

# Lecture 02

# Network Fundamentals

CS397/497 – Wireless Protocols for IoT  
Branden Gena – Winter 2023

Some slides borrowed from: Peter Steenkiste (CMU),  
Christian Poellabauer (Notre Dame)

Materials in collaboration  
with Pat Pannuto (UCSD)

# Administrivia

- Piazza
  - Everyone should have access to it
  - If you don't, try going to the Piazza tab on the sidebar in Canvas
  - If that still doesn't work, this is the exception when you should email me
  
- Canvas
  - Most important information is on the Canvas homepage
  - I'm posting slides there too

# Today's Goals

- Introduce OSI layer model of communication
- Provide background on Internet layering
- Overview of concerns for the Physical and Data link layers
  - Speak the “lingo” of wireless communication
  - Present technology aspects that we will return to in specific protocols
- Describe Medium Access Control mechanisms

# Outline

- **OSI Layers**
- Internet Architecture (Upper Layers)
- Physical Layer
- Data Link Layer

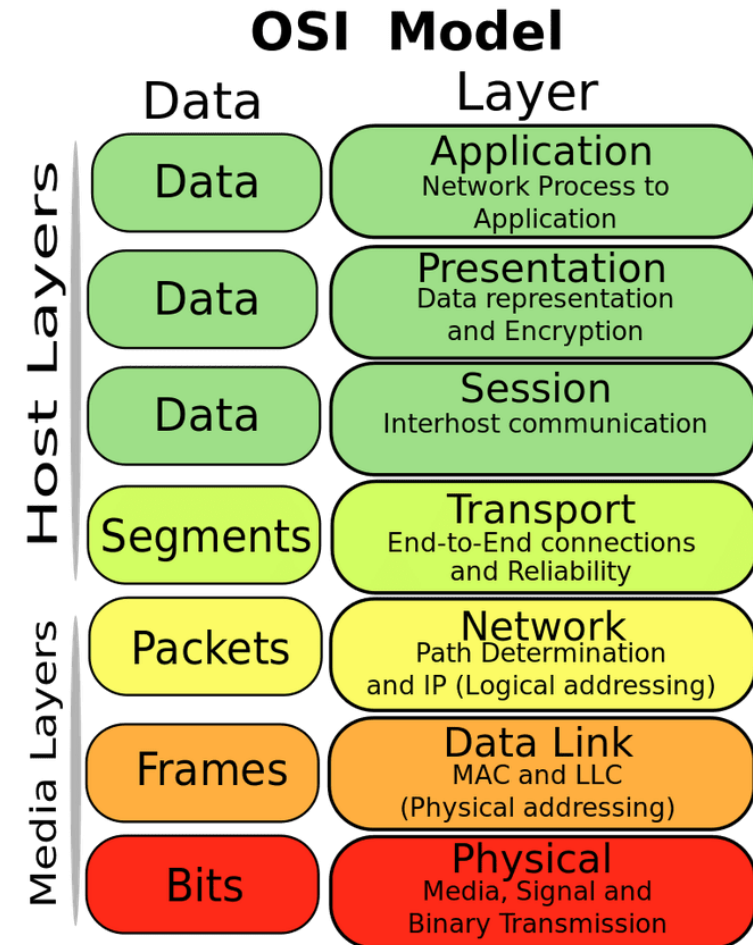
# Communication layers

- Application
- Presentation
- Session
- Transport
- Network
- Data Link
- Physical

**What goes on at each of these?**

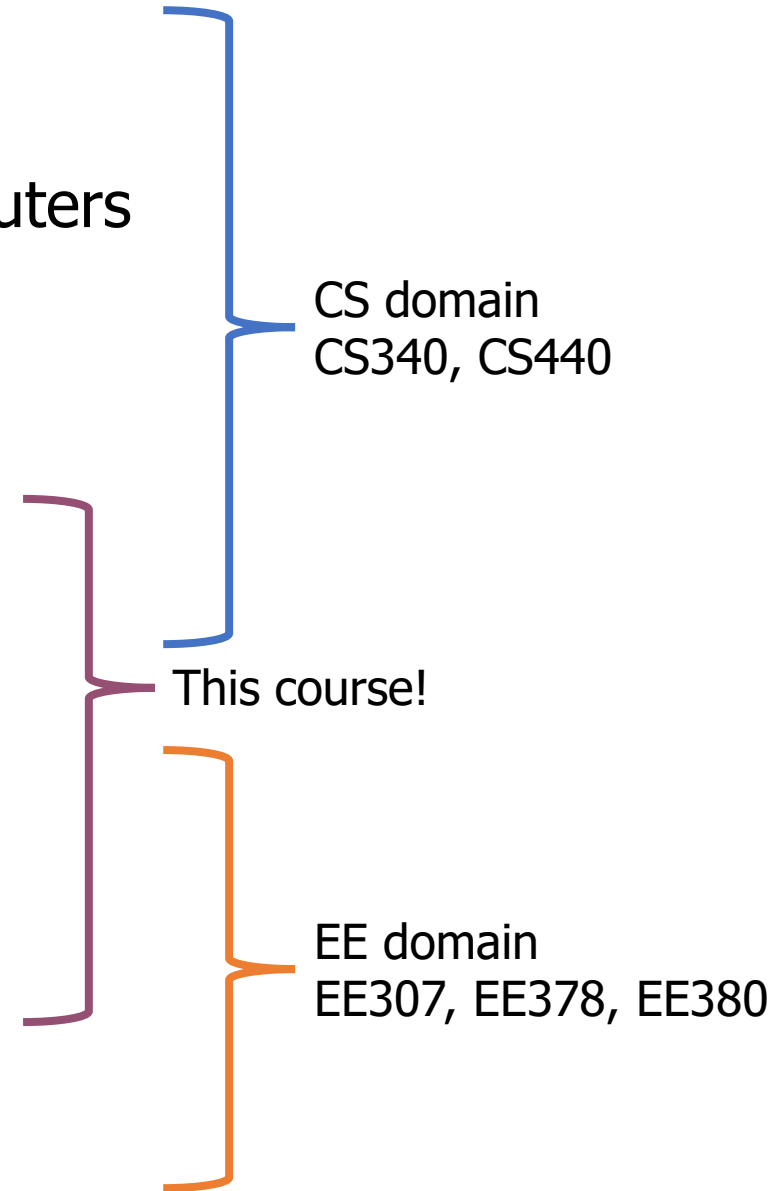
# OSI model of communication layers

- Transport
  - How to form connections between computers
  - TCP and UDP
- Network
  - How to send packets between networks
  - IP
- Data Link
  - How to send frames of data
  - Ethernet, WiFi
- Physical
  - How to send individual bits
  - Ethernet, WiFi



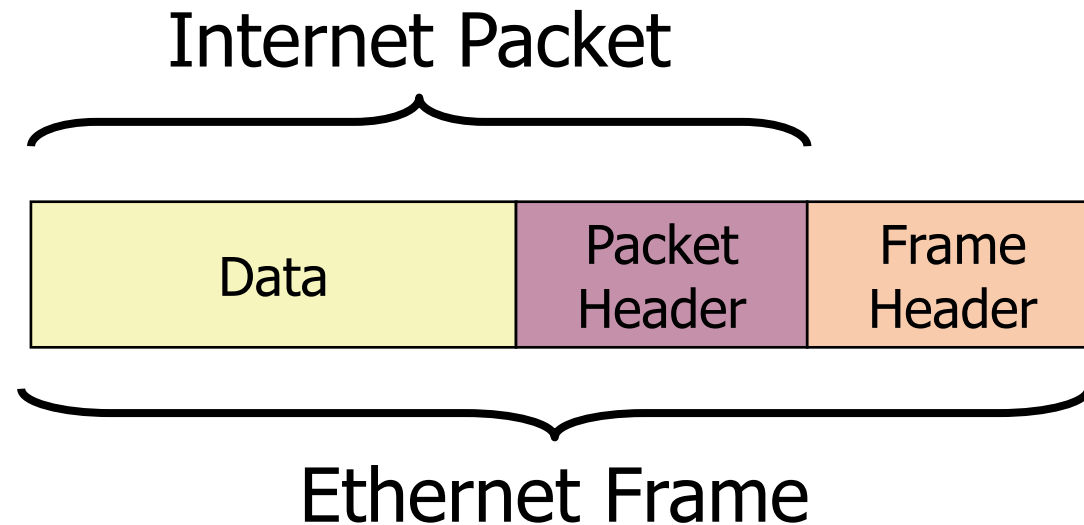
# OSI model of communication layers

- Transport
  - How to form connections between computers
  - TCP and UDP
- Network
  - How to send packets between networks
  - IP
- **Data Link**
  - **How to send frames of data**
  - **Ethernet, WiFi**
- Physical
  - How to send individual bits
  - Ethernet, WiFi



# Protocols are “layered”

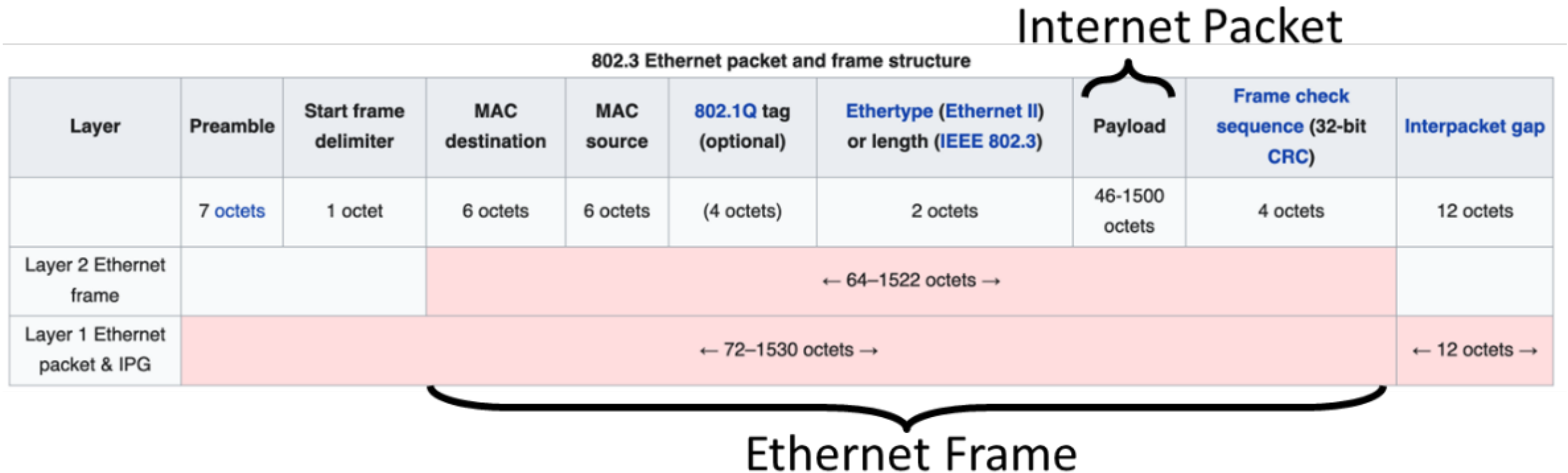
- Headers for each layer of communication wrap data
  - Data is wrapped with header for the network to make a packet
  - Packet is wrapped with header for the link to make a frame





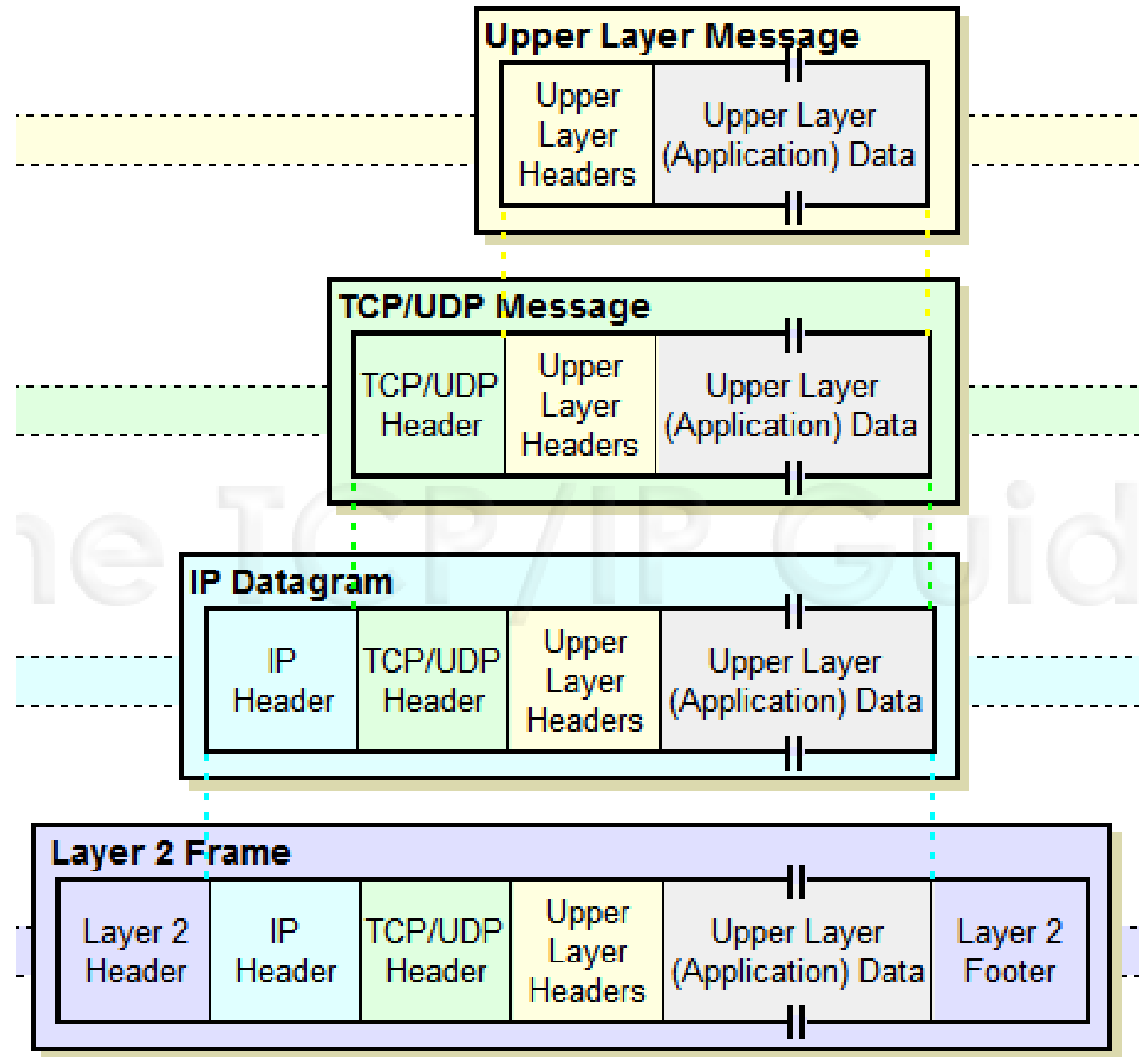
# Protocols are “layered”

- Headers for each layer of communication wrap data
  - Data is wrapped with header for network to make a packet
  - Packet is wrapped with a header for the link to make a frame

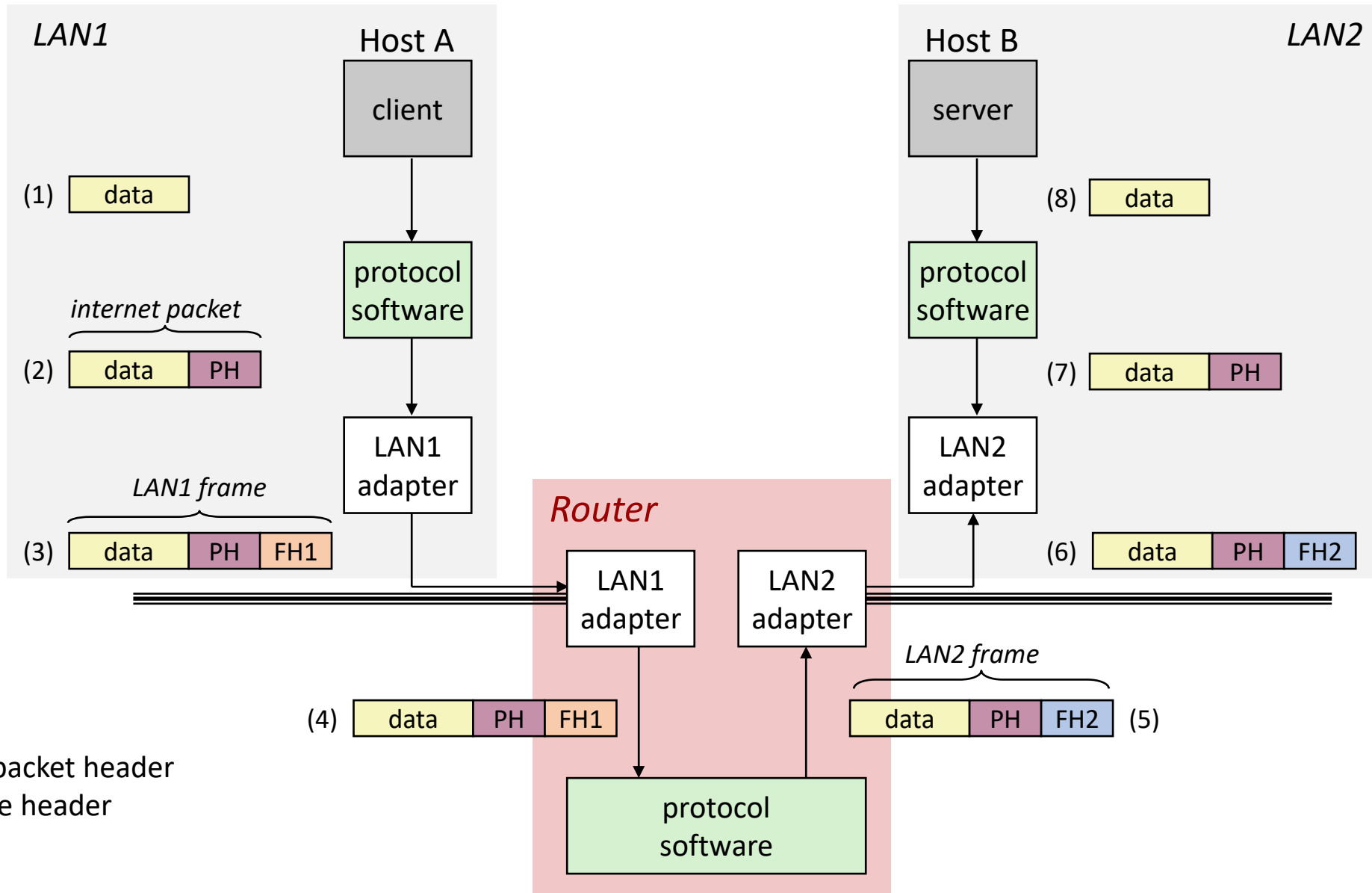


# Packet encapsulation

- Upper-layer packet is the payload for the lower-layer packet

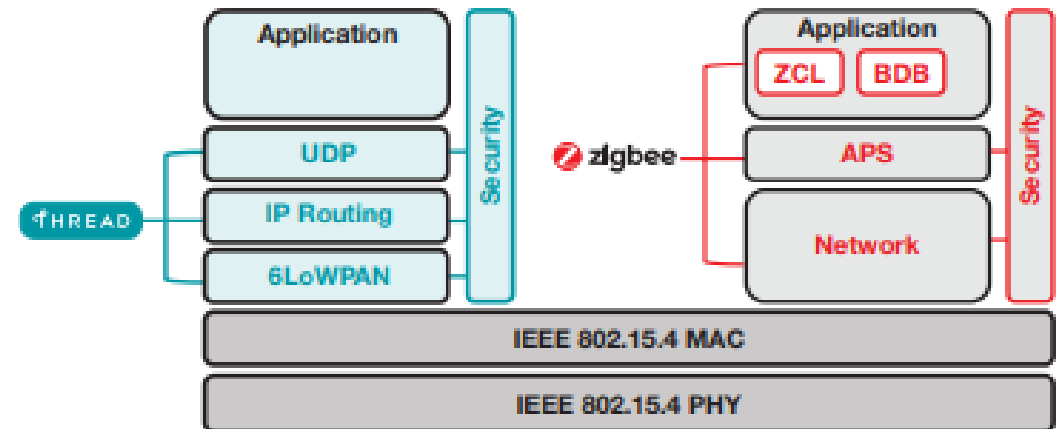
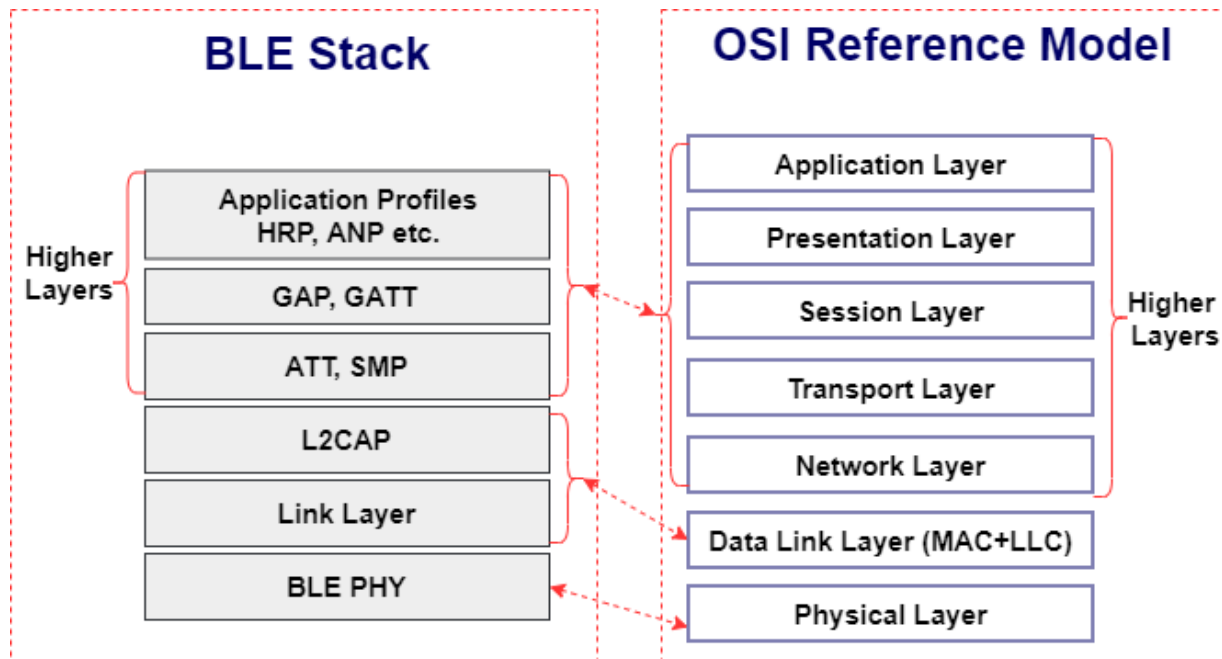


# Transmitting data between networks

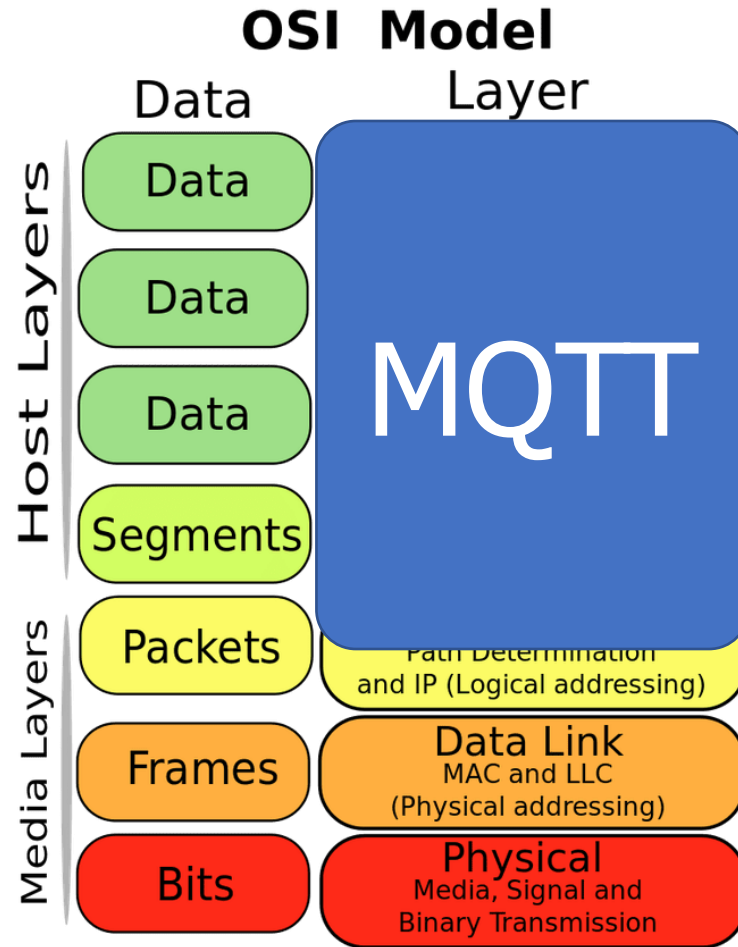


# Model does not equal reality

- Wireless protocols don't always split between layers cleanly
  - Usually explain parts of physical, data link, and possibly upper layers
- Model still helps conceptualize stack-up though
  - Layering of some type still occurs



# Layering for IoT (joke) (kind of)



MQTT is a  
publish/subscribe  
message broker

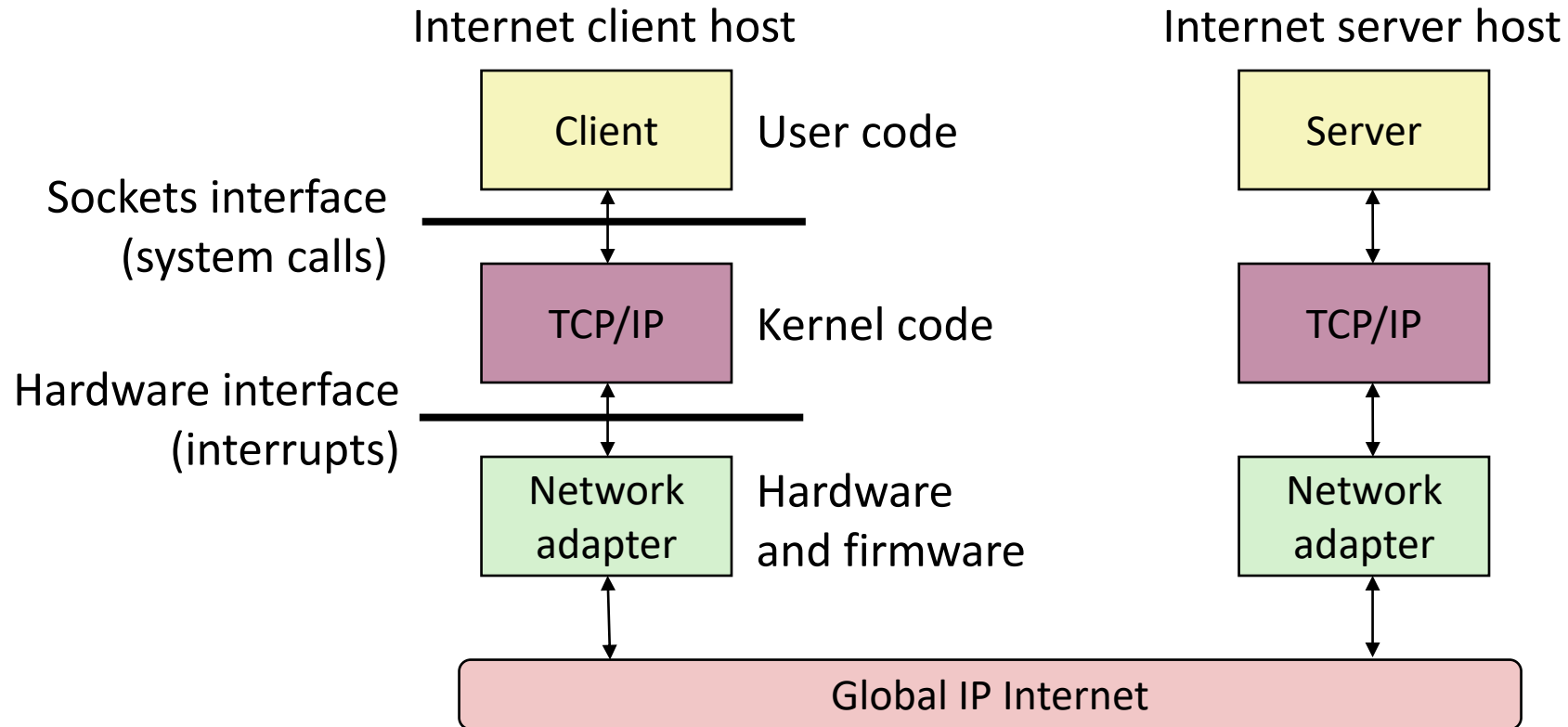
# Outline

- OSI Layers
- **Internet Architecture** (Upper Layers)
- Physical Layer
- Data Link Layer

# The global Internet

- Most famous example of an internet (uppercase to distinguish)
- Based on the TCP/IP protocol family
  - **IP** (Internet Protocol)
    - Provides a *naming scheme* and unreliable *delivery of packets* from **host-to-host**
  - **UDP** (Unreliable Datagram Protocol)
    - Uses IP to provide *unreliable data delivery* from **process-to-process**
  - **TCP** (Transmission Control Protocol)
    - Uses IP to provide *reliable data delivery* from **process-to-process**
- Accessed via a mix of Unix file I/O and the **sockets** interface

# Hardware and software organization of an Internet application





# A programmer's view of the internet

1. Hosts are mapped to a set of 32-bit **IP addresses**
  - 129.105.7.30
2. The set of IP addresses is mapped to a set of identifiers called Internet **domain names**
  - 129.105.7.30 is mapped to moore.wot.eecs.northwestern.edu
3. A process on one Internet host can communicate with a process on another Internet host over a **connection**

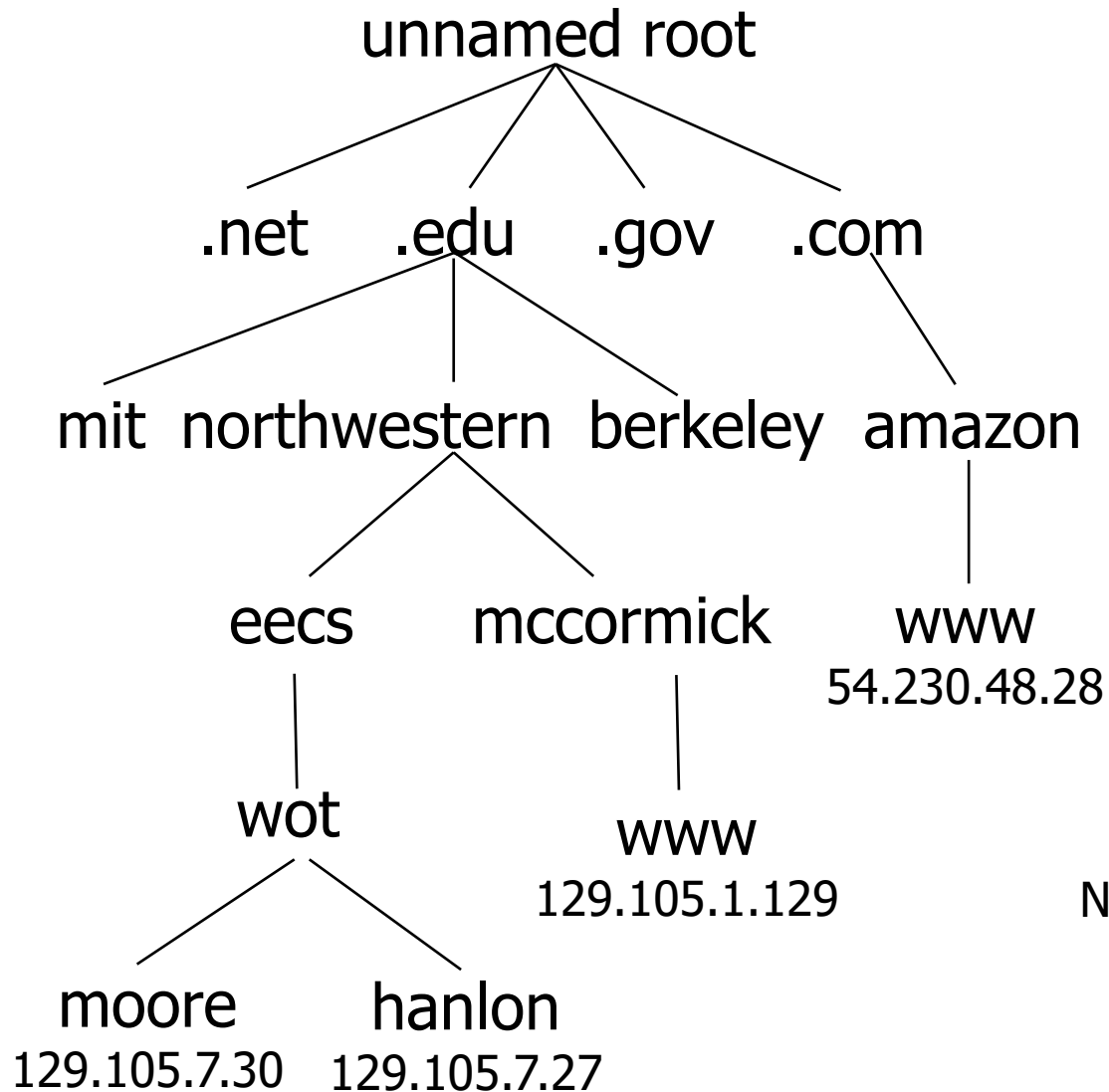
# 1. IP addresses

- 32-bit IP addresses are stored in an **IP address struct**
  - IP addresses are always stored in memory in *network byte order* (big-endian)
    - Remember: most computers use little-endian (🤪)
  - True in general for any integer transferred in a packet header from one machine to another
    - E.g., the port number used to identify an Internet connection

```
/* Internet address structure */
struct in_addr {
    uint32_t    s_addr; /* network byte order (big-endian) */
};
```

- By convention, each byte in a 32-bit IP address is represented by its decimal value and separated by a period
  - IP address:  $0x8169071E = 129.105.7.30$

## 2. Internet domain names



Top-level domain names

Second-level domain names

Third-level domain names  
and onwards...

Note: Northwestern owns 129.105.x.x

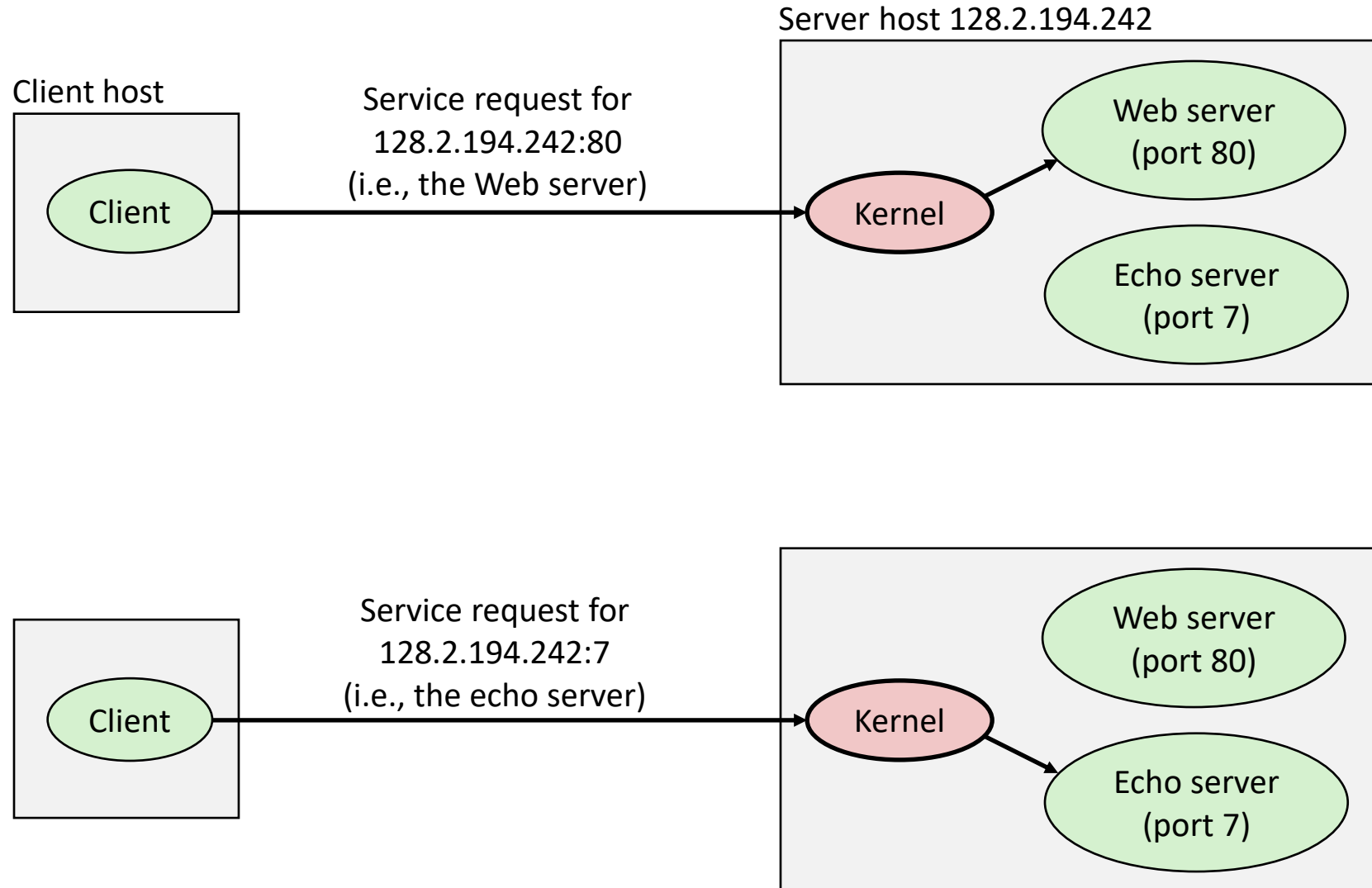
# Domain Naming System (DNS)

- The Internet maintains a mapping between IP addresses and domain names in a huge worldwide distributed database called **DNS**
- Conceptually, programmers can view the DNS database as a collection of millions of **host entries**
  - Each host entry defines the mapping between a set of domain names and IP addresses
- A special name: **localhost**
  - Refers back to the computer being used (IP address 127.0.0.1)

### 3. Internet connections

- A socket is an endpoint of a connection
  - Socket address is an **IPAddress:port** pair
    - IP address identifies the computer
    - Port identifies the process on the computer
- Clients and servers communicate by sending streams of bytes over **connections**. Most connections are:
  - Point-to-point: connects a pair of processes.
  - Full-duplex: data can flow in both directions at the same time,
  - [TCP adds] Reliable: stream of bytes sent by the source is eventually received by the destination in the same order it was sent.

# Ports are used to identify services to the kernel



# How does the Internet handle routing packets?

- IP layer
  - Describes the overall goal
    - Packets from my computer <---> Google
- Link layer (Ethernet)
  - Describes individual links
    - Packets from my computer <---> my router
- **Routing**
  - Using link-layer building blocks to get packets from one IP to another

# Addressing

- How to solve the routing problem?
  - I need to know how to get data from me to you
- How does the post office work?
  - I know where you live (your address)
    - Zip Code
    - City
    - Street
    - House Number
    - Name



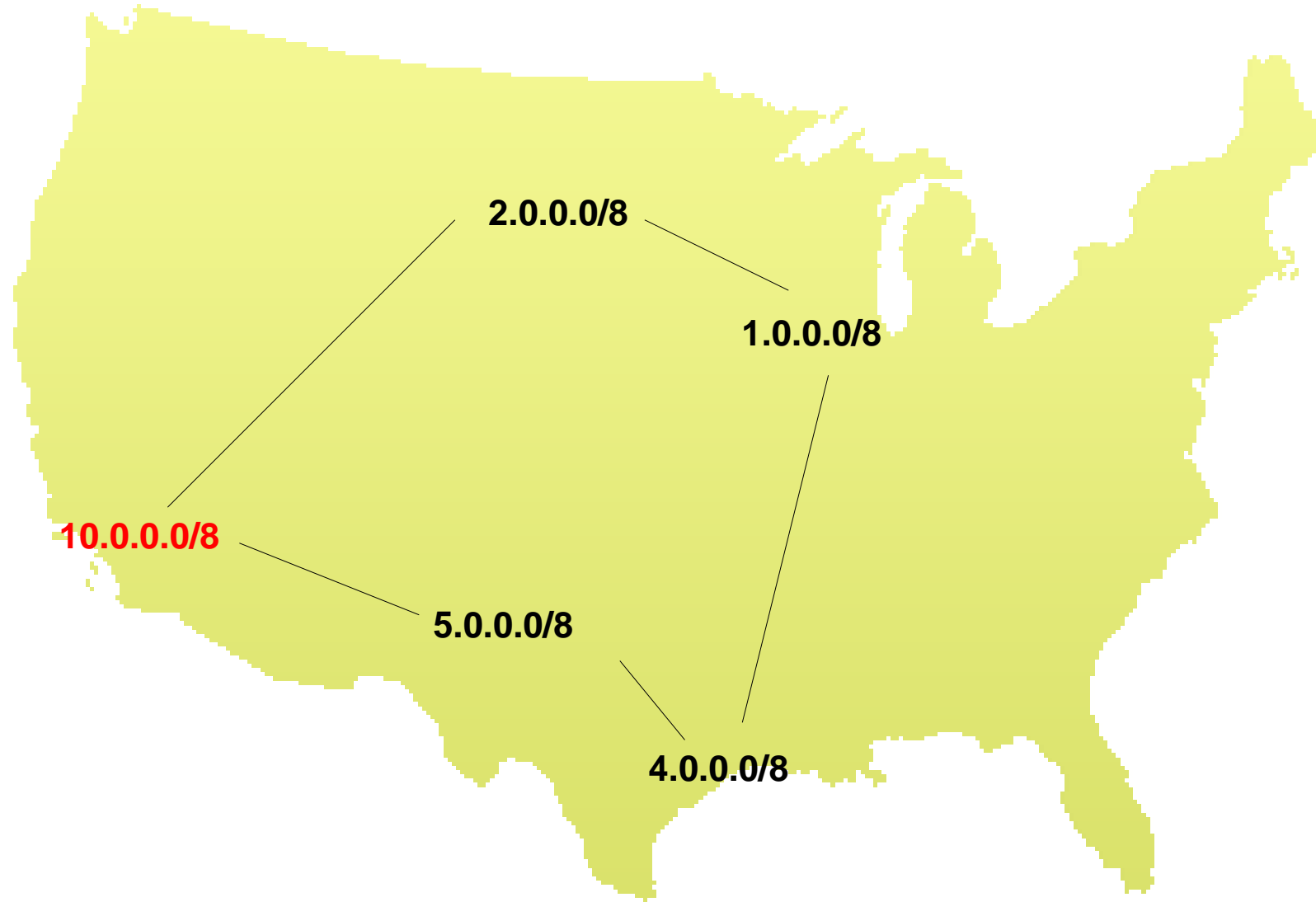
# The problem with addressing

- Your computer moves all the time
  - Home, school, Starbucks...

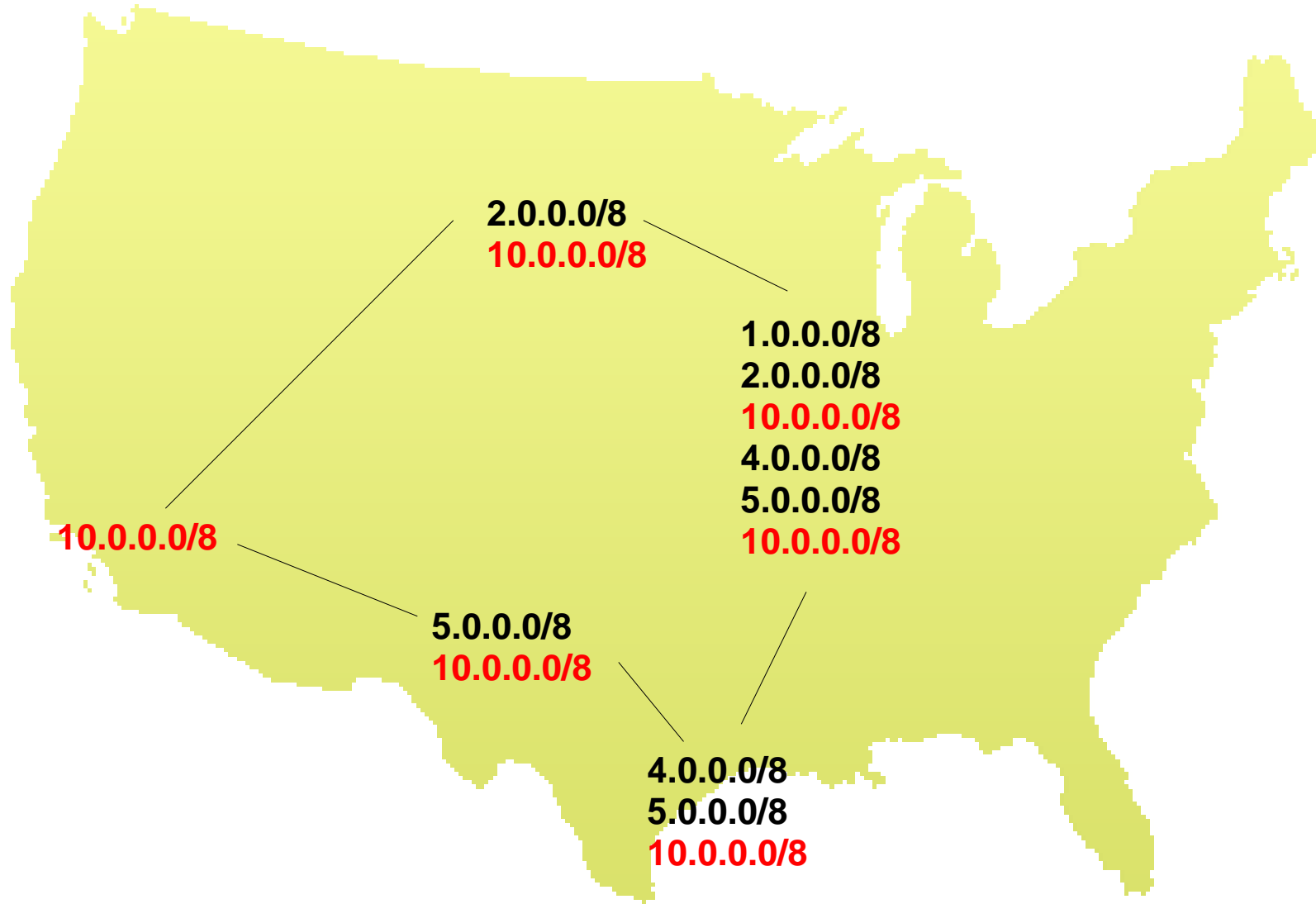
# Assigning and finding IP address ranges

- In general, network operators don't change that often
- Solution:
  - Tie IP addresses to network operators
  - Assign computers IPs as they join networks
- Key Point:
  - Networks "own" a block of IP address space
  - "The Internet" is a network of networks

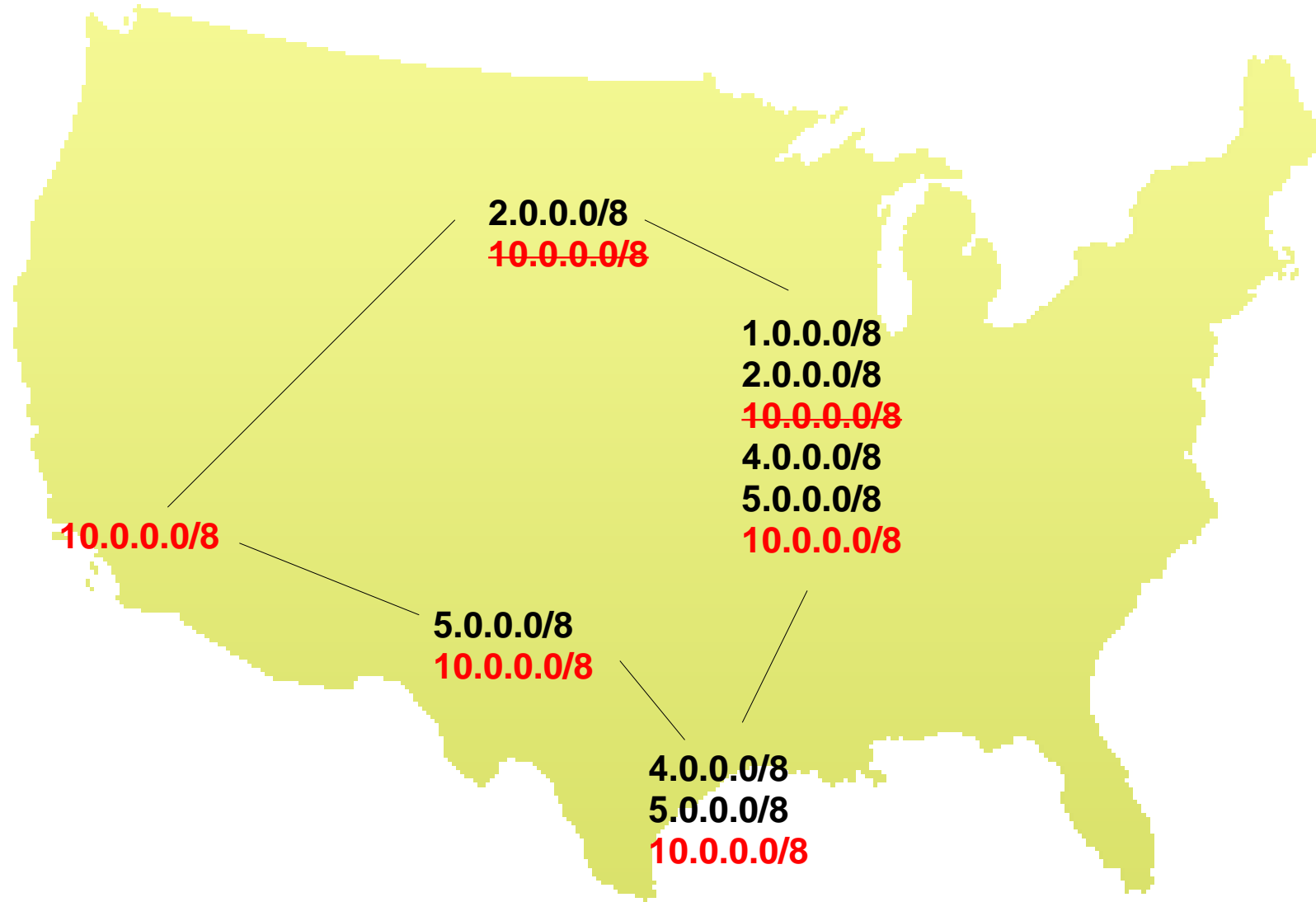
# Routing



# Routing



# Routing – “Adaptive”



# Identifying your computer?

- Every network card has its own MAC address
  - IPs are (somewhat) dynamic, "owned" by local networks
  - MACs are hardware and static, "owned" by specific computers
    - Manufacturers own blocks of MACs, "spend" them each time they make a device
- "Connecting" to a network
  - Your computer leases an IP from the local network
  - Only the local router knows your MAC, everyone else sees your IP
    - Note: this overview ignores NATs, which are commonplace today

# So how does the Internet of Things fit into the Internet?

- “IP is the Narrow Waist of the Internet”
  - [IP is Dead, Long Live IP for Wireless Sensor Networks](#)
- A recurring theme in this class:
  - How does this actually attach to the Internet
    - Physically  
[hello Hue Hub, Wyze Hub, August Hub, ...]
    - Logically  
[are BLE devices *really* part of the IoT?]

**IP is Dead, Long Live IP for Wireless Sensor Networks**

Jonathan W. Hui  
University of California at Berkeley  
Arch Rock Corporation  
jhui@cs.berkeley.edu

David E. Culler  
University of California at Berkeley  
Arch Rock Corporation  
culler@cs.berkeley.edu

**ABSTRACT**

A decade ago, wireless sensor network research took off more researchers in the field abandoned the use of IP as inadequate and incommensurate to the needs of sensor network networking. Since then the field has pursued, mostly in vain, sensor, and IP has evolved. In this paper, we present the design of a concrete IPv6-based network architecture for wireless sensor networks. We validate the architecture with a production-quality implementation that incorporates many techniques pioneered in the sensor network community, including relay-enabled link protocols, router compression, hop-by-hop forwarding, and efficient routing with efficient link estimation. In addition to providing interoperability with existing IP devices, this implementation was able to achieve an average duty cycle of 0.55%, average per-hop latency of 60ms, and a data reception rate of 55.6% over a period of 4 weeks in a real-world home monitoring application, while exhibiting general applicability to other protocols. Our results indicate that existing systems that do not adhere to any particular standard or architecture. In light of this demonstration of full IPv6 capability, we review the central arguments that led the field away from IP. We believe that the presence of an architecture, specifically an IPv6-based one, provides a strong foundation for various sensor networks going forward.

**Categories and Subject Descriptors**

C.2.1 [Computer-Communications Networks]: Network Architecture and Design—Wireless communication; C.2.2 [Computer-Communications Networks]: Network Protocols; C.2.3 [Computer-Communications Networks]: Interconnecting—standards

**General Terms**

Design, Measurement, Performance, Reliability, Security, Standards

**Keywords**

network architecture; Internet; Interconnecting; wireless; sensor networks; IPv6; WSN; WSN; network management

**1. INTRODUCTION**

As wireless sensor networks (WSNs) research has matured, many researchers in the field agreed that the “narrow waist” of the Internet is the narrow waist of the sensor network. In particular, the use of IP as inadequate and incommensurate to the needs of sensor network networking. In 1996, RFC 2442 argued IPv6 [11]. The large address space was only proposed for a large number of devices, it eliminated many of the address management issues. This is what facilitated the use of a unique local unicast IPv6 address (FE80::) for all IPv6 nodes using small EUI-64 short addresses [2]. The IPv6 prefix provided the address of the sensor. The sensor uses the sensor network (SN) discovery and network management mechanisms used with IPv6. It was established that the IPv6 framework was closely tied to the core of the network and needed devices in a changing environment. Thus, the emergence of options provided for compact headers in the common case,

known learned from Internet and mobile networks design and deployment (e.g., routing, security, sensor network applications, ...). Sensor networks have different security requirements to support monitoring the overall structure of applications and services [13]. The Internet architecture was designed for several reasons including the following [13]:

- The sensor network community may want to design the layered architecture.
- The sheer numbers of these devices, and their unattended deployment, will probably require a a broadcast communication for the configuration needed to connect and operate network devices.
- Local and global end-to-end network connectivity will be required to achieve robustness and scalability.
- Unlike traditional networks, sensor networks are not used as identity (e.g., an address). Naming will be decentralized.
- Traditional networks are designed to accommodate a wide range of applications. WSNs will be tailored to the sensing task at hand.

In addition, it was argued that protocols for all types of WSNs, the traditional Internet and layers of system abstraction should not be assumed [26]. By providing a framework for defining structures and offering the community to propose, new network architectures were expected to emerge [11]. Indeed, by introducing the Active Message Dispatch and the network management interface, a conventional header-based, Transport (TCP) [14] had away from IP. The new areas of protocols developed by the community operate at the link layer, rather than the network layer. The central interface to a base station (or the use of application level gateway) at the core of the WSN, as WSNs were considered by a number of researchers [20, 25], rather than an IP network similar to Internet or WiFi.

Since these beginnings, the field has matured substantially, a huge collection of protocols have been invented and evaluated, and we have gained experience in how WSNs are used in practice. Over the same period, the Internet has evolved as well. In 1996, RFC 2442 argued IPv6 [11]. The large address space was only proposed for a large number of devices, it eliminated many of the address management issues. This is what facilitated the use of a unique local unicast IPv6 address (FE80::) for all IPv6 nodes using small EUI-64 short addresses [2]. The IPv6 prefix provided the address of the sensor. The sensor uses the sensor network (SN) discovery and network management mechanisms used with IPv6. It was established that the IPv6 framework was closely tied to the core of the network and needed devices in a changing environment. Thus, the emergence of options provided for compact headers in the common case,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made for commercial or promotional purposes, the copyright notice, this notice, and the article title are included in the copy. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.  
Proceedings of the ACM, 2010, Berkeley, CA, USA.  
Copyright 2010 ACM 978-1-60558-989-9/10/0001...\$5.00

# Break + Thinking

- What are the steps for viewing a website?



# Break + Thinking

- What are the steps for viewing a website?
  1. You enter a domain name for the website
  2. Computer looks up domain name to get IP Address
  3. Computer sends request to IP\_address:80
  4. Computer gets back data, which it renders into a website

# ALL the layers

- A 'famous' interview question
  - "What happens when you type google.com into your browser's address bar and press enter?"
  - <https://github.com/alex/what-happens-when> (11 pages!)
    - Keyboard events
    - Parsing URL
    - DNS lookup
    - Opening socket
    - HTTP protocol
    - HTML parsing
    - GPU rendering

# Outline

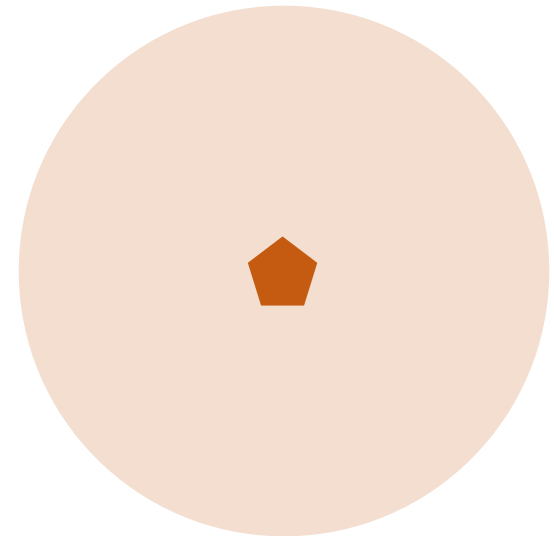
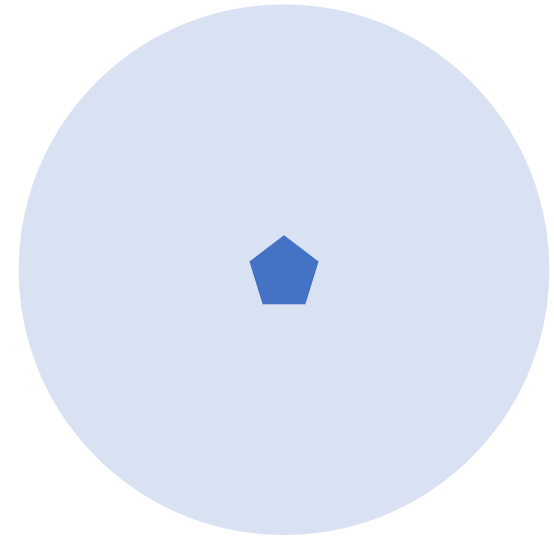
- OSI Layers
- Internet Architecture (Upper Layers)
- **Physical Layer**
- Data Link Layer

# Physical Layer

- How bits are transmitted
  - Wireless makes this entirely different from wired cases
- Important considerations
  - Signal strength
  - Modulation
  - Frequency

# Model of RF communication

- Energy that radiates spherically from an antenna
- Attenuation with distance
  - Density of energy reduces over time, distance
  - Signal strength is reduced, errors go up
- Two key features
  - Error rates depend on distance
  - Spatial reuse of frequencies



# Signal qualities

## 1. Signal strength

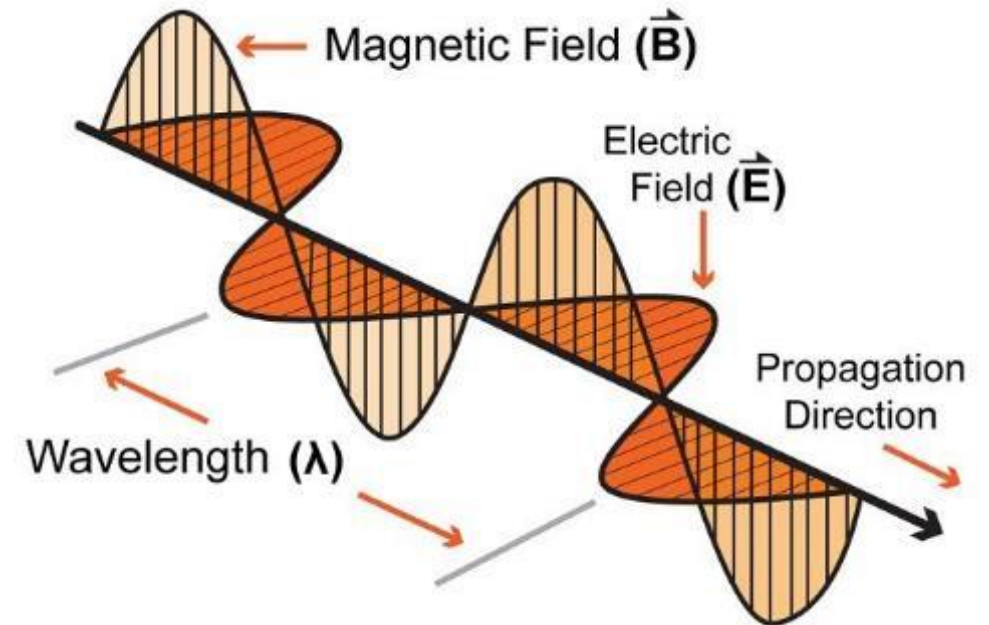
- The amount of energy transmitted/received

## 2. Signal frequency and bandwidth

- Which "channel" the signal is sent on

## 3. Signal modulation

- How data is encoded in the signal



# Signal qualities

## 1. Signal strength

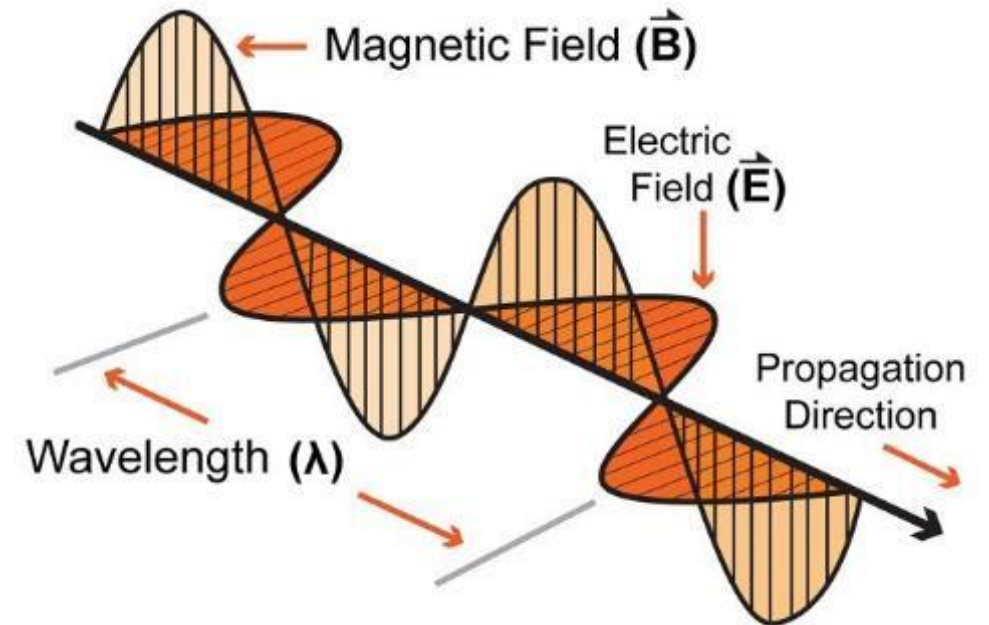
- The amount of energy transmitted/received

## 2. Signal frequency and bandwidth

- Which "channel" the signal is sent on

## 3. Signal modulation

- How data is encoded in the signal



# Signal strength is measured in decibels

- Power is measured in Watts or dBw or dBm
  - $Power_{dBw} = 10 * \log_{10}(Power_{Watts})$
  - $Power_{dBm} = 10 * \log_{10}(Power_{milliwatts})$
- dBm is most relevant to the IoT domain
  - 0 dBm equals 1 mW transmit power
  - Example
    - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
    - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)
- Rule of thumb: +3 dB is double the power



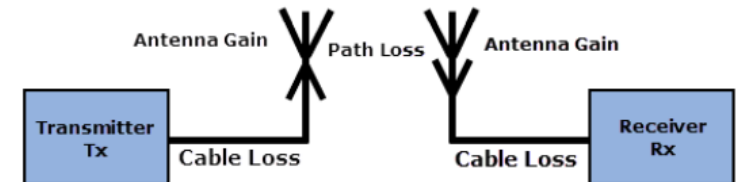
# Signal strength varies significantly across technologies

- Bluetooth Low Energy (local area)
  - nRF52840 transmit power: 8 dBm (6.31 mW)
  - nRF52840 receive sensitivity: -95 dBm (316.2 fW)
- LoRa (wide area)
  - SX127X LoRa transmit power: 20 dBm (100 mW)
  - SX127X LoRa receive sensitivity: -148 dBm (1.6 attoWatt)

# Propagation degrades RF signals

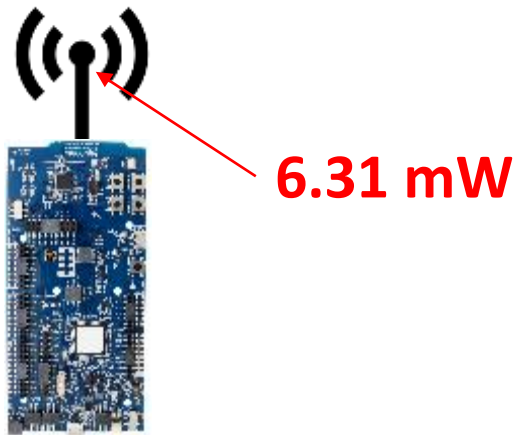
- Attenuation in free space
  - Signals get weaker as they travel over long distances
  - Signal spreads out → Free Space Path Loss (FSPL)

$$FSPL = 20 \log_{10}(d) + 20 \log_{10}(f) + 20 \log_{10}\left(\frac{4\pi}{c}\right) - G_t - G_r$$



# Some intuitions for signal propagation, power, gain, etc

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)



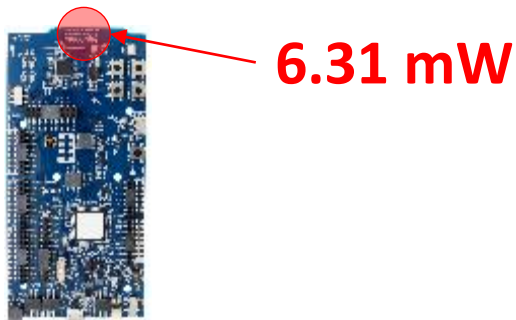
Wait,  is not an antenna

- Indeed, this little strip of metal is the actual antenna
  - Receiver only recovers the part of the signal that hits its antenna (“aperture”)



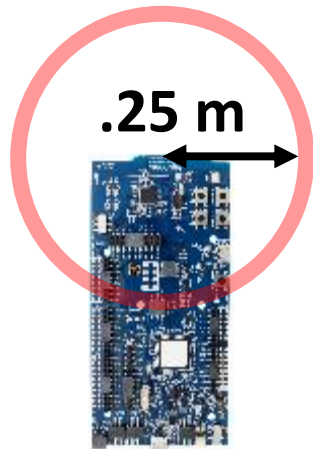
# Some Intuitions for Signal Propagation, Power, Gain, etc.

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)



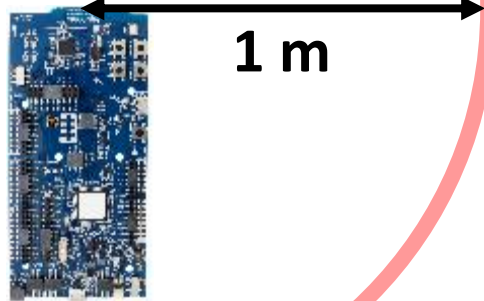
# Some Intuitions for Signal Propagation, Power, Gain, etc.

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)



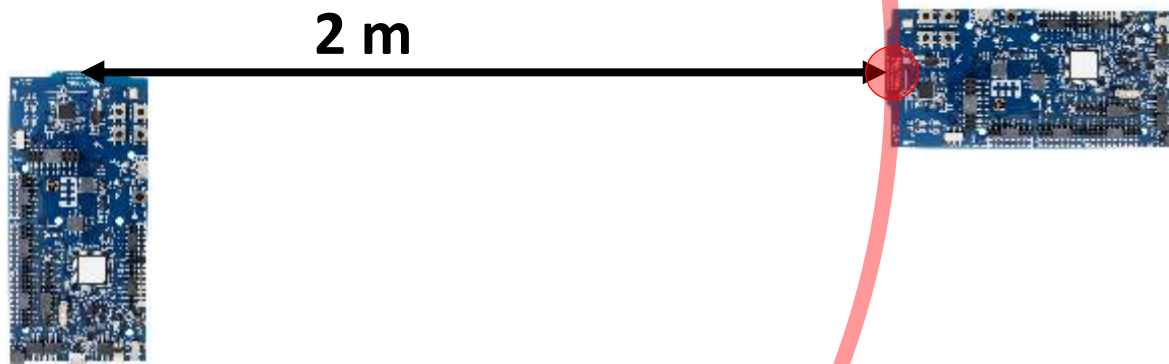
# Some Intuitions for Signal Propagation, Power, Gain, etc.

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)



# Some Intuitions for Signal Propagation, Power, Gain, etc.

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)





# Some Intuitions for Signal Propagation, Power, Gain, etc.

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)



# Okay.. So what's the limit?

- We will use the nrf52840 in lab:
  - Max BLE transmit power for nRF52840: 8 dBm (6.31 mW)
  - Min BLE receive sensitivity for nRF52840: -95 dBm (316.2 fW)
- $8 \text{ dBm} - -95 \text{ dBm} = 103 \text{ dB}$  link margin
- For FSPL alone for a 2.4 GHz signal, 103 dB is 1,400 m!

Bluetooth does not go 1.4 km...

# Propagation is *one thing* that degrades RF signals

- Attenuation in free space
  - Signals get weaker as they travel over long distances
  - Signal spreads out -> free space path loss
- Important: distance is NOT the only signal strength loss
  - Free space path loss calculation will not give you accurate range for a signal
- Obstacles can weaken signal through absorption or reflection
  - Precise quantitative details are in the EE domain
  - We'll use examples to develop qualitative instincts in this class

# Many factors affect the ability to actually receive data

- Here's some examples, from DW1000 [ultra wideband transceiver]

**3.4 Receiver Sensitivity Characteristics**

*T<sub>amb</sub> = 25 °C, all supplies centered on typical values. 20 byte payload*

**Table 6: Typical Receiver Sensitivity Characteristics**

Packet Error Rate	Data Rate	Typical Receiver Sensitivity	Units	Condition/Note		
1%	110 kbps	-106	dBm/500 MHz	Preamble 2048	Carrier frequency offset ±1 ppm. Requires use of the "tight" Rx operating parameter set – see [2]	All measurements performed on Channel 5, PRF 16 MHz. Channel 2 is approximately 1 dB less sensitive
10%	110 kbps	-107	dBm/500 MHz	Preamble 2048		
1%	110 kbps	-102	dBm/500 MHz	Preamble 2048	Carrier frequency offset ±10 ppm	
	850 kbps	-101	dBm/500 MHz	Preamble 1024		
	6.8 Mbps	-93 (*-97)	dBm/500 MHz	Preamble 256		
10%	110 kbps	-106	dBm/500 MHz	Preamble 2048		
	850 kbps	-102	dBm/500 MHz	Preamble 1024		
	6.8 Mbps	-94 (*-98)	dBm/500 MHz	Preamble 256		

# ITU model for Indoor Attenuation

$$L = 20 \log_{10} f + N \log_{10} d + P_f(n) - 28$$

where,

$L$  = the total path loss. Unit: decibel (dB).

$f$  = Frequency of transmission. Unit: megahertz(MHz).

$d$  = Distance. Unit: meter (m).

$N$  = The distance power loss coefficient.

$n$  = Number of floors between the transmitter and receiver.

$P_f(n)$  = the floor loss penetration factor.

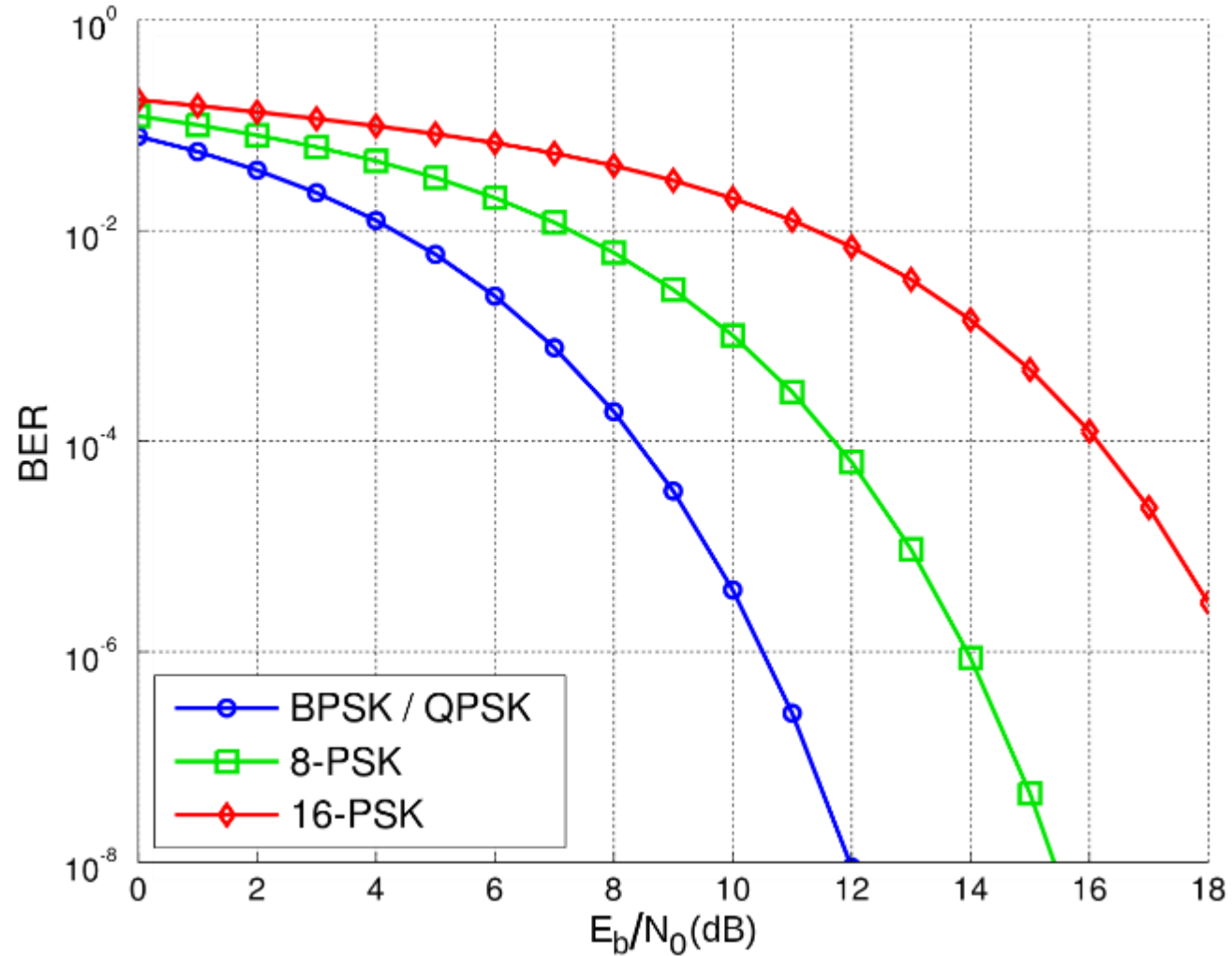
- Models like this are *less bad* than Free-Space Path Loss
  - [https://en.wikipedia.org/wiki/ITU\\_model\\_for\\_indoor\\_attenuation](https://en.wikipedia.org/wiki/ITU_model_for_indoor_attenuation)

# Lower received energy increases error rates

More Errors



Less Errors



BER:  
Bit Error Rate

Odds that a  
transmitted bit  
will be  
received  
incorrectly

Less Energy  
Received



More Energy  
Received

# Signal qualities

## 1. Signal strength

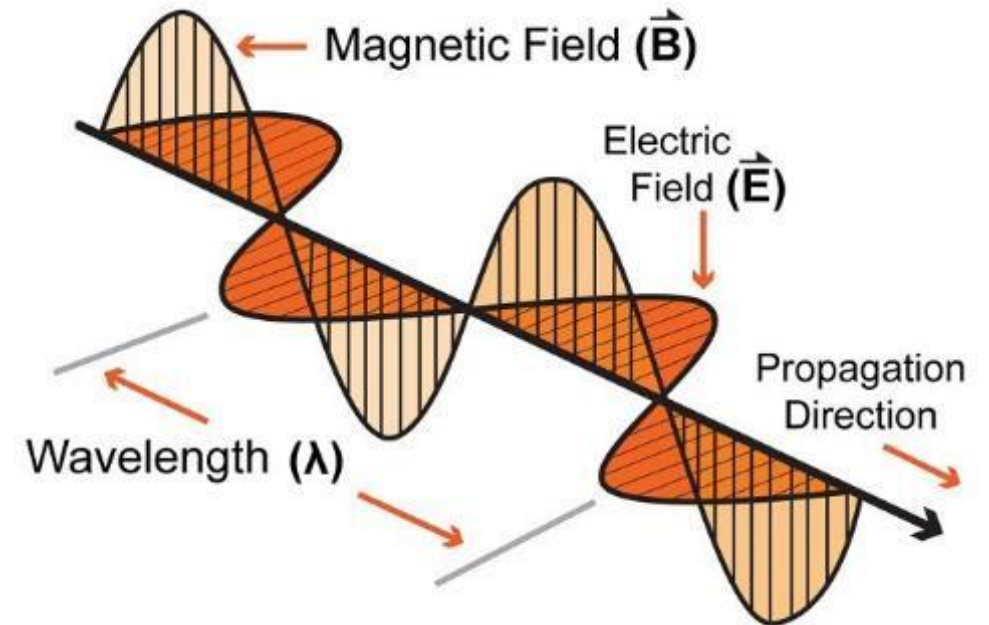
- The amount of energy transmitted/received

## 2. Signal frequency and bandwidth

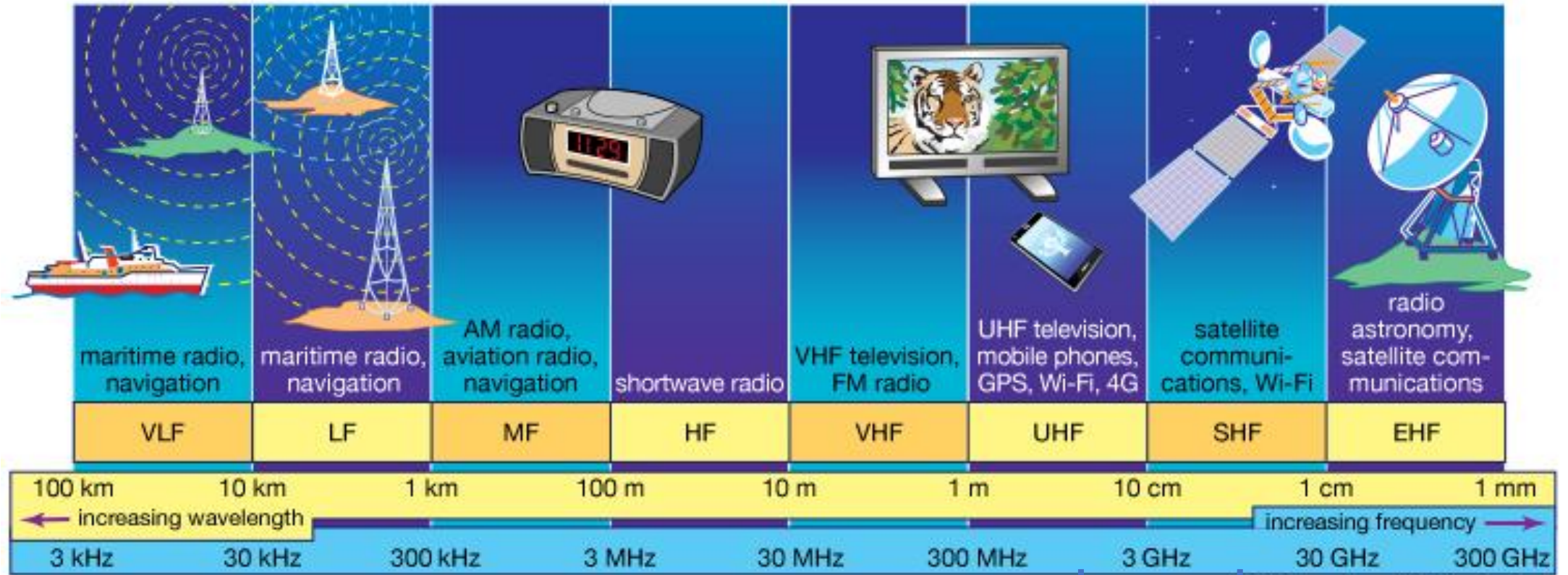
- Which "channel" the signal is sent on

## 3. Signal modulation

- How data is encoded in the signal



# RF communication frequencies



© 2013 Encyclopædia Britannica, Inc.

**IoT focus**



# Wireless spectrum is allocated to specific uses

## UNITED STATES FREQUENCY ALLOCATIONS

### THE RADIO SPECTRUM

**RADIO SERVICES COLOR LEGEND**

AERONAUTICAL MOBILE	HYPER SATELLITE	RADIO ASTRONOMY
AERONAUTICAL MOBILE SATELLITE	LAND MOBILE	AERONAUTICAL MOBILE SATELLITE
AERONAUTICAL MOBILE/TERRESTRIAL	LAND MOBILE SATELLITE	RADIOLOCATION
JOCHEVSE	MARITIME MOBILE	RADIOLOCATION SATELLITE
AERONAUTICAL SATELLITE	MARITIME MOBILE SATELLITE	RADIOLOCATION
BROADCASTING	MARITIME BROADCASTING	RADIOLOCATION SATELLITE
BROADCASTING SATELLITE	METEOROLOGICAL	SPACE RESEARCH
SPACE EXPLORATION SATELLITE	METEOROLOGICAL SATELLITE	SPACE RESEARCH
FIXED	MOBILE	STANDARD FREQUENCY AND TIME SIGNAL
FIXED SATELLITE	MOBILE SATELLITE	STANDARD FREQUENCY AND TIME SIGNAL SATELLITE

**ACTIVITY CODE**

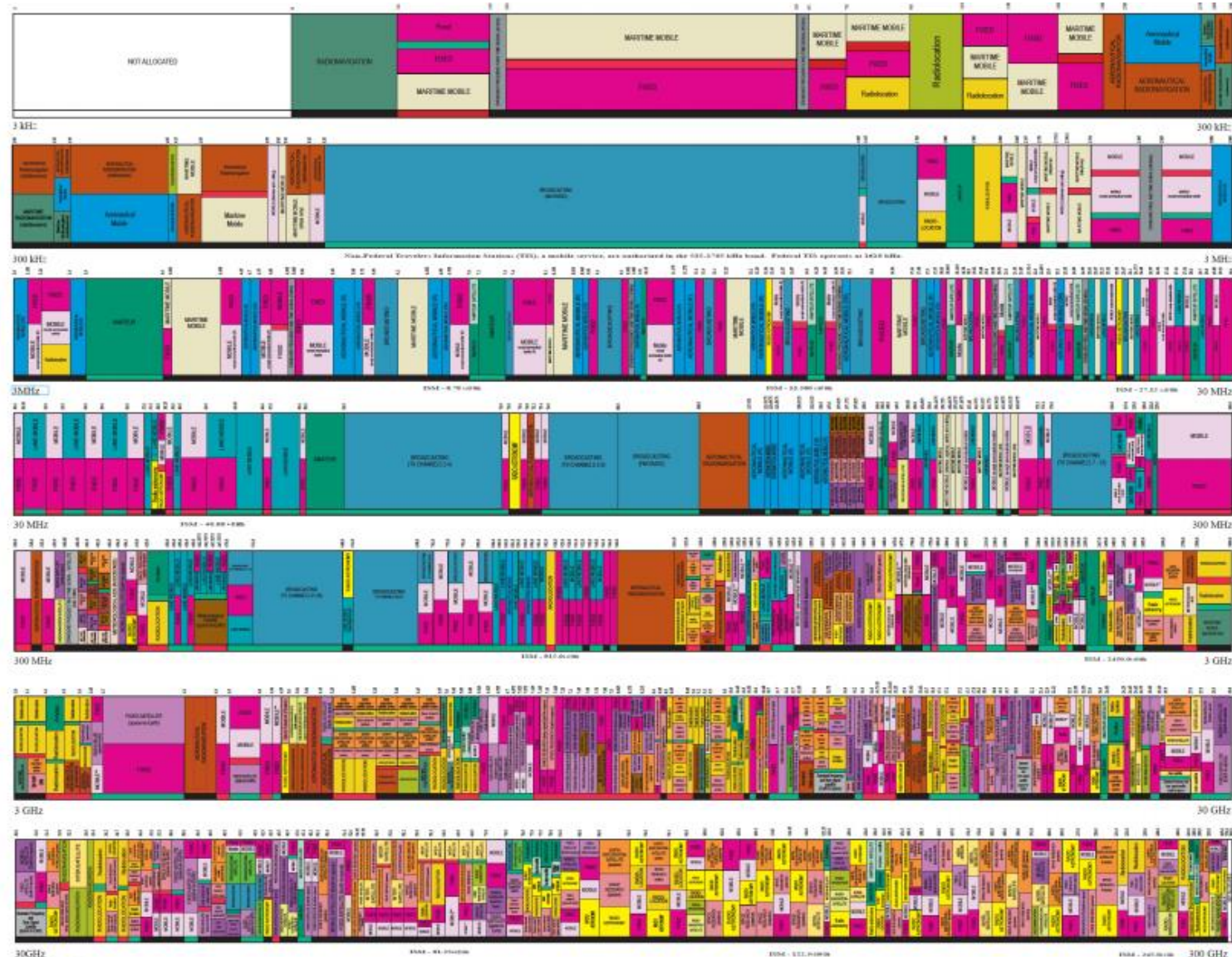
GOVERNMENT ESTABLISHMENT	GOVERNMENT-ASSIGNED SERVICE
NON-GOVERNMENT ESTABLISHMENT	

**ALLOCATION USAGE DESIGNATION**

<b>OFFICE</b>	<b>EXAMPLE</b>	<b>DESCRIPTION</b>
Primary	3120	Land Mobile
Secondary	3120	Land Mobile

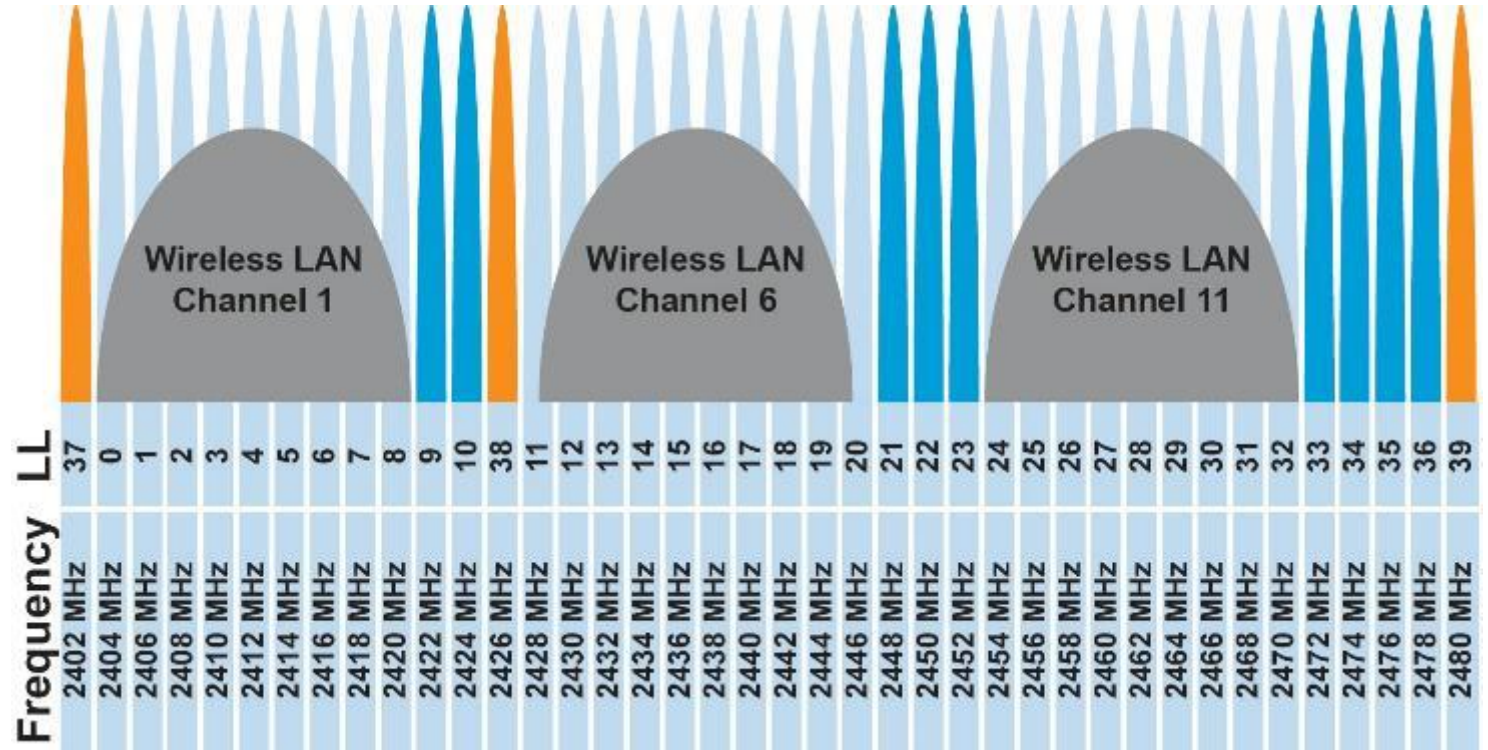
The radio spectrum is a public resource of the United States of America. It is a finite resource and its use is regulated by the Federal Communications Commission (FCC) under the authority of the Federal Communications Act of 1934 and the Communications Act of 1935. The FCC is responsible for the management and allocation of the radio spectrum to ensure that it is used in a fair and equitable manner. The FCC also issues licenses to individuals and organizations for the use of the radio spectrum. The FCC's decisions are subject to review by the United States Court of Appeals for the District of Columbia Circuit. The FCC's website is [www.fcc.gov](http://www.fcc.gov).

**U.S. DEPARTMENT OF COMMERCE**  
National Telecommunications and Information Administration  
Office of Spectrum Management  
August 2011



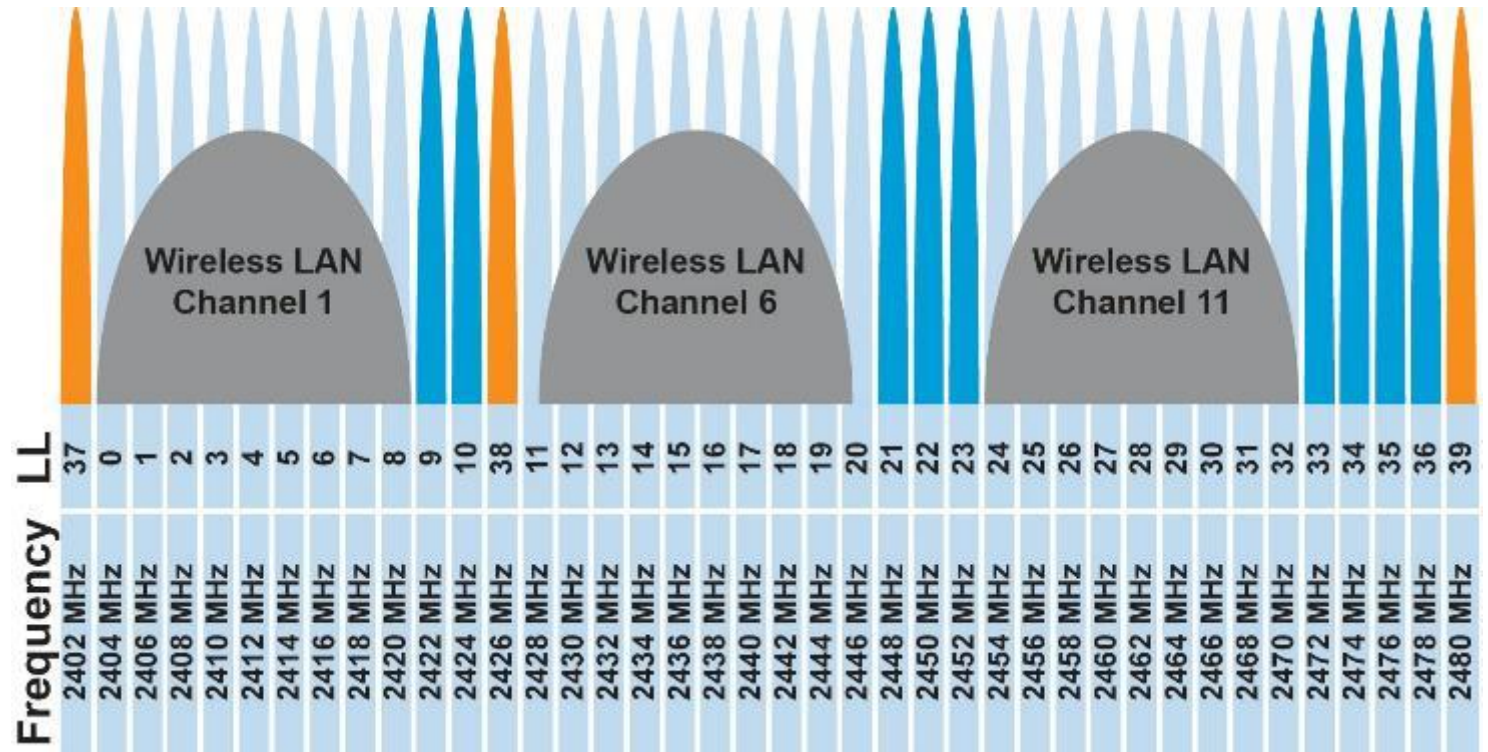
# Unlicensed bands are where IoT thrives

- 902 MHz – 928 MHz
  - LPWANs
- 2.4 GHz to 2.5 GHz
  - WiFi, BLE, Thread
- 5 GHz
  - Faster WiFi
- Cellular uses licensed bands at great cost
  - **Why?**

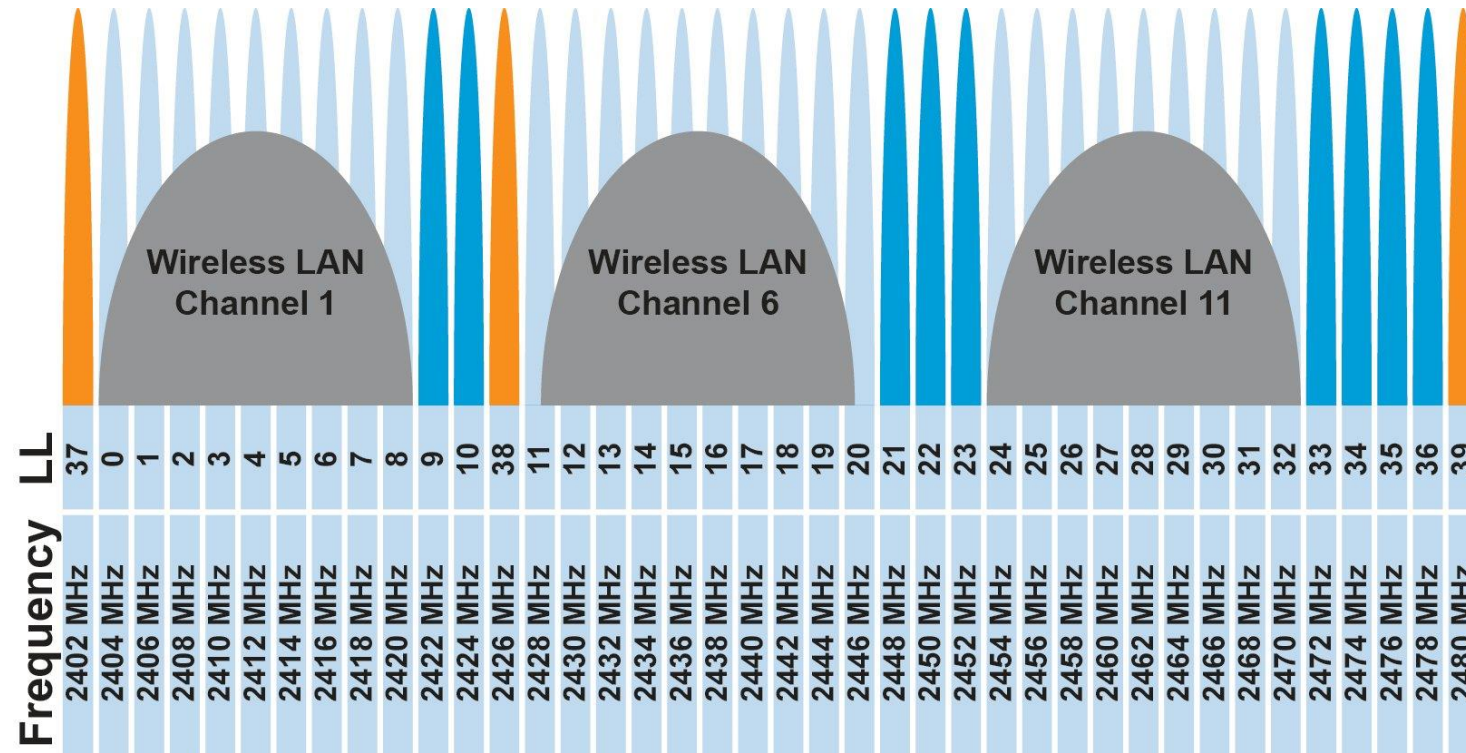


# Unlicensed bands are where IoT thrives

- 902 MHz – 928 MHz
  - LPWANs
- 2.4 GHz to 2.5 GHz
  - WiFi, BLE, Thread
- 5 GHz
  - Faster WiFi
- Cellular uses licensed bands at great cost
  - **Why? No interference from other users**



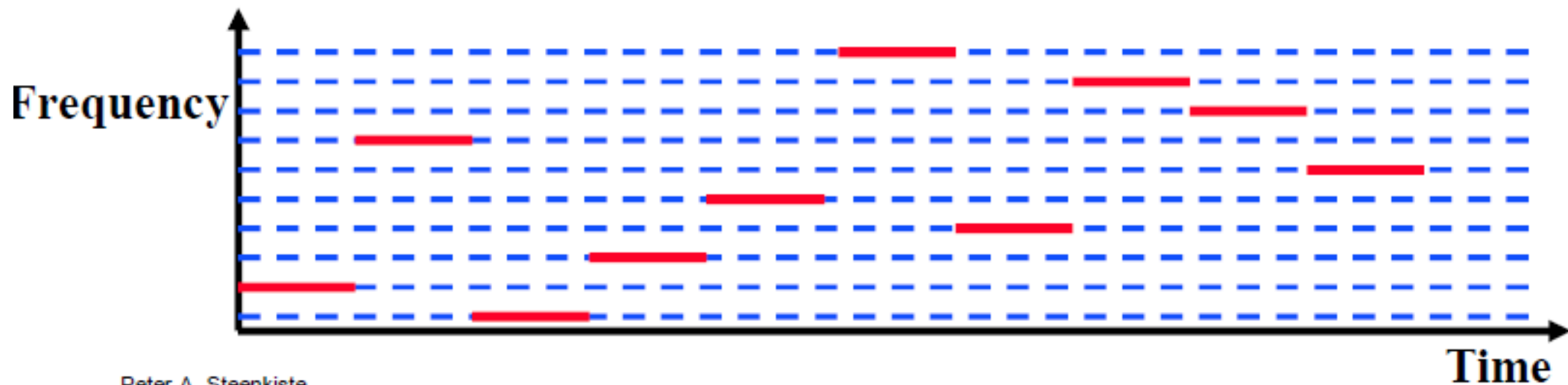
# Different technologies use spectrum in different ways



- How spectrum is used affects: cost (\$), robustness, throughput...
  - We will talk about how each technology uses spectrum, and implications
- This graphic shows how BLE and WiFi interoperate; more on this next week

# Frequency Hopping Spread Spectrum

- Transmitter hops through a sequence of transmit channels
  - Spend some "dwell time" on each channel before hopping again
  - Receiver must know the hopping pattern
- Avoid causing or receiving prolonged interference



Peter A. Steenkiste

## Sidebar: inventor of FHSS – Hedy Lamarr

- Actress and Inventor
  - Designed FHSS with George Antheil during WWII
  - Idea: torpedo control can't be easily jammed if it jumps around
  
- [https://en.wikipedia.org/wiki/Hedy\\_Lamarr#Inventor](https://en.wikipedia.org/wiki/Hedy_Lamarr#Inventor)

# Signal qualities

## 1. Signal strength

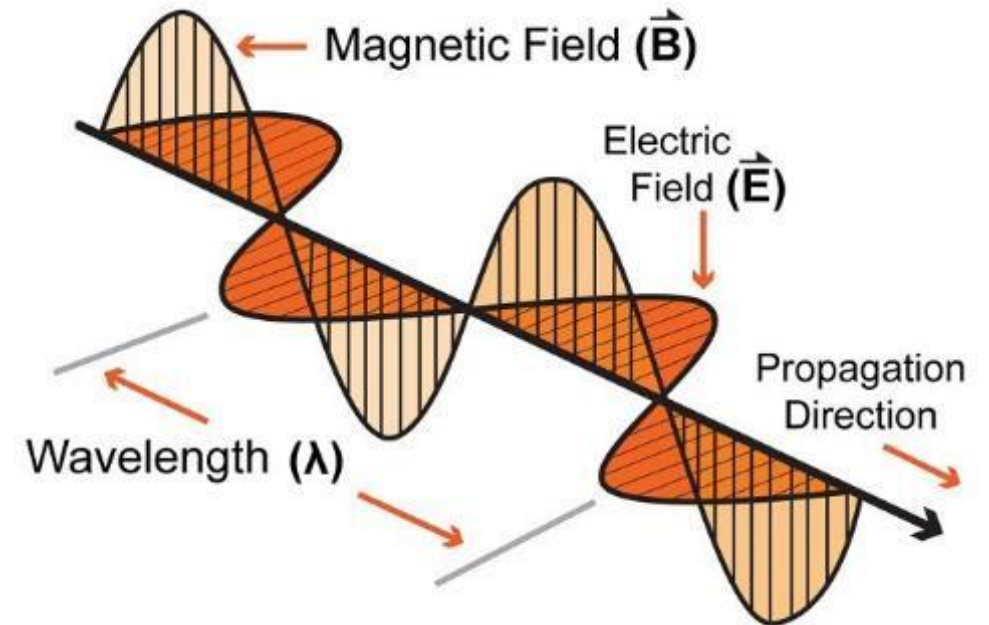
- The amount of energy transmitted/received

## 2. Signal frequency and bandwidth

- Which "channel" the signal is sent on

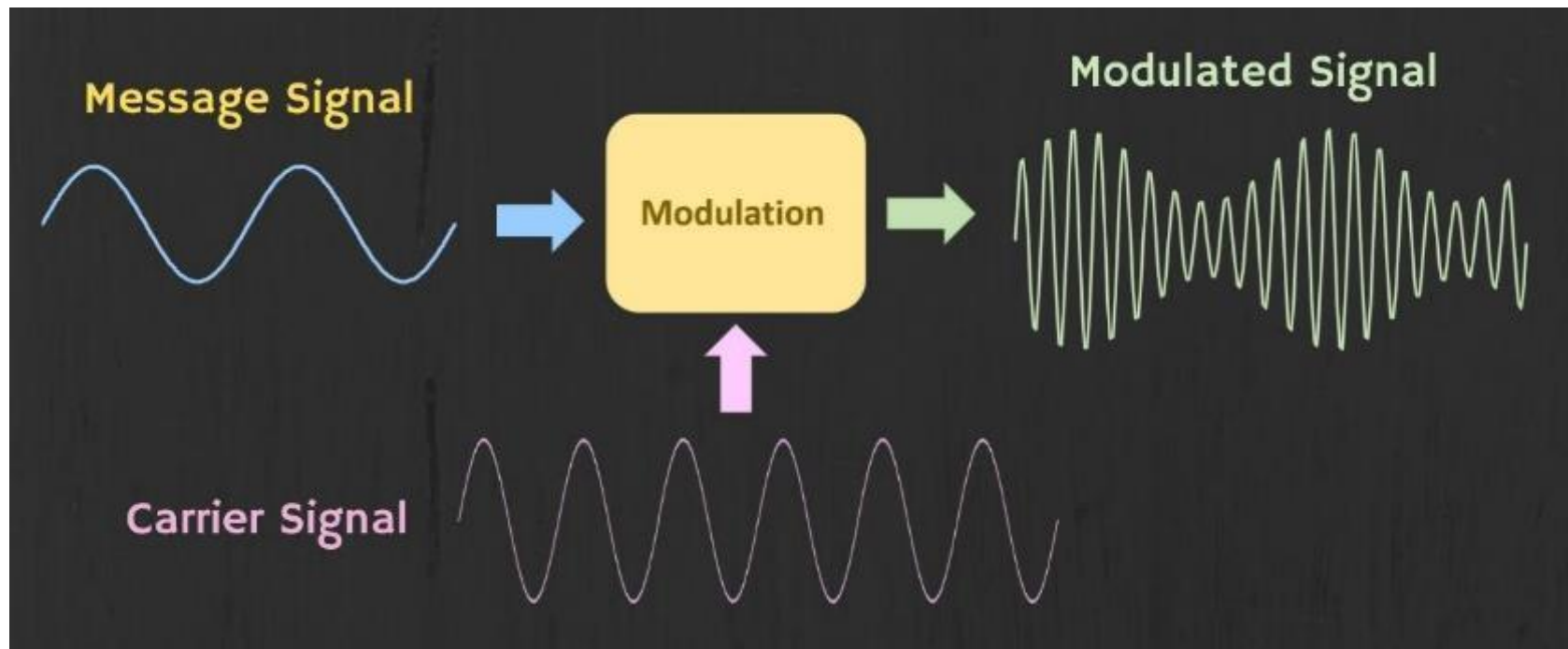
## 3. Signal modulation

- How data is encoded in the signal



# Modulation

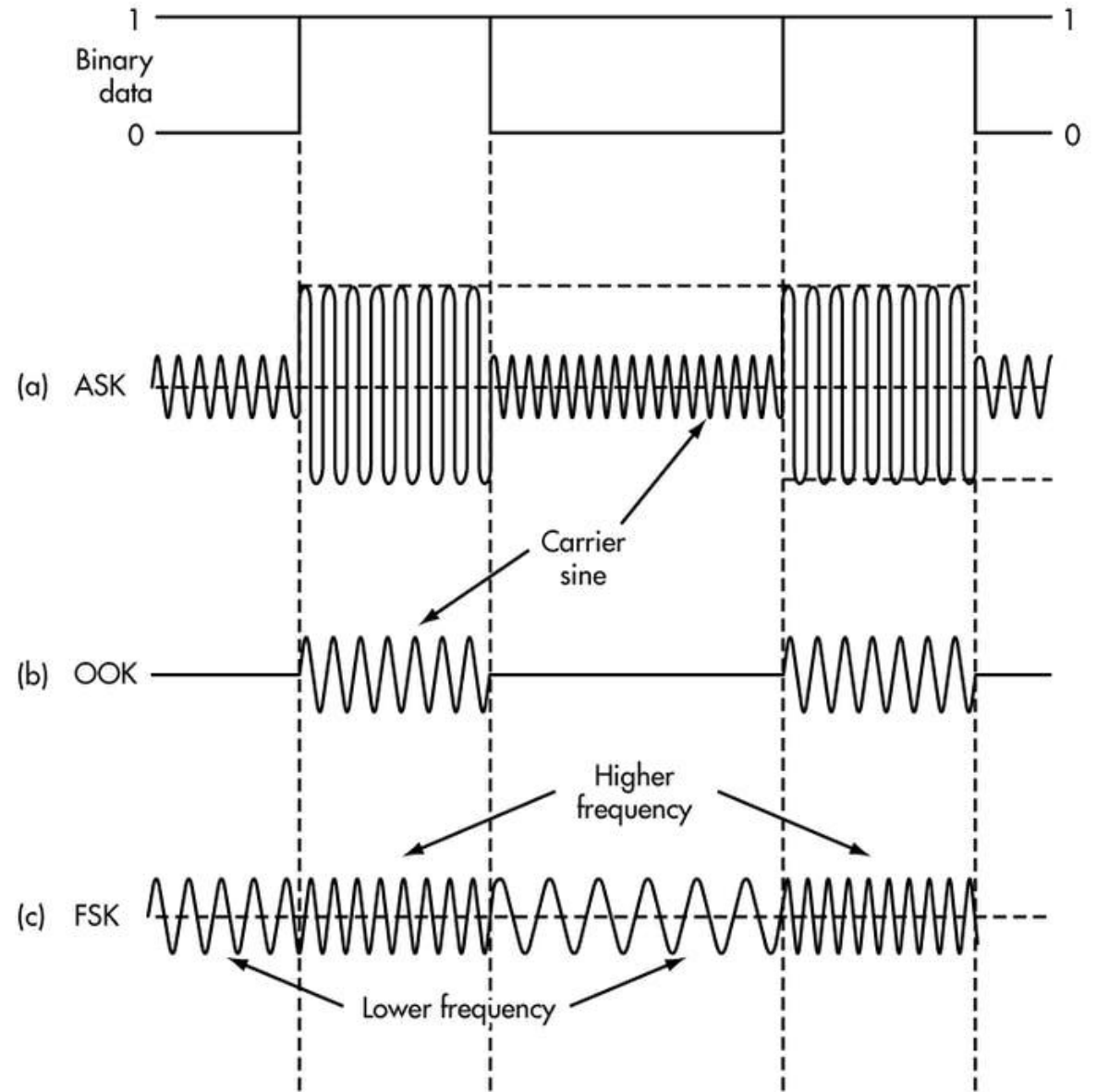
- Encoding signal data in an analog “carrier” signal
  - Carrier signal defines the frequency
  - Modulation scheme + data define bandwidth required





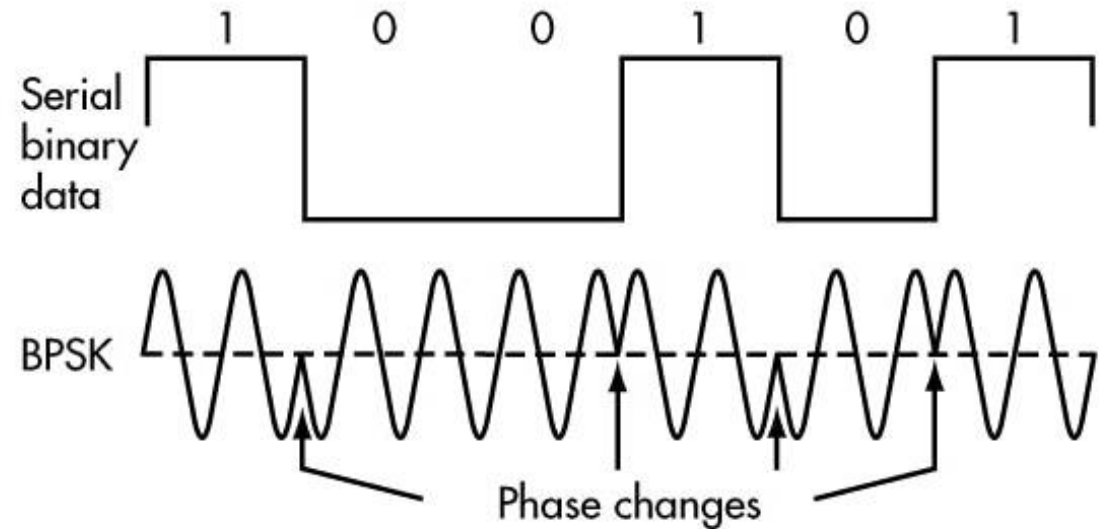
# Modulation types

- Encoding binary data on a signal
- Amplitude-shift Keying (ASK)
  - Modify amplitude of carrier signal
  - On-Off Keying (OOK) is an extreme example
- Frequency-shift Keying (FSK)
  - Modify frequency of carrier signal



# Modulation types

- Phase-shift keying (PSK)
  - Modify phase of carrier signal
  - Usually differential:  
the change signifies data



- More complicated possibilities exist
  - QAM (Quadrature Amplitude Modulation) combines amplitude and phase shift keying
    - Allows for more than one bit per "symbol"

# Modulation tradeoffs

- Various tradeoffs between different modulation schemes
  - Bandwidth requirements, transceiver hardware, immunity to noise, etc.
- ASK (amplitude) is simple but susceptible to noise
  - Noise exists in the real world
- FSK (frequency) is relatively simple and robust to noise, but uses more bandwidth
  - Bandwidth is limited, but still commonly used
- PSK (phase) energy efficient and robust, but more complex hardware
  - More expensive hardware, but very commonly used

# Break + Say hi to your neighbors

- Things to share
  - Name
  - Major
  - One of the following
    - Favorite Candy
    - Favorite Pokemon
    - Favorite Emoji

# Break + Say hi to your neighbors

- Things to share
  - Name -Branden
  - Major -EE, CE, and CS
  - One of the following
    - Favorite Candy - Twix
    - Favorite Pokemon - Eevee
    - Favorite Emoji - 🍷

# Outline

- OSI Layers
- Internet Architecture (Upper Layers)
- Physical Layer
- **Data Link Layer**

# Data Link Layer

- Framing
  - Combine arbitrary bits into a “packet” of data
- Logical link control
  - Manage transfer between transmitter and receiver
  - Error detection and correction
- Media access
  - Controlling which device gets to transmit next
- Inherently coupled to PHY and its decisions

# Framing

- Typical packet structure
  - Preamble - Existence of packet and synchronization of clocks
  - Header - Addresses, Type, Length
  - Data - Payload plus higher layer headers (e.g. IP packet)
  - Trailer - Padding, CRC



- Wireless considerations
  - Control information for Physical Layer
  - Ensure robustness for header
  - Explicit multi-hop routing
  - Possibly different data rates for different parts of packet



# Error control: detection and recovery

- Detection: only detect errors
  - Make sure corrupted packets get discarded
  - Cyclical Redundancy Checks
    - Detect single bit errors
    - Detect "burst" errors of several contiguous bits
- Recovery: also try to recover from small bit errors
  - Forward error correction
  - Retransmissions
  - Far more important for wireless because the cost of transmission is higher

# Medium Access Control

- How does a network determine which transmitter gets to transmit?
- Remember: the wireless medium is inherently broadcast
  - Two simultaneous transmitters may lose both packets

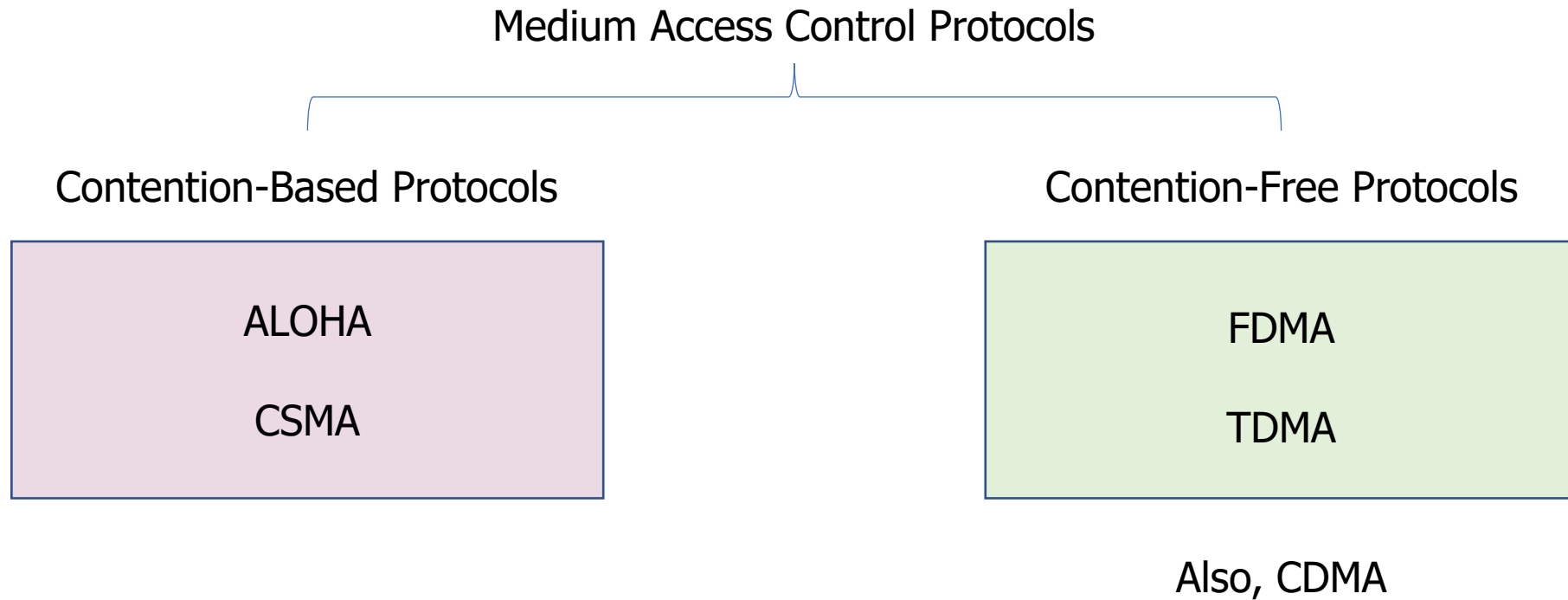
# Analogy: wireless medium as acoustic

- **How do we determine who gets to speak?**
  - Two simultaneous speakers also lose both “transmissions”
- Task: in one minute you will have to recite the alphabet
  - We’ll jump by tables, one person per letter
  - You all fail if two people speak at the same time
  - I will ban any strategy that two tables use

# Analogy: wireless medium as acoustic

- How do we determine who gets to speak?
  - Two simultaneous speakers also lose both “transmissions”
- Eye contact (or raise hand) -> out-of-band communication
- Wait until it's quiet for some time -> carrier sense multiple access
- Strict turn order -> time division multiple access
- Just speak and hope it works -> ALOHA
- Everybody sing at different tones -> frequency division multiple access (stretching the metaphor)
- Everyone speak in different languages -> code division multiple access
  
- Others?

# MAC protocol categorization

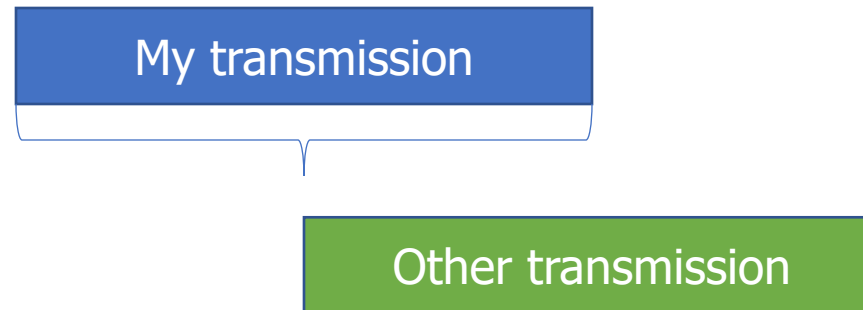


# ALOHA

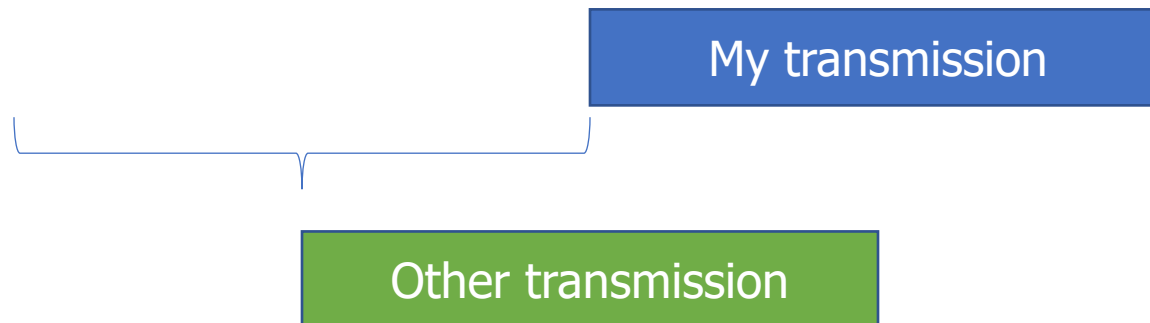
- ALOHAnet (1971)
  - University of Hawaii – Norman Abramson
  - First demonstration of wireless packet network
- Rules
  1. If you have data to send, send it
- Two (or more) simultaneous transmissions will collide and be lost
  - Wait a duration of time for an acknowledgement
  - If transmission was lost, try sending again “later”
    - Want some kind of exponential backoff scheme here

# Packet collisions

- Each packet transmission has a window of vulnerability
  - Twice the on-air duration of a packet
  - Transmissions during the packet are bad

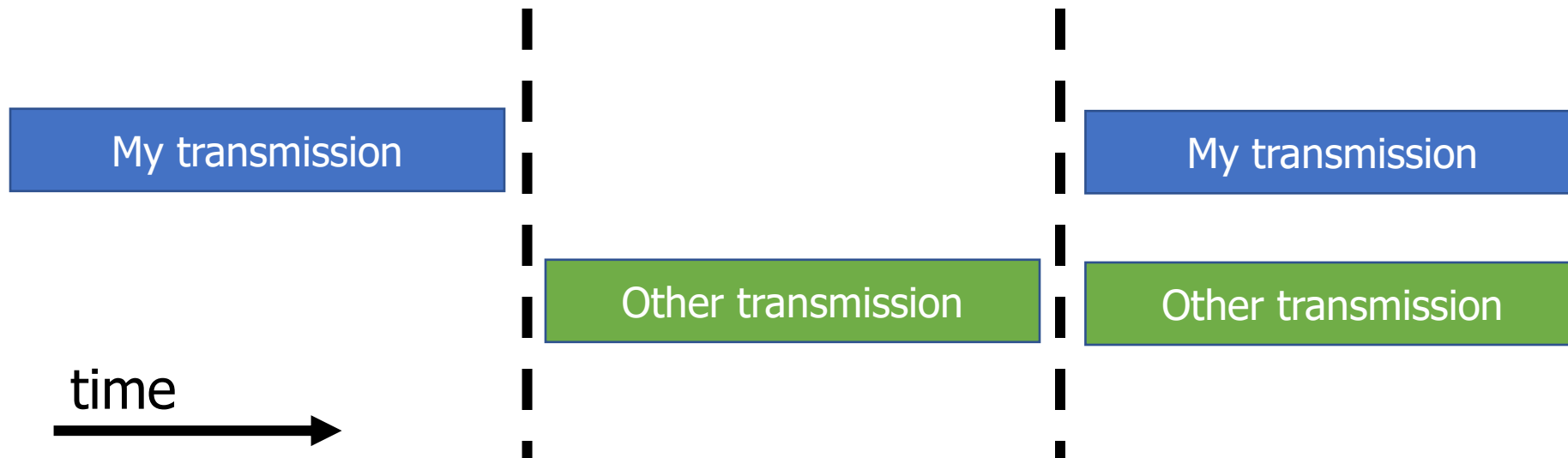


- Transmissions before packet can also be bad



# Slotted ALOHA

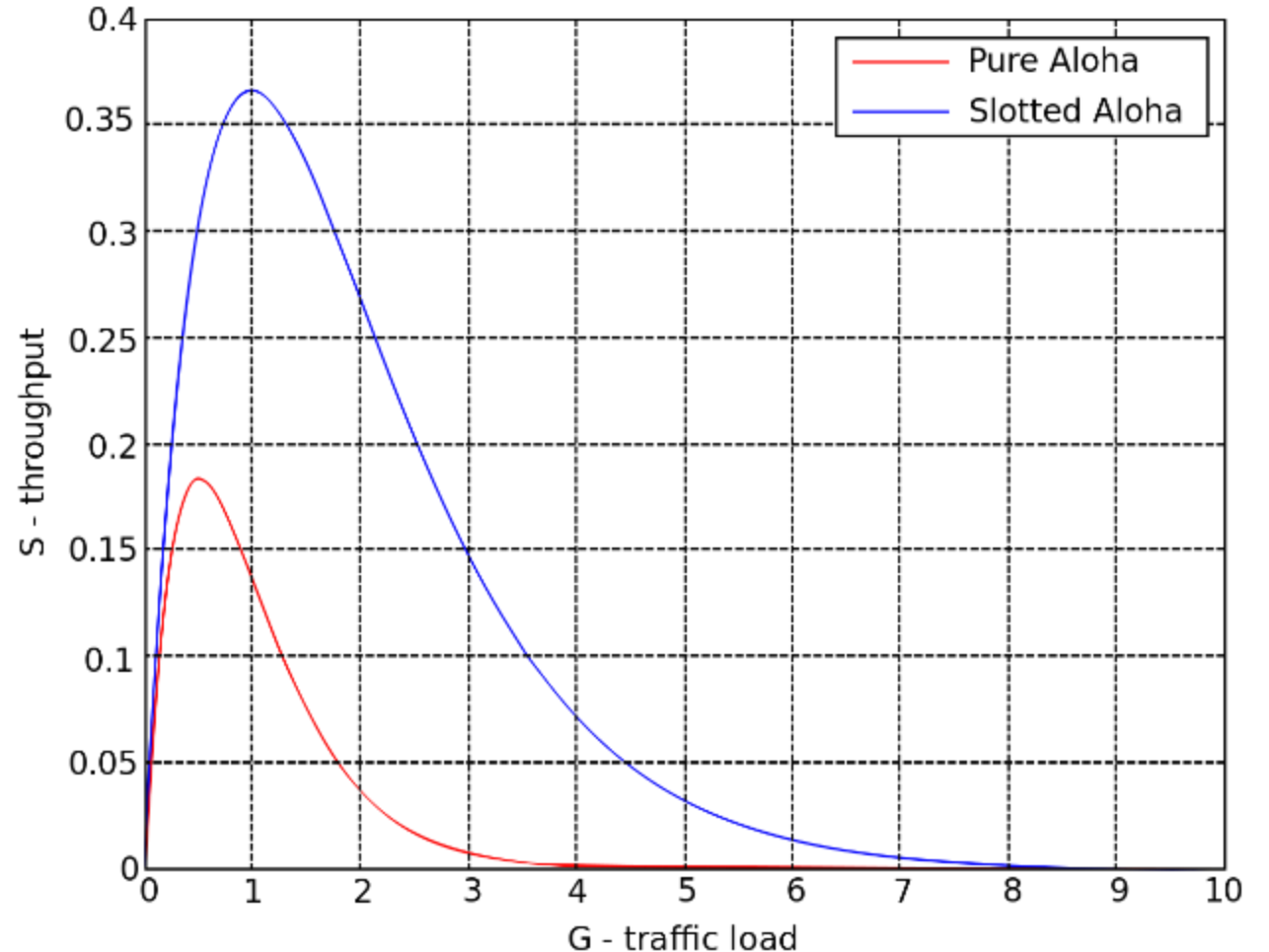
- Split time into synchronized “slots”
- Any device can transmit whenever it has data
  - But it must transmit at the start of a slot
  - And its transmission cannot be longer than a slot
  - Removes half of the possibilities for collisions!
    - At the cost of some synchronization method





# ALOHA throughput

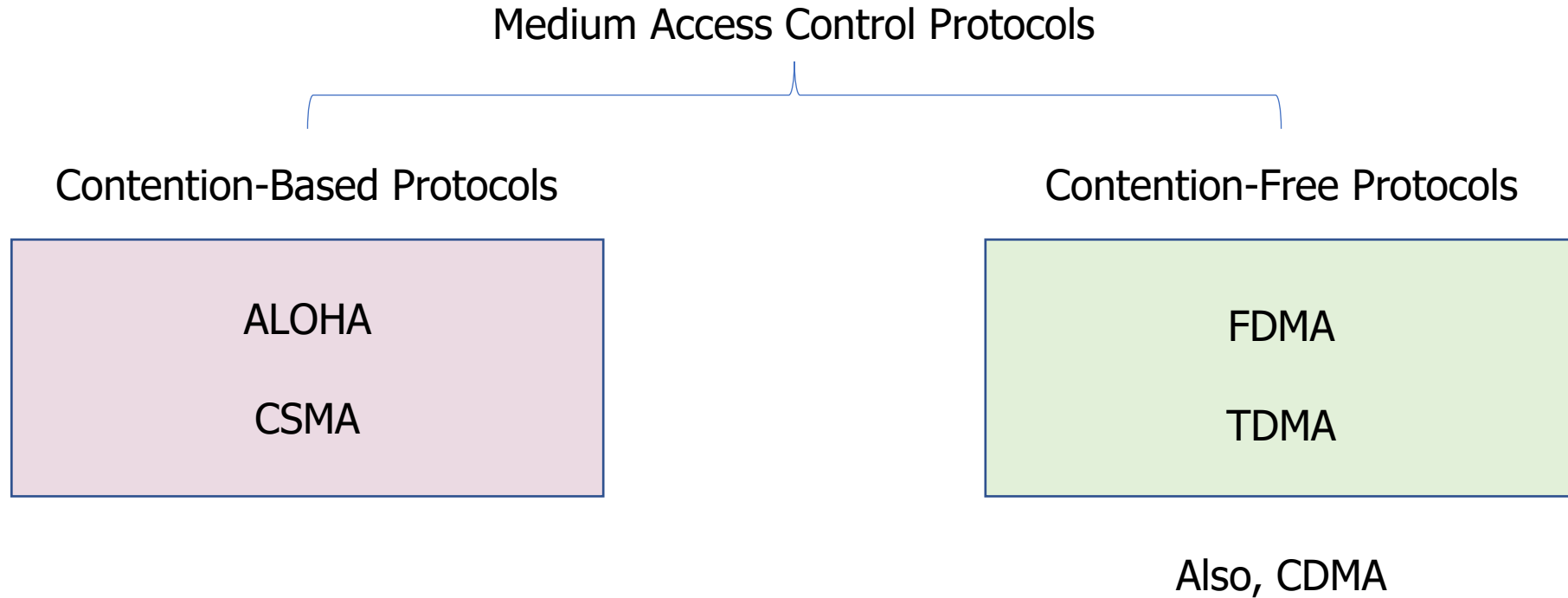
- It can be shown that traffic maxes out at
  - ALOHA: 18.4%
  - Slotted ALOHA: 36.8%
- Assuming Poisson distribution of transmission attempts
- Slotted throughput is double because the “before” collisions can no longer occur



# Capture effect

- Actually, two packets at once isn't *always* a total loss
  - The louder packet can still sometimes be heard if loud enough
- How much louder?
  - Ballpark 12-14 dB
- When does this work?
  - Depends on the radio hardware
  - Louder packet first almost always works
  - Louder packet second *sometimes* works

# MAC protocol categorization



# CSMA/CA – Carrier Sense Multiple Access with Collision Avoidance

- First listen for a duration and determine if anyone is transmitting
  - If idle, you can transmit
  - If busy, wait and try again later
- “listen before send”
- Can be combined with notion of slotting
  - If current slot is idle, transmit in next slot
  - If current slot is busy, follow some algorithm to try again later

# CSMA/CD – CSMA with Collision Detection

- Detect collisions during your own transmission
  - Works great on wired mediums (Ethernet, I2C)
- Very challenging for wireless systems
  - Transmit and receive are usually the same antenna
  - Receiving while transmitting would be drowned out by transmission
    - Remember: TX at 8 dBm and RX at -95 dBm
- Area of active research!

2017 18th Annual Conference on Wireless On-demand Network Systems and Services (WONS)

## On the Feasibility of Collision Detection in Full-Duplex 802.11 Radio

Michele Segata, Renato Lo Cigno

Dept. of Information Engineering and Computer Science, University of Trento, Italy  
{msegata, lo.cigno}@disi.unitn.it

**Abstract**—Full-duplex radios are becoming a feasible reality thanks to recent advances in self-interference cancellation. Switching from half- to full-duplex requires a major re-design of many network features and characteristics, including the MAC layer. The literature provides several new proposals or improvements that are applicable in different topologies; the notion of channel itself becomes blurred, as there is intrinsic spatial reuse and stations very far one another hidden with respect to stations in between. Still, the possibility of detecting collisions and avoid the waste of channel is appealing and has only partially been investigated [5].

2014 IEEE 22nd International Conference on Network Protocols

## Concise Paper: Semi-Synchronous Channel Access for Full-Duplex Wireless Networks

Xiaofeng Xie and Xinyu Zhang  
University of Wisconsin-Madison  
Email: {xiufeng,xyzhang}@ece.wisc.edu

## Throughput Analysis of CSMA With Imperfect Collision Detection in Full Duplex-Enabled WLAN

Megumi Kaneko

**Abstract**—As an alternative to carrier sense multiple access (CSMA) with collision avoidance in half-duplex wireless local area network (WLAN) that incurs heavy control overhead, full-duplex WLANs enabling wireless collision detection (WCD) by simultaneous carrier sensing and data transmission are getting

through CSMA with perfect characteristics, actual performance in wireless environment, user patterns. Hence, we derive a throughput analysis for imperfect WCD, that can be expected from conventional random access

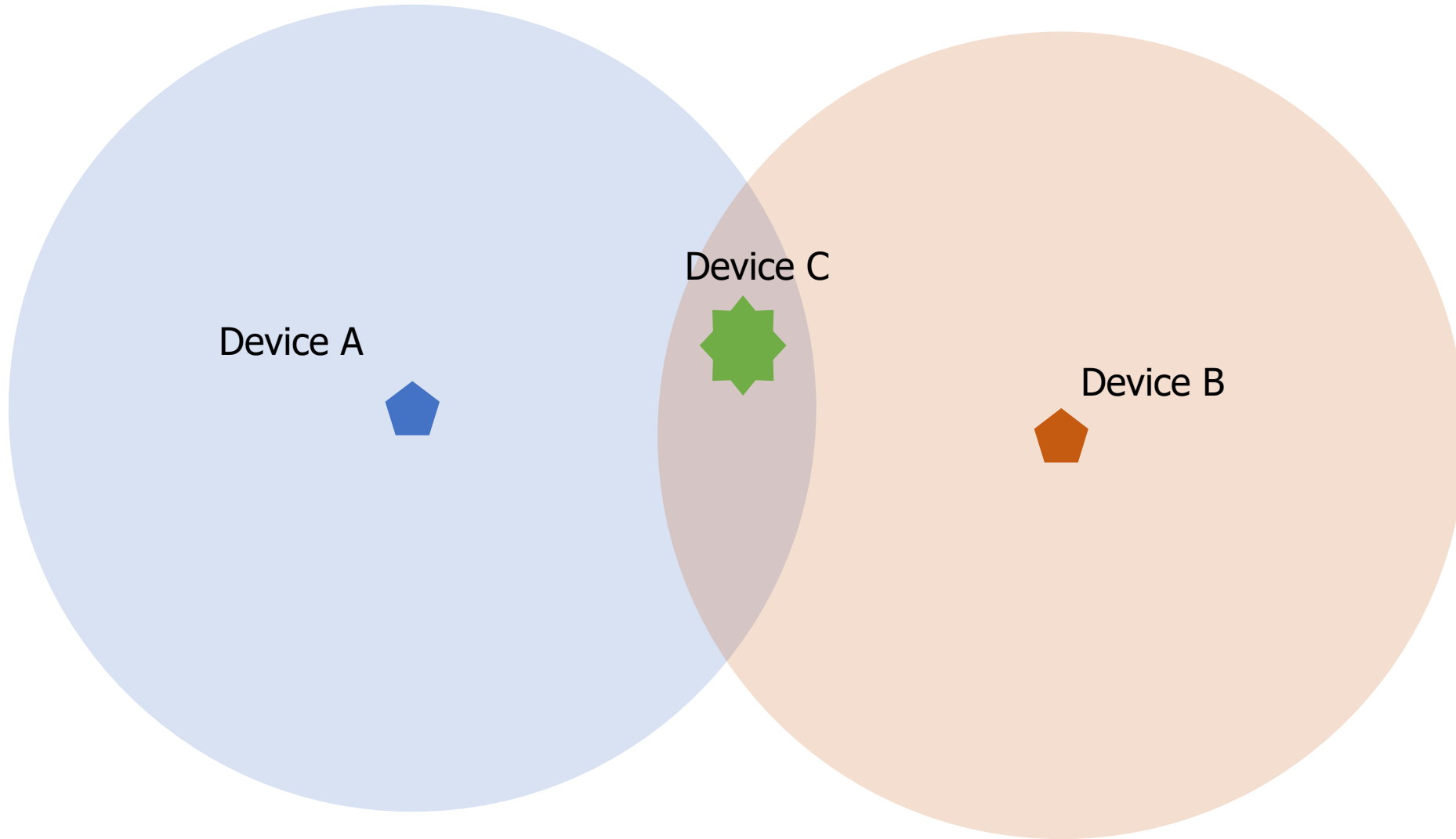
multiple access, wireless collision, full duplex.

work (WLAN) systems are problems due to the error

main reasons. Firstly, a collision detected at the transmitter does not necessarily imply a collision at the receiver due to the nature of wireless channels such as large/small-scale fading. Secondly, detecting simultaneous transmissions during one's own transmission is very challenging, as the transmitter's self-interference signal power is several orders of magnitudes higher than that of collision signals to be detected.

Thus, a number of PHY layer WCD schemes have been proposed [7], [8]. A MIMO-based scheme is designed in [7] for detecting an interfering preamble signal at one of the transmit antennas, and a self-interference canceller is designed in [8] which enables the transmitter to detect simultaneous transmissions even under very high self-interference. Such schemes allow the UTs to detect potential collisions during transmission, and hence to immediately revert to the retransmission process without any delay, leading to large throughput improvements compared to CSMA/CA [5]. Note that [3] assumed an ideal WCD where any collision can be perfectly detected at the transmitter. In [9], the impact of interference on full-duplex transmitter-receiver pairs in ad-hoc

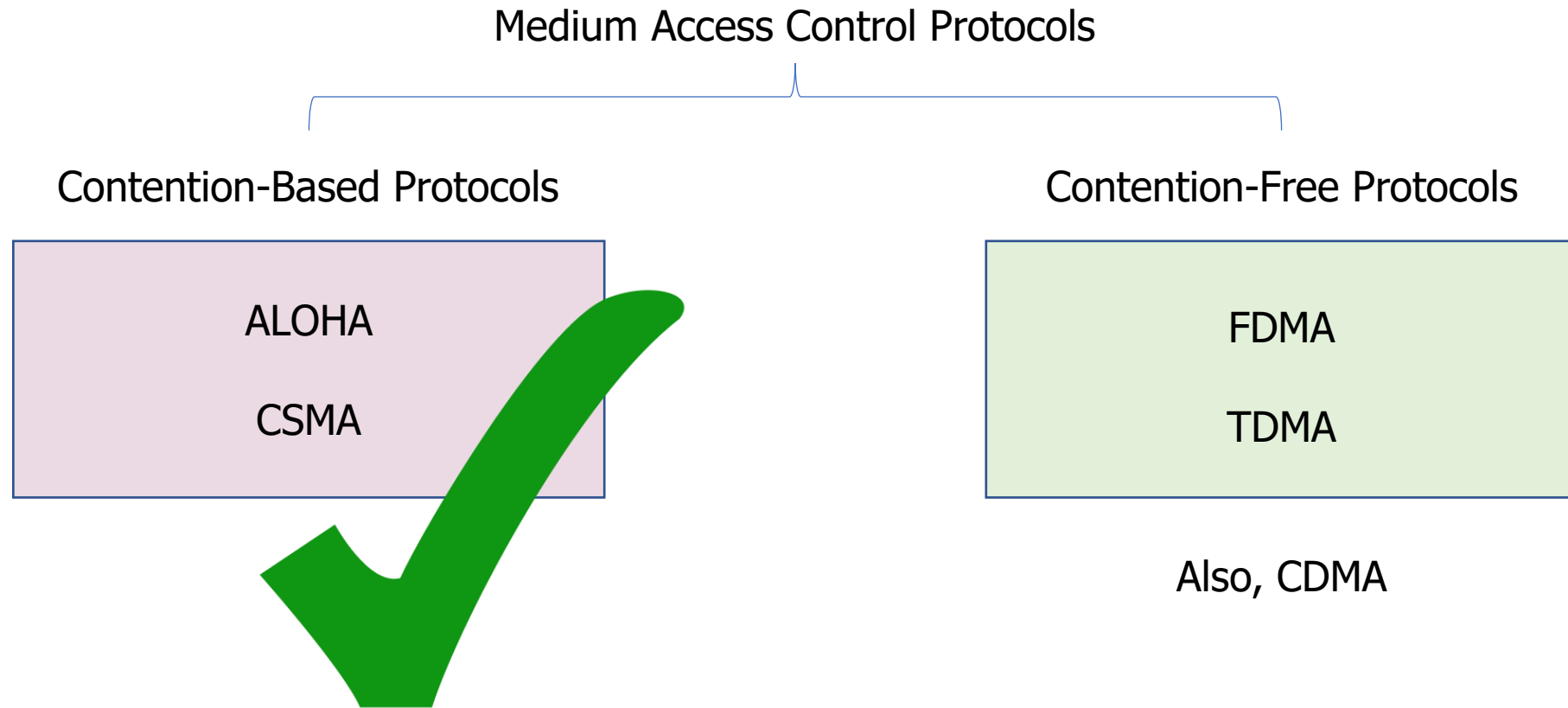
# Hidden terminal problem



# CSMA with RTS/CTS

- Hidden terminal problem means that two transmitters might never be able to detect each other's transmissions
- A partial solution
  - When channel is idle, transmitter sends a short Request To Send (RTS)
  - Receiver will send a Clear To Send (CTS) to only one node at a time
  - RTS collisions are faster and less wasteful than hidden terminal collisions
  - Downside: overhead is high for waiting for CTS when contention is low

# MAC protocol categorization



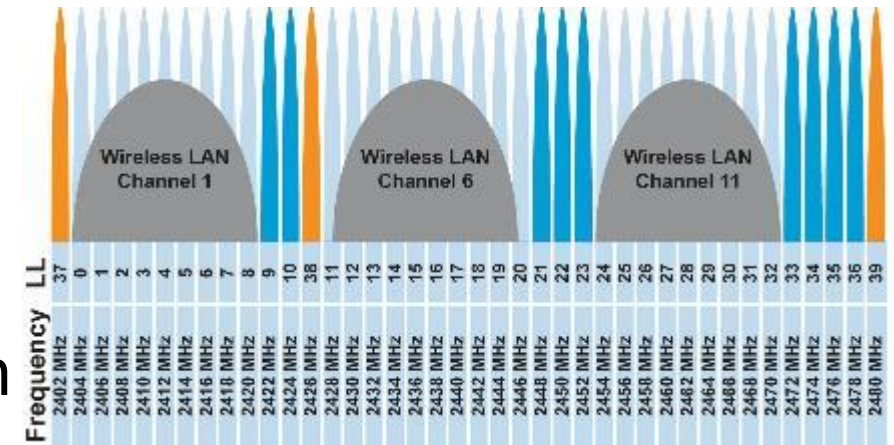


# Contention-free access control protocols

- Goal: split up communication such that devices will not conflict
- Can be predetermined or reservation-based
  - Devices might request to join the schedule and be given a slot
    - Devices lose their slot if it goes unused for some amount of time
    - Reservations often occur during a dedicated CSMA contention slot
  - Assignment of schedules can be complicated
- Really efficient at creating a high-throughput network
  - Assuming they are all following the same protocol
  - Otherwise, interference can be very problematic

# FDMA – Frequency Division Multiple Access

- Split transmissions in frequency
  - Different carrier frequencies are independent
  - Fundamentally how RF spectrum is split
- Technically, each device uses a separate, fixed frequency
  - Walkie-talkies
- Conceptually, how RF channels work
  - WiFi networks pick different bands
  - 802.15.4 picks a channel to communicate on



# TDMA – Time Division Multiple Access

- Split transmissions in time
  - Devices share the same channel
- Splits time into fixed-length windows
  - Each device is assigned one or more windows
  - Can build a priority system here with uneven split among devices
- Requires synchronization between devices
  - Often devices must listen periodically to resynchronize
  - Less efficient use of slots reduce synchronization
    - Large guard windows. E.g. 1.5 second slot for a 1 second transmission

# Real-world protocol access control

- ALOHA
  - BLE advertisements
  - Unlicensed LPWANs: Sigfox, LoRaWAN
- CSMA
  - WiFi (slotted, CSMA/CA)
- TDMA
  - BLE connections
  - Cellular LPWANs: LTE-M and NB-IoT

# Outline

- OSI Layers
- Internet Architecture (Upper Layers)
- Physical Layer
- Data Link Layer