Unit – For now think of it as a basic data structure that outlines the key properties of a given protocol
- Protocol – We will be studying how network protocols are layered (aka) they stack on each other. For the ones we are using in this initial assignment:



Notice that the protocol is the stacking of various PDUs. The frame is the entire unit that is transferred over the network. The frame often has protocol specific data at the end. We will see other protocol types in this class like TCP/IP, UDP/IP, HTTP, etc

PDU – Basically a data structure

Protocol – Stacks PDUs to represent useful information over the wire

BUT… What governs "state" are we allowed to send any Protocol Frame at any Time?

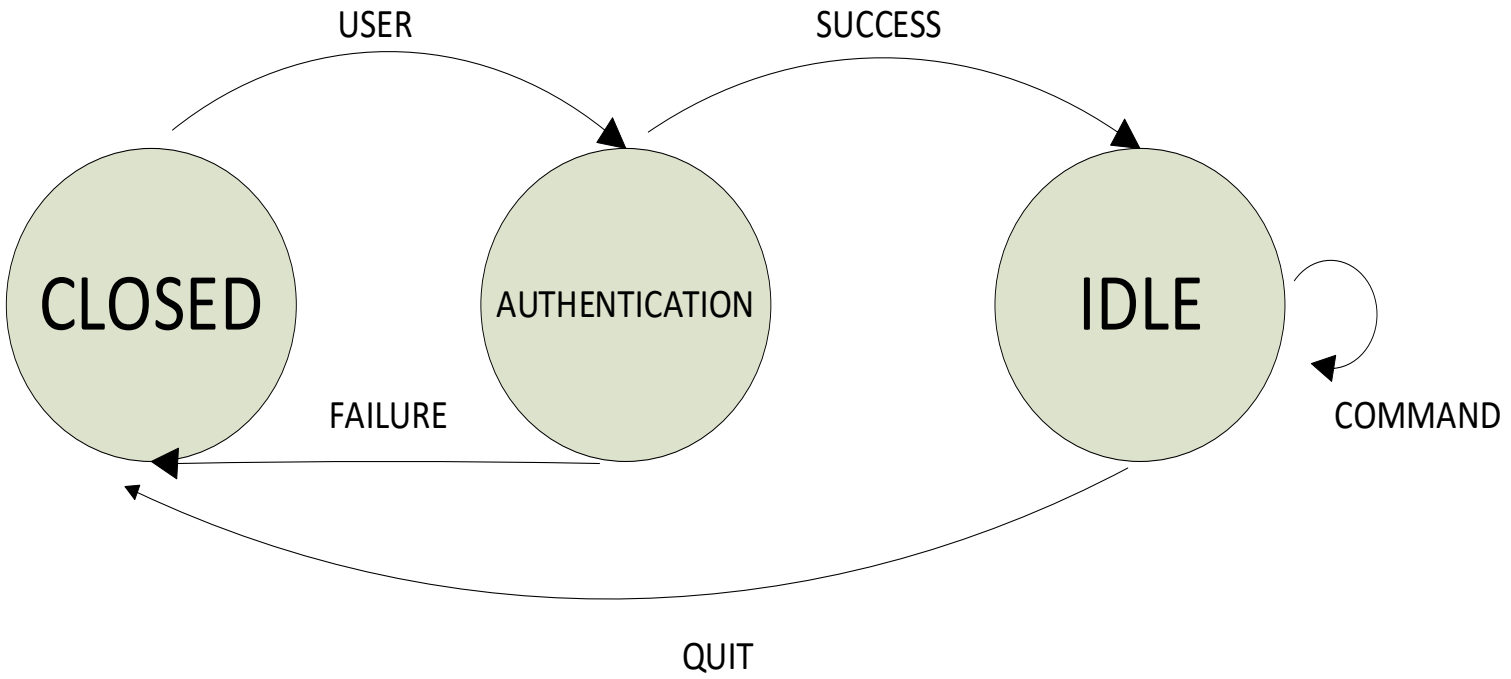
What is a DFA?

- Deterministic Finite Automata (also known as a Finite State Machine)
  - Graphical representation of a protocol where:
    - The nodes are the states of the protocol
    - The directed edges are messages sent or received, or events which happen in the protocol
    - NOT A FLOWCHART!

- Why do we need a graphical representation?
  - To help each side follow the "conversation" easily
  - Nodes are the stages of the conversation.
- **DFAs are used to model stateful protocols** – DFA specifications provided in RFC

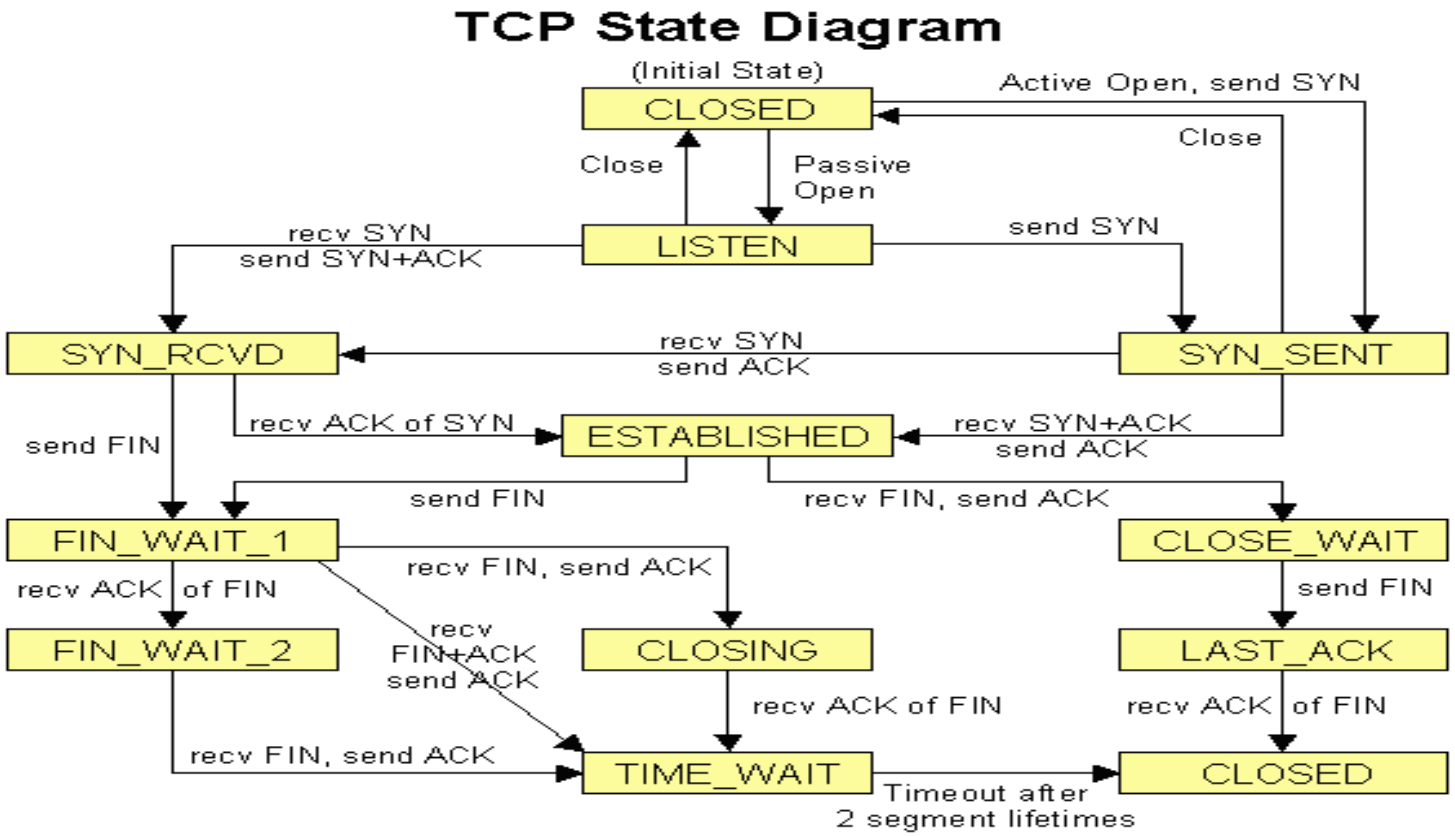
Is a DFA needed for a stateless protocol?

Simple DFA

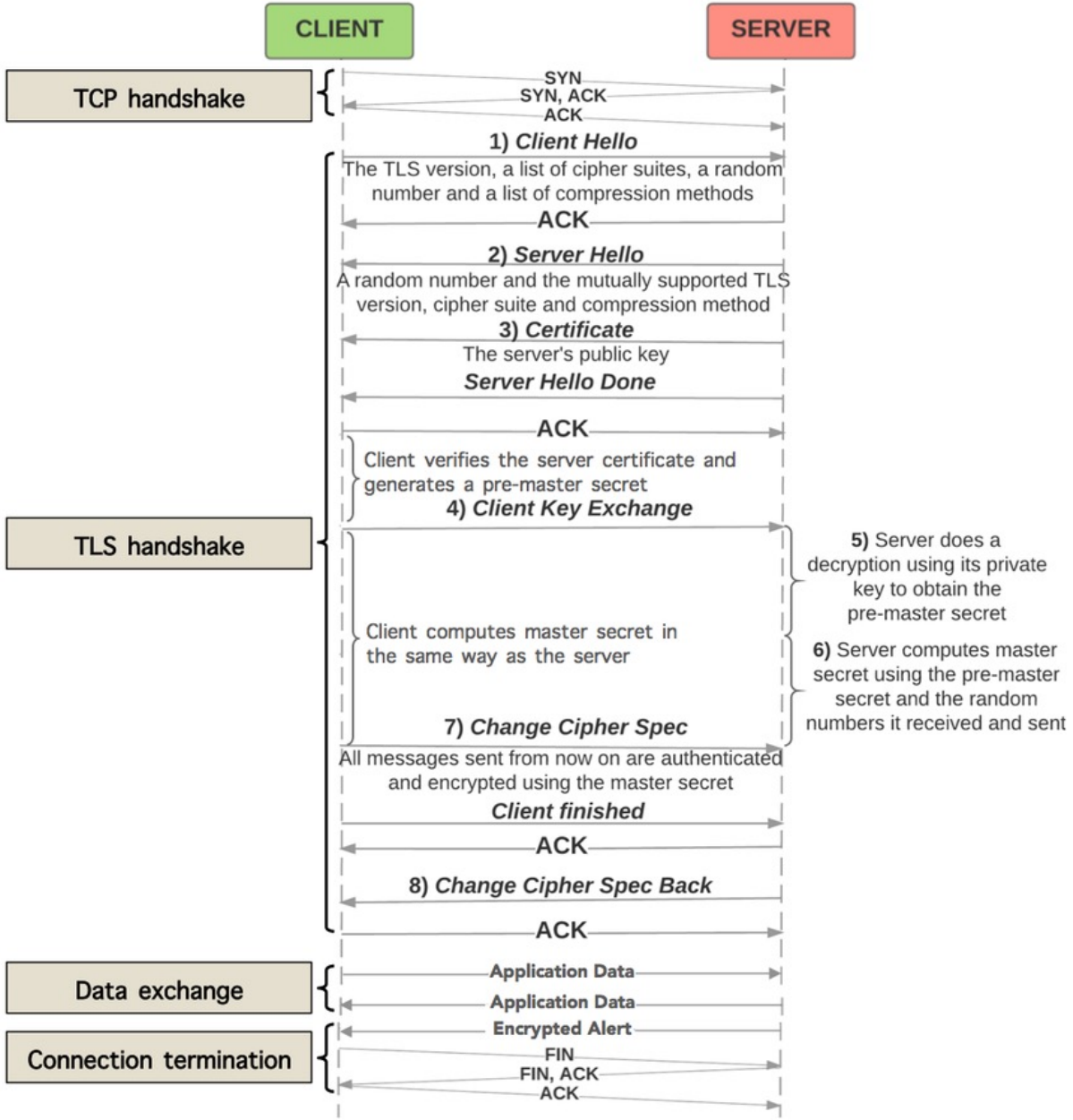


DFAs are governed by well defined states, and well-defined state transitions.
At any given state, what can we do?
What happens if we encounter an error?
Are there different types of errors that we must deal with?

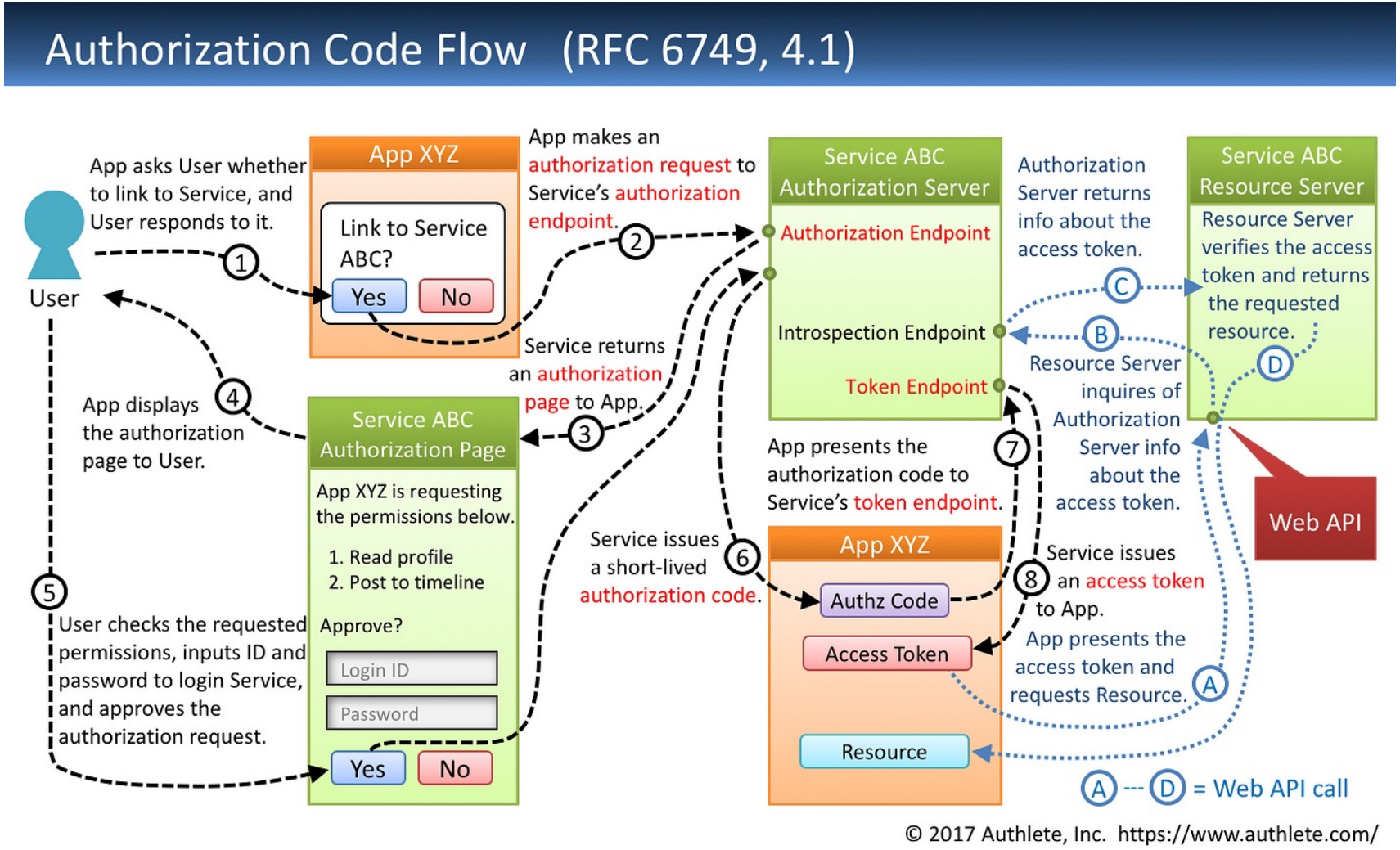
A more complicated DFA



Another DFA, shown as a sequence diagram



Another DFA you use often

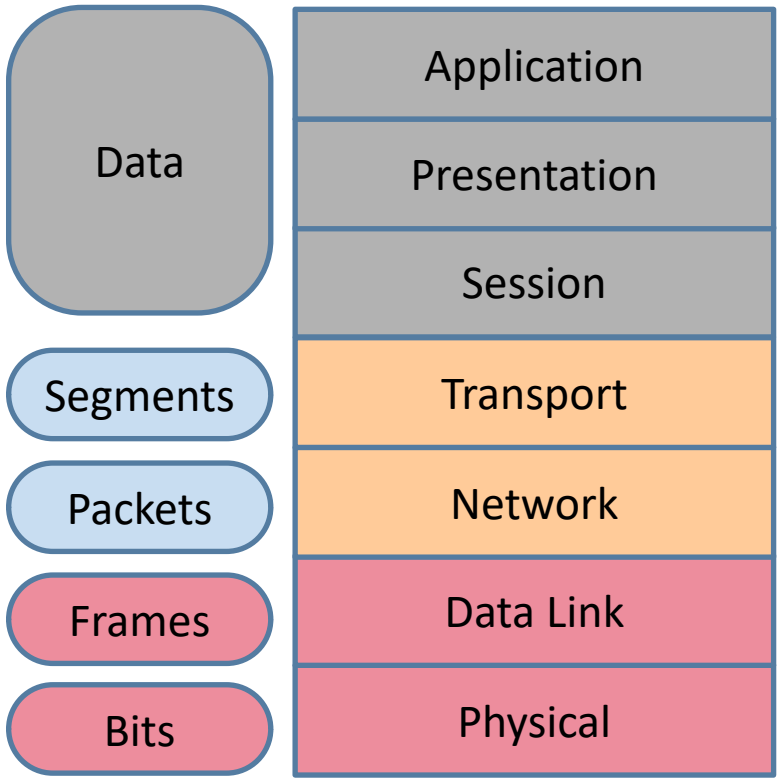


Network Protocols – Stateful or Stateless

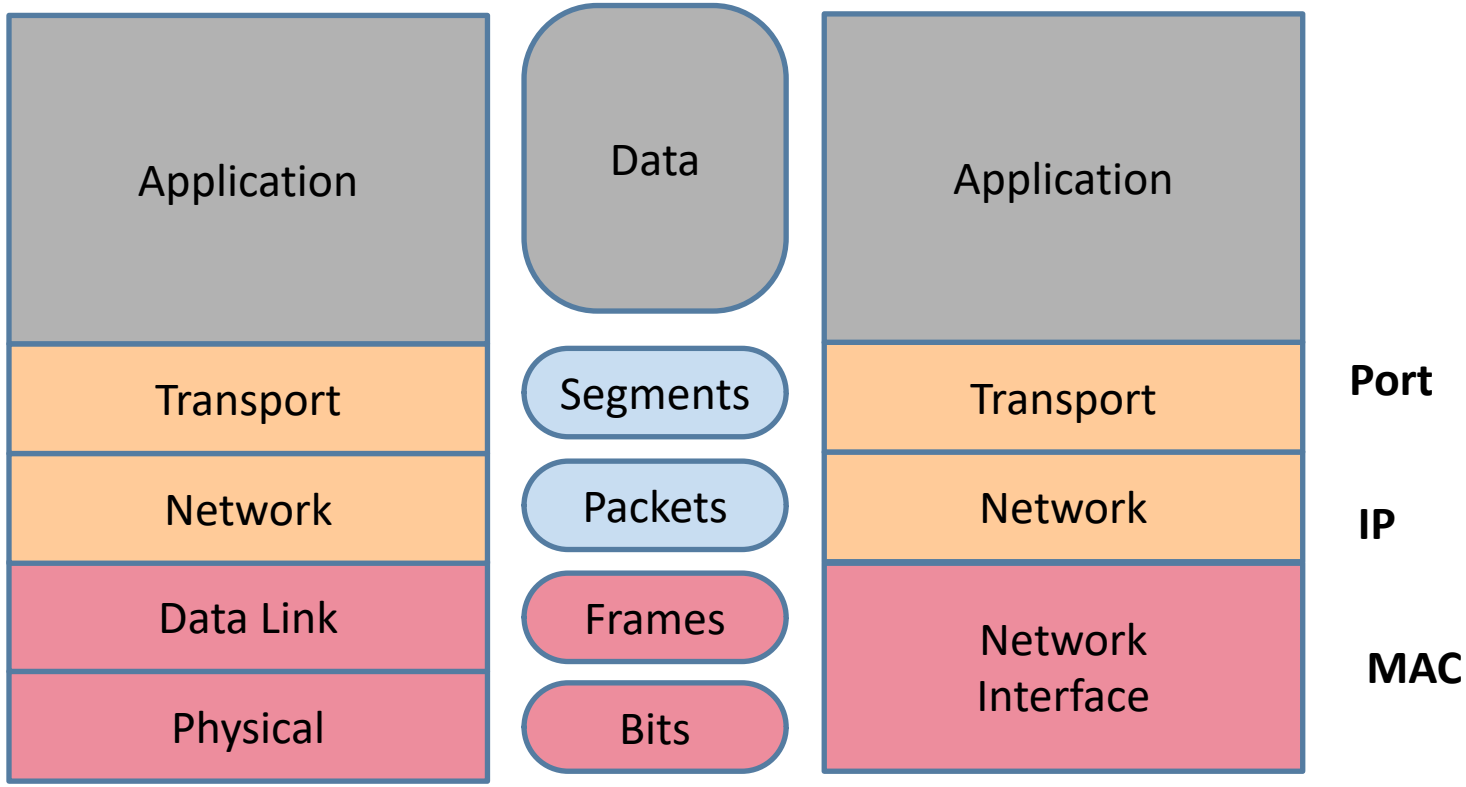
- ARP?
- ICMP?
- HTTP?

Network Reference Models

OSI Model

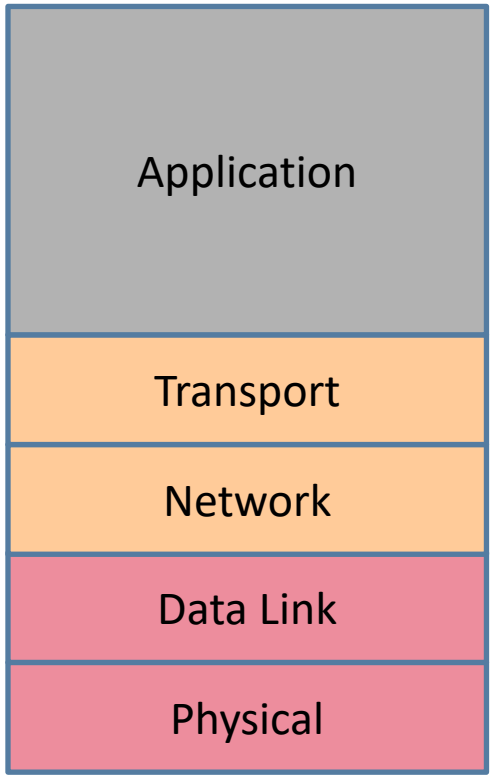


TCP/IP Model



Sometimes its better to think about the Data Link and Physical separately; sometimes its easier to think about combining them into a single Network Interface Level

TCP Model – Key Concepts



Think about abstractions for running applications that can communicate with other applications on any node in the network (of course with security permissions)

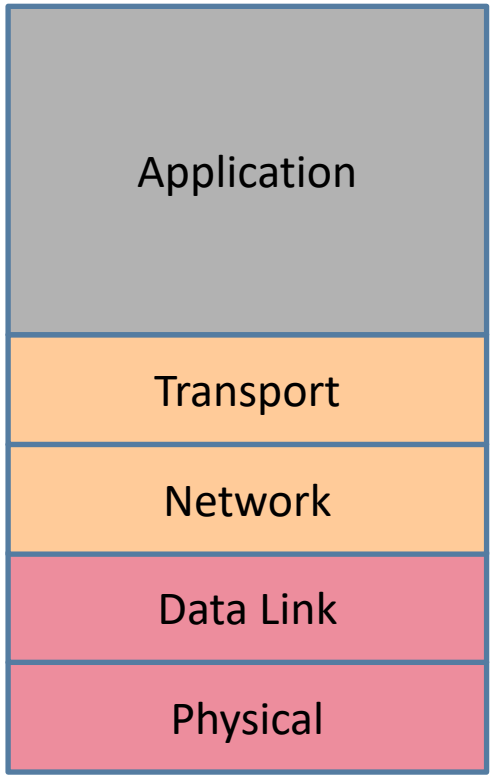
Think about how data gets routed to an application once it gets to a node

Think about getting data to any node in a network

Think about transferring data between adjacent nodes in a network

Think 1's and 0's on a physical media

TCP Model – Key Concepts



Socket API

Port Number

IP Address

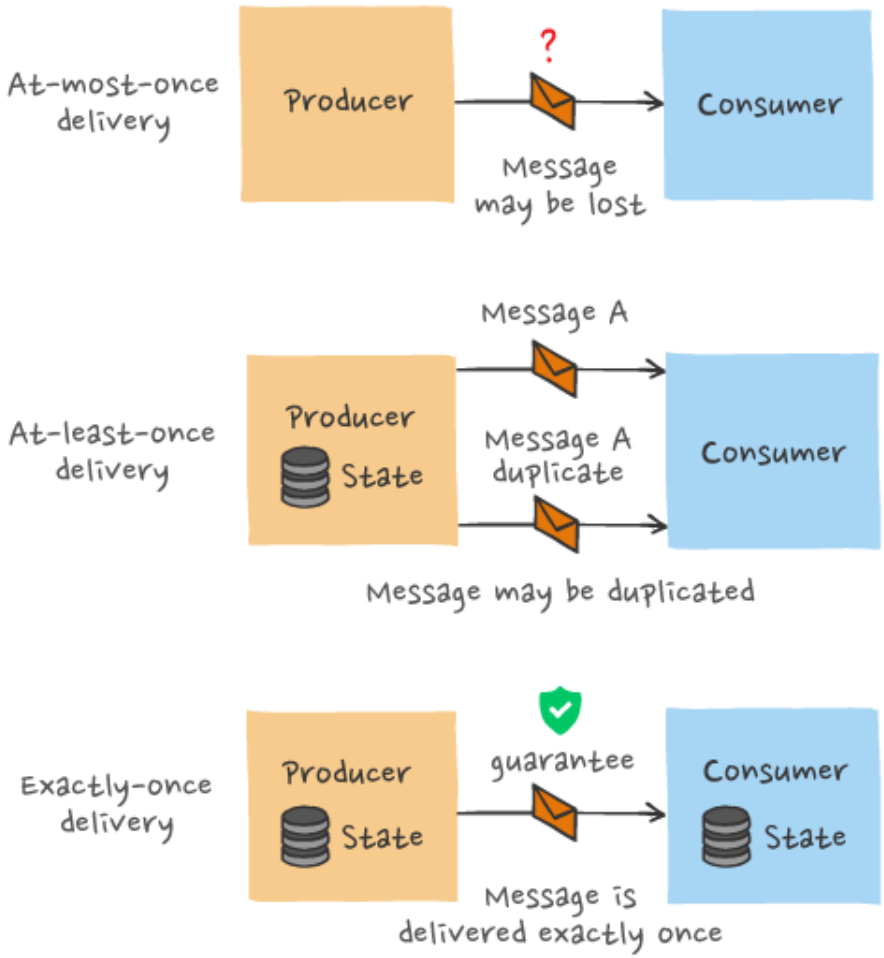
MAC Address

PAM-2/ PAM-4 / QAM, etc

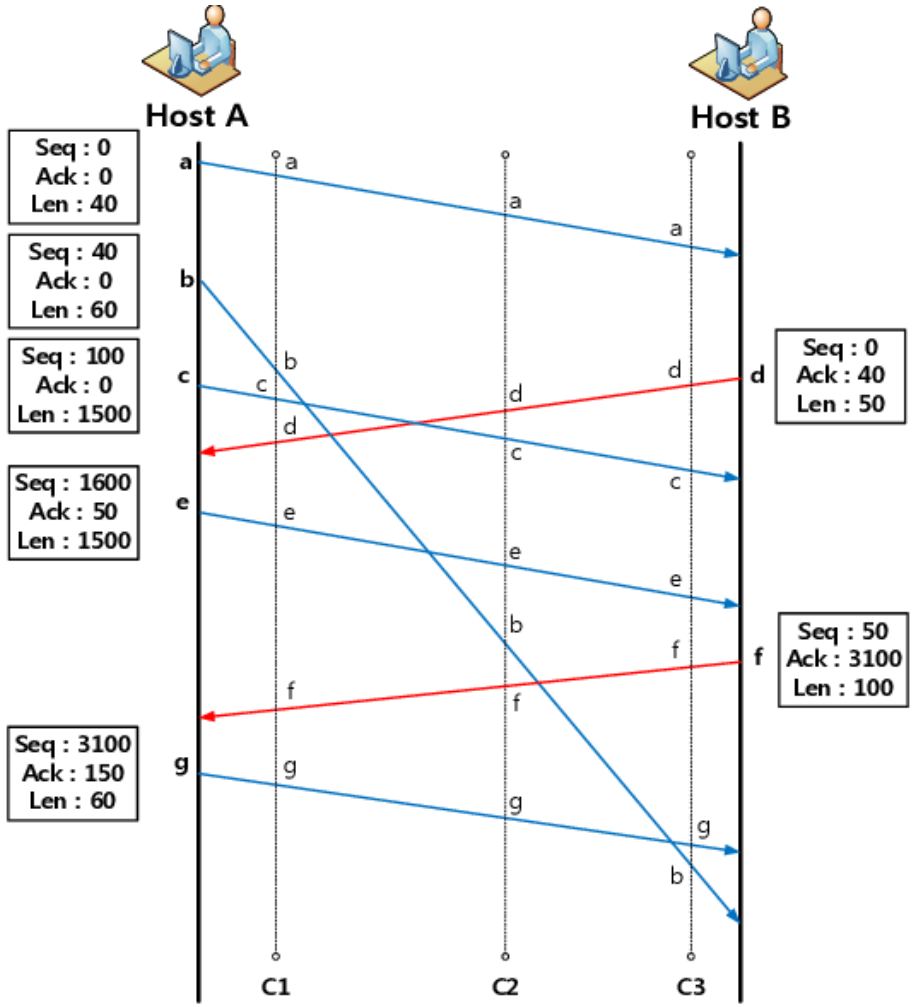
Questions, What layer is …

1. Ethernet
2. ARP
3. ICMP
4. IP
5. TLS
6. HTTP?

Other issues! Delivery Semantics and Out-of-Order Messages



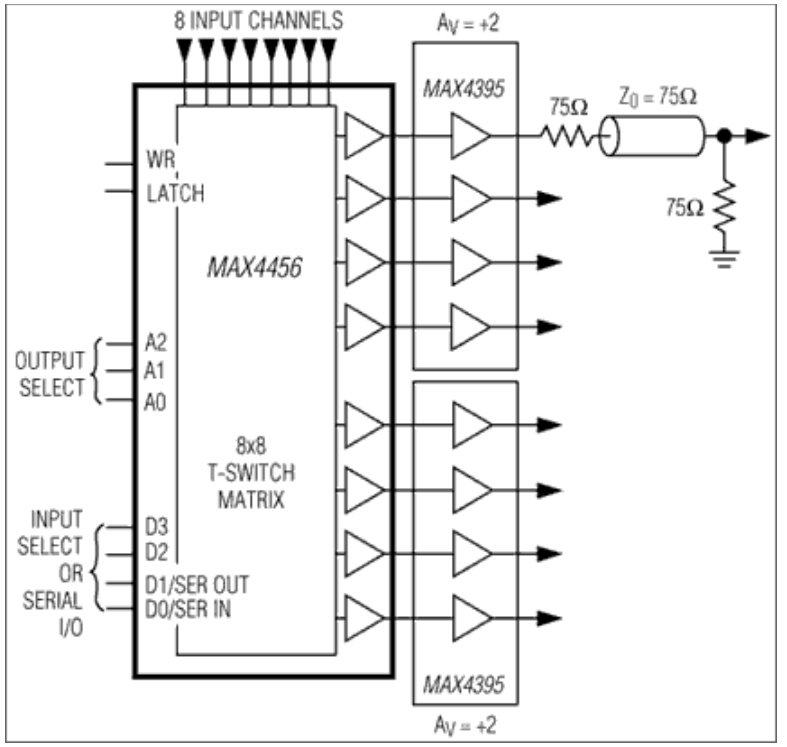
The ideal of exactly-once is pretty hard, That's more of a topic for a distributed computing class

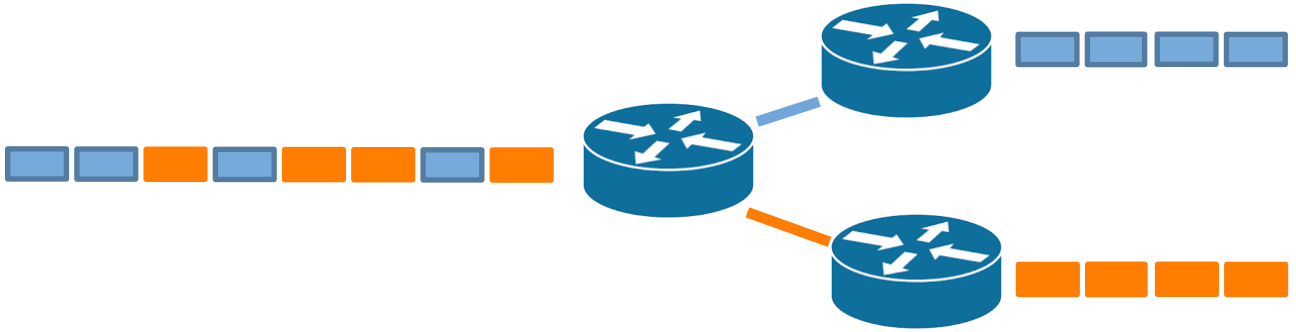


Google – Lamport Logical Clock if you are interested in learning more about this now

Network connections between links

- Circuit Switched
  - Establish a dedicated "electrical" path
  - Use it
  - Tear it down
- Packet Switched
  - Take a message and break it up to parts of reasonable size
  - Route each part to where it needs to go
  - Reassemble original message
- For all practical purposes, modern networks require packet switching, why?





Network terminology

- Two types of hosts
  - Server – provide services (WWW, etc.)
  - Client – use services

- Three types of applications
  - Server – provides information
  - Client
  - Peer (both server and client at the same time)
    - Have different approaches to the network

The Network

- Review: Applications (client, server, peer) communicate via the network
  - Connection-oriented (TCP)
  - Connection-less
- Connection-less = UDP (RFC 768 and others) - https://www.rfc-editor.org/rfc/rfc768.html (Aug 1980)
  - User Datagram Protocol
  - Just send

- What is an RFC? (Request for Comments) – Question on Hw1

RFC1 – April 1969; RFC 9311 – September 2022 (https://www.rfc-editor.org/rfc-index.txt)
Many of the RFCs we will be looking at are over 40 years old!

Hmmm – UDP is Unreliable, why would we want that?

The reliable network

- Connection-oriented = TCP (RFC 793 and many others)
  - Transmission Control Protocol
  - https://www.rfc-editor.org/rfc/rfc793.html. Also old, September 1981. Based on a 1974 research paper
  - Not really connected – just a handshake (a "virtual" connection)
  - Other features of TCP:
    - Reliable Data Transfer
    - Flow Control / Back Pressure
    - Congestion Control
    - Define each

- There are other types (SCTP, DCCP, NORM, etc.)
  - Why? Adoption Challenges?
- Newer approaches – QUIC – RFC 9000 (May 2021)

https://www.cs.princeton.edu/courses/archive/fall06/cos561/papers/cerf74.pdf

Network core

- How do we get between systems (either computers or devices?)
- US Mail-like system
  - No connection, each envelope has full address, no reliability
- Telephone-like system
  - Connection – each envelope goes along the same path (circuit) reliably.
- So, what is the Internet?

The Internet

- All information is broken up into PACKETS
- The best "route" is found from source to destination
  - Which links (connections) are the best to use
  - Think of driving to the shore – which roads are best?

- This "routing" is the same for both US-mail and telephone based routing (just at different times)

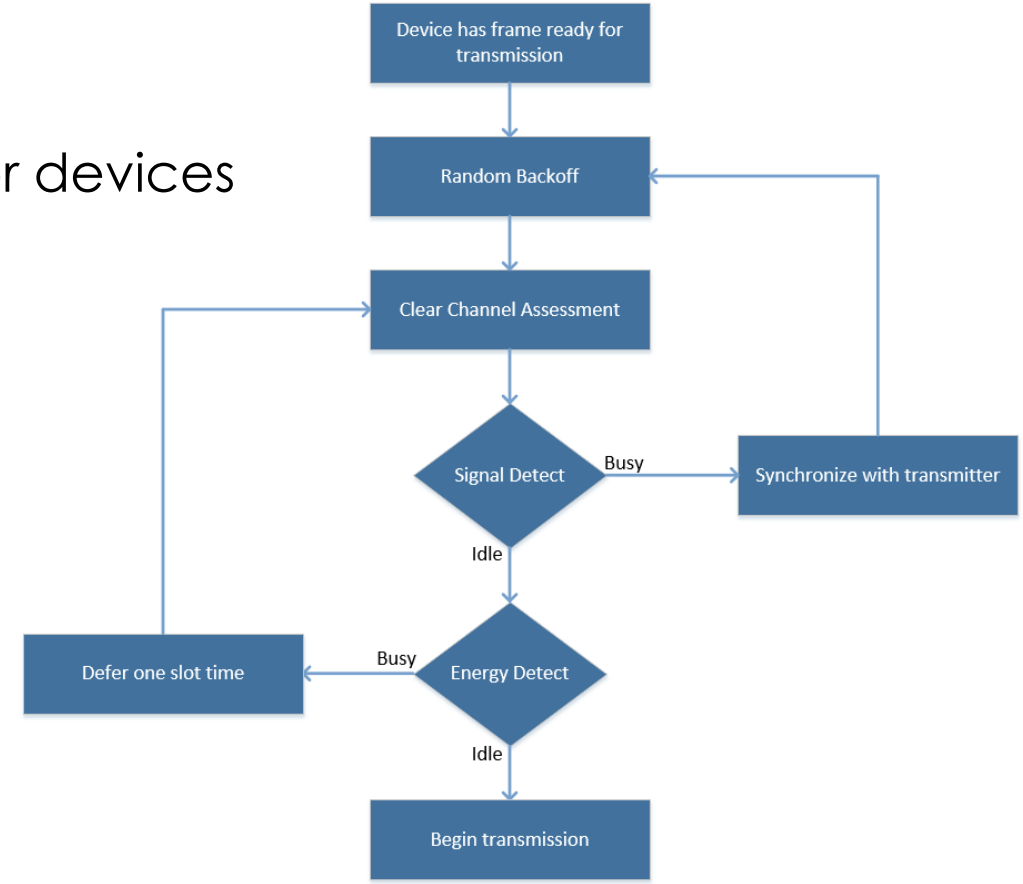
- How do you make sure that everyone is moving well?
  - This is where routing comes in – the optimal route depends on a number of factors
    - Number of hops
    - Link Speed
    - Congestion

Network Types

- US-Mail = Datagram network
- Telephone-based = Virtual Circuit network
- Which is better?

The Link

- The direct connection between computers or devices
- Two categories:
  - Point-to-point
  - Shared (e.g. Air, some cables)
- Also two types:
  - Wired (guided)
  - Wireless (unguided)



Wired: Full Duplex, Wifi: Half Duplex -> Cant have collisions – managed by things like CSMA/CA

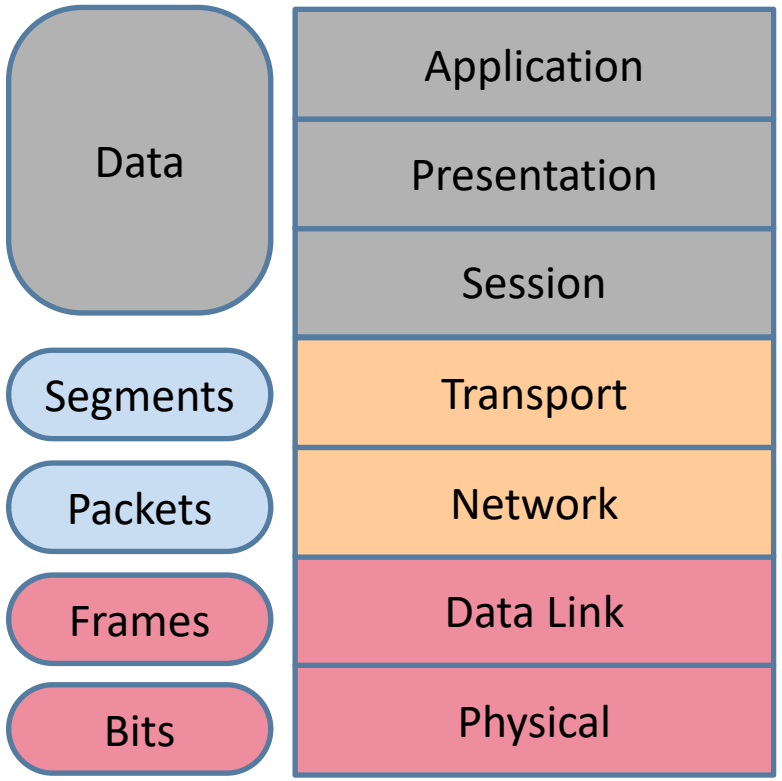
The Link

Difference between Guided and Unguided Media

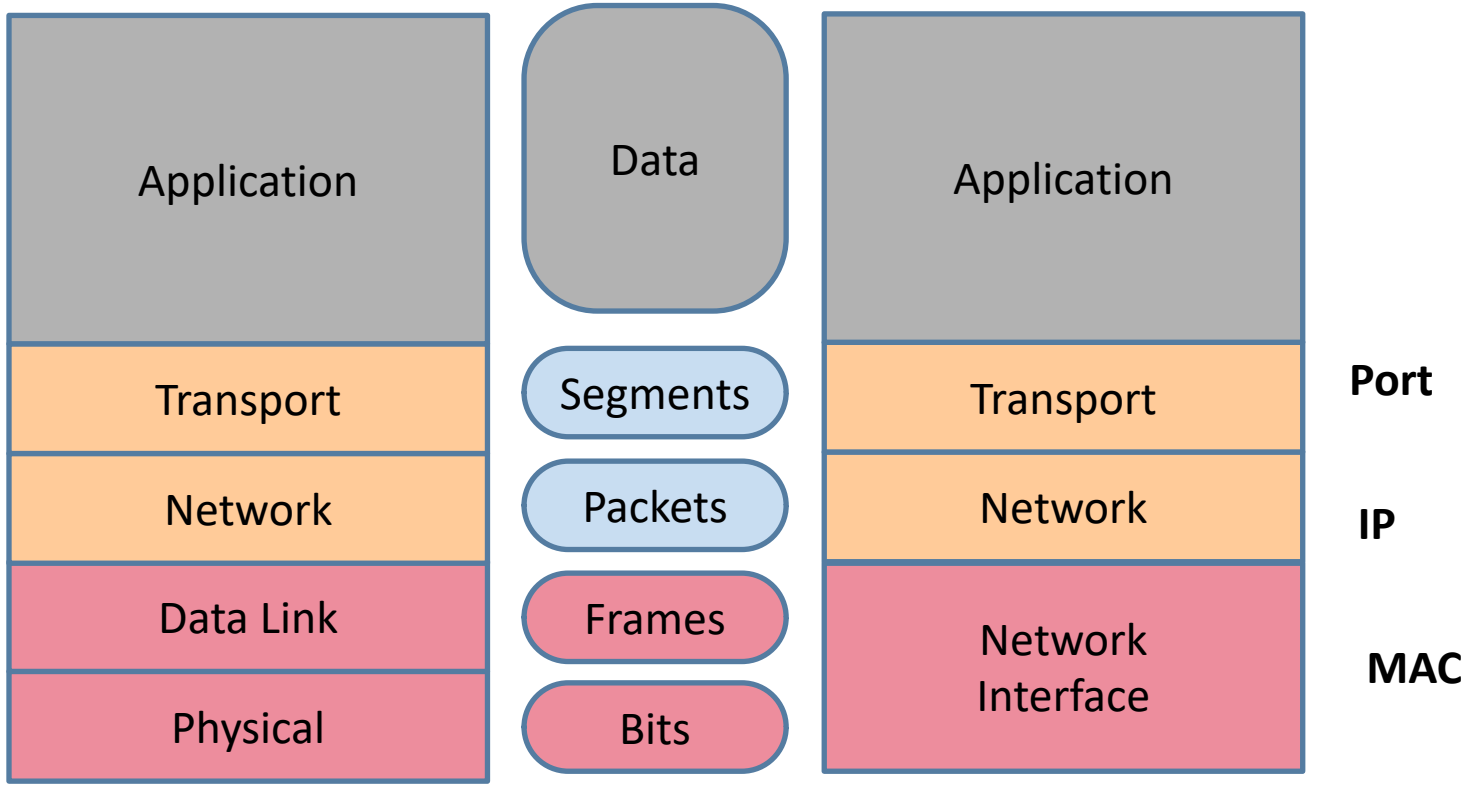
S.No.	Guided Media	Unguided Media
1.	In guided media, the signal energy communicates via wires.	In unguided media, the signal energy communicates through the air.
2.	Guided media is generally preferred when we want to execute direct communication.	Unguided media is generally preferred for radio broadcasting in all directions.
3.	The guided media formed the different network topologies.	The unguided media formed the continuous network topologies.
4.	Here, the signals are in the state of current and voltage.	Here, the signals are in the state of electromagnetic waves.
5.	In the case of guided media, the transmission capacity can be boosted by counting more wires.	In the case of unguided media, it is not feasible to acquire more capacity.
6.	Open Wire, Twisted Pair, Coaxial Cable, and Optical Fibre are the different kinds of guided media.	Microwave Transmission, Radio Transmission, and Infrared Transmission are the types of unguided media.

Network Reference Models

OSI Model



TCP/IP Model



Sometimes its better to think about the Data Link and Physical separately; sometimes its easier to think about combining them into a single Network Interface Level

TCP/IP, UDP/IP MAC, IP Address, Port & US Mail

Sender Address **Recipient Address** **Quality of Service**

[Your Name]
[House Number] [Street Name]
[City] [State] [Zipcode]

TCP and UDP
Will be the Key
Protocols we will be
studying

[Your Name] = Port Number
[House Number] = Host Part of IP Address
[Street Name] = Network Part of Host Address
[House Number, Street Name] = MAC Address
[City] [State] [Zipcode] = Network Identifier/Routing point (post office)

TCP/IP Reference Model

- Application Layer (HTTP (1, 2, 3), FTP, SMTP, IMAP4, BitTorrent, etc.)
- Transport Layer (TCP, UDP, QUIC,SCTP, DCCP, NORM, etc.)
- Internetworking Layer (IPv4, IPv6)
- Link Layer (Ethernet, 4G, LTE, HSPA+, 5G, Wifi 5, Wifi 6, etc.)
- Physical Layer (these two are sometimes together as Host to Network Layer or NIC Layer)
    - Transport Layer Security sublayer added later between application and transport layer

Layering

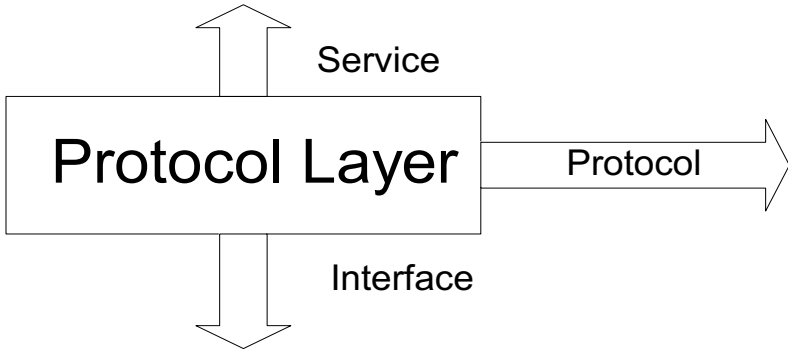
- Why layers?
  - Concentrates similar functions together
  - Presents a "building block" approach to networks
  - Allows "Mix & Match" of boxes.

- Sublayers – layers can be broken down into sublayers if functions can be grouped.

- Other terms:
  - Service Access Points (SAP)
  - Protocol Data Unit (PDU)

Layering concepts

- Service
- Interface
- Protocol



Sublayers

- Dividing up one layer into multiple sublayers because of functionality.
- Can we add layers or sublayers to our models?

Interfaces Between Layers

- Sometimes we will see things added between layers
  - TLS. (Sits between Application and Transport)
  - ARP (Sits between Data Link and Physical Layers)

Common themes

- We'll see the same concepts used for the same reasons at multiple layers
  - Addressing
  - Connection Control (do I have a connection or not?)
  - Ordered Delivery (up)
  - Segmentation & Reassembly
  - PDU definition
  - Error Detection / Correction
  - Flow Control
  - Multiplexing (more than one lower or upper service)

The BIG themes

- SECURITY!
  - Data Integrity
  - Confidentiality
  - Access Control
  - Availability
  - Authentication
  - Non-Repudiation

- Is there an implied security in each conversation that you have?

- Quality of Service (QoS):
  - What you get out of a protocol (DIFFERENT THAN MOST BOOKS)

Other reference models

- The three major models:
  - OSI Basic Reference Model
  - TCP/IP (seen before)
  - Asynchronous Transfer Mode (ATM)
- Others
  - Infiniband
  - SDN
  - NDN

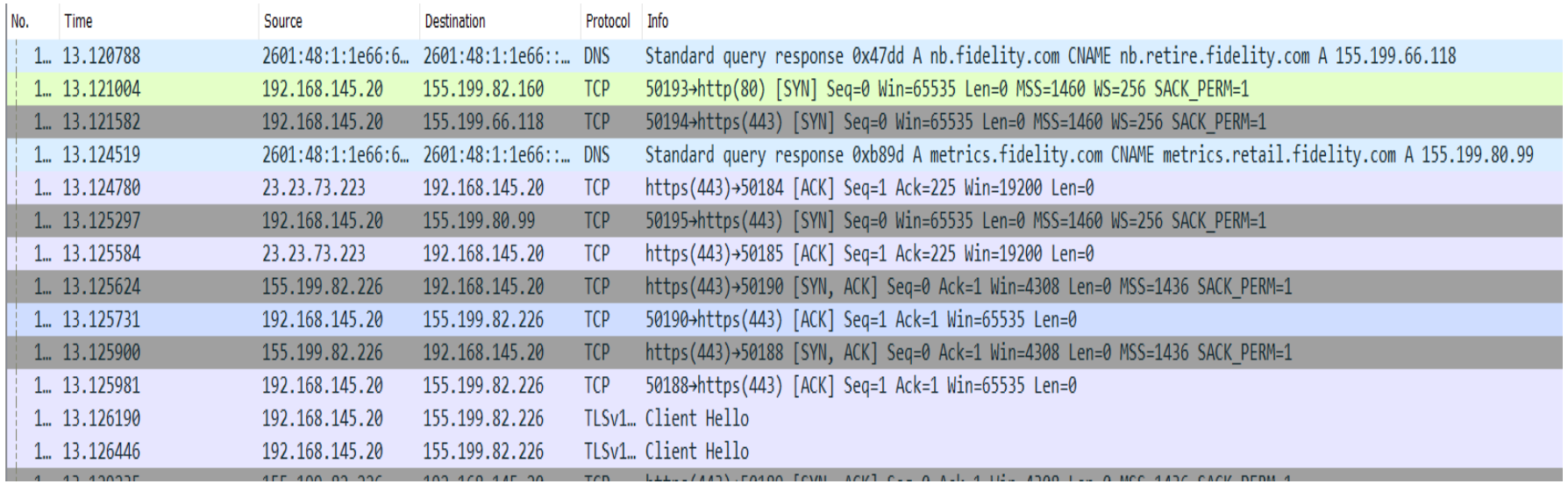
- Many others both old (BNA, SNA, DecNet, AppleTalk) and new (HTTP 2.0, NDN, SDN, etc.)

- MANY DIFFERENT WAYS TO BREAK UP THE PROBLEM OF NETWORKING

Protocol stacks

- How do we link the stack together?
  - Usually a reference in the PDU header
  - Which is set on the way down the stack
  - For use on the way up the stack on the other side

- Wireshark example – Get that installed and start playing with it ASAP – there are some good tutorials on YouTube, Ill be sending out some links)



No.	Time	Source	Destination	Protocol	Info
1…	13.120788	2601:48:1:1e66:6…	2601:48:1:1e66::…	DNS	Standard query response 0x47dd A nb.fidelity.com CNAME nb.retire.fidelity.com A 155.199.66.118
1…	13.121004	192.168.145.20	155.199.82.160	TCP	50193→http(80) [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1…	13.121582	192.168.145.20	155.199.66.118	TCP	50194→https(443) [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1…	13.124519	2601:48:1:1e66:6…	2601:48:1:1e66::…	DNS	Standard query response 0xb89d A metrics.fidelity.com CNAME metrics.retail.fidelity.com A 155.199.80.99
1…	13.124780	23.23.73.223	192.168.145.20	TCP	https(443)→50184 [ACK] Seq=1 Ack=225 Win=19200 Len=0
1…	13.125297	192.168.145.20	155.199.80.99	TCP	50195→https(443) [SYN] Seq=0 Win=65535 Len=0 MSS=1460 WS=256 SACK_PERM=1
1…	13.125584	23.23.73.223	192.168.145.20	TCP	https(443)→50185 [ACK] Seq=1 Ack=225 Win=19200 Len=0
1…	13.125624	155.199.82.226	192.168.145.20	TCP	https(443)→50190 [SYN, ACK] Seq=0 Ack=1 Win=4308 Len=0 MSS=1436 SACK_PERM=1
1…	13.125731	192.168.145.20	155.199.82.226	TCP	50190→https(443) [ACK] Seq=1 Ack=1 Win=65535 Len=0
1…	13.125900	155.199.82.226	192.168.145.20	TCP	https(443)→50188 [SYN, ACK] Seq=0 Ack=1 Win=4308 Len=0 MSS=1436 SACK_PERM=1
1…	13.125981	192.168.145.20	155.199.82.226	TCP	50188→https(443) [ACK] Seq=1 Ack=1 Win=65535 Len=0
1…	13.126190	192.168.145.20	155.199.82.226	TLSv1…	Client Hello
1…	13.126446	192.168.145.20	155.199.82.226	TLSv1…	Client Hello

Security

- Security is sometimes (mostly) not totally thought out
  - Protocols leak data
  - Protocols don't secure data
  - Implementations don't check boundary conditions in protocols
    - 3 way transfer example of FTP
  - Protocols don't validate other endpoint
  - Protocols assume trust

Appendix :
Introducing some basic concepts and terminology

Command Line Tools we will be using in this demo

- Note you may or may not have these tools on your computer, they are available online as open source for Mac, Windows and Linux – just google if you don't have them
- **Ifconfig** (ipconfig on windows) – tool to explore interfaces on the device you are using
- **arp** – tool to explore devices on link-local network that your machine has interacted with
- **arp-scan** – tool to explore all devices on the link-local network. This tool generally requires admin access on your device
- **nmap** – tool that does a lot of things, but we will use it for exploring the network.

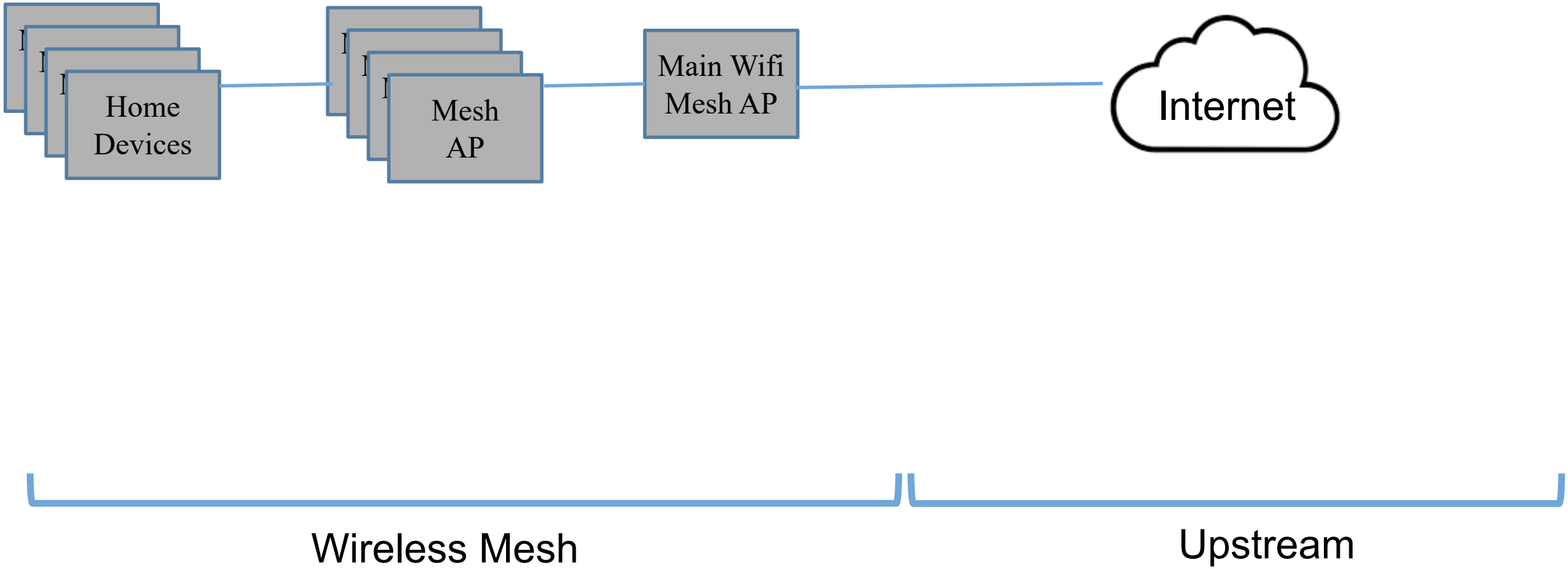
Note you may not have some of these tools on your computer, but they are available for free for Mac, Windows and Linux. Also note, some of these tools require you to have sys-admin level access – tools that snoop down into the raw protocols need additional access

Querying Network Interfaces

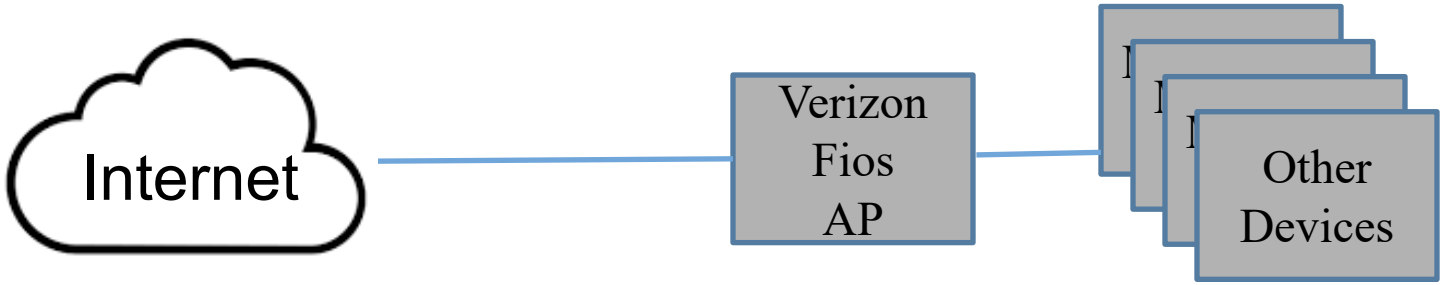
```
→  ~ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6463<RXCSUM,TXCSUM,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSU
M>
        ether f0:18:98:51:15:18
        inet6 fe80::88b:fd4f:4c9e:95f%en0 prefixlen 64 secured scopeid 0x6
        inet 192.168.50.25 netmask 0xffffff00 broadcast 192.168.50.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

```
→  ~ ifconfig lo0
lo0: flags=8049<UP,LOOPBACK,RUNNING,MULTICAST> mtu 16384
        options=1203<RXCSUM,TXCSUM,TXSTATUS,SW_TIMESTAMP>
        inet 127.0.0.1 netmask 0xff000000
        inet6 ::1 prefixlen 128
        inet6 fe80::1%lo0 prefixlen 64 scopeid 0x1
        nd6 options=201<PERFORMNUD,DAD>
```

Demo 1 – my Home Network



Demo 1 – my Home Network

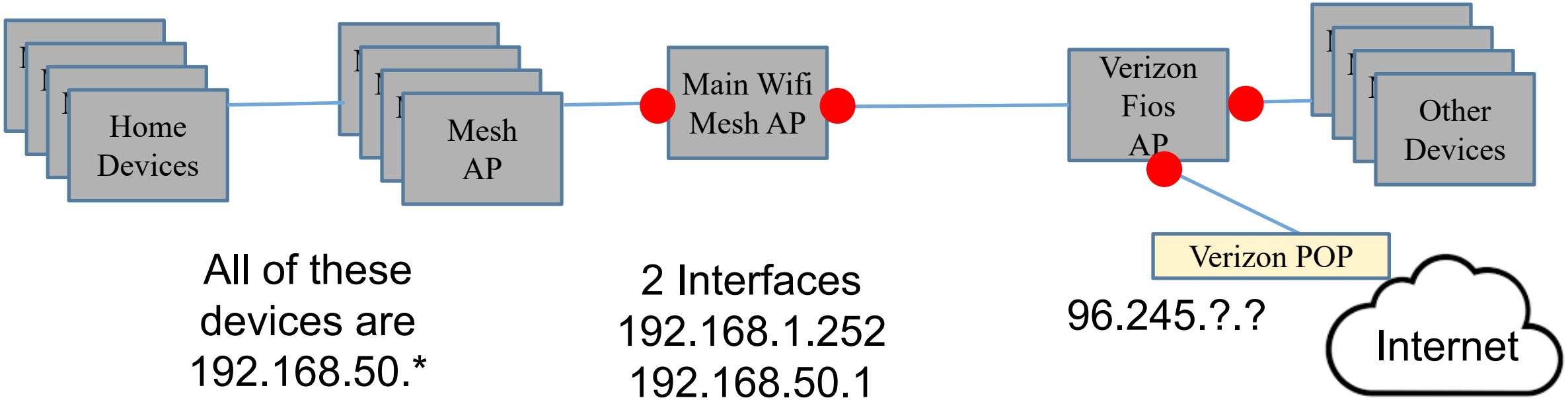


Upstream

My 2nd Network

Demo 1 – my Home Network

All of these devices are 192.168.1.*



All of these devices are 192.168.50.*

2 Interfaces
192.168.1.252
192.168.50.1

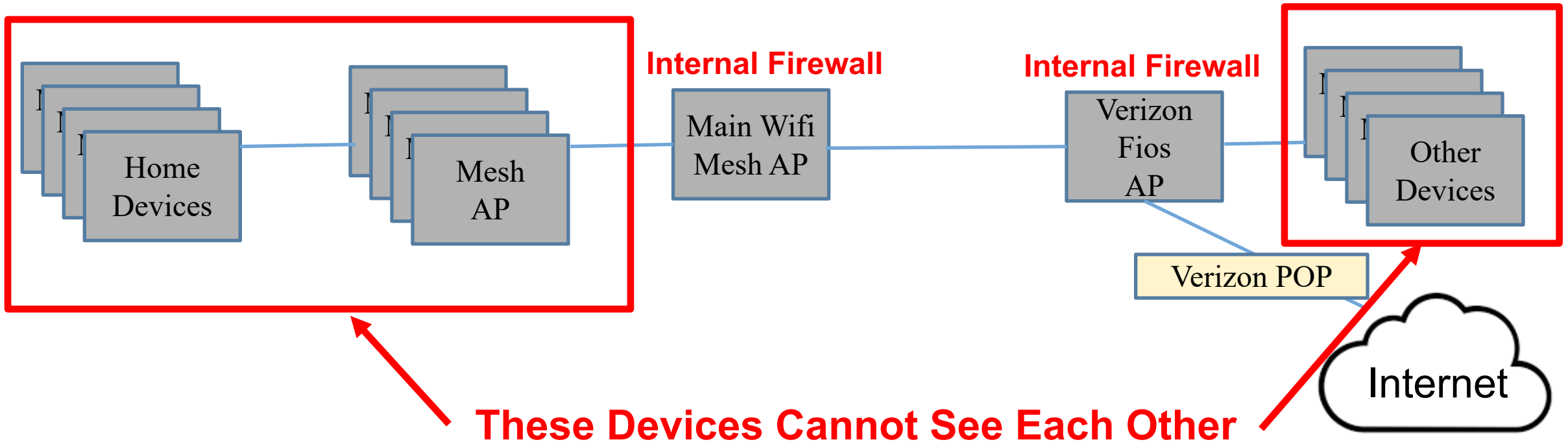
96.245.?.?

Wifi and Wired Network 2

Wifi and Wired Network 1

●Think about these as “Routable” Interfaces

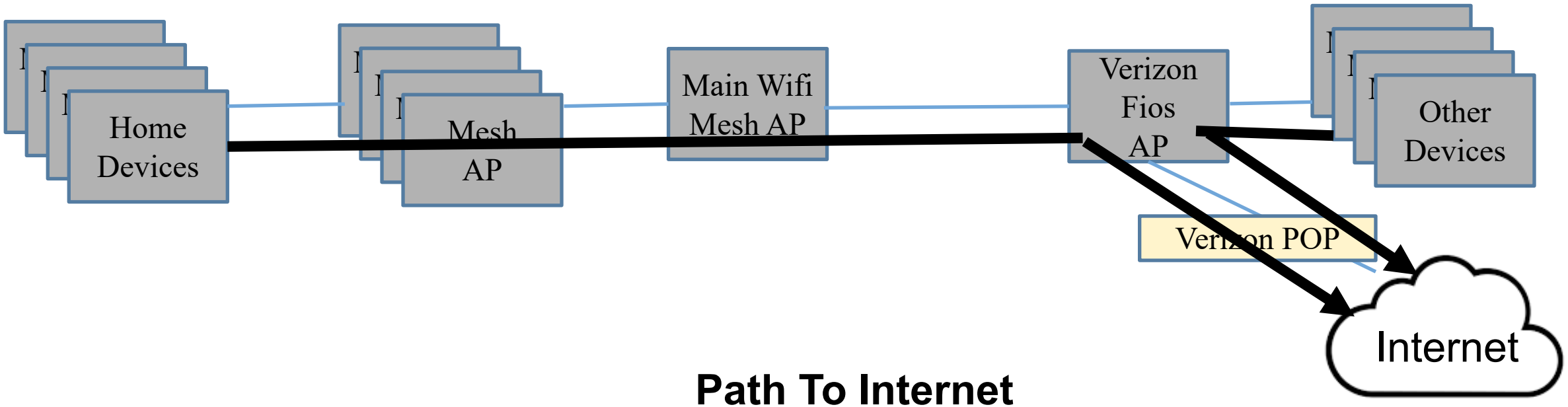
Demo 1 – my Home Network



Wifi and Wired Network 2

Wifi and Wired Network 1

Demo 1 – my Home Network



Path To Internet

Wifi and Wired Network 2

Wifi and Wired Network 1

Demo 1

```
→  ~ ifconfig en0
en0: flags=8863<UP,BROADCAST,SMART,RUNNING,SIMPLEX,MULTICAST> mtu 1500
        options=6463<RXCSUM,TXCSUM,TSO4,TSO6,CHANNEL_IO,PARTIAL_CSUM,ZEROINVERT_CSU
M>
        ether f0:18:98:51:15:18
        inet6 fe80::88b:fd4f:4c9e:95f%en0 prefixlen 64 secured scopeid 0x6
        inet 192.168.50.25 netmask 0xffffff00 broadcast 192.168.50.255
        nd6 options=201<PERFORMNUD,DAD>
        media: autoselect
        status: active
```

On my home network over wifi my IP address is 192.168.50.25, Notice the netmask of 0xFFFFFF00 – remember hex F is 1111 in binary so 6 hex F is 24 1's in a row

So the CIDR(More on this later) of my local network is 192.168.50.0/24

Demo 1

Which are the link local devices?

```
→  ~ sudo arp-scan --interface=en0 192.168.50.0/24
Interface: en0, type: EN10MB, MAC: f0:18:98:51:15:18, IPv4: 192.168.50.25
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.50.1    04:d9:f5:2d:8b:f0       ASUSTek COMPUTER INC.
192.168.50.7    38:2c:4a:cc:2b:18       ASUSTek COMPUTER INC.
192.168.50.44   20:ef:bd:5f:04:c8       (Unknown)
192.168.50.93   54:60:09:6f:7f:0c       Google, Inc.
192.168.50.97   88:63:df:99:a4:35       Apple, Inc.
192.168.50.139  04:d9:f5:2b:5b:40       ASUSTek COMPUTER INC.
192.168.50.151  04:d4:c4:c7:27:b8       ASUSTek COMPUTER INC.
192.168.50.101  ba:f0:7f:d9:31:4b       (Unknown: locally administered)
192.168.50.101  ba:f0:7f:d9:31:4b       (Unknown: locally administered) (DUP: 2)
192.168.50.180  04:d4:c4:c5:82:48       ASUSTek COMPUTER INC.
192.168.50.210  dc:72:23:89:82:00       (Unknown)
192.168.50.158  14:0a:c5:73:9a:4a       (Unknown)
192.168.50.158  14:0a:c5:73:9a:4a       (Unknown) (DUP: 2)
192.168.50.198  a8:6d:aa:c3:72:38       Intel Corporate
192.168.50.198  a8:6d:aa:c3:72:38       Intel Corporate (DUP: 2)
192.168.50.225  a4:83:e7:5f:da:cf       Apple, Inc.
192.168.50.231  66:93:cb:00:63:47       (Unknown: locally administered)
192.168.50.36   f8:b4:6a:0f:c5:7c       Hewlett Packard
192.168.50.46   62:0d:96:3c:39:44       (Unknown: locally administered)
192.168.50.46   62:0d:96:3c:39:44       (Unknown: locally administered) (DUP: 2)
```

Demo 1

```
→  ~ nmap -sn 192.168.50.0/24
Starting Nmap 7.93 ( https://nmap.org ) at 2022-09-12 19:32 EDT
Nmap scan report for RT-AX92U-8BF0 (192.168.50.1)
Host is up (0.072s latency).
Nmap scan report for 192.168.50.7
Host is up (0.078s latency).
Nmap scan report for BRIANs-MBP (192.168.50.25)
Host is up (0.0011s latency).
Nmap scan report for NPI0FC57C (192.168.50.36)
Host is up (0.026s latency).
Nmap scan report for 192.168.50.62
Host is up (0.084s latency).
Nmap scan report for Google-Home (192.168.50.93)
Host is up (0.0033s latency).
Nmap scan report for Mitchells-iMac (192.168.50.97)
Host is up (0.013s latency).
Nmap scan report for EcoNet-7066554EFA90 (192.168.50.129)
Host is up (0.053s latency).
Nmap scan report for 192.168.50.139
Host is up (0.080s latency).
Nmap scan report for 192.168.50.151
Host is up (0.0048s latency).
Nmap scan report for amazon-6c56dc785 (192.168.50.158)
Host is up (0.0039s latency).
Nmap scan report for 192.168.50.180
Host is up (0.0037s latency).
Nmap scan report for HS-5CG9164DBS (192.168.50.198)
Host is up (0.053s latency).
```

Link local devices with nmap

Of interest, nmap took about 16 seconds because it basically pings all addresses, in my case 255.

arp-scan sends out an arp broadcast and listens for responses, which is much faster

Demo 1

```
→  ~ sudo arp-scan --interface=en0 --localnet
Password:
Interface: en0, type: EN10MB, MAC: f0:18:98:51:15:18, IPv4: 192.168.1.159
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-scan)
192.168.1.1     b8:f8:53:4d:f6:6f       Arcadyan Corporation
192.168.1.100   60:d2:48:89:e2:ba       ARRIS Group, Inc.
192.168.1.101   a8:97:cd:5e:bb:60       ARRIS Group, Inc.
192.168.1.102   a8:97:cd:5e:bd:24       ARRIS Group, Inc.
192.168.1.104   a8:97:cd:5e:bc:27       ARRIS Group, Inc.
192.168.1.162   4c:0b:be:53:0d:18       Microsoft
192.168.1.74    5e:f7:90:eb:1b:ae       (Unknown: locally administered)
192.168.1.215   20:4c:03:cb:55:b8       Aruba, a Hewlett Packard Enterprise Company
192.168.1.252   04:d9:f5:2d:8b:f0       ASUSTek COMPUTER INC.
192.168.1.244   28:39:5e:49:27:9c       Samsung Electronics Co.,Ltd
192.168.1.193   b2:ea:c7:71:5c:ee       (Unknown: locally administered)
```

This interface is known as 192.168.1.254 on the second router

All devices wired into here see the router as 192.168.50.1

Result of switching networks, notice 192.168.1.252 is an Asus AP, this is the same device that is 192.168.50.1 on the other network

Demo 1


```
→  ~ sudo arp-scan --interface=en0 --localnet
Password:
Interface: en0, type: EN10MB, MAC: f0:18:98:51:15:18, IPv4: 192.168.50.2
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-s
192.168.50.1    04:d9:f5:2d:8b:f0       ASUSTek COMPUTER INC.
192.168.50.7    38:2c:4a:cc:2b:18       ASUSTek COMPUTER INC.
192.168.50.44   20:ef:bd:5f:04:c8       (Unknown)
192.168.50.139  04:d9:f5:2b:5b:40       ASUSTek COMPUTER INC.
192.168.50.151  04:d4:c4:c7:27:b8       ASUSTek COMPUTER INC.
192.168.50.97   88:63:df:99:a4:35       Apple, Inc.
192.168.50.180  04:d4:c4:c5:82:48       ASUSTek COMPUTER INC.
192.168.50.107  dc:91:bf:a4:40:b4       (Unknown)
192.168.50.107  dc:91:bf:a4:40:b4       (Unknown) (DUP: 2)
192.168.50.158  14:0a:c5:73:9a:4a       (Unknown)
192.168.50.158  14:0a:c5:73:9a:4a       (Unknown) (DUP: 2)
192.168.50.210  dc:72:23:89:82:00       (Unknown)
192.168.50.231  66:93:cb:00:63:47       (Unknown: locally administered)
192.168.50.231  66:93:cb:00:63:47       (Unknown: locally administered) (D
192.168.50.203  fe:80:7d:73:16:f8       (Unknown: locally administered)
192.168.50.93   54:60:09:6f:7f:0c       Google, Inc.
192.168.50.93   54:60:09:6f:7f:0c       Google, Inc. (DUP: 2)
```

```
→  ~ sudo arp-scan --interface=en0 --localnet
Password:
Interface: en0, type: EN10MB, MAC: f0:18:98:51:15:18, IPv4: 192.168.1.159
Starting arp-scan 1.9.7 with 256 hosts (https://github.com/royhills/arp-sca
192.168.1.1     b8:f8:53:4d:f6:6f       Arcadyan Corporation
192.168.1.100   60:d2:48:89:e2:ba       ARRIS Group, Inc.
192.168.1.101   a8:97:cd:5e:bb:60       ARRIS Group, Inc.
192.168.1.102   a8:97:cd:5e:bd:24       ARRIS Group, Inc.
192.168.1.104   a8:97:cd:5e:bc:27       ARRIS Group, Inc.
192.168.1.162   4c:0b:be:53:0d:18       Microsoft
192.168.1.74    5e:f7:90:eb:1b:ae       (Unknown: locally administered)
192.168.1.215   20:4c:03:cb:55:b8       Aruba, a Hewlett Packard Enterprise
192.168.1.252   04:d9:f5:2d:8b:f0       ASUSTek COMPUTER INC.
192.168.1.244   28:39:5e:49:27:9c       Samsung Electronics Co.,Ltd
192.168.1.193   b2:ea:c7:71:5c:ee       (Unknown: locally administered)
```

Note: 2nd interface of Verizon Fios AP not shown, why do you think this is?

What we now should now know

- The basic concept of a modern network involves the small network where your device is located, and then a gateway to access everything else
- You can use command line tools to explore your local network
- With the IP protocol, every device has one or more interfaces, each interface has a unique MAC address, but can also have one or more IP addresses
- IP addresses can change over time, but MAC addresses will not (except in some very rare cases with user intervention)
- MAC addresses are used to discover devices on the local network, IP addresses
- The "wire" protocol must use MAC addresses to establish point-to-point communications, IP addresses are used for broader network communication