CS-340 Introduction to Computer Networking

Lecture 13: Medium Access Control

Steve Tarzia

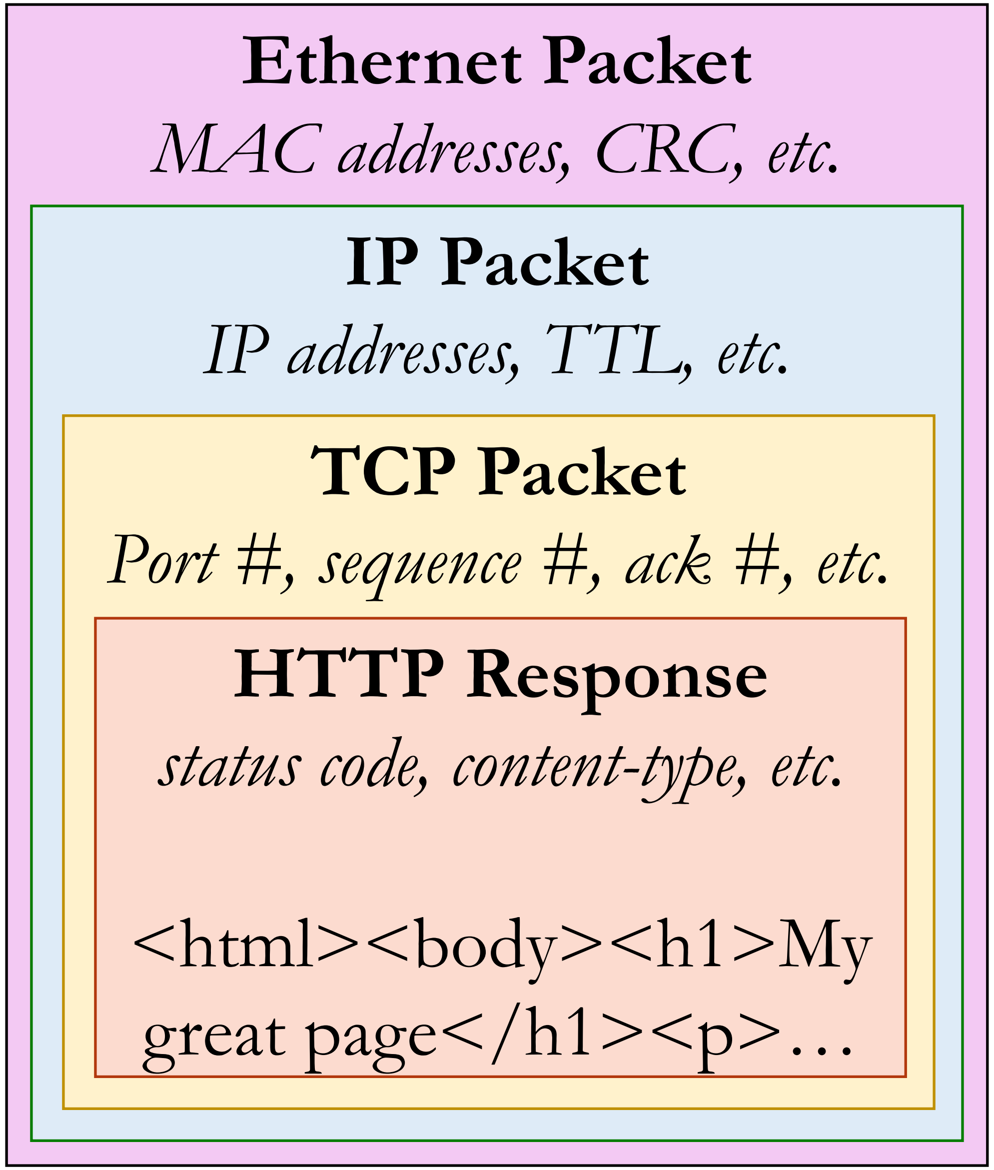
Fall 2020

Many diagrams adapted from those by J.F Kurose and K.W. Ross

Recap

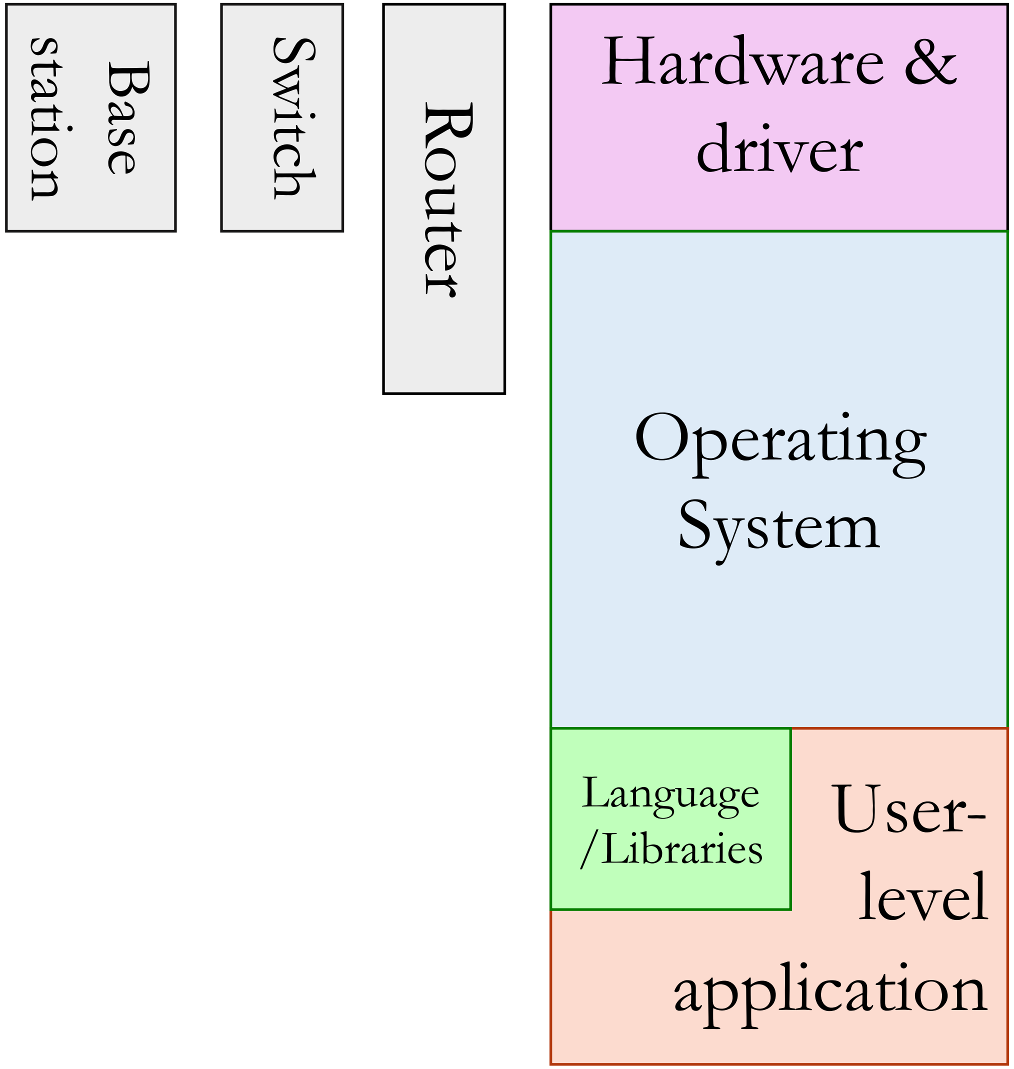
- **IP anycast**: BGP trick to send traffic for one IP address to two hosts.
- **Broadcasting** is sending a single message to every host.
  - **Multicast** is sending a single message to many (but not all) hosts.
- **Controlled flooding:** add a sequence number to messages, and retransmit only if you have not seen the received sequence number.
- **Spanning tree:** a graph without cycles that reaches all nodes. Broadcast can be done by transmitting along a spanning tree.
  - **Prim's algorithm** constructs a minimum-cost spanning tree
  - **Dijkstra's algorithm** constructs a shortest-path-from-root spanning tree

Each layer solves a particular set of problems



- **Link layer**: shares a physical channel among several transmitters/receivers
- Network layer: routes from source to destination, along many hops.
- Transport layer:
  - Multiplexing (>1 connection / machine)
  - Ordering, • Acknowledgement, • Pacing
- HTTP layer:
  - Resource urls, • Response codes,
  - Caching, • Content-types, • Compression

Who implements each layer?



- **Link layer**: shares a physical channel among several transmitters/receivers
- Network layer: routes from source to destination, along many hops.
- Transport layer:
  - Multiplexing (>1 connection / machine)
  - Ordering, • Acknowledgement, • Pacing
- HTTP layer:
  - Resource urls, • Response codes,
  - Caching, • Content-types, • Compression

Terminology

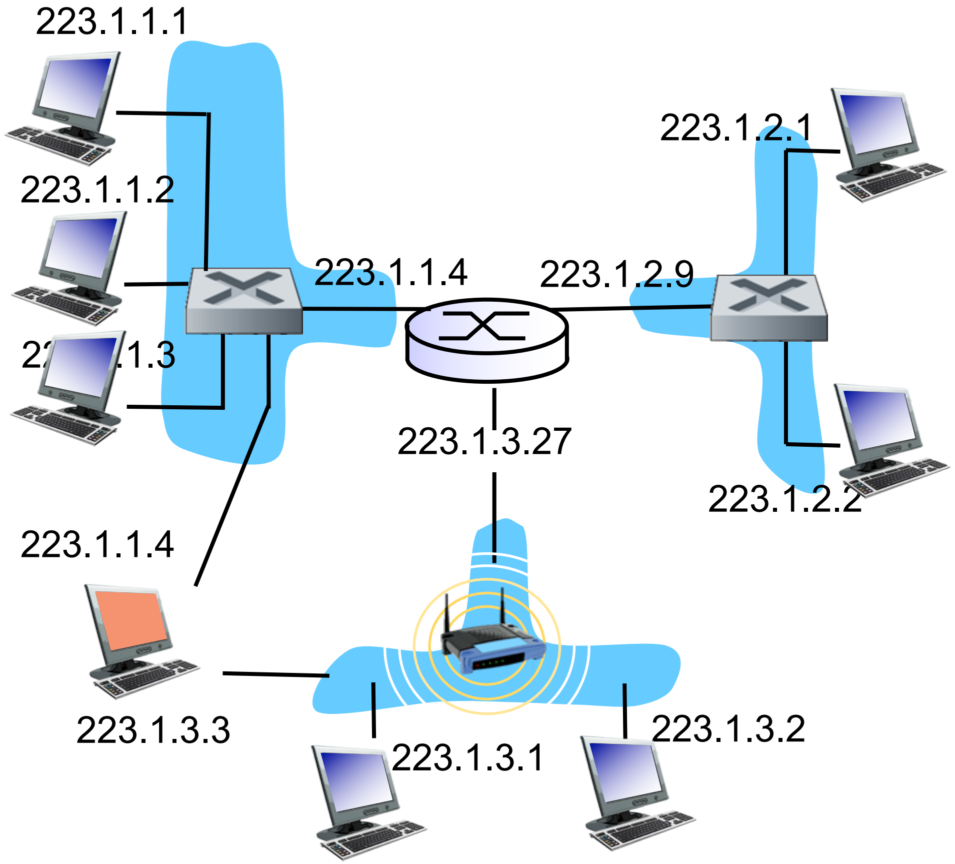
- ***Nodes*** are hosts and routers.
- ***Links*** are communication channels that connect adjacent nodes along communication path
  - wired links
  - wireless links
  - LANs
- Physical connectivity is called Layer 1.
- Link layer is called ***Layer 2***.
- IP layer is ***Layer 3***.
- Transport (TCP) layer is Layer 4.

cables connect two interfaces

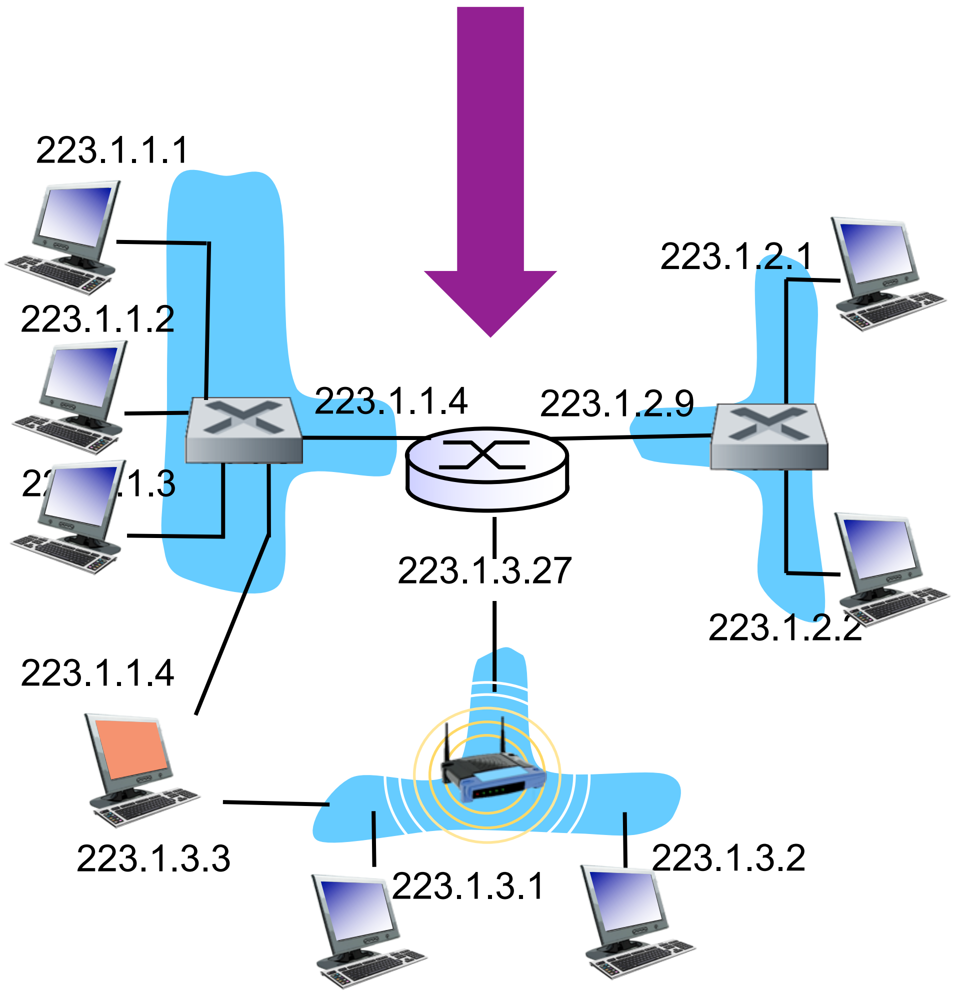
switches connect multiple Ethernet interfaces

base stations connect multiple wireless interfaces.

routers connect subnets



Why does a router have multiple IP addresses?

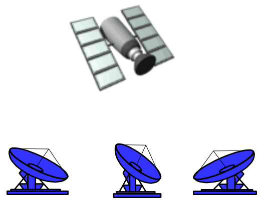


- Nodes on a given subnet can all communicate via link-layer
- Router *routes* packets between subnets.
- So, it must have an IP address on each subnet it is participating in

Medium Access Control (MAC)

shared wire (e.g., cabled Ethernet)

shared RF (e.g., 802.11 WiFi)



shared RF (satellite)

- Communication happens in some physical **medium**. (Plural is **media**.)
- Wireless communication occurs in open space, in some freq. band.
- Wired communication occurs on a wire.
  - Many nodes are often connected to the same wire, physically or logically.
- MAC is needed in **broadcast links**, where more than one node is *sharing* a single channel.
- MAC is unnecessary in **point-to-point** links.
  - Eg., a wired link with separate send and receive wires
  - such connections are only common over long distances.

Multiple Access Protocols

- **Collision** is the fundamental problem in broadcast links:
    - Multiple nodes try to communicate at the same time.
    - Simultaneous messages interfere with each other.
    - None of the colliding messages can be received.
    - Precious time and bandwidth is wasted.
    - Messages must be retransmitted.
- If we're too aggressive in sharing, many collisions will occur.
- If we're too polite about sharing, may waste time/bandwidth waiting.

Multiple Access Protocol Goals

Assuming multiple nodes are sharing a link with throughput ***R*** bps:

- If only one node is communicating, then it should get the full bandwidth (***R***).
- When ***N*** nodes have data to send, they should each get ***R/N*** bandwidth.
- Protocol should be decentralized, with no single point of failure.
- It should be simple and inexpensive to implement.

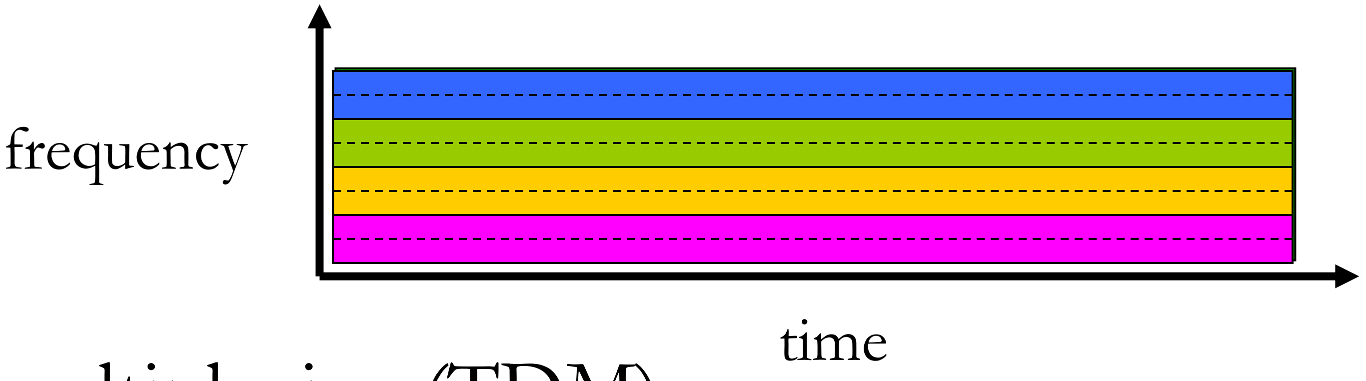
Three basic classes of multiple access protocols:

- Channel partitioning
- Random access
- Taking turns

Channel Partitioning Protocols

Frequency division multiplexing (FDM)

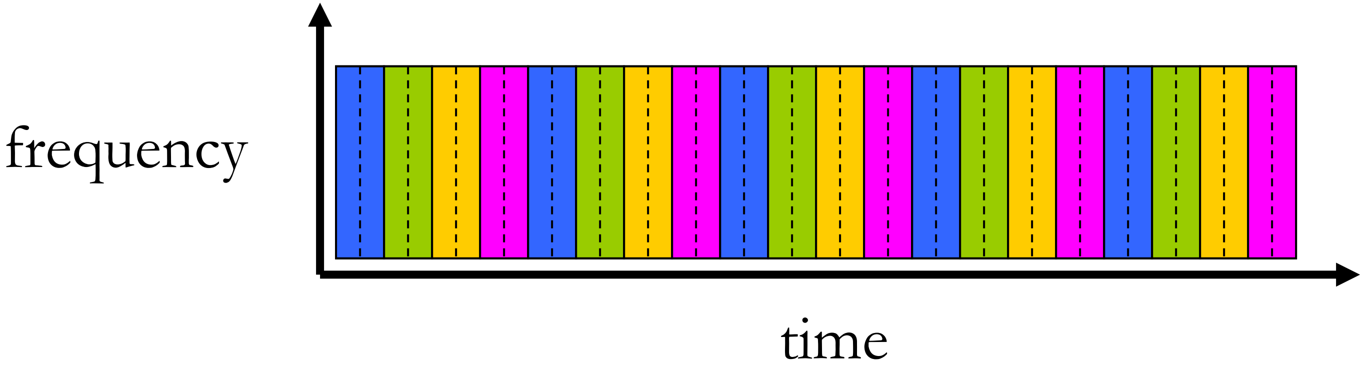
4 nodes



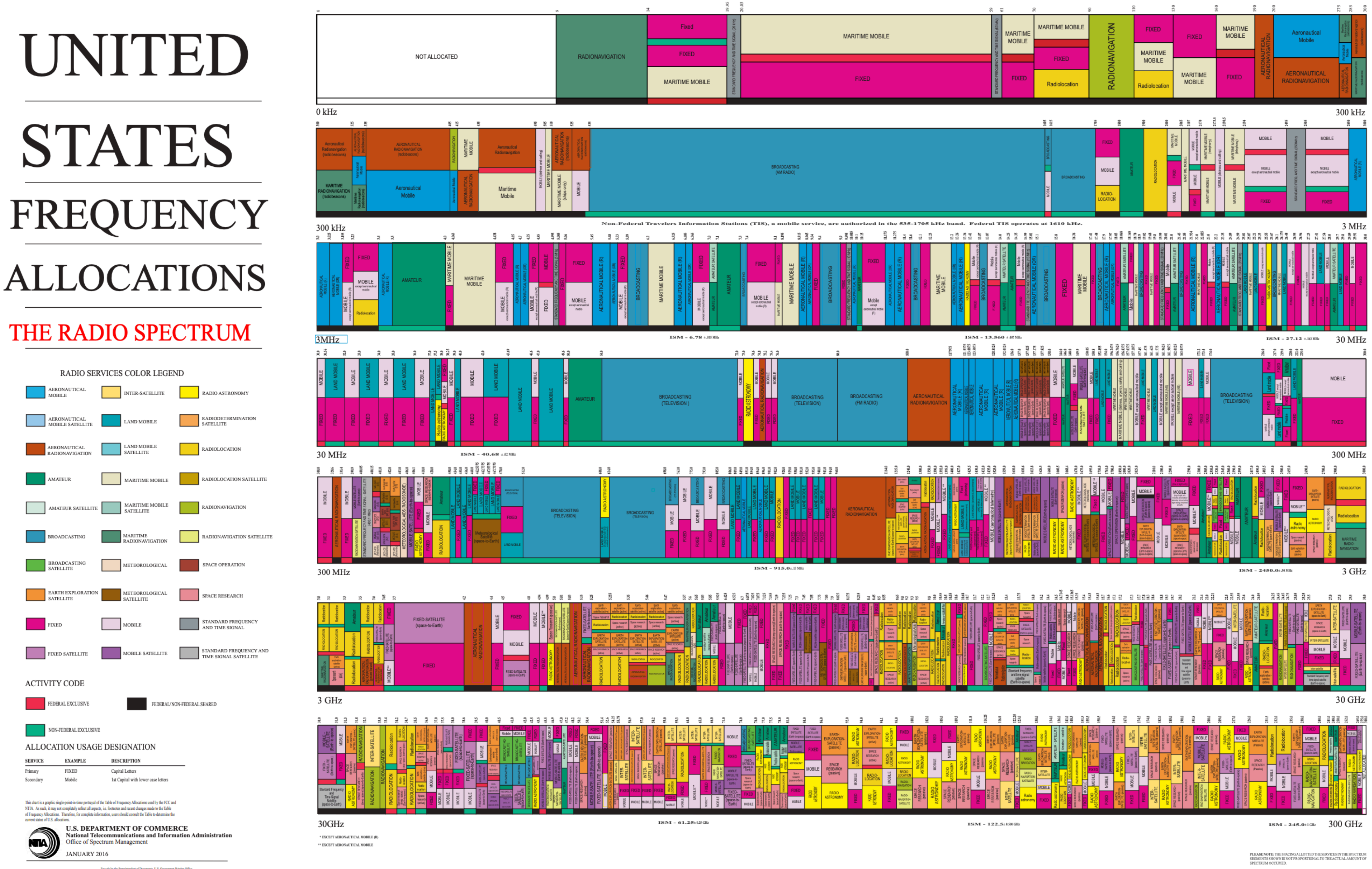
Each channel can only use ¼ of the frequency range, so it can send only ¼ of the total max throughput.

Why less data throughput?

Time division multiplexing (TDM)

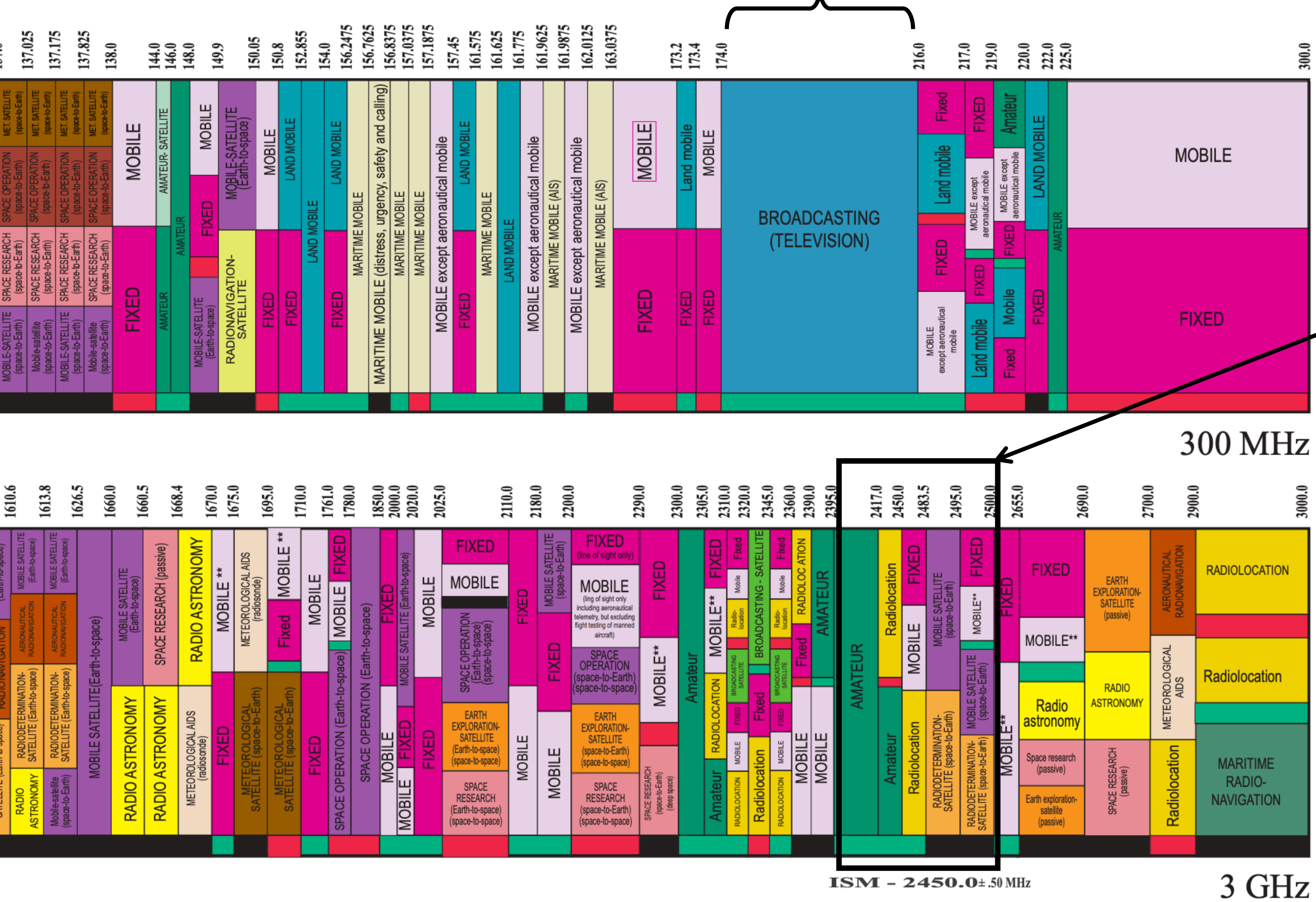


US radio **spectrum** (use of frequencies bands) is governed by the [FCC]



Zoomed-in view

Frequency ranges are called **bands.**



ISM band: Industrial, Scientific, and Medical bands are "unlicensed" & usable by random electronic devices.

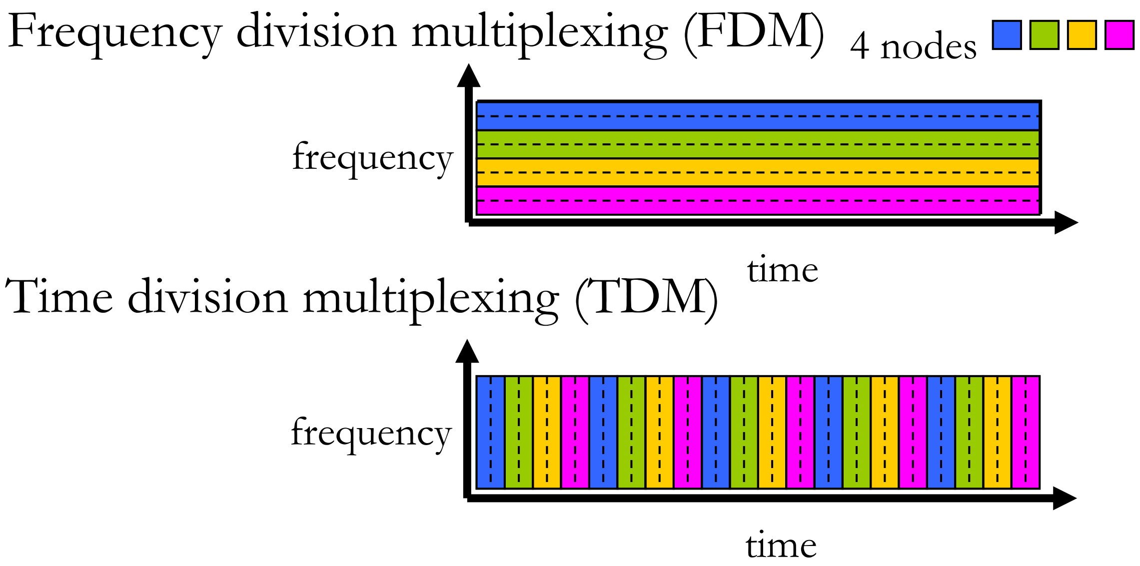
2.4Ghz band is used by:

- Microwave ovens
- 802.11b/g/n/ax WiFi
- Cordless telephones
- Radio-controlled toys
- Whatever!

Channel Partitioning Protocols

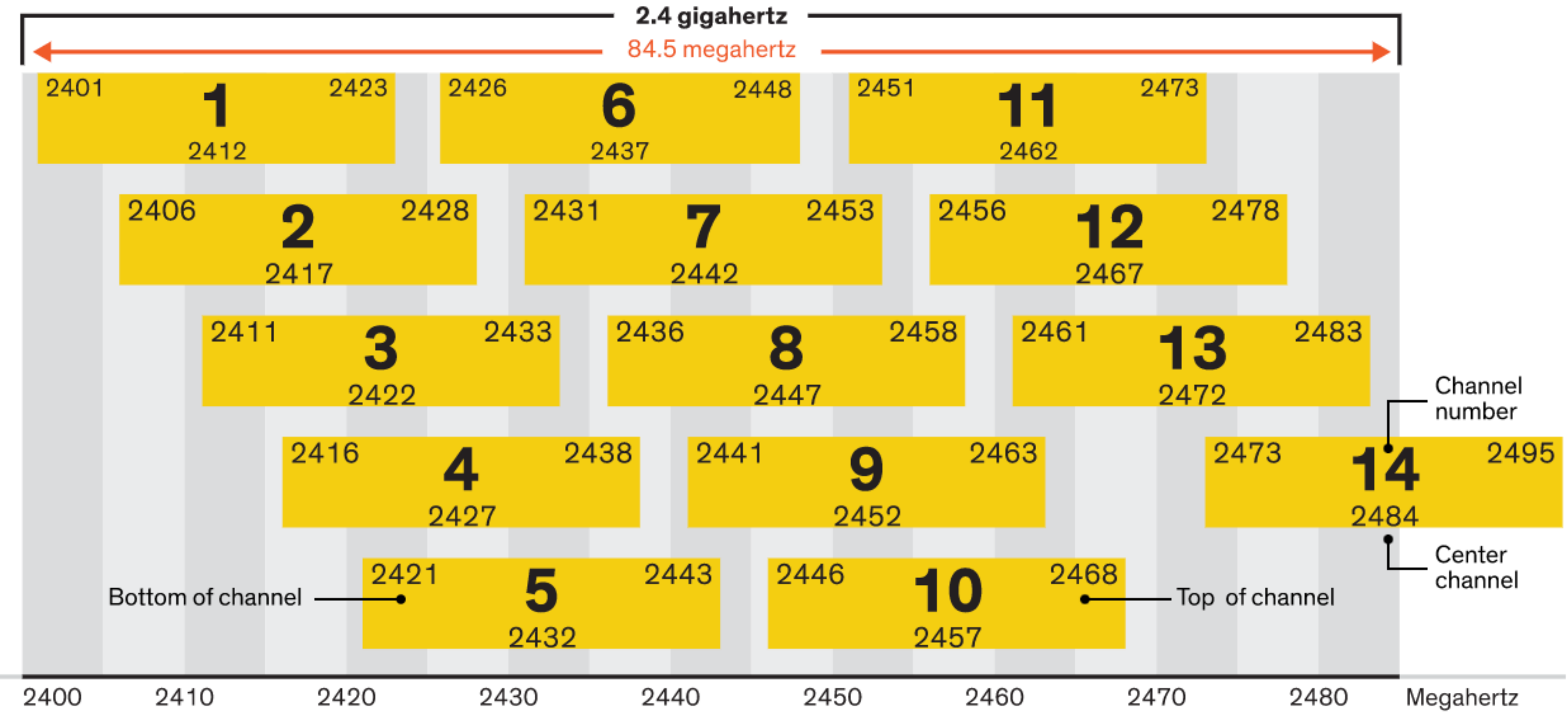
"Bandwidth" literally means the **size of a frequency range** and is measured in hertz. The word's use as a synonym for **throughput** derives from FDM

- Each node gets a reserved "slice" of the bandwidth.
- **FDM**: each node uses a fraction of the spectrum, thus less bandwidth
- **TDM**: each node communicates a fraction of the time
- These are **collision-free** and **fair**, but **inflexible**:
  - Max throughput is R/N, even if only one node wants to communicate!



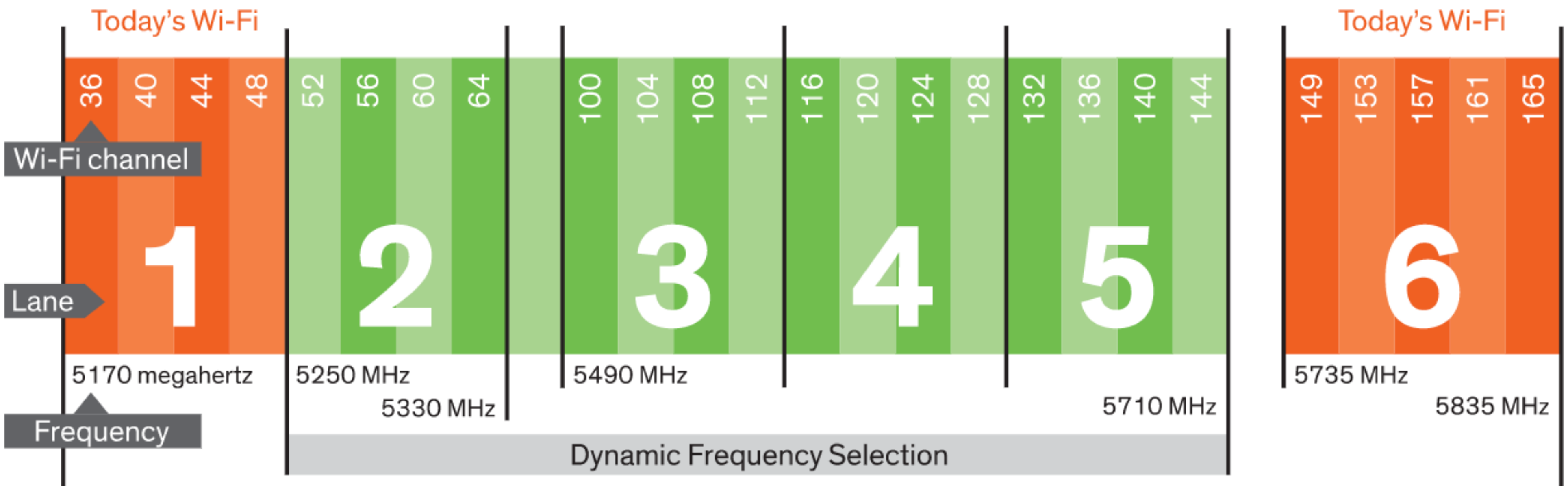
- Code division multiplexing (**CDM**) is similar, but more difficult to explain without EE background.

Wifi uses FDM to isolate different base stations



- **2.4Ghz WiFi** technically support 14 channels, but the overlap means that it's best to use only **three**: channels 1, 6, and 11.
- If your neighbor uses channel 9, their traffic will interfere with channels 5-13, leaving just one free channel (1, 2, 3 or 4)

5Ghz WiFi adds another 9 to 25 channels



- In the 5Ghz band, 9 channels are dedicated to WiFi (red) and 16 more can be used during times when weather and military radar is not active (DFS/green).
- 802.11ac merges channels (perhaps all 9 red ones above) to get up to 1.3Gbps
- For more details, see: https://spectrum.ieee.org/telecom/wireless/why-wifi-stinksand-how-to-fix-it

Wi-Fi Scanner Tool Demo

- Option-click wifi icon in tray
- Open wireless diagnostics
- Window → Scan

Random Access Protocols

- Abandons strict partitioning of so that individual sender can operate faster.
- Anyone can try sending immediately at full bitrate.
- In the event of collision, wait a **random delay** before retransmitting.

  Host 1: send *(collision)* ………send *(success)*

  Host 2: send *(collision)* ……………………… send *(success)*

- Randomization will likely cause the two hosts to retry at different moments in the future:
  - What happens if we foolishly use a constant retransmission delay?

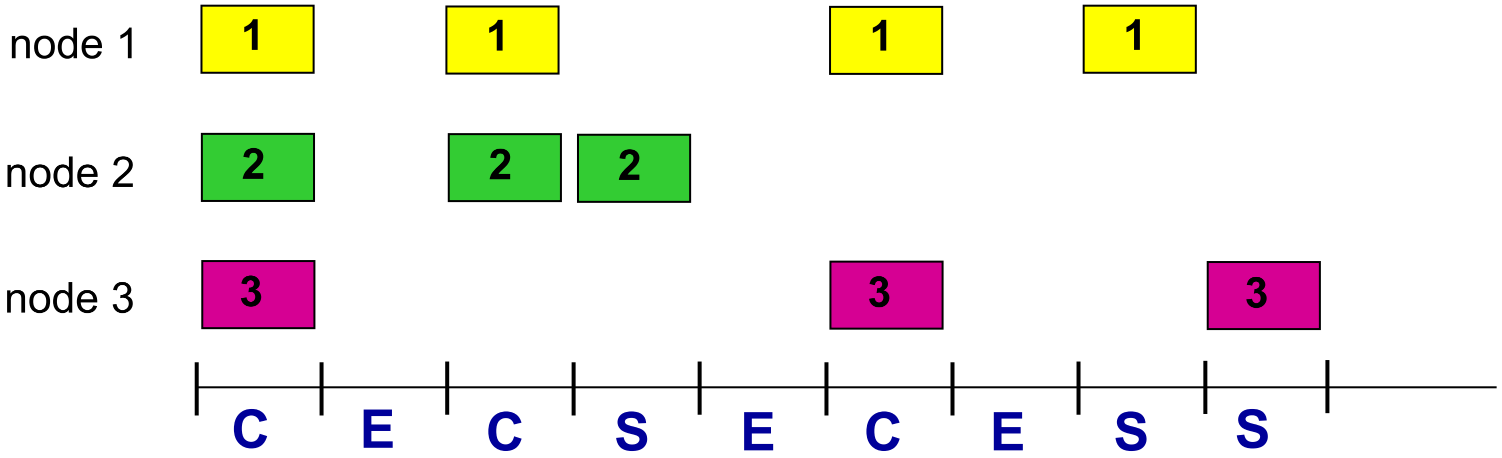
  

  Host 1: send *(collision)* …… send *(collision)* …… send *(collision)* …
  Host 2: send *(collision)* …… send *(collision)* …… send *(collision)* …

- But, how to choose the random delay?

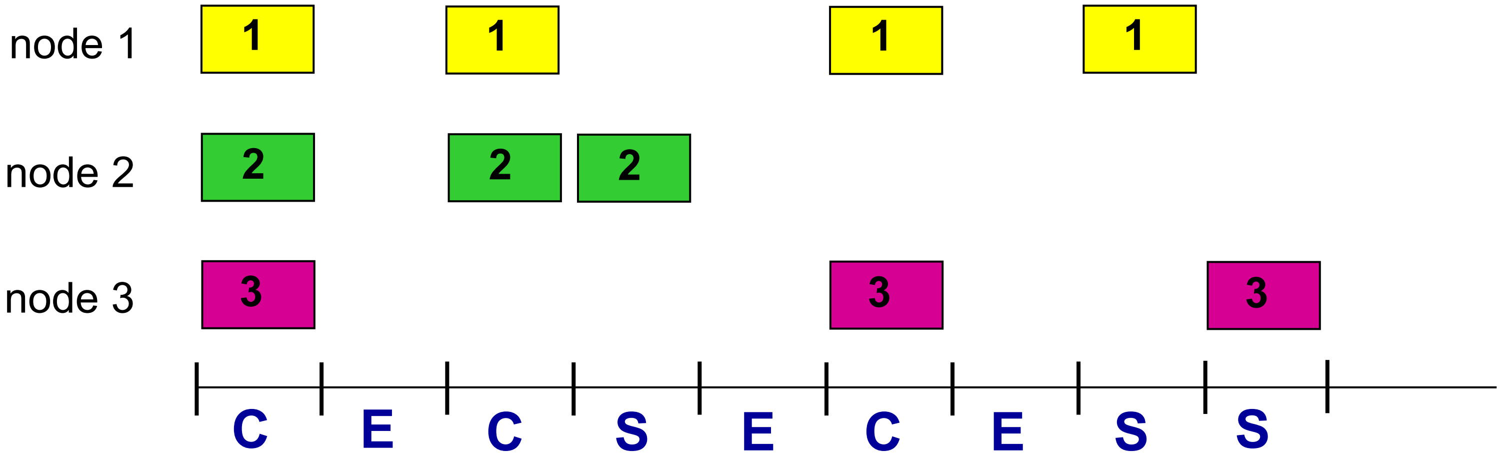
ALOHA: an early random access protocol

- Invented in 1970, for radio communication between Hawaiian islands.
- Send immediately. If collision, retransmit the packet with probability *p* or wait (with probability *p-1)* for the time needed to send one packet.
- Keep trying until the packet is sent.
- Senders have no collision detection. Receiver uses checksum to drop corrupted packets. Retry if no ACK was received.
- Don’t listen before broadcasting – just assume channel is free



ALOHA: expected throughput

- Delay probability p should be tuned according to the expected number of concurrent senders. (Lower if channel is busier.)
- If the channel is busy, then on average we expect:
  - 18% of the peak throughput. (see book for formula derivation)
- **Slotted** ALOHA requires time synchronization among senders.
  - Achieves 37% of the peak throughput. (sent packet blocks one slot, not two)



ALOHA

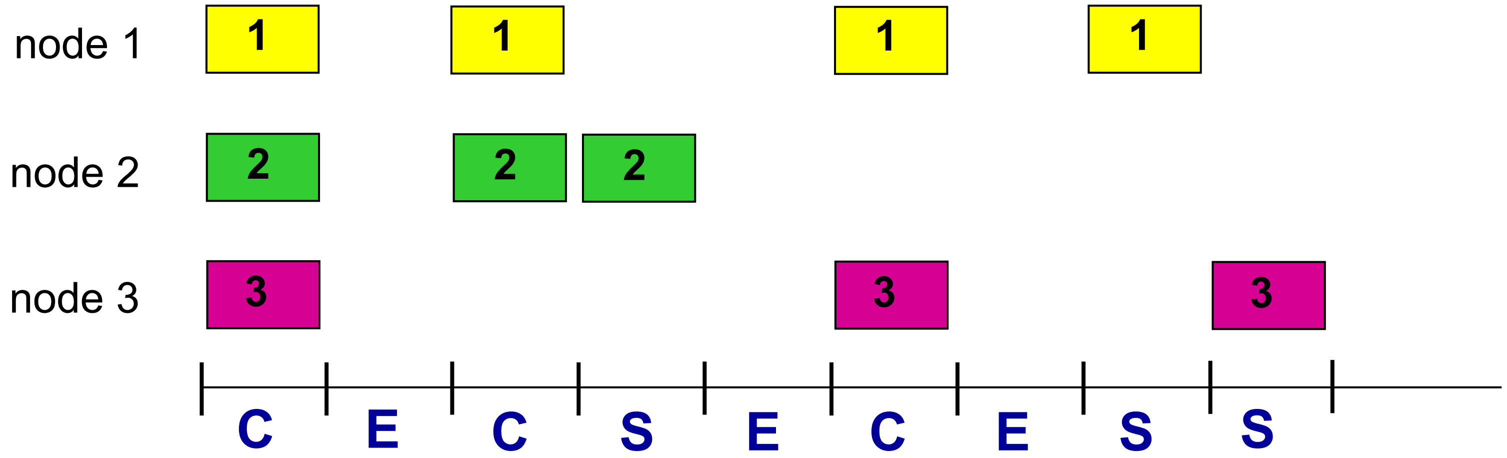
Ideas for improvement?

Pros

- One sender can use full bitrate
- Decentralized
- Simple

Cons

- Collisions are possible
- Link may be idle while waiting
- May interrupt another sender simply because didn't listen first.
- Poor throughput when busy



CSMA/CD

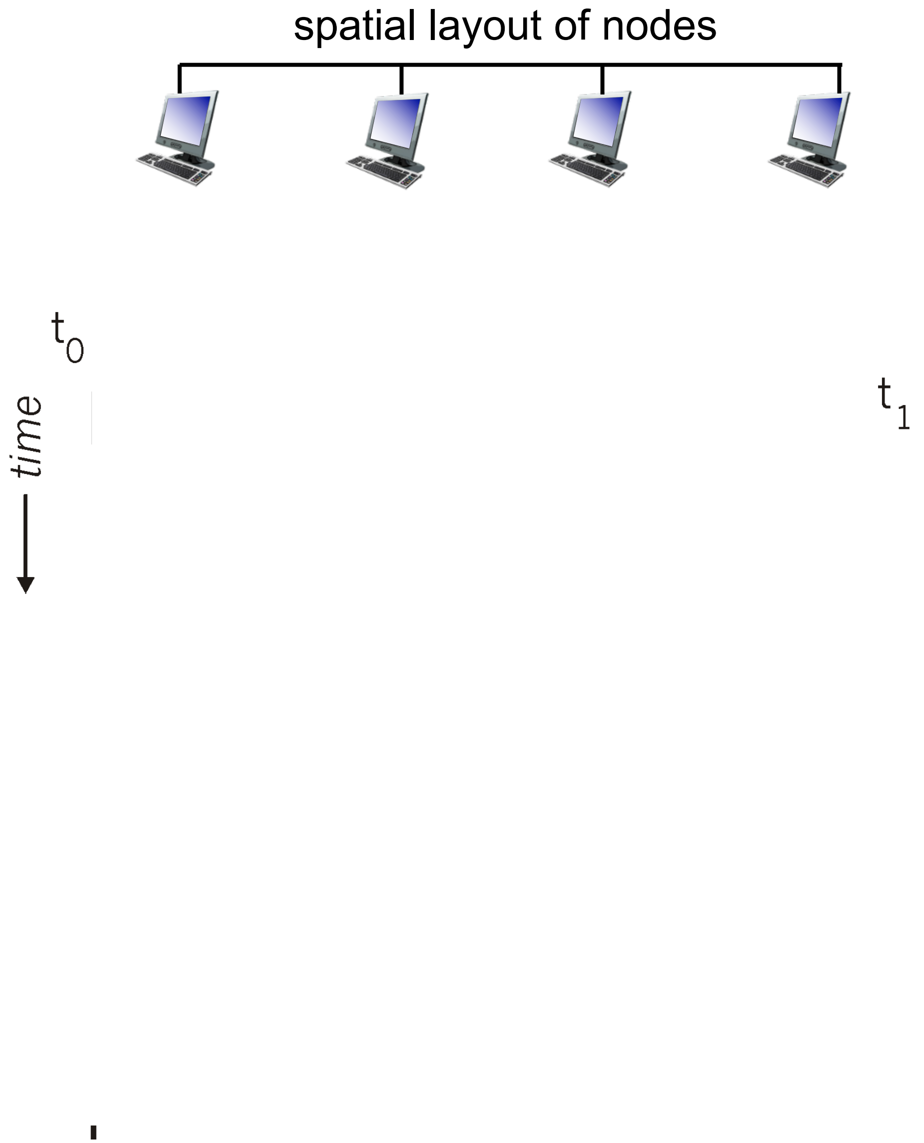
Carrier Sense Multiple Access with Collision Detection

- **Carrier Sensing:** listen to the channel before sending,
  - If it's busy, then wait. This anticipates and prevents collision.
- **Collision Detection:** listen while transmitting,
  - Stop transmission immediately if another transmission is heard.
  - This minimizes the channel-time wasted due to collision.
- Requires hardware that can detect signals on the transmission channel.
- Used by wired Ethernet *(very effectively)* and 802.11 WiFi *(less effectively)*.

Why does Collision Detection work better for wired links?

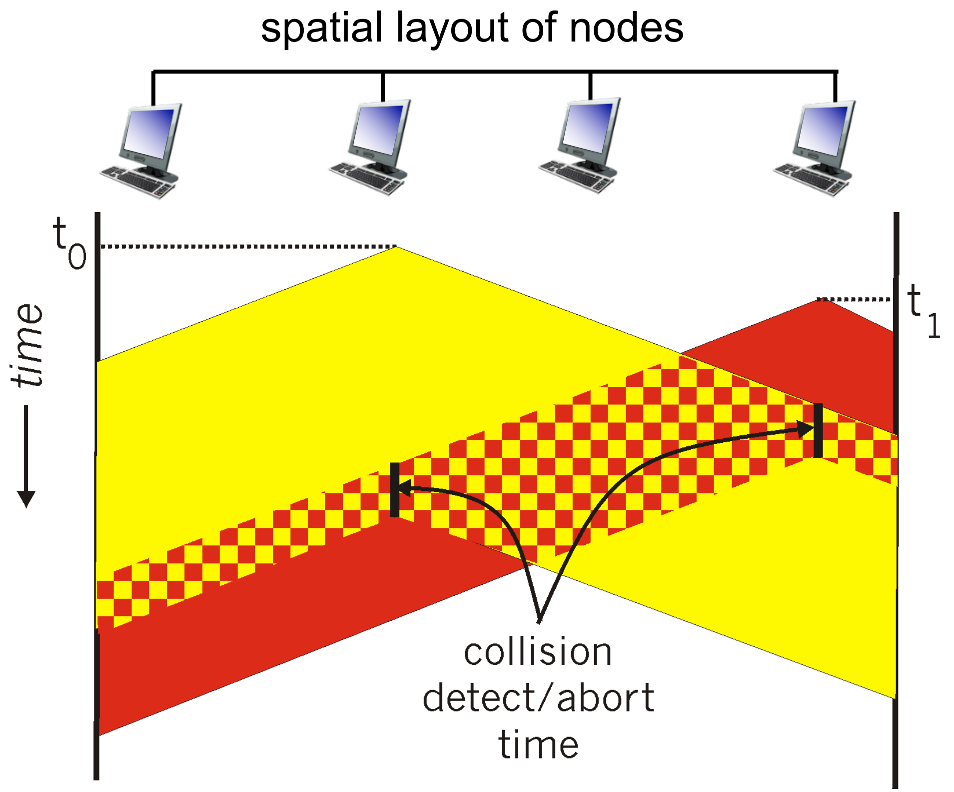
Collisions are unavoidable

- You might think that carrier sensing is enough to prevent collisions.
- However, *propagation delay* across the channel delays carrier sensing observations.
  - Also, there is a small delay between sensing and transmission.
- A node's knowledge of channel state is always slightly out-of-date.
- Longer propagation delay leads to more collision in CSMA.



Collision Detection

- Reduces the channel-time wasted by collisions.
- How?
  - Measure channel signal while sending.
  - If energy is greater than transmission energy, then there must be some added signal.
- Works well for wired channels.
- Received radio signals are much weaker than transmitted signals, so collision detection is difficult for wireless links.



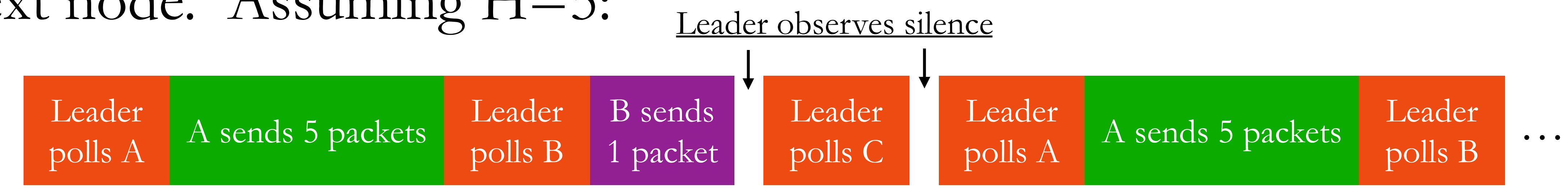
CSMA/CD steps

- **Carrier Sensing:** listen to the channel before sending,
  - If it's busy, then wait. Start sending when channel is free.
- **Collision Detection:** listen while transmitting,
  - Stop transmission immediately if another transmission is heard.
- **Binary exponential backoff** determines when transmission is retried.
  - If packet has collided n times, choose a **random** delay between 0 and $C*(2^n-1)$ For Ethernet, the constant C=512/bitrate, and n cannot grow past 10.
  - Exponential backoff resets for each new packet.

Taking-Turns Protocols

- Random access protocols do not guarantee a fair share of bandwidth.
- Taking-Turns protocols are like TDM in the sense that time slots are reserved, but the reservations are dynamic, not pre-scheduled.

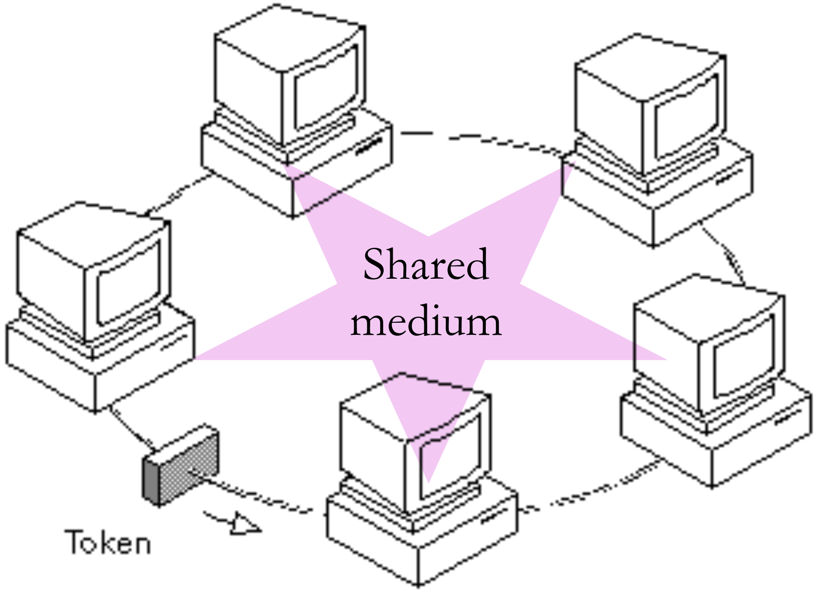
- *Bluetooth* is a type of taking-turns protocol called a **Polling Protocol***:*
  - One node is designated the **Leader**.
  - Leader **polls** the nodes in a round-robin manner, sending a message to each node telling it to send up to *H* packets.
  - Polling messages add some **coordination overhead** (C*N) where C is a constant
- If leader sees that a node stopped sending packets early, it polls the next node. Assuming H=5:



Token-passing protocols

- Like a polling protocol, but without the special leader node.
- Nodes have a designated order 1…N.
- One nodes transmits up to some maximum number of packets, then sends a special message (a **token**) giving the next node a turn.
- Nodes only send packets while "holding" the token, so collisions are avoided.

- If token-holding node crashes, the entire network is crashed. (like crashing the leader node in a polling protocol).



Multiple access protocol summary

	Channel Partitioning		Random Access		Taking-Turns	
	FDM	**TDM**	**ALOHA**	**CSMA/CD**	**Polling**	**Token-Passing**
Single Sender throughput	R/N	R/N	R	R	R – C*N	R – C*N
Busy throughput	R	R	~37% R *(slotted)* or ~18% R	$\frac{R}{1+5d_{prop}/d_{trans}}$	R – C*N	R – C*N
Collisions			yes	unlikely		
Centralized	yes	yes			yes	
Crash-sensitive					yes	yes
Requires time synchronization		yes	optional			
Requires carrier sensing				yes	yes	

Recap

- Link-layer handles sharing a physical link/medium with multiple nodes.
- Medium Access Control / Multiple Access Protocol
  - Decide how to share the link.
  - Two nodes sending simultaneously is a **collision.** Packets are lost.

Three classes of sharing protocols:

- **Channel Partitioning**:
  - Frequency Division Multiplexing - *WiFi*
  - Time Division Multiplexing
- **Turn-Taking**:
  - Polling - *Bluetooth*
  - Token-passing
- **Random Access**:
  - ALOHA (simple historical example)
  - CSMA/CD (Carrier Sense Multiple Access/Collision Detection) - *Ethernet, Wifi*