Link Layer I: Link Establishment, Medium Access Control

COS 463: Wireless Networks
Lecture 4
Kyle Jamieson

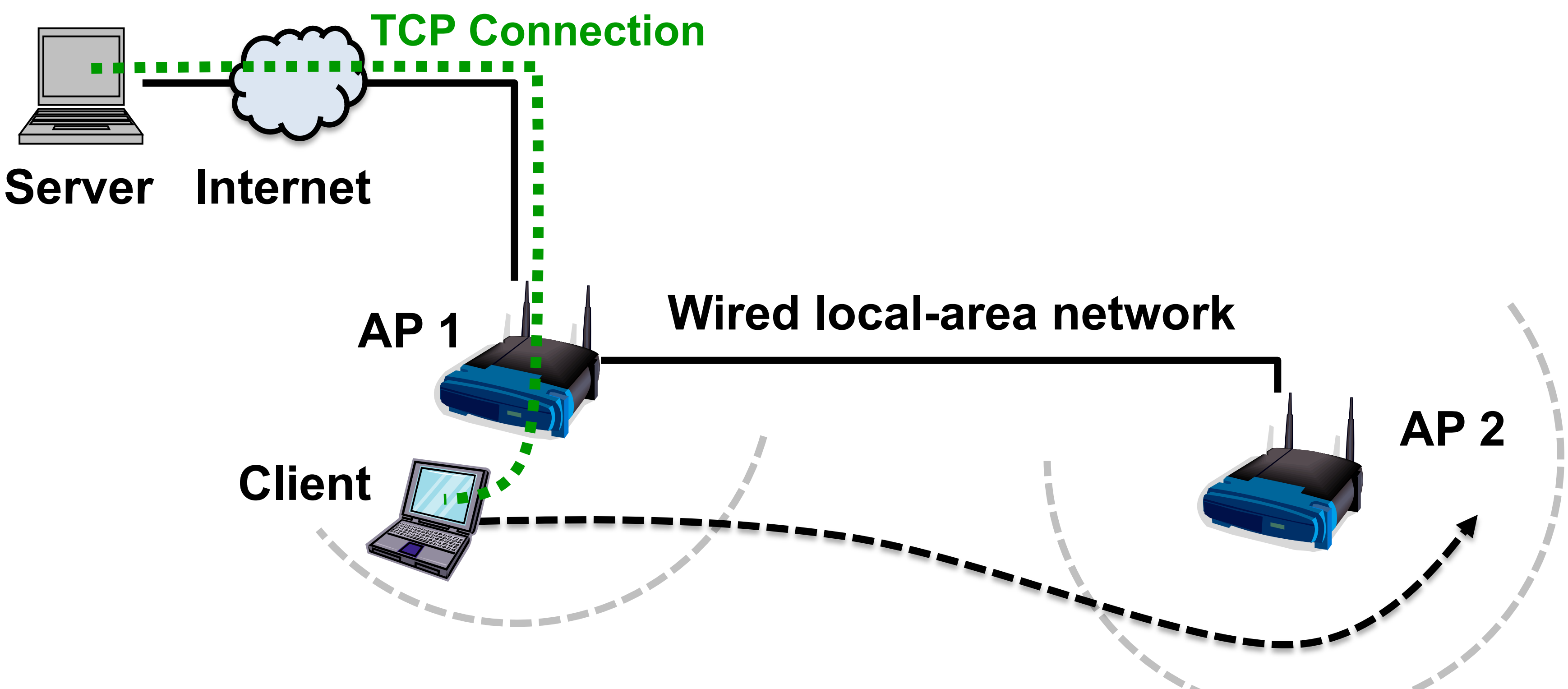
Review: The Data Link Layer (L2)

- Enables exchange of atomic messages (frames) between end hosts on the same network

- **Functions in L2:**
  - Determine start and end of bits and frames (***framing***)
  - **Establish link** and deliver information reliably
    - **Today: How Wi-Fi establishes a link**
  - **Control errors**

  - If needed, **share the medium**
    - *e.g.*, Shared-wire Ethernet, satellite uplink, Wi-Fi
    - **Today: *Medium access control* to share the medium**

Motivation: Link-Layer Handoff



How to minimize TCP segment loss, TCP timeouts?

- Client moves out of AP 1's coverage, into AP 2's coverage
- **Ongoing TCP connections** between client and server

Connecting to a Wi-Fi AP

- Notion of link-layer network: an ***Access Point*** (AP) and a **set of *connected* clients**
  - **Named** by the ***service set identifier*** (***SSID***)
  - APs generally **drop data** from clients, APs **outside the set**

*How is the wireless link **connection** established?*

1. **Discovery:** Client detects presence of AP
2. **Authentication:** Establish identity of AP, client
3. **Association:** Establish shared state between AP, client

1. Discovery

How do clients find access points, and vice-versa?

- **Access Points (APs)** send short ***beacon*** frames every 100 milliseconds

- **Clients scan** to **discover the AP.** Two ways:

1. ***Passive scan:*** Switch channels, **listen** for beacons, **wait, repeat on another channel**

2. ***Active scan:*** Send packets to **probe** for the AP's presence

Discovery: Active scan

- **Don't want to wait** *ca.* 100 milliseconds for the next beacon from an AP that may or may not be present

- **Active scan** protocol: **On each channel:**

1. **Client** broadcasts ***probe request*** frame

2. **AP responds** with ***probe response*** frame containing its SSID (network name), **data rates** supported
   - Multiple APs may respond

3. **Clients chooses** AP to continue with

2. Authentication

- AP **establishes its identity** to the client, **and vice-versa**
  - A security problem!

1. ***Open system* authentication:** Trivial, client sends authentication frame, AP responds “success”

2. ***Shared key* authentication:** Configure both client and AP with a **shared secret key**
   - **Doesn’t scale** too well

3. ***Enterprise* authentication:** Use public key certificates, akin to web site authentication

3. Association

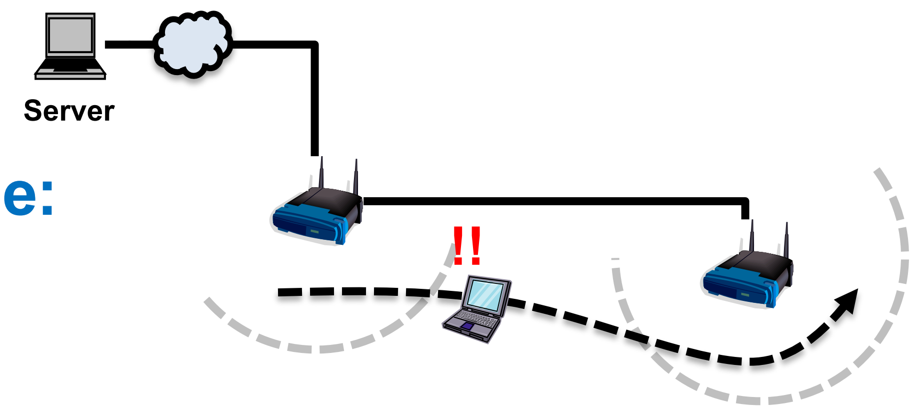
- The “commit” step that establishes shared AP-client state

1. Client sends ***association request*** frame to AP

2. AP sends ***association response*** frame to client

- **Data now may flow,** both to the AP (***uplink***) and to the client (***downlink***)

Initiating The Wi-Fi Handoff

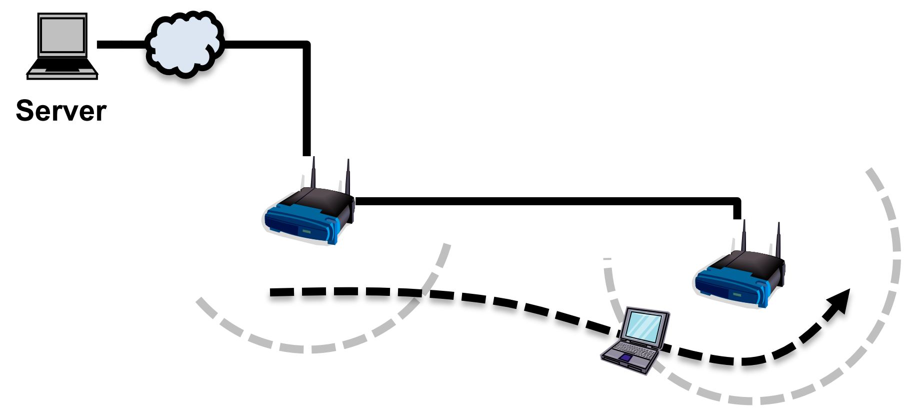


- Back to the **handoff example:**

How is the handoff initiated?

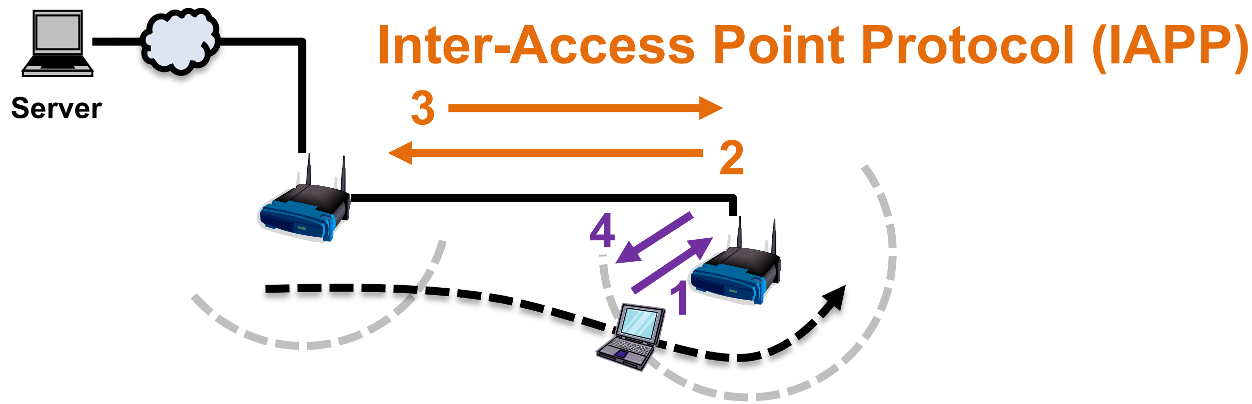
- Client tracks **received signal strength** from AP's frames
  - Client initiates handoff if and when signal strength **falls below a threshold**

Wi-Fi Handoff Process: High-Level View



- **Discovery step**
  - Same as before
- **Authentication step**
  - Same as before
- ***Reassociation step***
  - Replaces **association step** of the connection process discussed before

Wi-Fi Handoff: Reassociation



1. **Reassociation Request:** Client asks new AP to connect
   – Supplies old AP identifier
2. **IAPP Move request:** New AP asks for old AP's state
3. **IAPP Move response:** Old AP supplies state to new AP
4. **Reassociation Response:** "Commit" step, data may flow

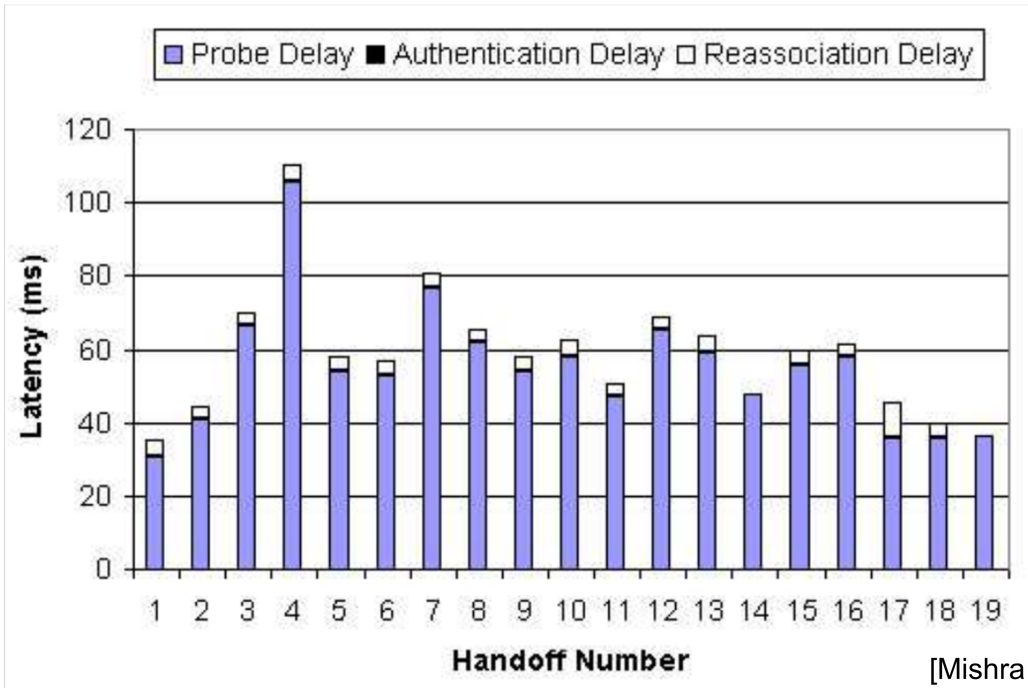
Reassociation: Latency

- ***For how long** is the client's link-layer connection **interrupted?***
  - This time duration is the ***reassociation latency***

- **Beginning:** frames get dropped from old AP
  - **Imprecise:** Link-layer retransmissions recover some losses

- **End:** Reassociation protocol completes with new AP
  - **Precise:** Reassociation response message received

802.11 Reassociation Performance



[Mishra et al., 2003]

- Conventional 802.11 reassociation takes *ca.* 40 to 100 ms
  - Long enough to trigger TCP **duplicate ACKs, timeouts**

Improvements to Wi-Fi Handoff

- Wi-Fi standard 802.11r: **Fast Roaming**
  - Store encryption keys on all APs in the network
    - So **no need** for client to perform **complete authentication process** on reassociation

- Wi-Fi Standard 802.11k: **Assisted Roaming**
  - AP tells client a list of nearby other APs and their channels
    - So **no need** for the client to **scan**

Medium Access Control

1. **Sharing by partitioning the medium**
   – **Introduction, Time and Frequency division**
   – Code division

2. Contention-based sharing
   – ALOHA
   – The Ethernet

Medium access: The Problem

- **Two questions:**
  1. How should the shared medium be divided?
  2. Who gets to talk on a shared medium, and when?

- A **medium access control (MAC)** protocol specifies the above

Medium access: Metrics of Success

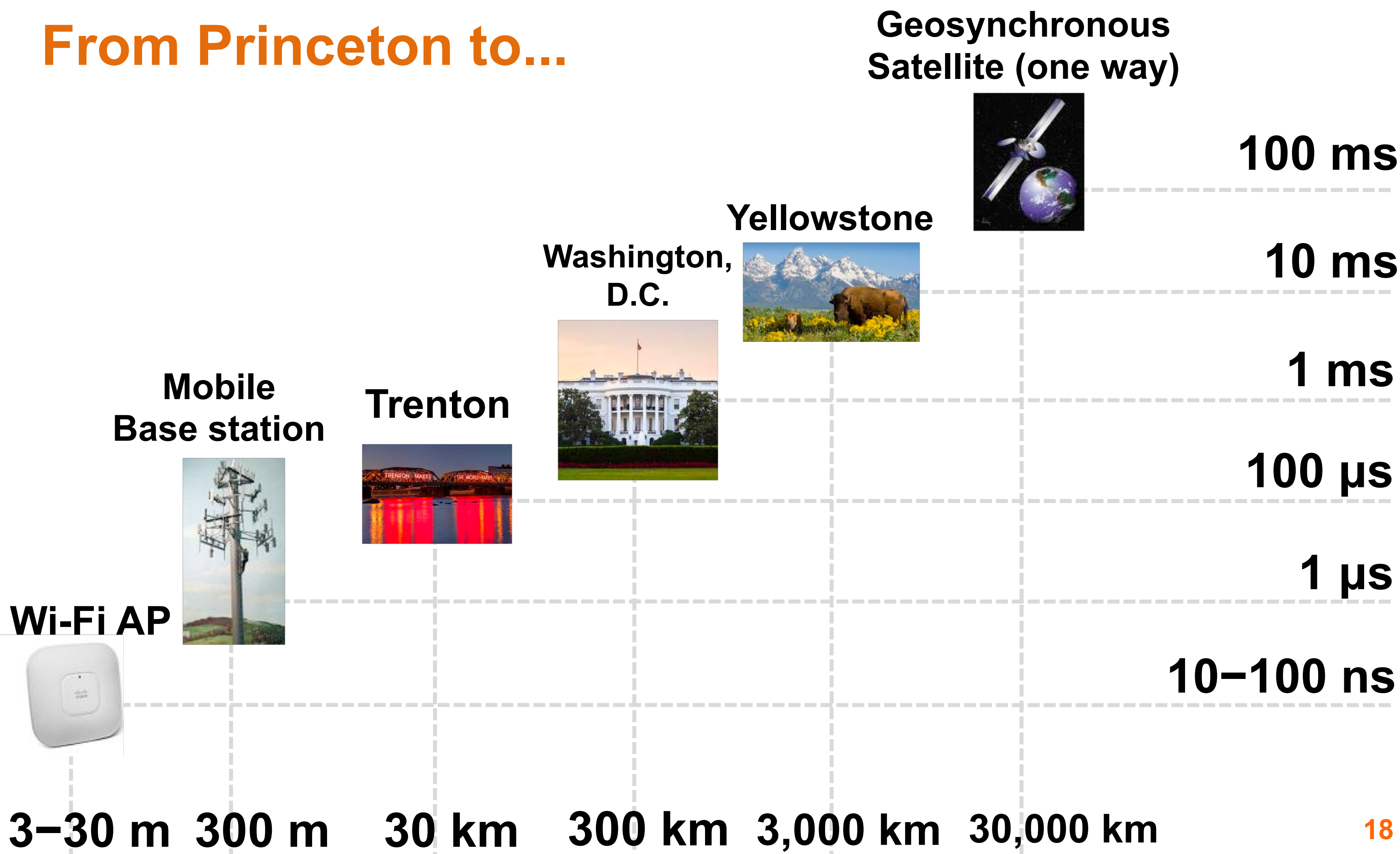
1. **Efficiency**
   – High ***throughput*** (bits/second successfully **received**)
     - *i.e.* high ***utilization*** (throughput / raw channel rate)

2. **Fairness:** All hosts with data to send should get a roughly equal share of the medium over time

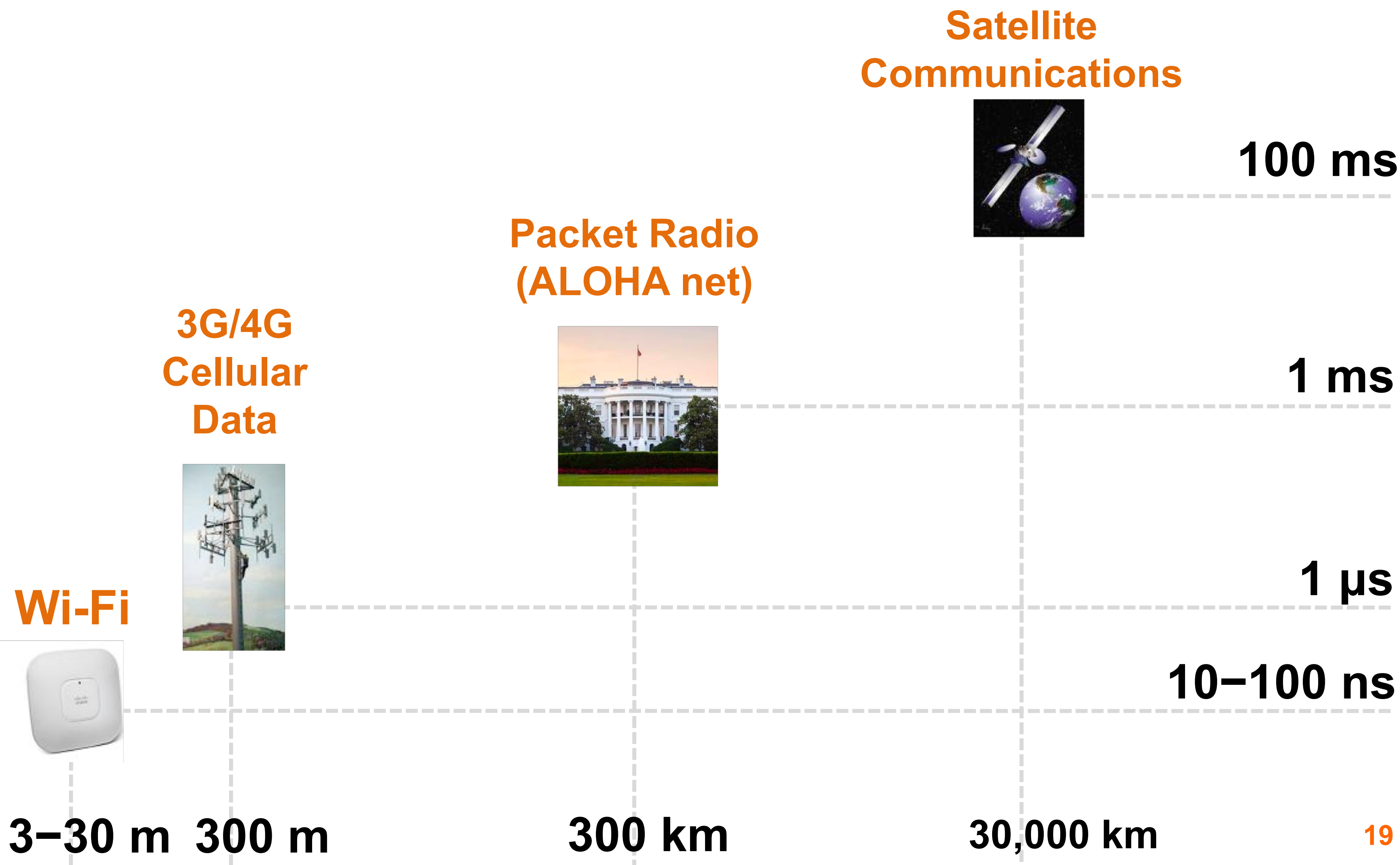
3. **Latency:** Want to minimize the time a host waits before being granted permission to talk on the shared medium

Physical Limitation: Finite speed of light

From Princeton to...

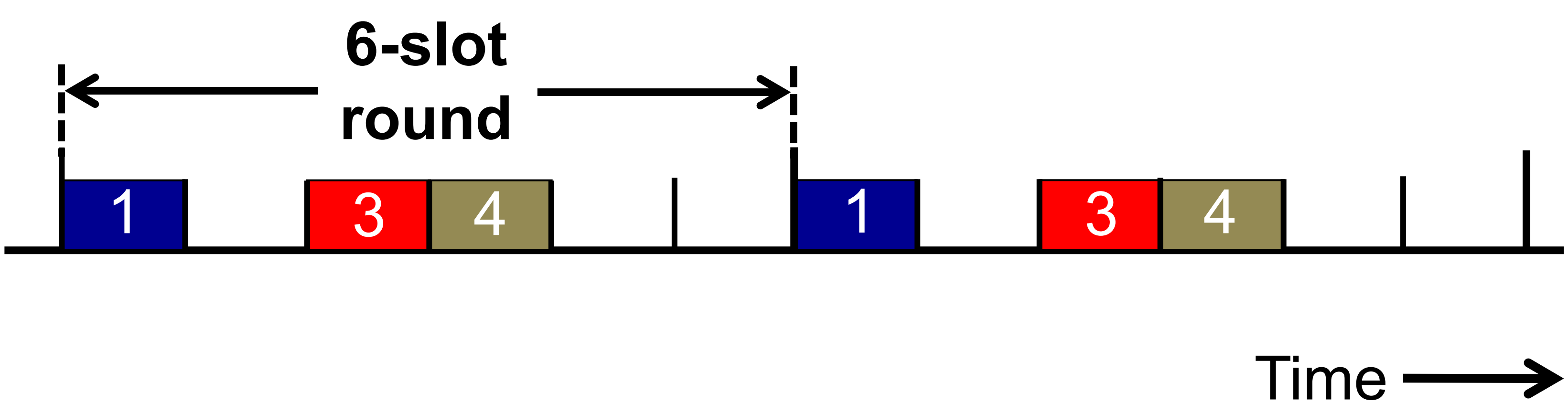


Vastly Different Timescales, Same Medium Access Protocol!



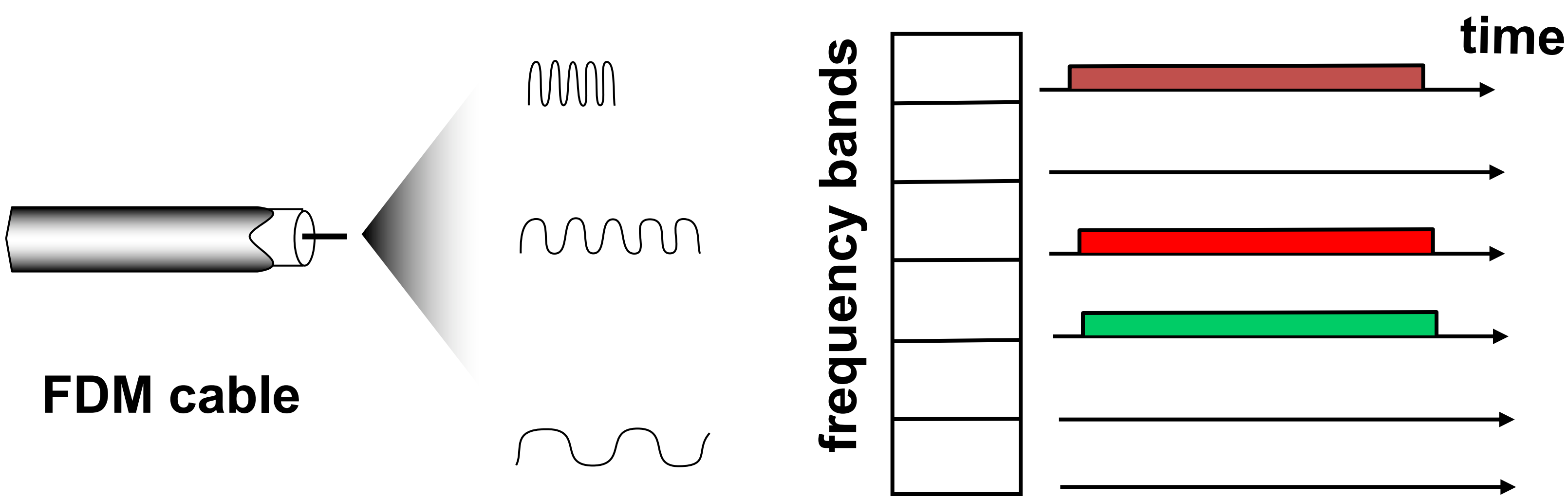
TDMA: Time Division Multiple Access

- Channel time is divided fixed-period, repeating ***rounds***
- Each user gets a fixed-length ***slot*** (packet time) in each round (**unused slots are wasted**)
- **Out-of-band:** Mechanism for allocating/de-allocating slots
- ***e.g.:*** six stations, only 1, 3, and 4 have data to send



FDMA: Frequency Division Multiple Access

- Channel spectrum divided into frequency bands
- Each user gets a fixed frequency band (**unused frequency slots are wasted**)
- ***e.g.:*** six stations, only 1, 3, and 4 have data to send



TDMA and FDMA: Considerations

- **Advantages**

1. Users are **guaranteed** to be able to send bits, continuously (FDMA) or periodically (TDMA)

- **Disadvantages**

1. Unused time slots or frequency bands **reduce channel utilization**

2. An out-of-band mechanism is needed to allocate slots or bands (which **requires another channel**)

3. Guard bands or guard times **reduce channel utilization**

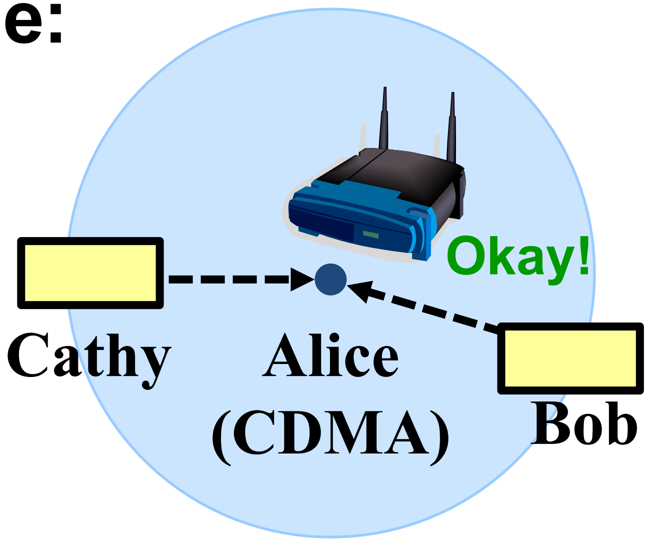
Medium Access Control

1. **Sharing by partitioning the medium**
   – Introduction, Time and Frequency division
   – **Code division**

2. Contention-based sharing
   – ALOHA
   – The Ethernet

CDMA: Code Division Multiple Access

- All users transmit over the **same frequencies,** and **at the same time:**



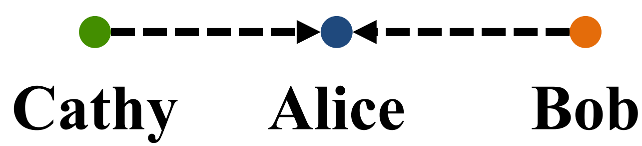
- Allows multiple users to coexist and transmit simultaneously with **no interference, in theory**
- **In practice:** also **performs well**
  - Some **cellular data networks** have used CDMA

Representing bits as binary levels

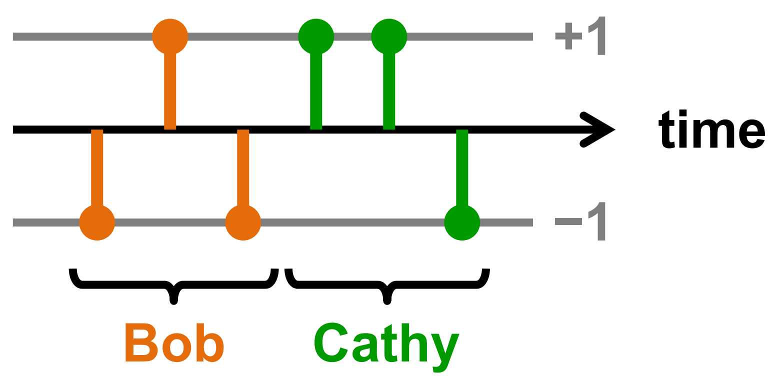
- Let's represent bits with two (binary) **levels** as follows:

  0 bit ←→ +1 level   1 bit ←→ −1 level

- **Scenario: Alice** receives data from **Bob** and **Cathy:**

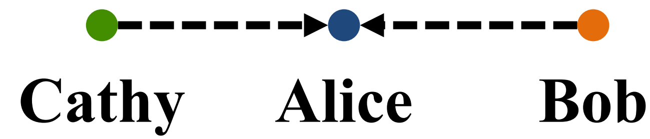


  – **TDMA *e.g.*: Bob** sends bits **101**, **Cathy** sends **001:**



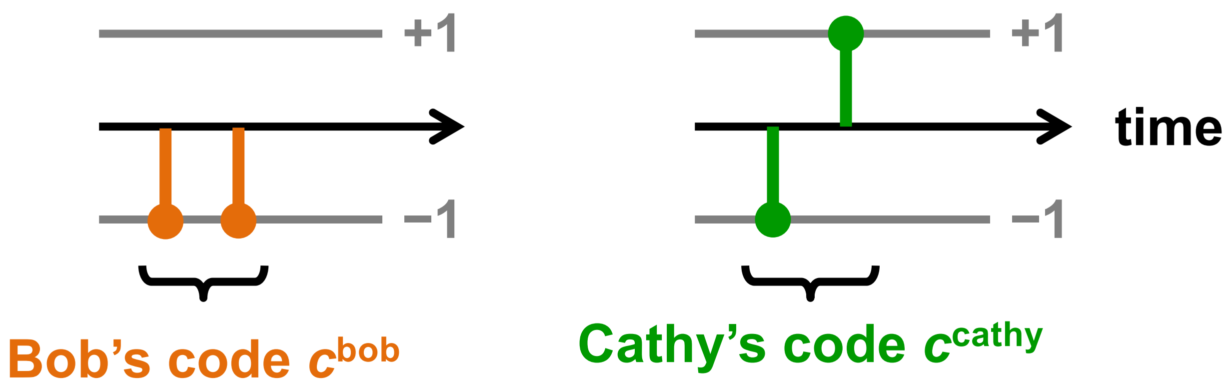
TDMA timeslots:

CDMA: User codes

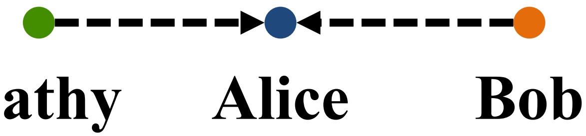


- Assign each user a unique binary sequence of bits: ***code***
  - Call each code bit a ***chip*** (convention)
  - Call the code length ***M***

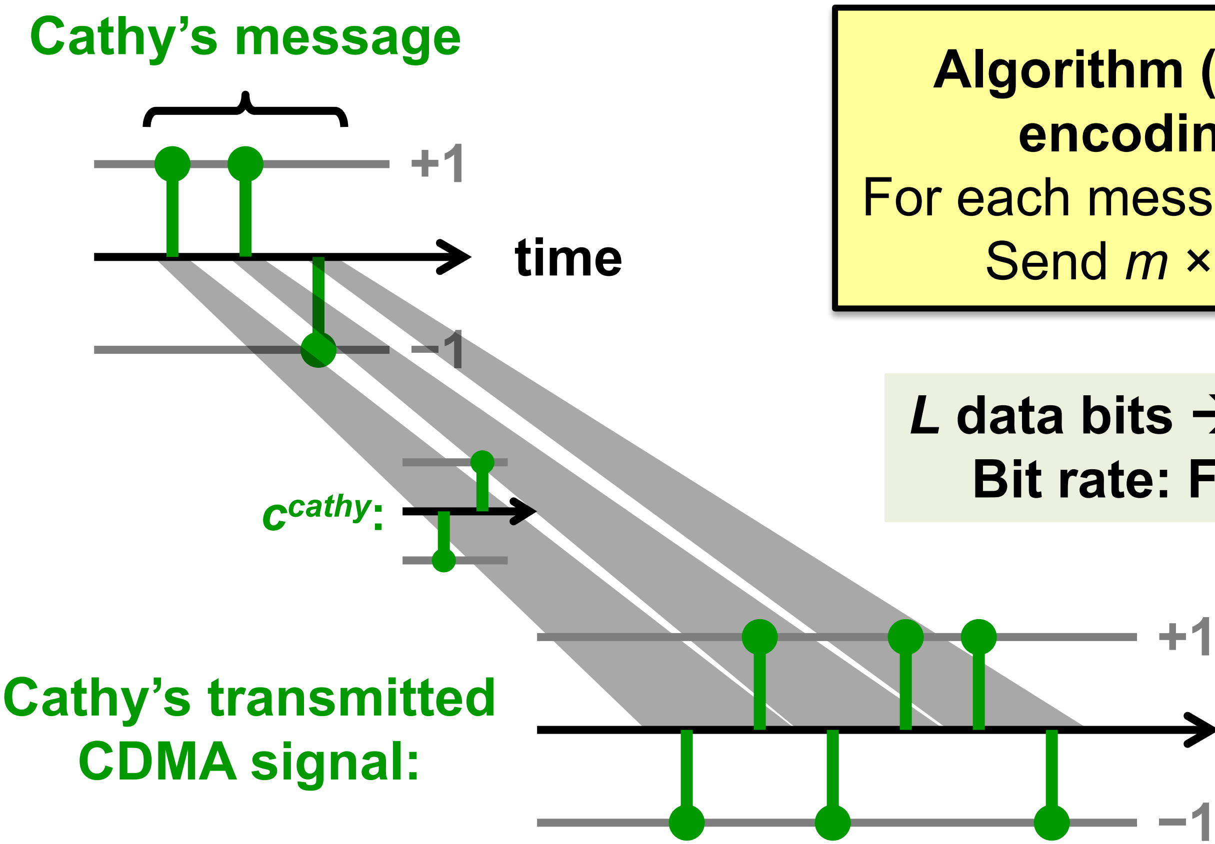
- **CDMA example:**



CDMA: Cathy Sending



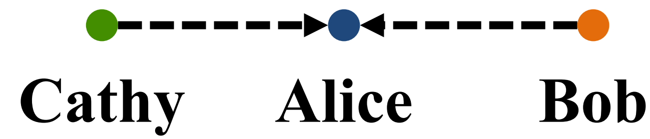
- Suppose Cathy alone sends message bits **`001`:**



Algorithm (CDMA encoding):
For each message bit m:
Send $m \times c^{user}$

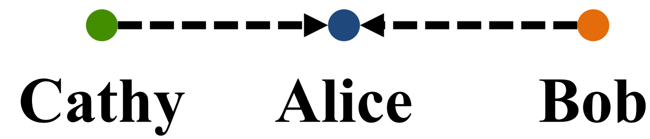
L data bits → $M \times L$ CDMA chips
Bit rate: Factor of M slower

CDMA: Assumptions

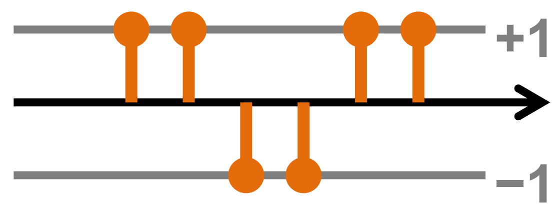


- Let's assume we have a way of:
  - Synchronizing **Cathy's** and **Bob's** **data bits** in time
  - Synchronizing **Cathy's** and **Bob's** **CDMA chips** in time
  - Estimating and correcting the **effect of the wireless channel** between **Cathy** and **Bob** to **Alice**

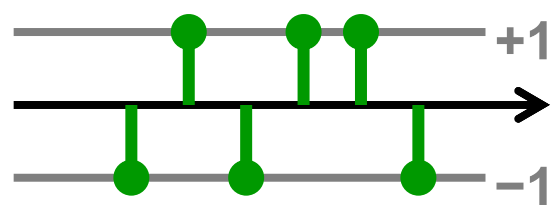
What Alice Hears



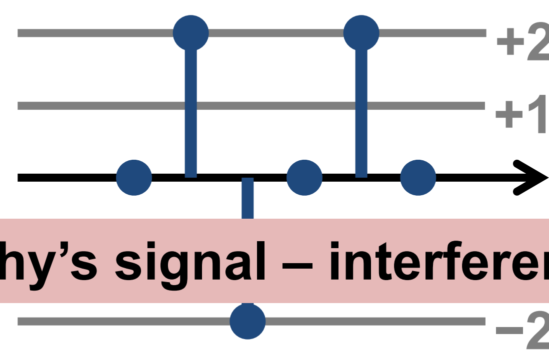
Bob's transmitted CDMA signal:



Cathy's transmitted CDMA signal:

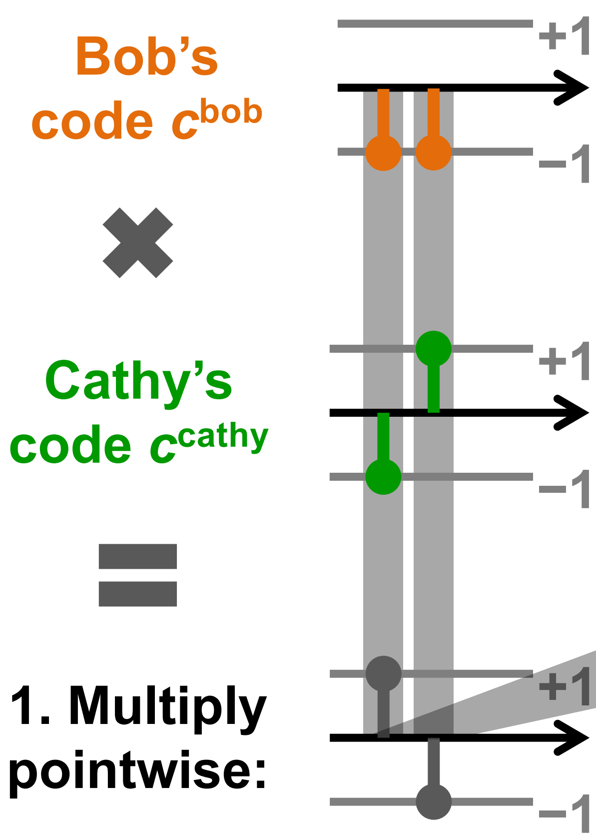


What Alice hears:



Result: Neither Bob nor Cathy's signal – interference!

Tool: Correlation



Algorithm (correlation):

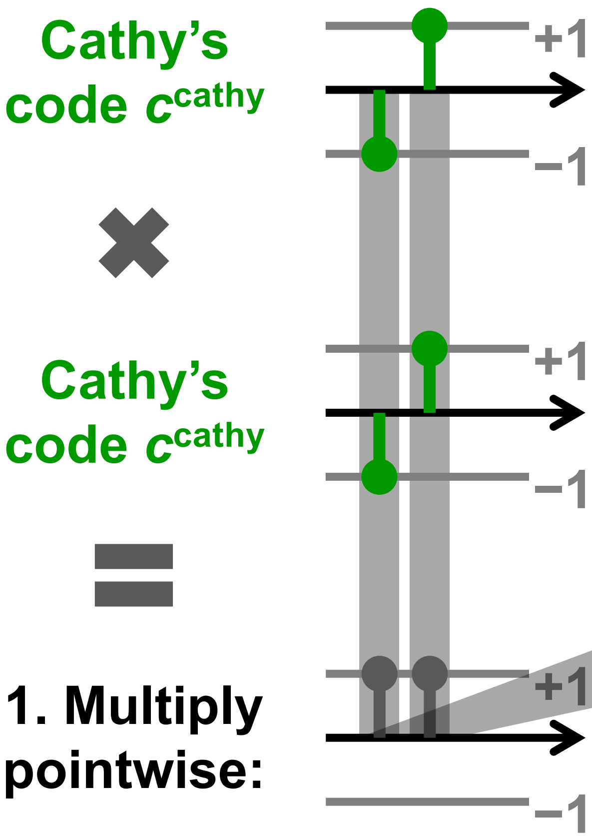
1. **Multiply** two signals **pointwise, across time**
2. **Sum** the result across time
3. **Normalize** (divide) by the signal length

Sum: 0 → Correlation: 0

2. Sum across time

3. Normalize (÷2)

Tool: Correlation



Algorithm (correlation):

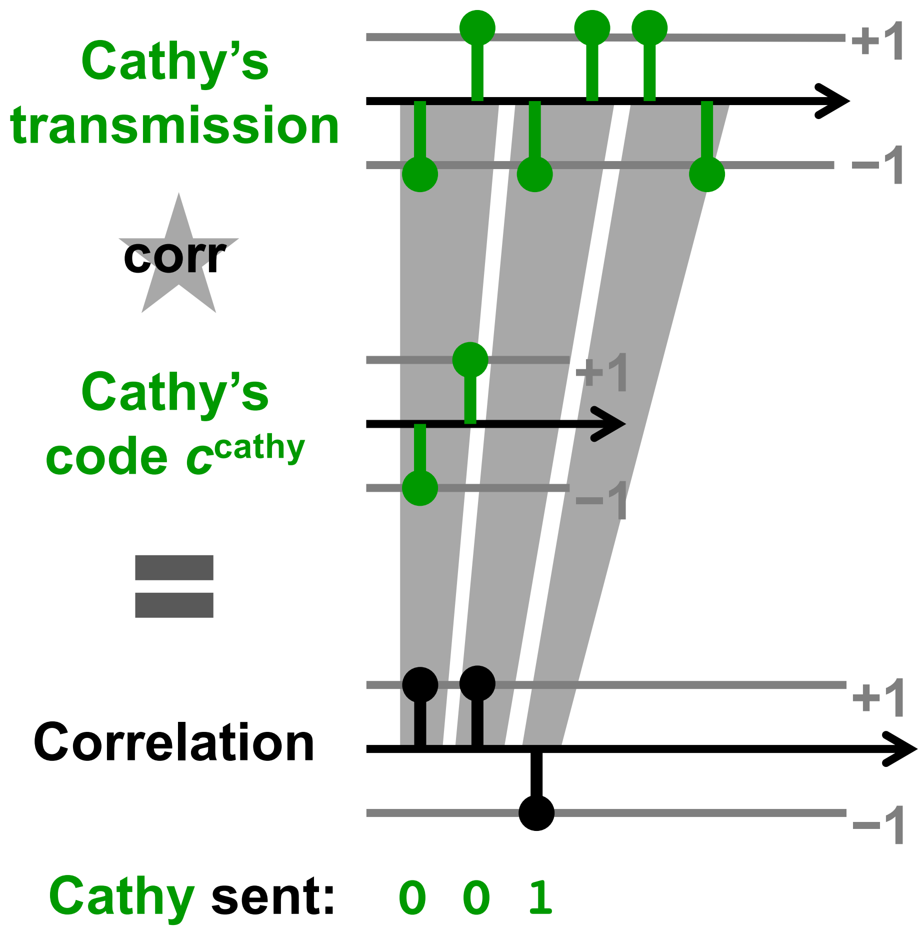
1. **Multiply** two signals **pointwise, across time**
2. **Sum** the result across time
3. **Normalize** (divide) by the signal length

Sum: 2 → <u>Correlation: 1</u>

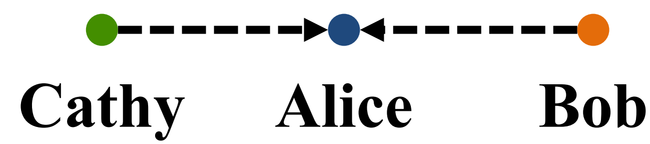
2. Sum across time

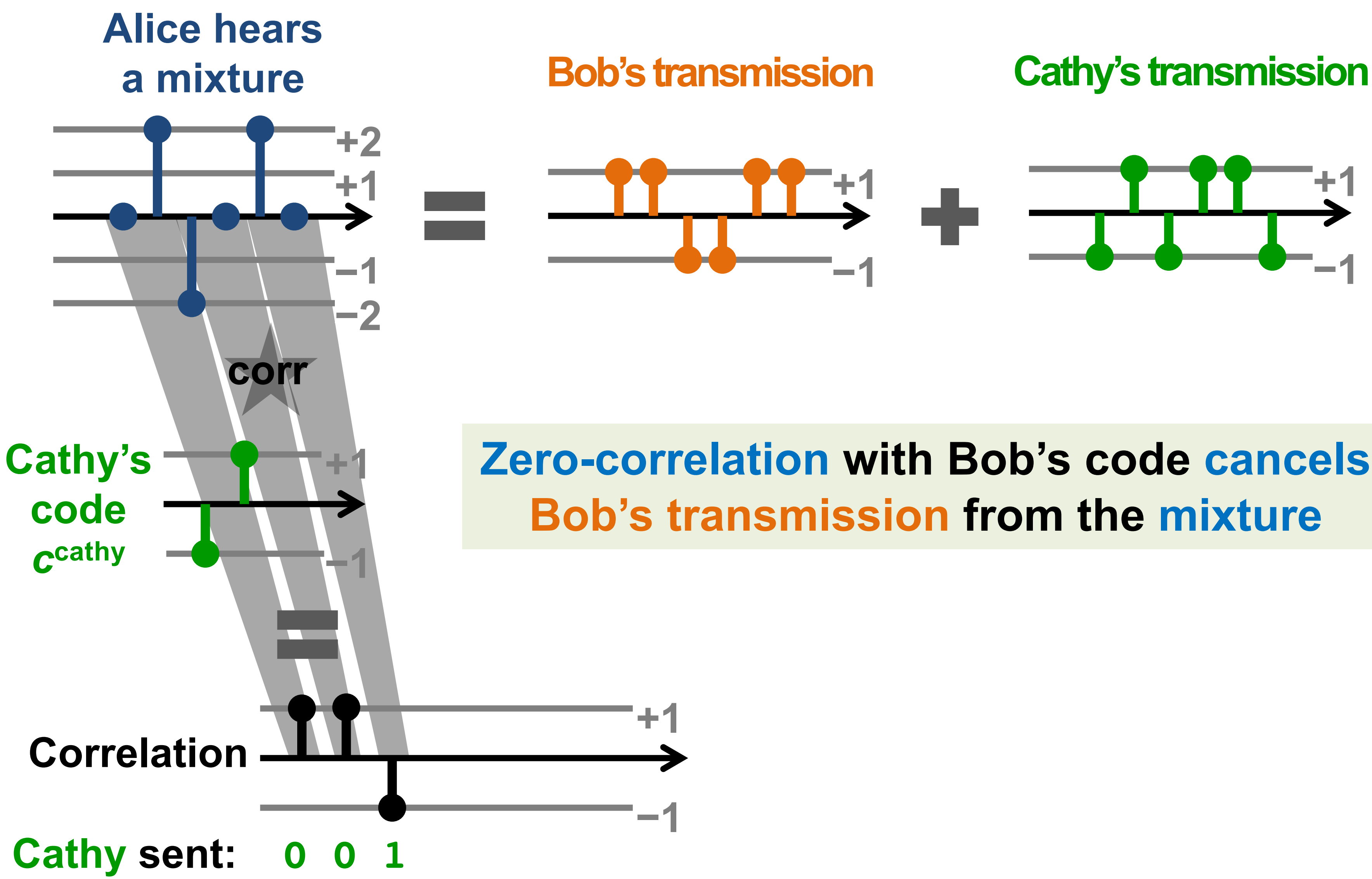
3. Normalize (÷2)

Correlating Cathy's Code with Cathy's CDMA transmission



Listening to Cathy





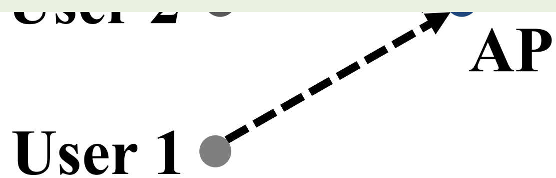
Zero-correlation with Bob's code **cancels**
Bob's transmission from the **mixture**

CDMA: How to choose codes?

- Let's **generalize** the Alice, Bob, Cathy scenario:
  - ***N* users,** each user *n* has code c_m^n, n = 1...*N*
    - (m = 1...Code length M)

Zero mutual correlation condition:

$$c_m^{n_1} \star c_m^{n_2} = 0, n_1 \neq n_2$$



- **Goal:** Ensure **cancellation of all other users** when **correlating** against (each) **one**

Example CDMA code: Walsh Codes

- Start with the **Bob** / **Cathy** code, write as **rows in a matrix**

$$\begin{bmatrix} c^{bob} \\ c^{cathy} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- **Recursive rule:** given matrix **M,** form $\begin{bmatrix} \mathbf{M} & \mathbf{M} \\ \mathbf{M} & -\mathbf{M} \end{bmatrix}$

- *e.g.* **four** users:

$$\begin{bmatrix} c^1 \\ c^2 \\ c^3 \\ c^4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

CDMA: Considerations

- CDMA **advantages:**
  - Sending over entire channel frequency bandwidth
    - Some parts of frequency band **interfered?** **Okay!**

- FDMA, TDMA, CDMA **disadvantages:**
  - Rigid allocation of channel resources, requires **advance coordination** (frequency, time, code)
  - **Partitioning** the channel → **reduced rate**

- Can we have the best of both worlds, perhaps?

Stretch Break with CDMA Calculation!

- Recall the two-user Walsh code $\begin{bmatrix} c^{bob} \\ c^{cathy} \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$, and recursive rule: given matrix **M,** form $\begin{bmatrix} \mathbf{M} & \mathbf{M} \\ \mathbf{M} & -\mathbf{M} \end{bmatrix}$ to double the number of users in the system.

What's the **second user's** Walsh code
in an **eight-user** CDMA system?

Medium Access Control

1. Sharing by partitioning the medium
   – Introduction, Time and Frequency division
   – Code division

2. **Contention-based sharing**
   – **Unslotted ALOHA,** Slotted ALOHA
   – The Ethernet

Contention-based sharing

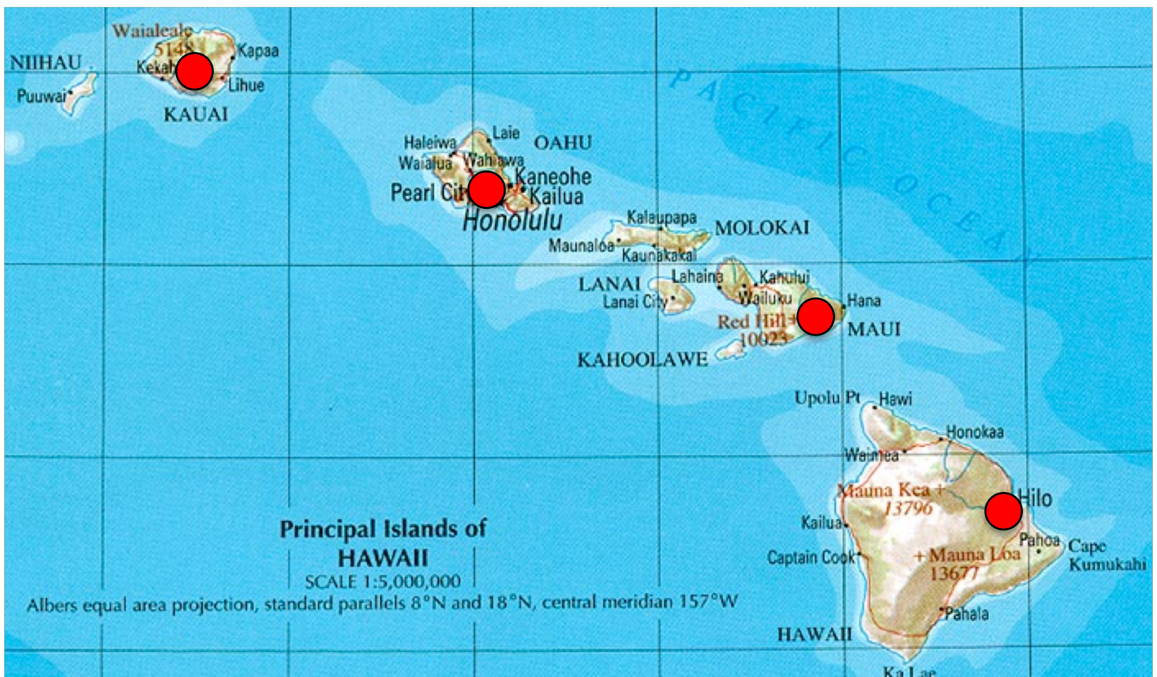
- When a station has a frame to send:
  - Transmit at **full channel data rate** *B*
  - No *a priori* coordination among nodes

- Two or more frames overlapping in time: **collision**
  - **Both frames lost,** resulting in **diminished throughput**

- A random access MAC protocol specifies:
  - How to detect collisions
  - How to recover from collisions

ALOHAnet: Context

- Norm Abramson, 1970 at the University of Hawaii
  - Seven campuses, on four islands
  - Wanted to **connect** campus terminals and mainframe
  - Telephone costs high, so built a **packet radio network**



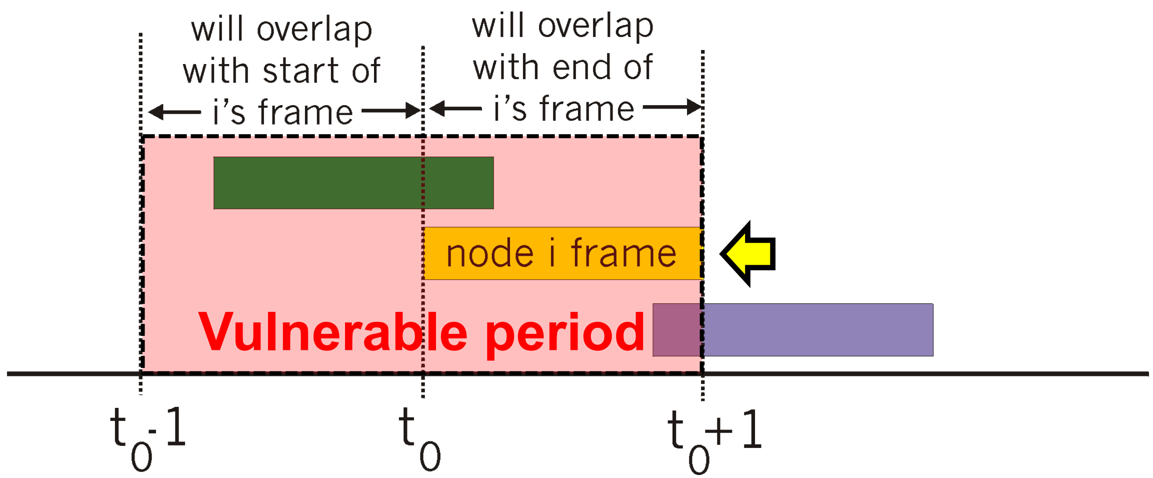
Unslotted ALOHA

- **Simplest possible medium access control:** no control at all, anyone can just transmit a packet without delay

- **Suppose:** Chance packet **begins in time interval Δt is $\lambda \times \Delta t$**
  - ***N*** senders in total, sending frames of time duration **1**
- λ is the **aggregate rate** from **all *N* senders**
- ***Individual rate*** **λ/N** for **each sender**

Unslotted ALOHA: Performance

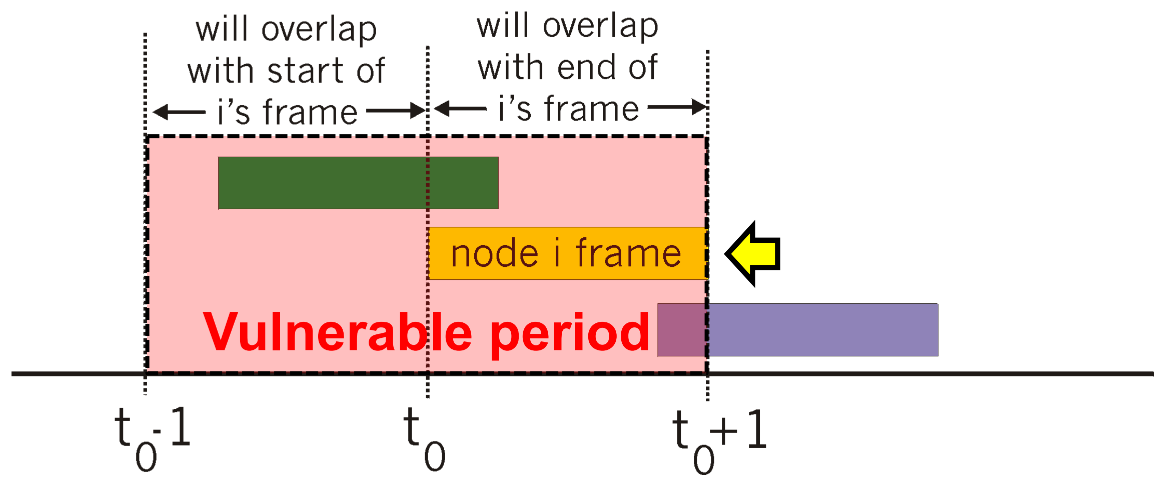
- Suppose some node ***i*** is transmitting; let's focus on ***i* 's frame**



I. Others send in [t_0−1, t_0]: overlap *i* 's frame **start** → **collision**
II. Others send in [t_0, t_0+1]: overlap *i* 's frame **end** → **collision**
III. Otherwise, **no collision**, node *i* 's frame is delivered

- Therefore, ***vulnerable period*** of length **2** around *i* 's frame

Unslotted ALOHA: Performance



- *What's the chance no one else sends in the vulnerable period (length 2)?*

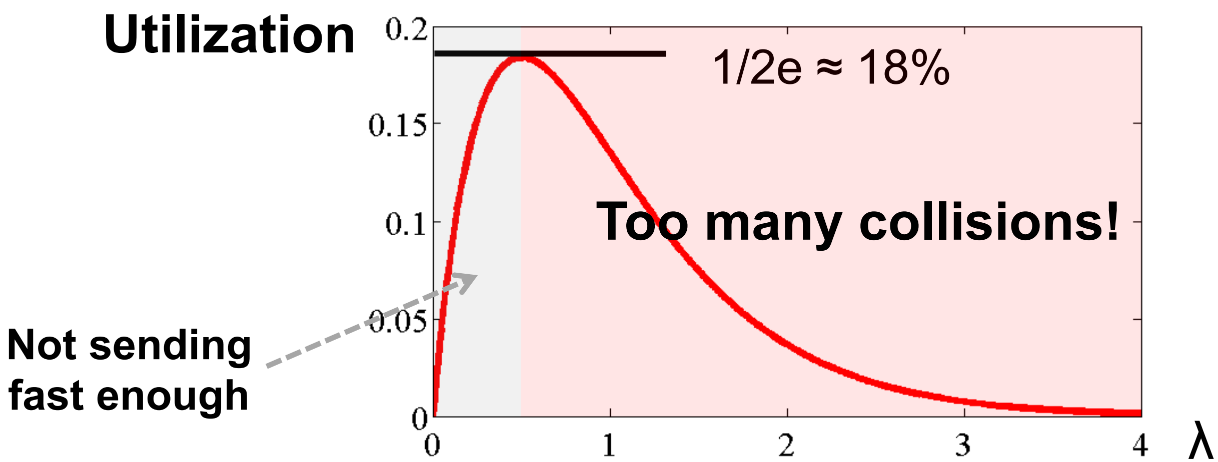
$$\Pr(\text{no send from } \textit{one} \text{ node in } 2) = 1 - \frac{2\lambda}{N}$$

$$\Pr(\text{no send } \textit{at all} \text{ in } 2) = \left(1 - \frac{2\lambda}{N}\right)^{N-1}$$

$$\lim_{N \to \infty} \left(1 - \frac{2\lambda}{N}\right)^{N-1} \rightarrow e^{-2\lambda}$$

$$\lim_{x \to \infty} \left(1 + \frac{1}{x}\right)^{x} = e$$

Unslotted ALOHA: Utilization

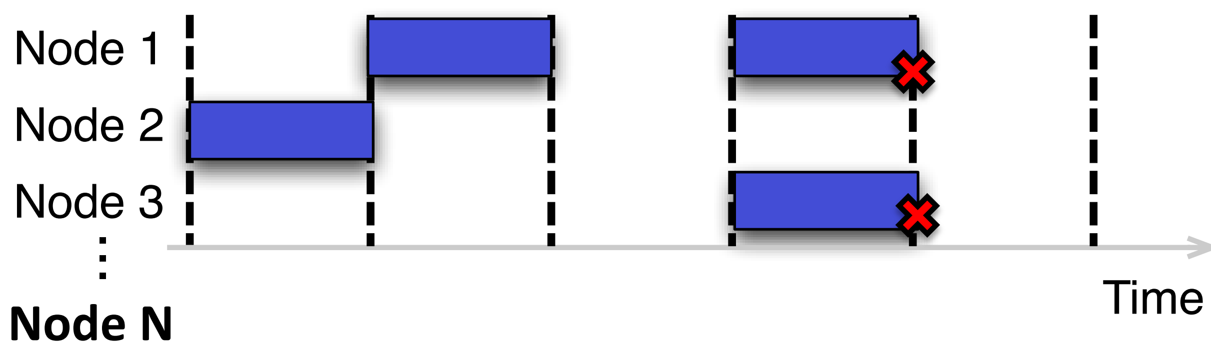


- Recall ***λ*** is the **aggregate rate** from **all senders**
- So, utilization = λ × Pr(no other transmission in 2)

  = $\lambda e^{-2\lambda}$

Slotted ALOHA

- **Divide time into slots** of duration 1, **synchronize** so that nodes transmit **only** in a slot
  - Each of ***N* nodes** transmits with probability ***p*** in each slot
  - So **aggregate transmission rate λ** = $N \times p$
- As before, if **exactly one** transmission in slot, **can receive;** if **two or more** in slot, **no one can receive** (**collision**)

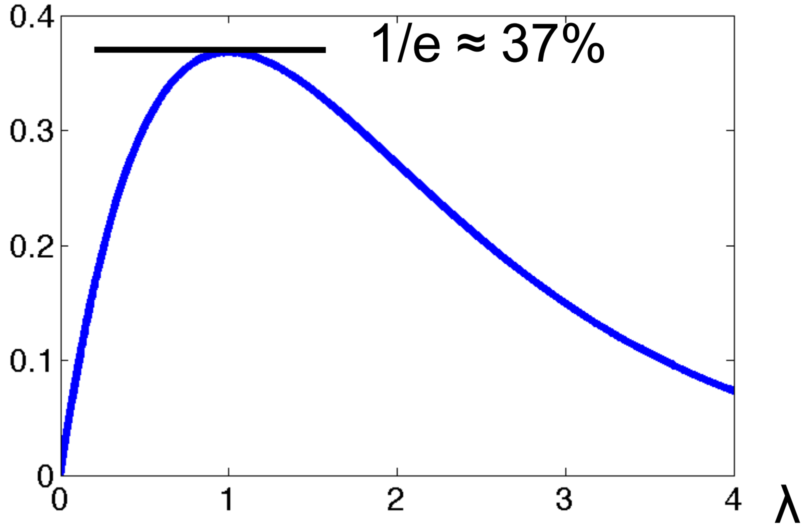


Slotted ALOHA: Utilization
(*N* nodes, each transmits with probability *p* in each slot)

What is the utilization as a function of aggregate rate $\lambda = N \times p$?

- Pr[A node is successful in a slot] = $p(1-p)^{N-1}$
- Pr[Success in a slot] = $Np(1-p)^{N-1}$

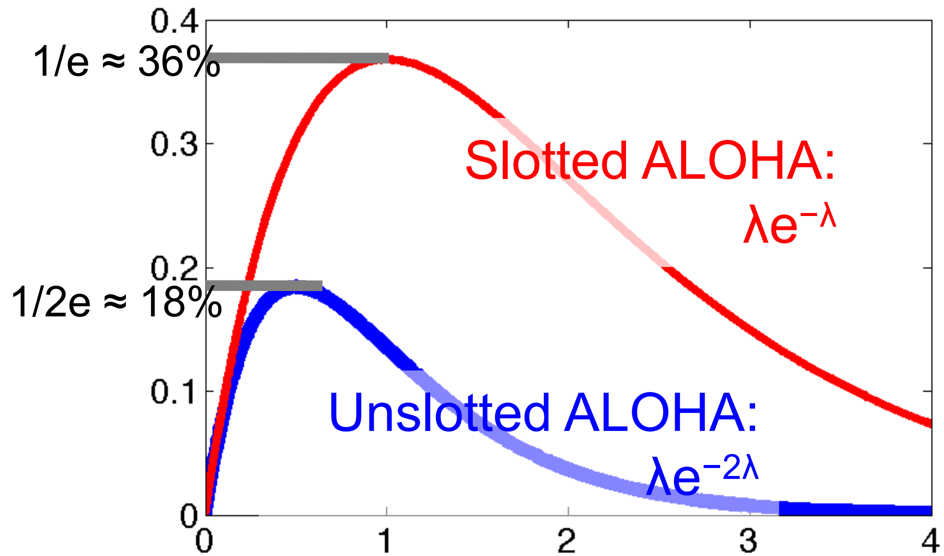
Utilization:
$\lambda e^{-\lambda}$



$$\Pr(\text{success}) = \lambda\left(1-\frac{\lambda}{N}\right)^{N-1}$$

$$\lim_{N\to\infty} \lambda\left(1-\frac{\lambda}{N}\right)^{N-1} = \lambda e^{-\lambda}$$

ALOHA throughput: slotted versus unslotted



Just by forcing nodes to transmit on slot boundaries, we double peak medium utilization!

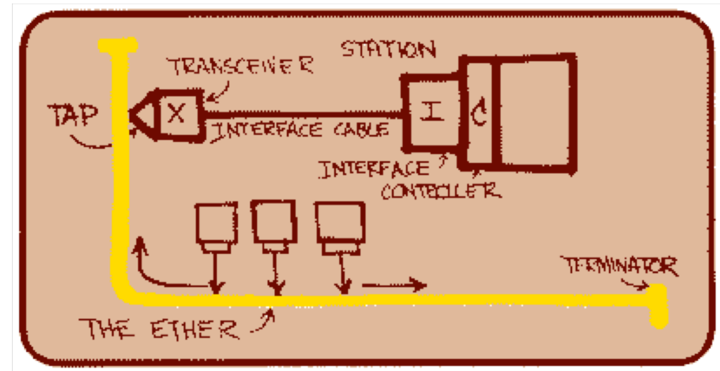
Medium Access Control

1. Sharing by partitioning the medium
   – Introduction, Time and Frequency division
   – Code division

2. **Contention-based sharing**
   – Unslotted ALOHA, Slotted ALOHA
   – **The Ethernet**

How did the Ethernet get built?

- Bob Metcalfe, PhD student at Harvard in early 1970s
  - Working on protocols for the ARPAnet
  - Intern at **Xerox Palo Alto Research Center (PARC),** 1973



  - Needed a way to network ≈100 *Alto* workstations in-building
  - Adapted ALOHA packet radio

- Metcalfe later founds *3Com*, acquired by HP in April ’10 for USD $2.7 bn

The Ethernet: Physical design

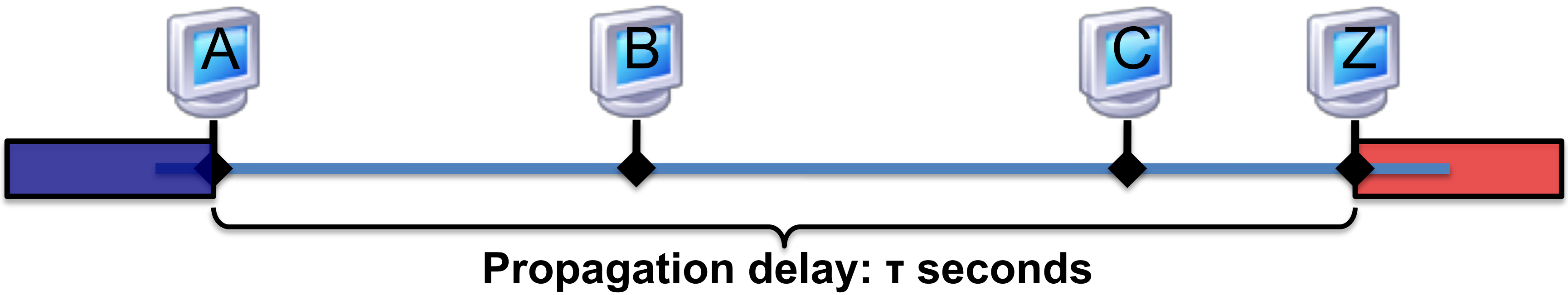
- **Coaxial cable,** with ***propagation time*** τ
  - Propagation speed: 3/5 × speed of light

- Experimental Ethernet
  - Data rate: *B* = 3 Mbits/s, maximum length: 1000 m

- **Goal:** Any **frame** a station injects onto the coaxial cable reaches all other stations **with high probability**

Propagation delay $\tau = \dfrac{10^3 \text{ m}}{\frac{3}{5}\left(3 \times 10^8 \text{ m/s}\right)} \approx 5\ \mu s$

Collisions on the Ethernet



Two Problems:

1. **Sender doesn't know** whether frame **collided or not**
2. **Sender doesn't know who received** a colliding frame

- Overlapping packets at **B** means **signals sum**
  - Not time-synchronized: result is **bit errors** at **B**
- **But: C receives OK** in this example

Who gets to transmit, and when?

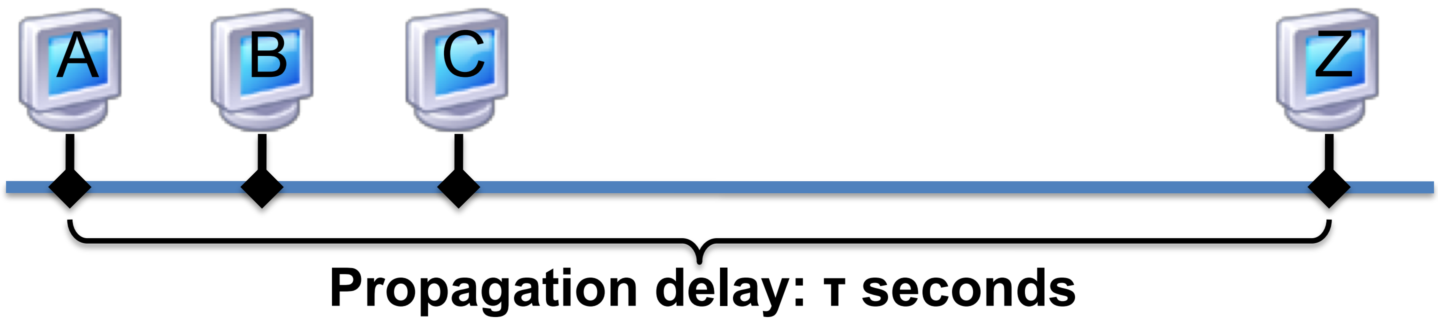
**Carrier Sense Multiple Access
with Collision Detection (CSMA/CD)**

1. Begin the transmission procedure at any time

2. ***Carrier sensing:*** **defer your transmission** if you sense that another station is transmitting

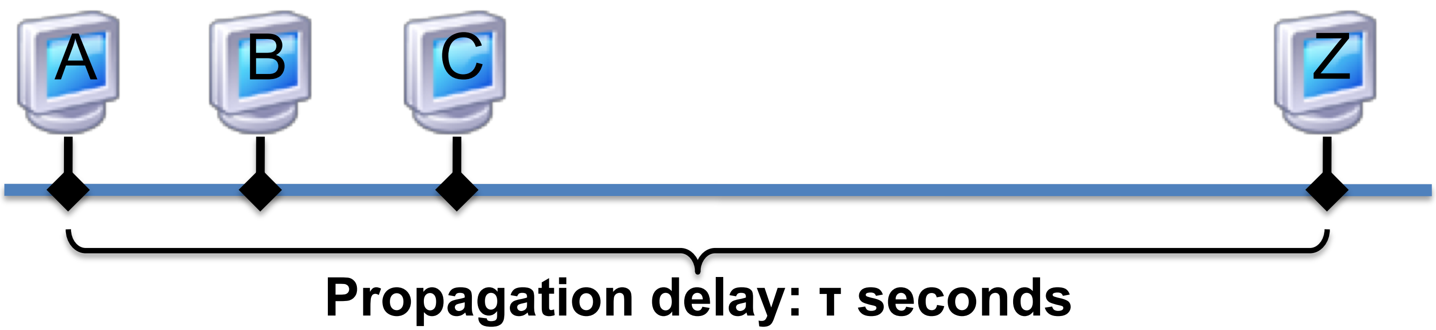
3. ***Collision detection:*** while sending, immediately **abort your transmission** if you detect another station transmitting

When might a collision happen?



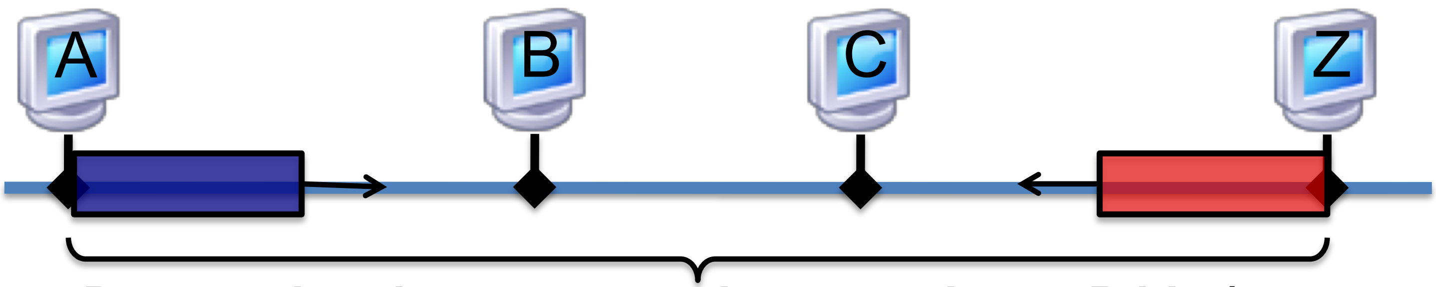
- Suppose Station A begins transmitting at time 0
- Assume that the packet lasts much longer than τ
- All stations sense transmission and defer by time τ
  - Don't begin any new transmissions

How long does a collision take to detect?



- Suppose **Station A** begins transmitting at **time 0**
- **Worst case: Z** begins transmitting **just before time τ**
- Just before **time 2τ, A and B** hear **Z's** transmission (hence **detect collision**)

Collision detection and packet size



Propagation delay: τ seconds, transmit rate B bits/sec

- If packets take **time 2τ**, **A** will still be transmitting when **Z's** packet arrives at **A,** so **everyone will detect collision**

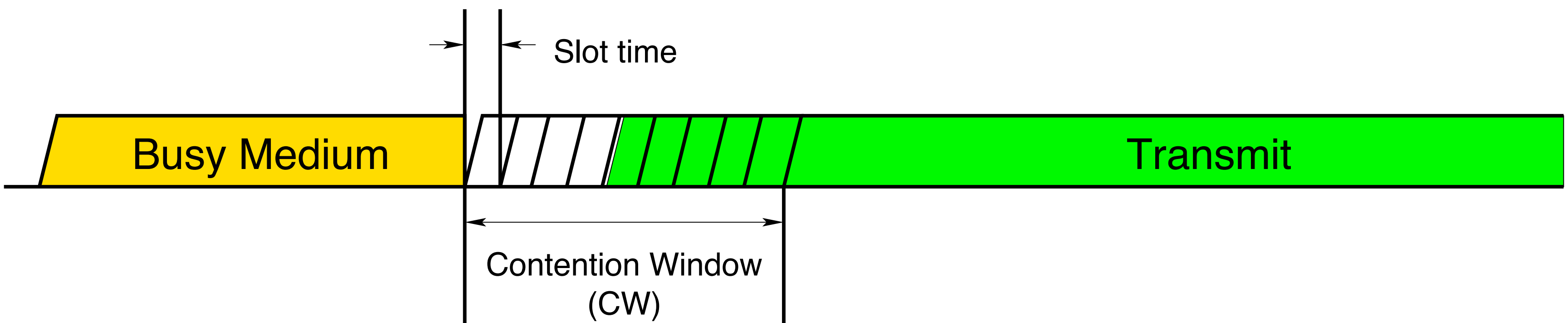
- So Ethernet enforces a **minimum packet size of 2τ*B* bits**
  - Experimental Ethernet:
    - τ = 5 μs, *B* = 3 Mbits/s → 2τ*B* = 30 bits

Resolving collisions

- Upon abort (carrier detect), station enters the **backoff state**
- **Key idea:** the colliding stations all wait a random time before carrier sensing and transmitting again
  - *How to pick the random waiting time?* (Should be based on how stations have data to send)
  - *How to estimate the number of colliding stations?*
- **Goal:** Engineer such that nodes will wait different amounts of time, carrier sense, and not collide

Slotted Ethernet backoff

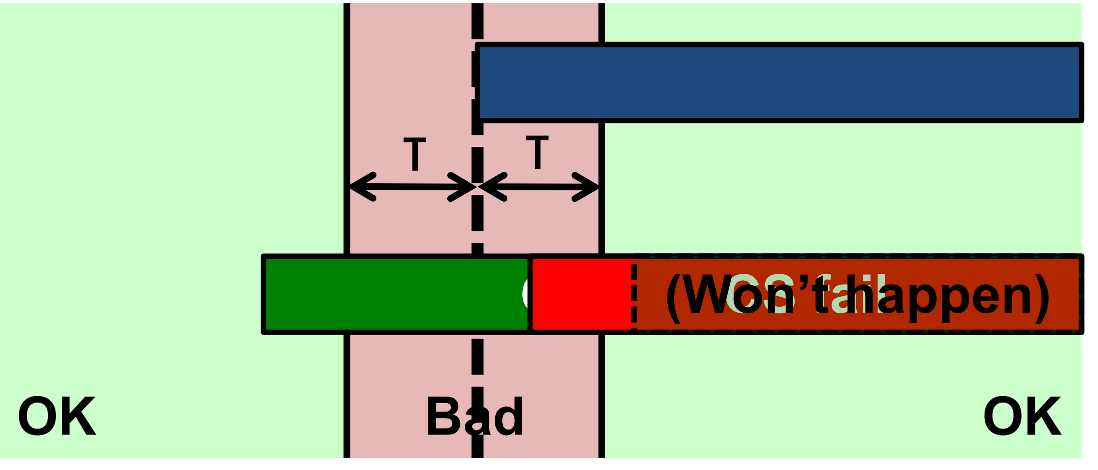
- Backoff time is **slotted** **(like slotted ALOHA)** and **random**
  - Station's view of the where the first slot begins is at the end of the busy medium
  - Random slot choice in ***contention window (CW)***



- **Goal:** Choose slot time so that **different** nodes picking **different** slots CS and defer → **don't collide**

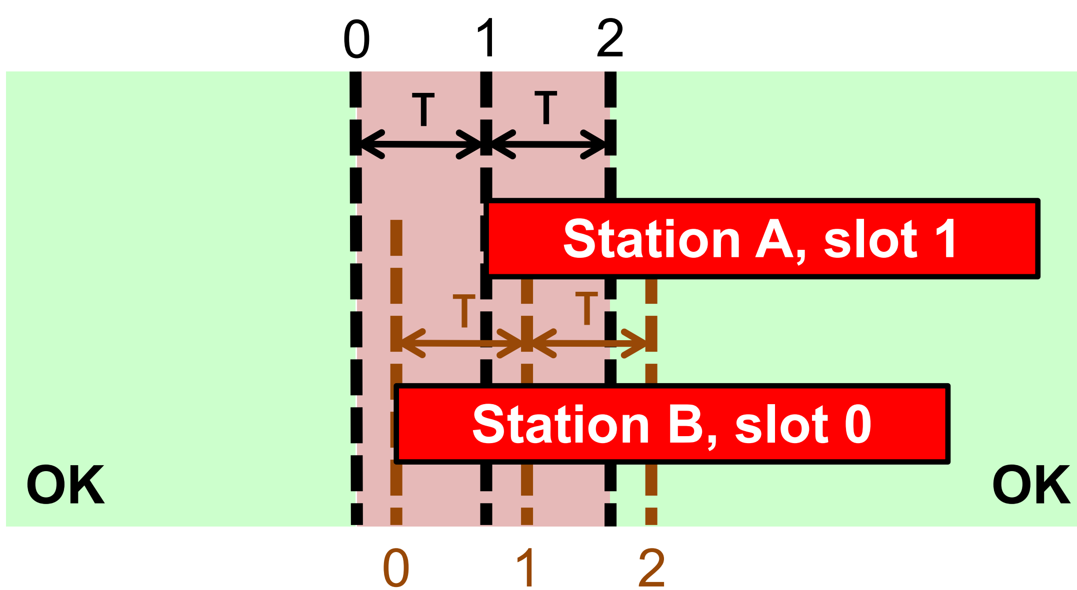
Picking the length of a backoff slot

- Consider from the perspective of **one packet at time t**
  1. Packets before t−τ will cause **packet** to **defer**
  2. Packets after t+τ **will not happen** *(why not?)*
- Packets beginning within time τ apart **will collide**
- *So should we pick a backoff slot length of τ?*



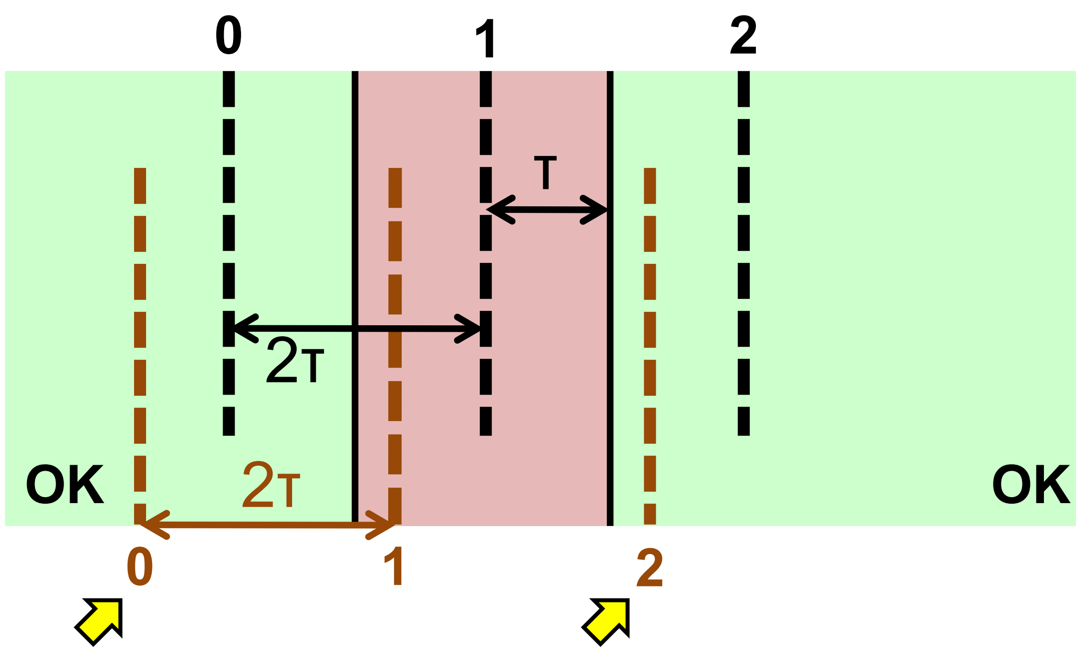
The problem of clock skew

- **No!** Slots are timed off the tail-end of the last packet
  - Therefore, **stations' clocks differ by at most τ**
- Suppose we use a backoff slot length of τ
  - **Different** stations picking **different** slots **may collide!**



Picking slot time in presence of clock skew

- Want **other** station's **other** slots to **all** be in "**OK**" region
  - Then, transmissions in **different slots won't collide**
  - Worst case clock skew: τ
  - So, pick a slot time of $\tau + \tau = 2\tau$



Binary Exponential Backoff

- Binary exponential backoff (**BEB**): double CW size on each consecutive collision

- Stations wait some number of slots chosen uniformly at random from CW = $[0, 2^{m}-1]$
  - Reset $m \leftarrow 1$ upon a successful transmission
  - First retransmit (m = 1): pick from [0, 1]
  - Second retransmit (m = 2): pick from [0, 1, 2, 3]

- **Observe:** Stations transmitting new frames don’t take into account recent collisions, might transmit before stations in backoff

Comparing CDMA vs ALOHA random access

- **CDMA wireless**
  - **No interference** between transmitting stations
  - Adaptation to varying numbers of users possible by changing codes
  - **Reduced rate** of individual transmissions
  - Unused codes **waste overall capacity**

- **ALOHA random access**
  - Stations can transmit using the entire medium, at full rate if alone
  - Almost-instant adaptation to varying traffic loads
  - Concurrent transmissions result in **collisions, reduced throughput**

Monday, Tuesday Precepts

Introduction to Lab 1

Tuesday Topic:

Link Layer II: Sharing the Medium, Wi-Fi Above the PHY