Link Layer I: ALOHA, Time-, Frequency-, and Code Division

COS 463: Wireless Networks
Lecture 4
Kyle Jamieson

Review: The Data Link Layer (L2)

- Enables exchange of atomic messages (frames) between end hosts

- Determine start and end of bits and frames (framing)
- Deliver information reliably
- Control errors

- Some link layers involve a **shared medium**
  - *e.g.*, Shared-wire Ethernet, satellite uplink, Wi-Fi
  - **Today: *Medium access control* to share the medium**

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: Goals

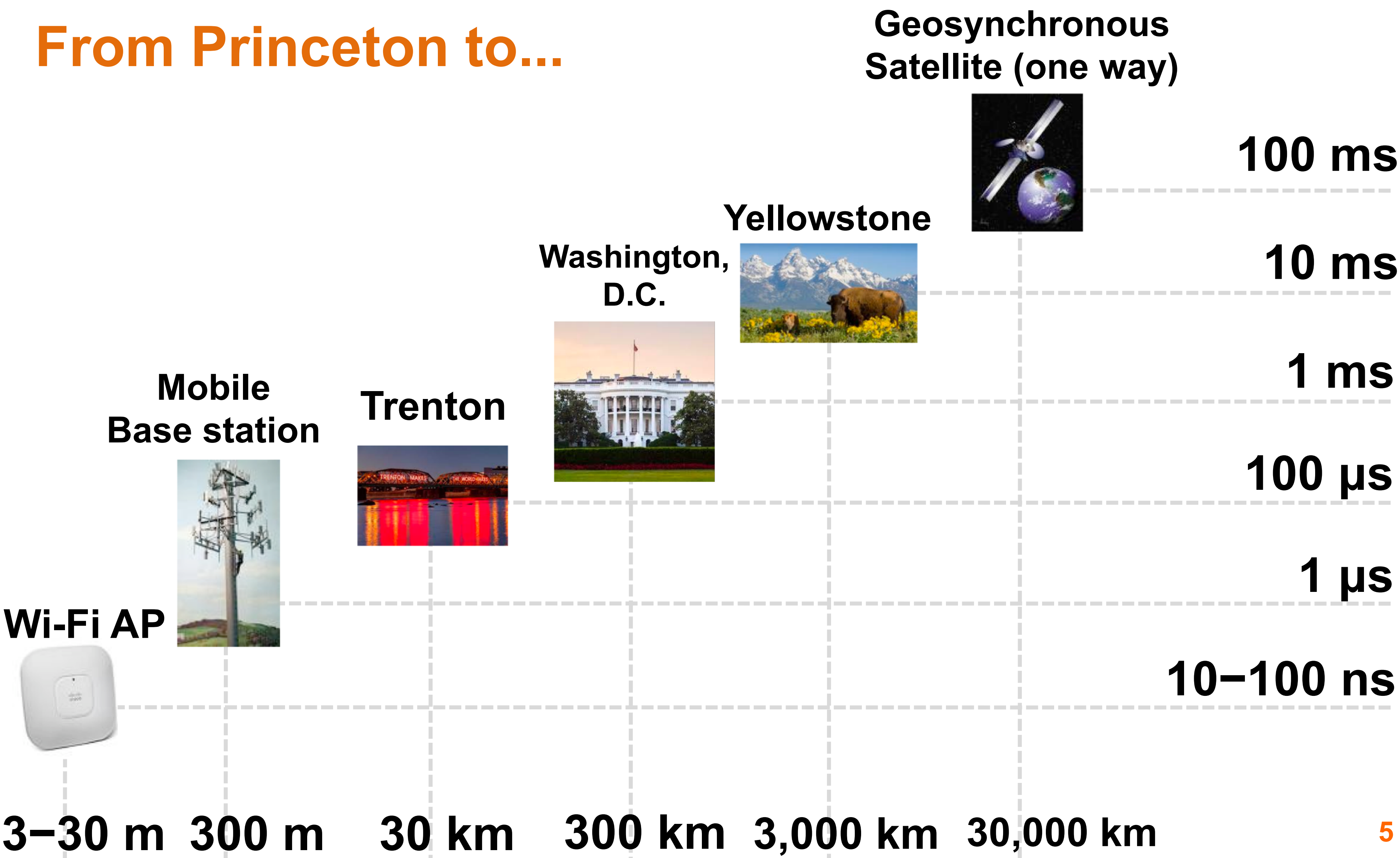
1. **Efficiency**
   - **High throughput** (bits/second successfully **received** through the channel)
     - *i.e.* high **utilization** (throughput/raw channel rate)

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

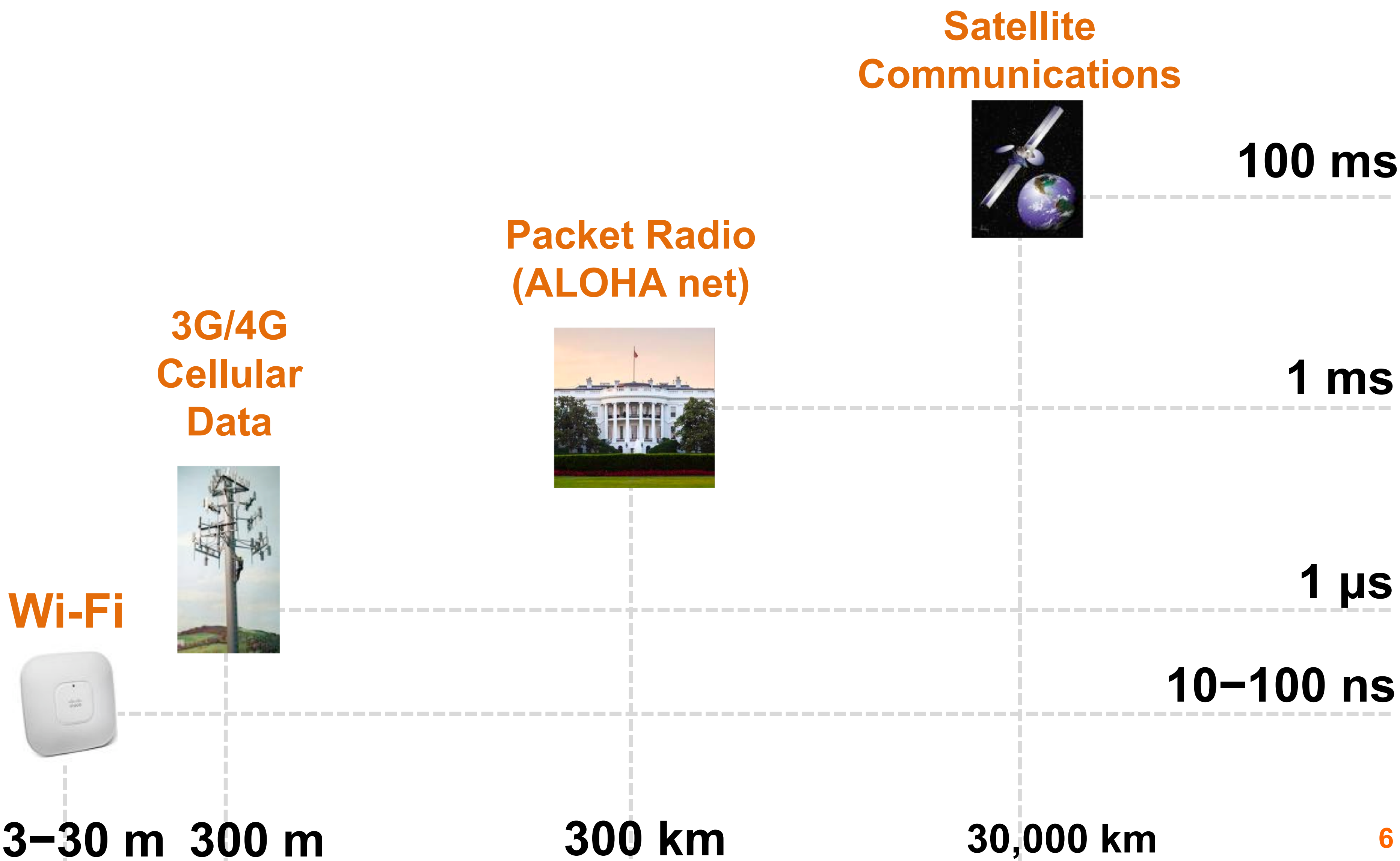
3. **Latency:** Want to minimize the time a host waits before being granted permission to talk on the shared medium

Physical Limitation: Finite speed of light

From Princeton to...



Vastly Different Timescales, Same Medium Access Protocol!



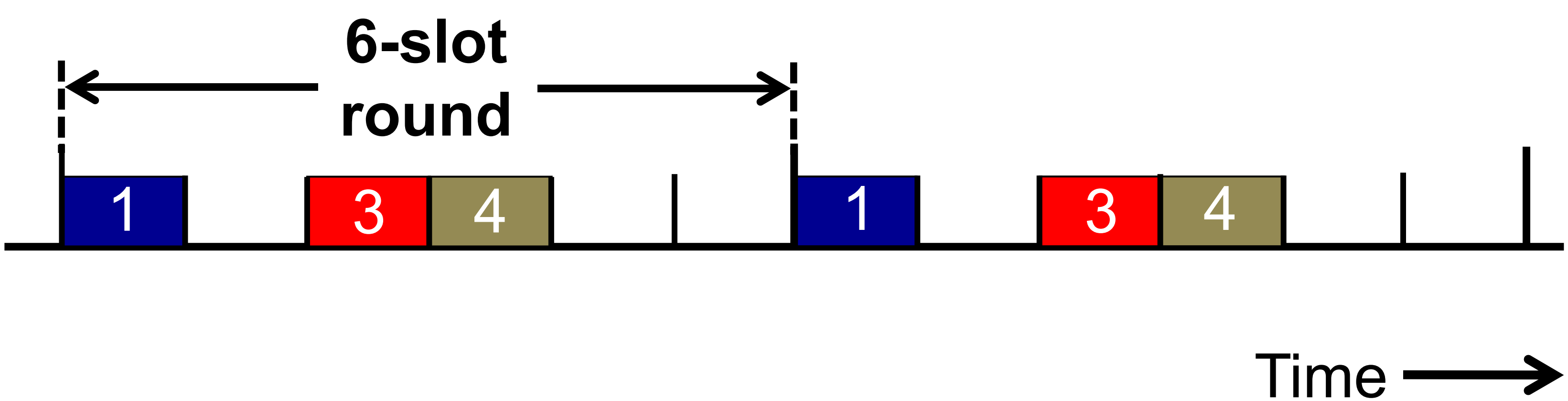
Today

1. **Sharing by partitioning**
   - **Time division**
   - **Frequency division**
   - Code division

2. Contention-based sharing
   - ALOHA
   - The Ethernet

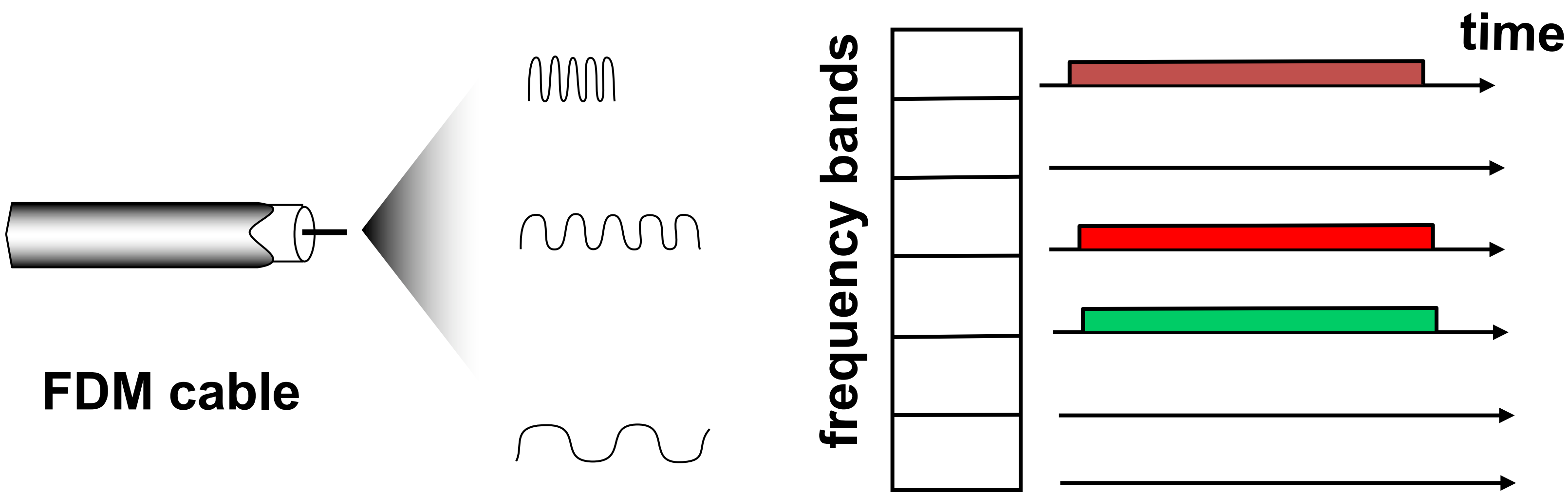
TDMA: Time Division Multiple Access

- 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**

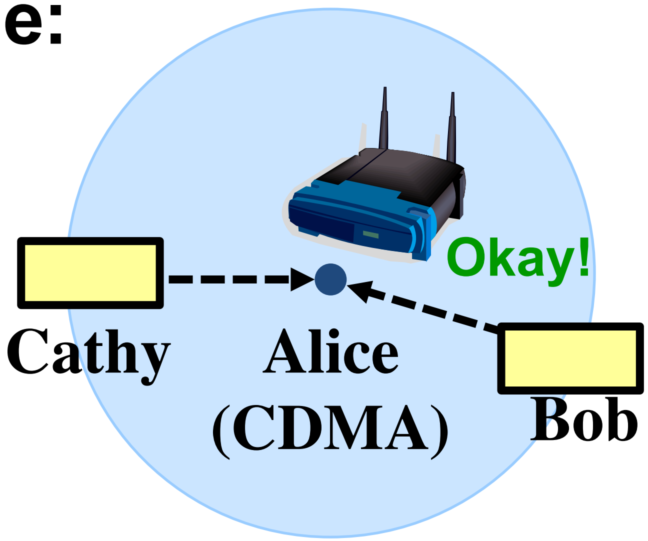
Today

1. **Sharing by partitioning**
   – Time division
   – 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

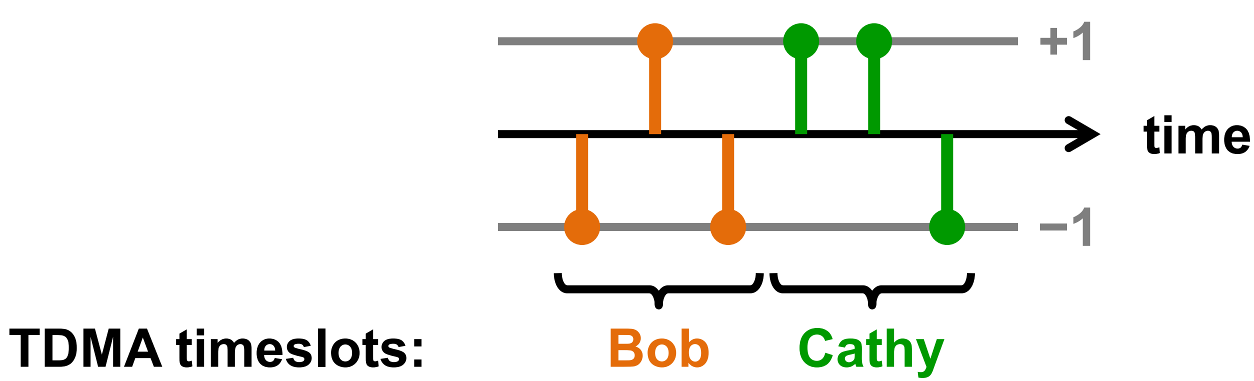
Representing bits as binary levels

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

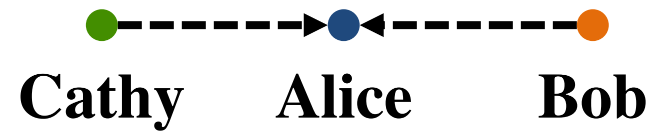
  0 bit ←→ +1 level　　　**1 bit ←→ −1 level**

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

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

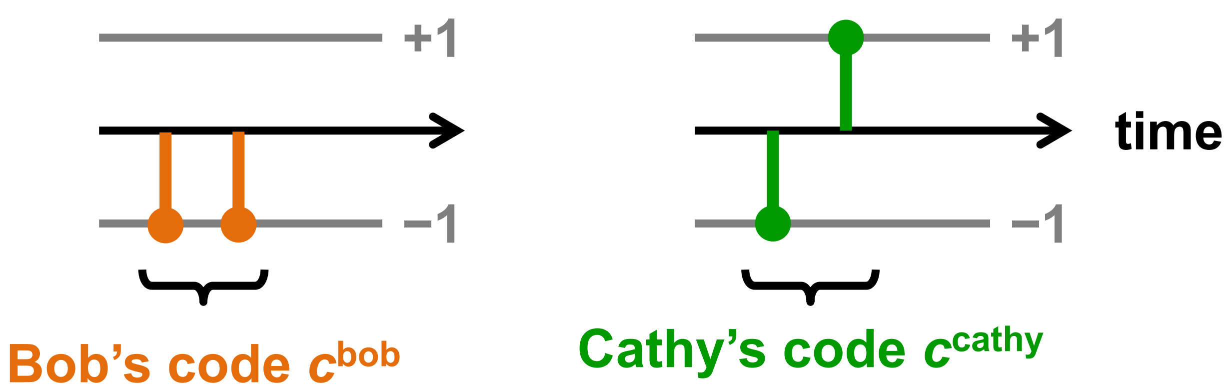


CDMA: User codes

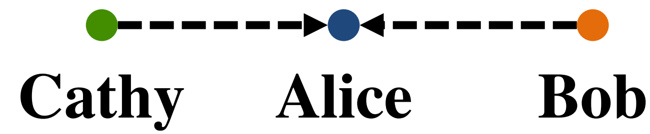


- Assign each user a unique binary sequence of bits: ***code***
  - Call each code bit a ***chip*** (convention)
  - Call the code length ***M***

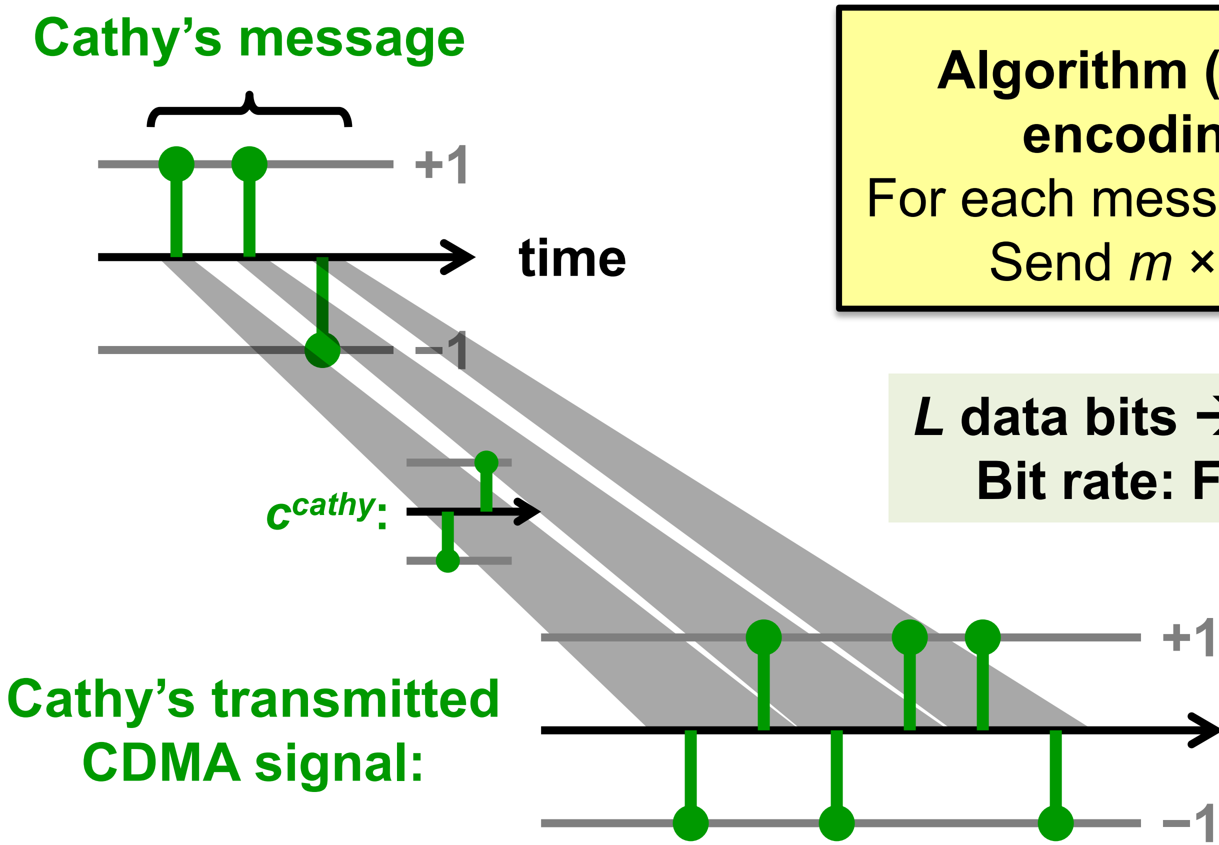
- **CDMA example:**



CDMA: Cathy Sending



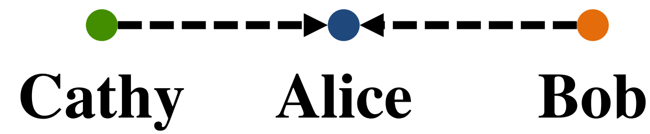
- Suppose Cathy alone sends message bits **`001`**:



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

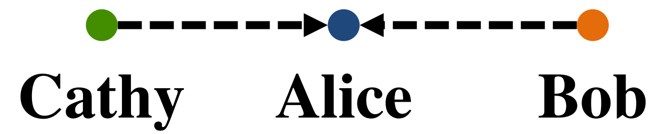
L **data bits → *M* × *L* CDMA chips**
Bit rate: Factor of *M* slower

CDMA: Assumptions

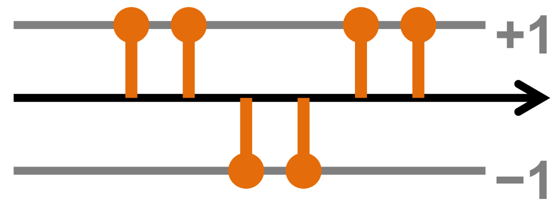


- Let's assume we have a way of:
  - Synchronizing **Cathy's** and **Bob's** **data bits** in time
  - Synchronizing **Cathy's** and **Bob's** **CDMA chips** in time
  - Estimating and correcting the **effect of the wireless channel** between **Cathy** and **Bob** to **Alice**

What Alice Hears

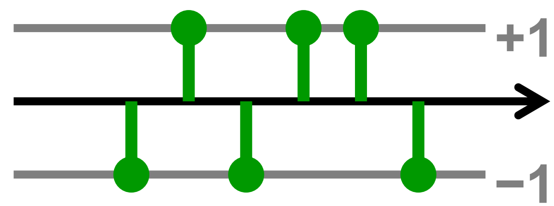


Bob's transmitted CDMA signal:



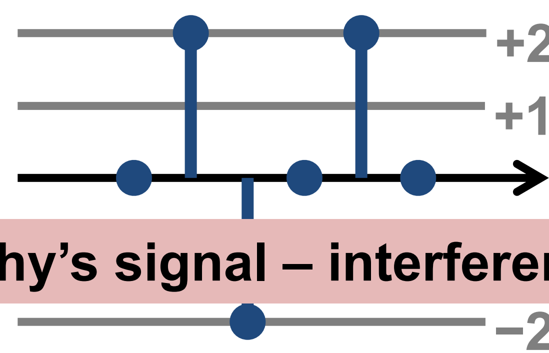
+

Cathy's transmitted CDMA signal:



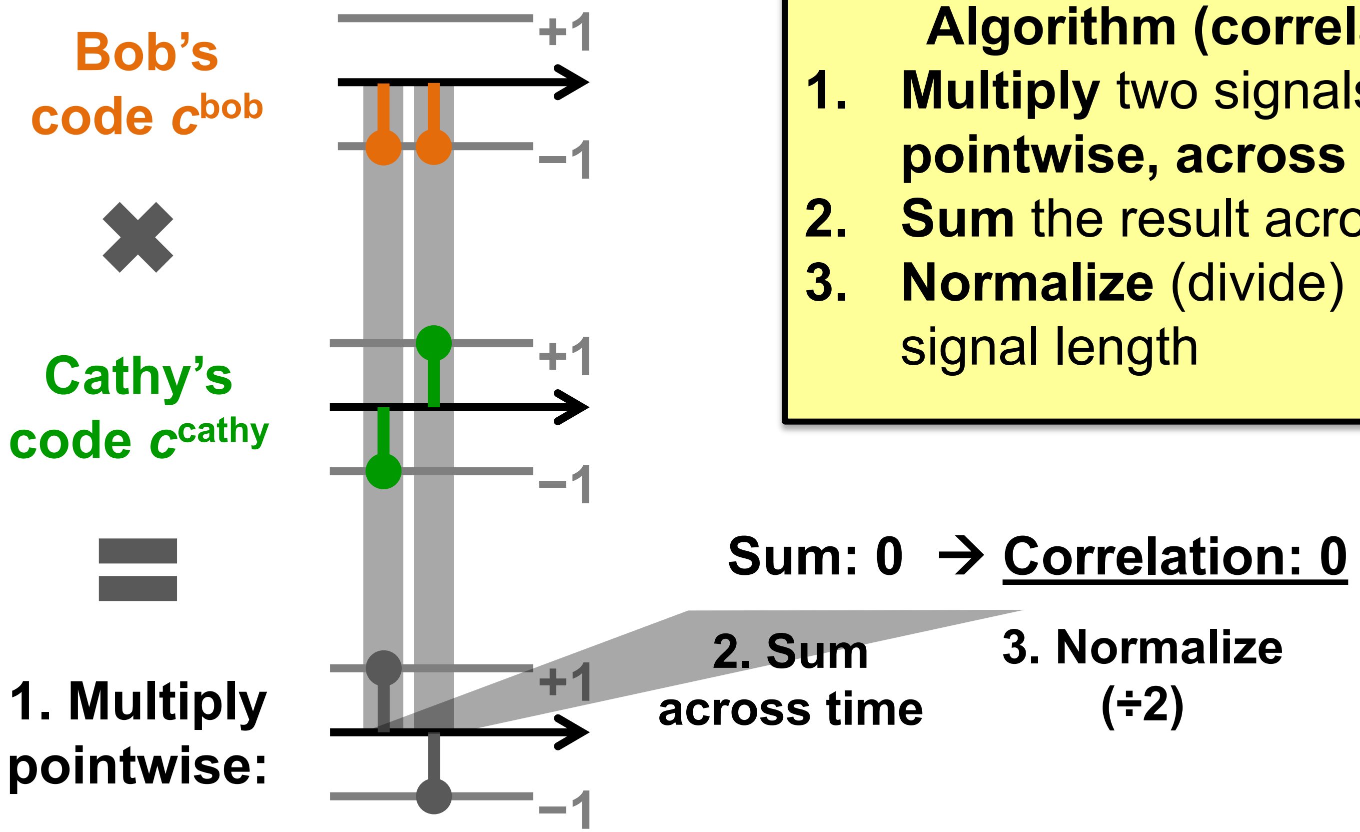
=

What Alice hears:



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

Tool: Correlation

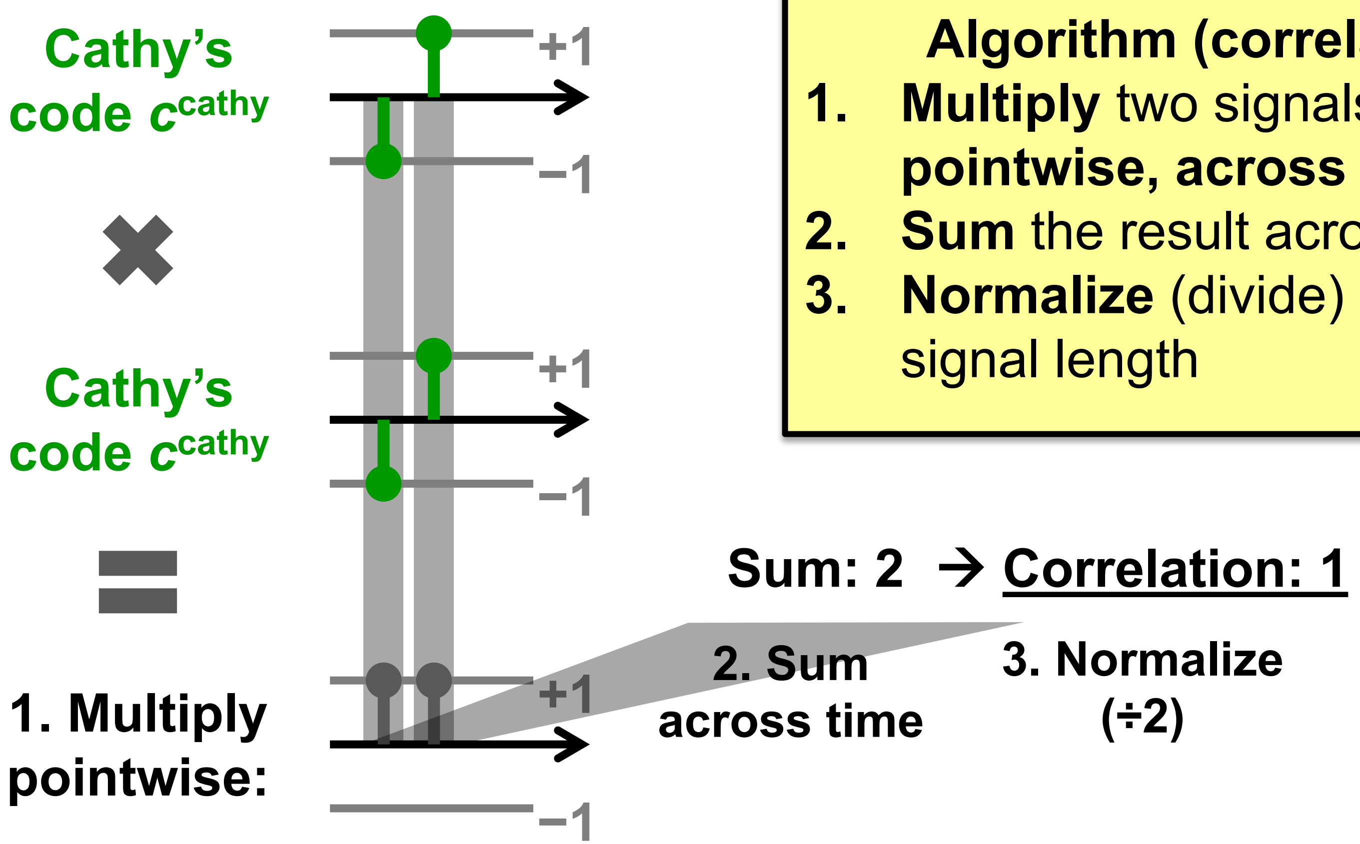


Algorithm (correlation):

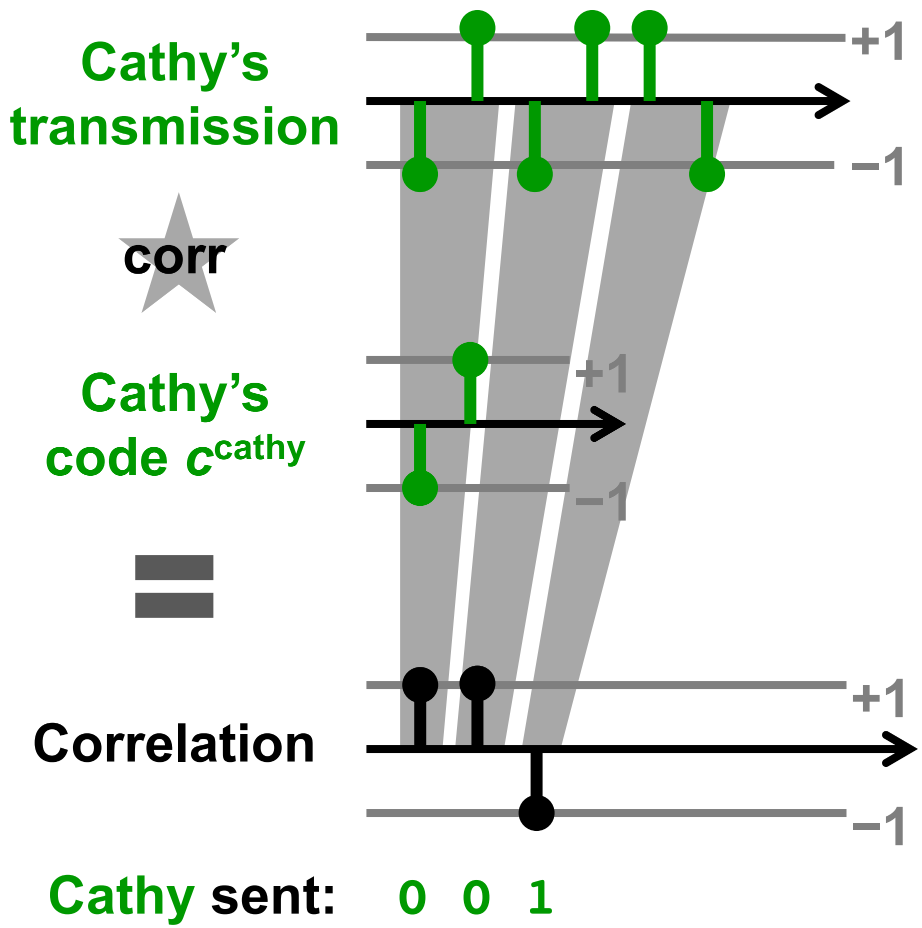
1. **Multiply** two signals **pointwise, across time**
2. **Sum** the result across time
3. **Normalize** (divide) by the signal length

Tool: Correlation

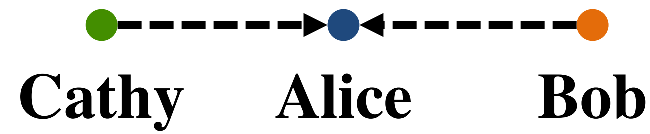
Algorithm (correlation):
1. **Multiply** two signals **pointwise, across time**
2. **Sum** the result across time
3. **Normalize** (divide) by the signal length

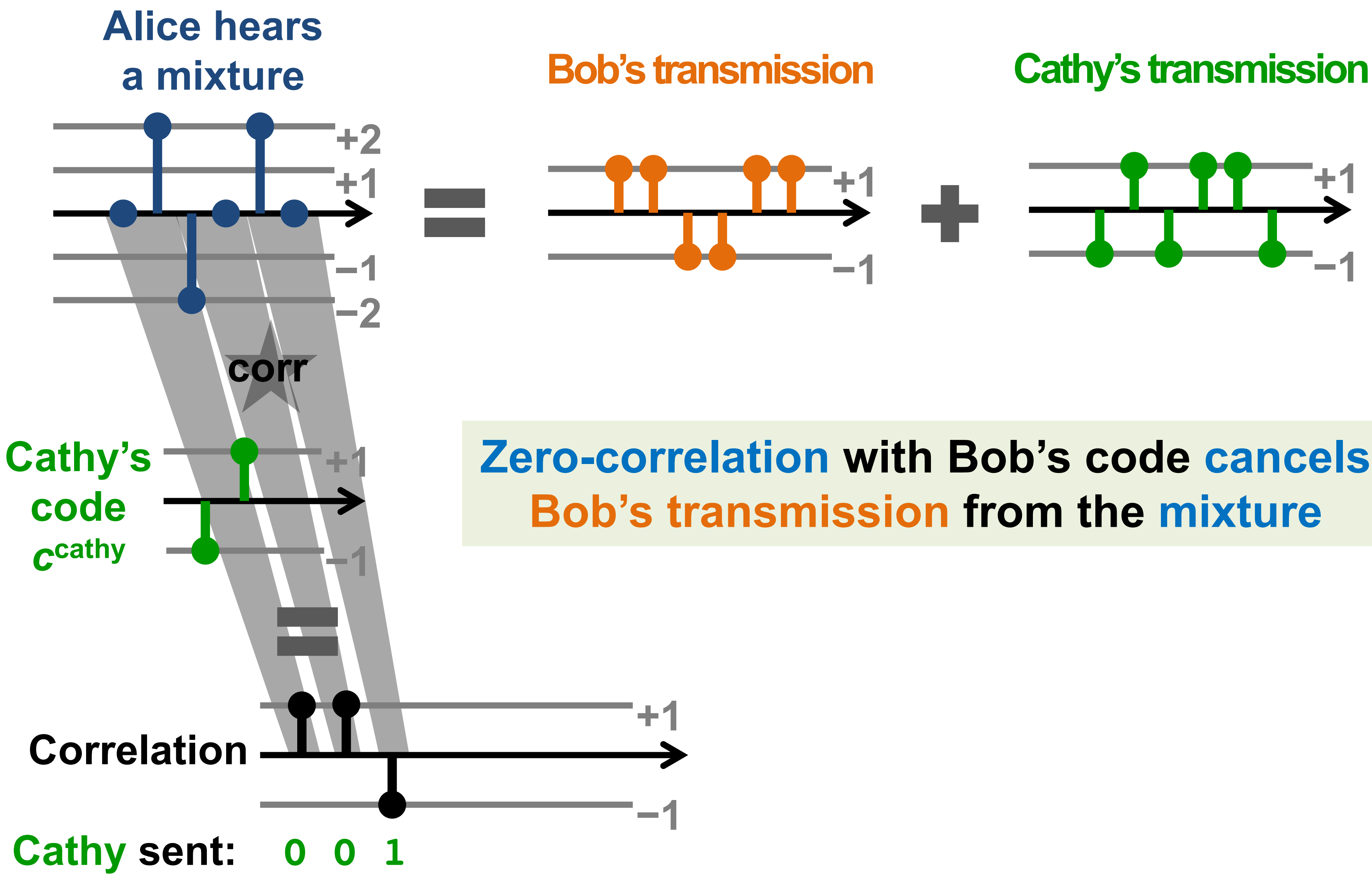


Correlating Cathy's Code and CDMA transmission



Listening to Cathy

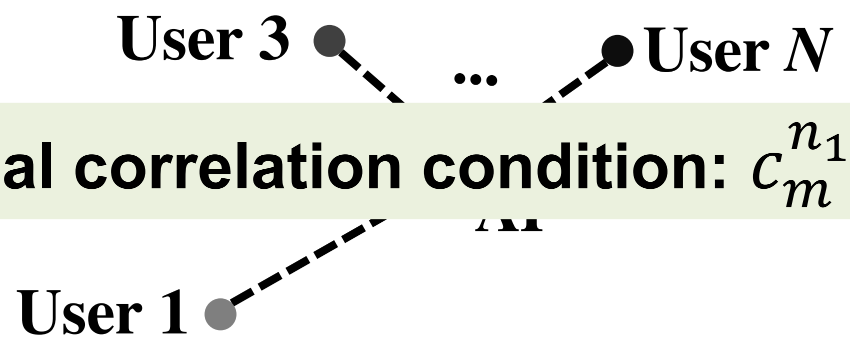




Zero-correlation with Bob's code **cancels Bob's transmission** from the **mixture**

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: $c_m^{n_1} \bigstar c_m^{n_2} = 0$

- **Recall:** Correlate against code c_m^n to decode user n
  - Correlate any user's code against itself: $c_m^n \bigstar c_m^n = 1$
- **Goal:** Ensure **cancellation of all other users** when **correlating** against (each) **one**

Example of CDMA 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?

Today

1. Sharing by partitioning
   – Time division
   – Frequency division
   – Code division

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

Contention-based sharing

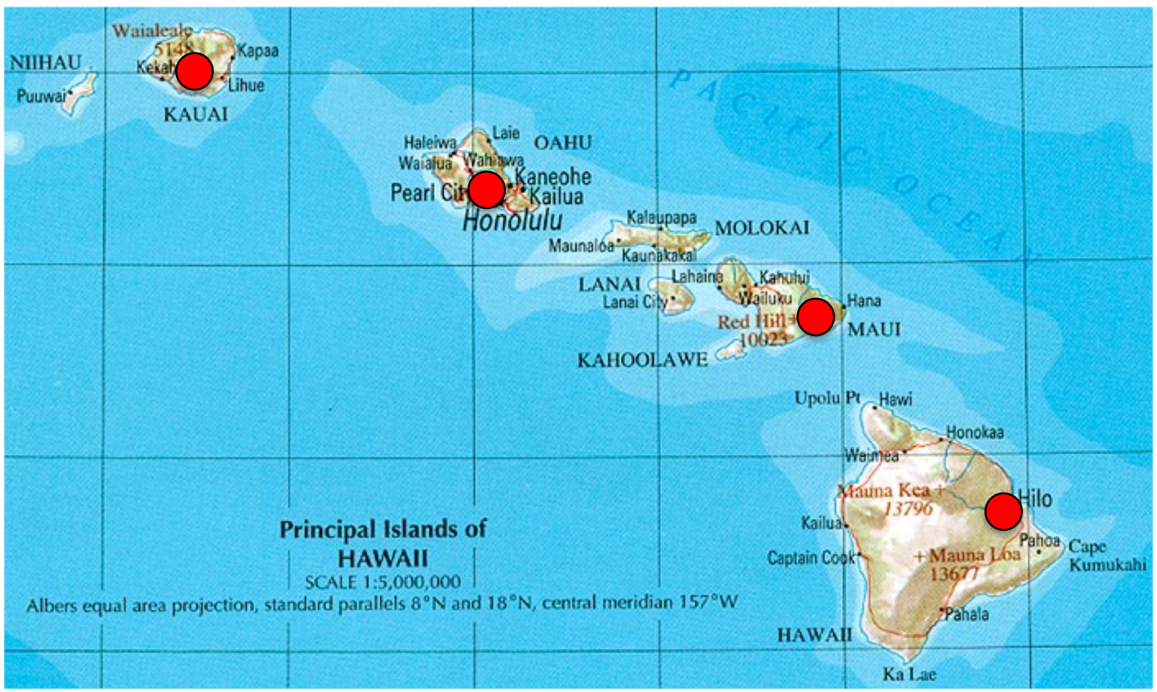
- When a station has a frame to send:
  - Transmit at **full channel data rate** *B*
  - No *a priori* coordination among nodes

- 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

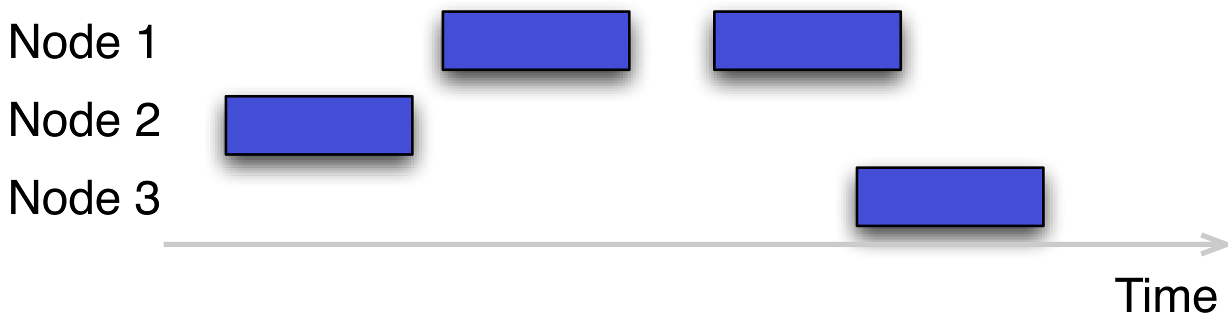
ALOHAnet: Context

- 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

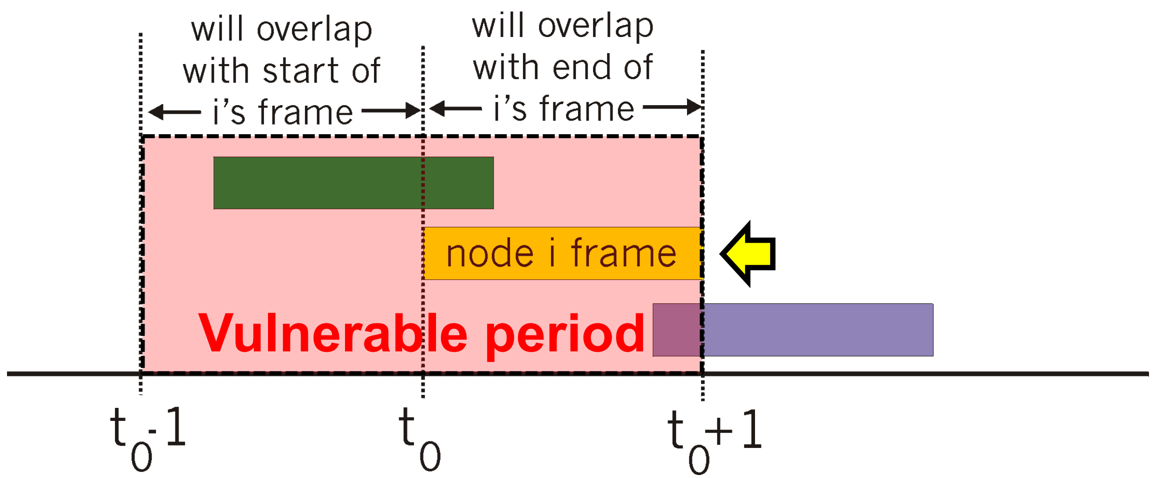
- **Simplest possible medium access control:** no control at all, anyone can just transmit a packet without delay



- **Suppose:** Probability packet **begins** in time interval $\Delta t = \lambda \times \Delta t$
  - ***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

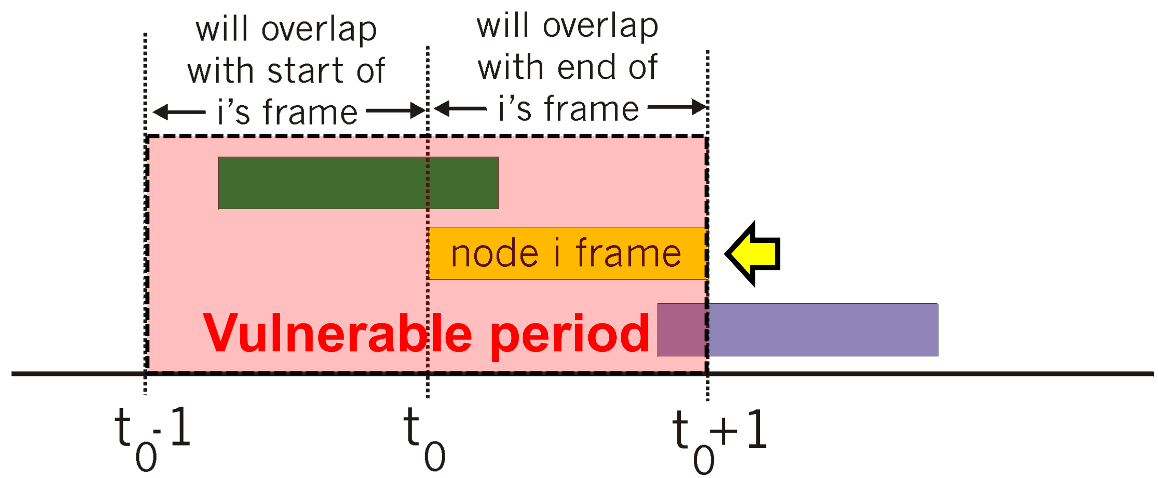
- Suppose some node ***i*** is transmitting; let's focus on ***i*** **'s frame**



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)?*

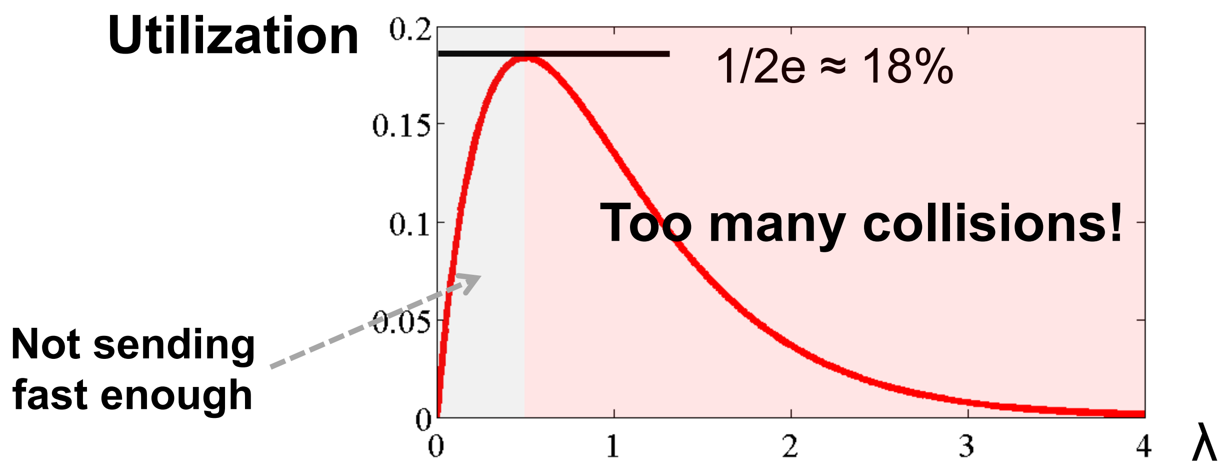
$$\Pr(\text{no send from } \textit{one} \text{ node in } 2) = 1 - \frac{2\lambda}{N}$$

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

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

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

Unslotted ALOHA: Utilization

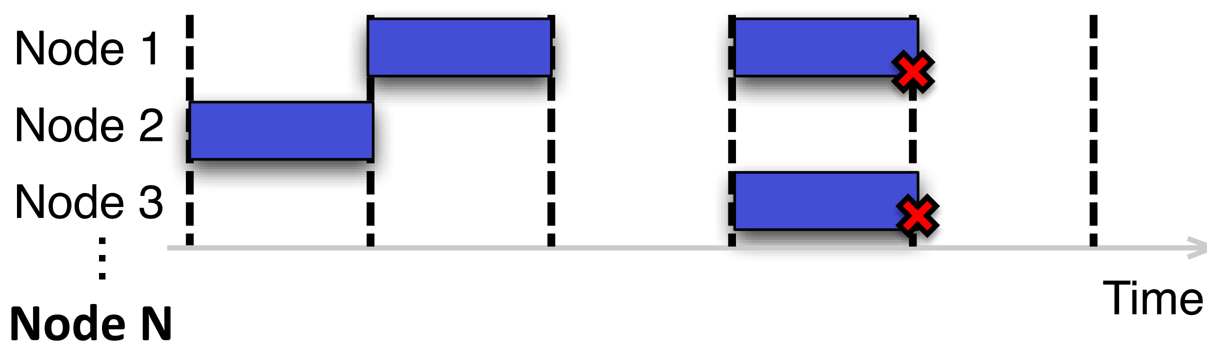


- Recall λ is the **aggregate rate** from **all senders**
- So, utilization $= \lambda \times \Pr(\text{no other transmission in 2})$
  $= \lambda e^{-2\lambda}$

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 $\lambda = N \times p$

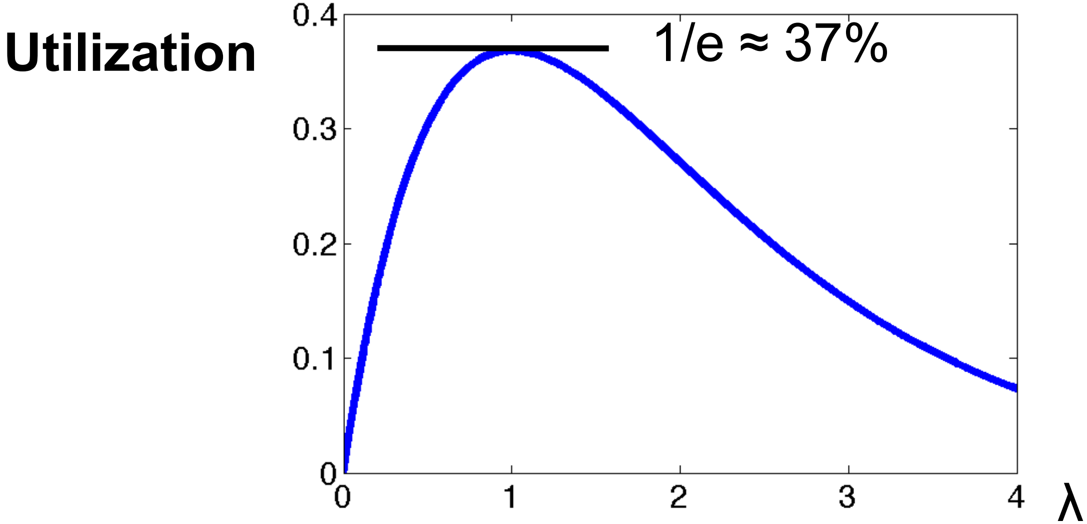
- As before, if there is exactly one transmission in a slot, can receive; if two or more in a slot, no one can receive (**collision**)



Slotted ALOHA: Utilization

Suppose N nodes, each transmit with probability p in each slot. What is the utilization as a function of aggregate rate $\lambda = N \times p$?

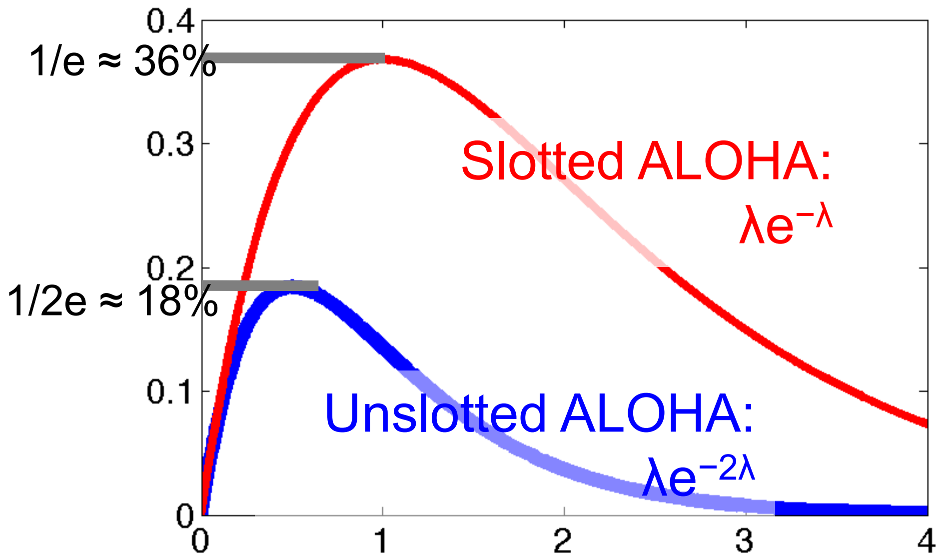
- Pr[A node is successful in a slot] = $p(1-p)^{N-1}$
- Pr[Success in a slot] = $Np(1-p)^{N-1}$



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

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

ALOHA throughput: slotted versus unslotted



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

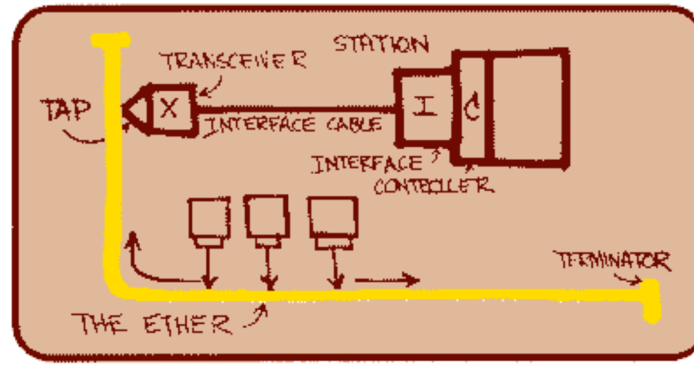
Today

1. Sharing by partitioning
   – Time division
   – 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

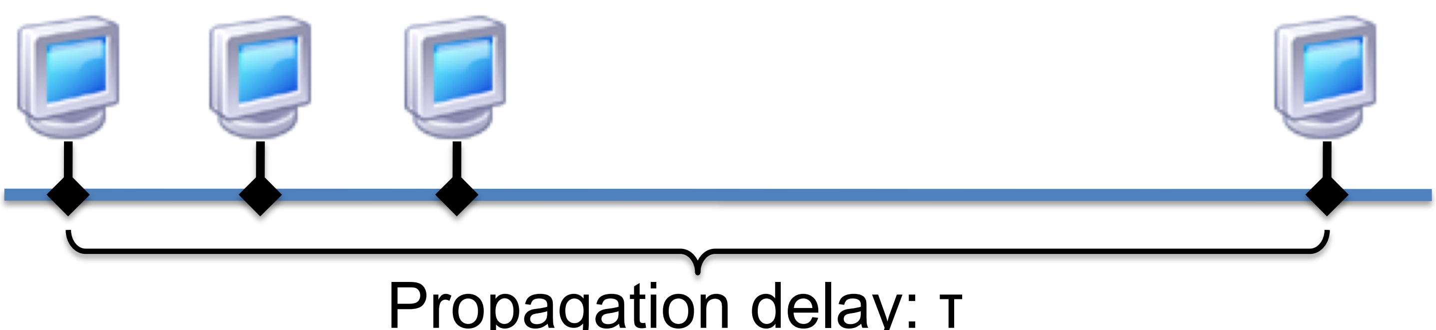
- Metcalfe later founds *3Com*, acquired by HP in April ’10 for USD $2.7 bn

The Ethernet: Physical design

- Coaxial cable, propagation delay τ
  - Propagation speed: 3/5 × speed of light

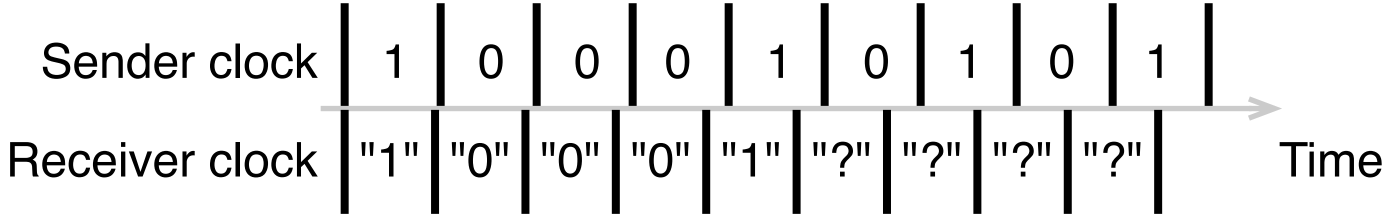
- Experimental Ethernet
  - Data rate: *B* = 3 Mbits/s
  - Maximum length: 1000 m

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

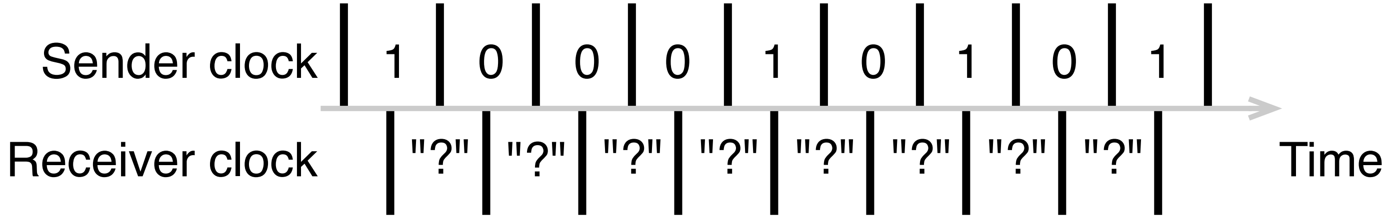


Building the link: Framing bits

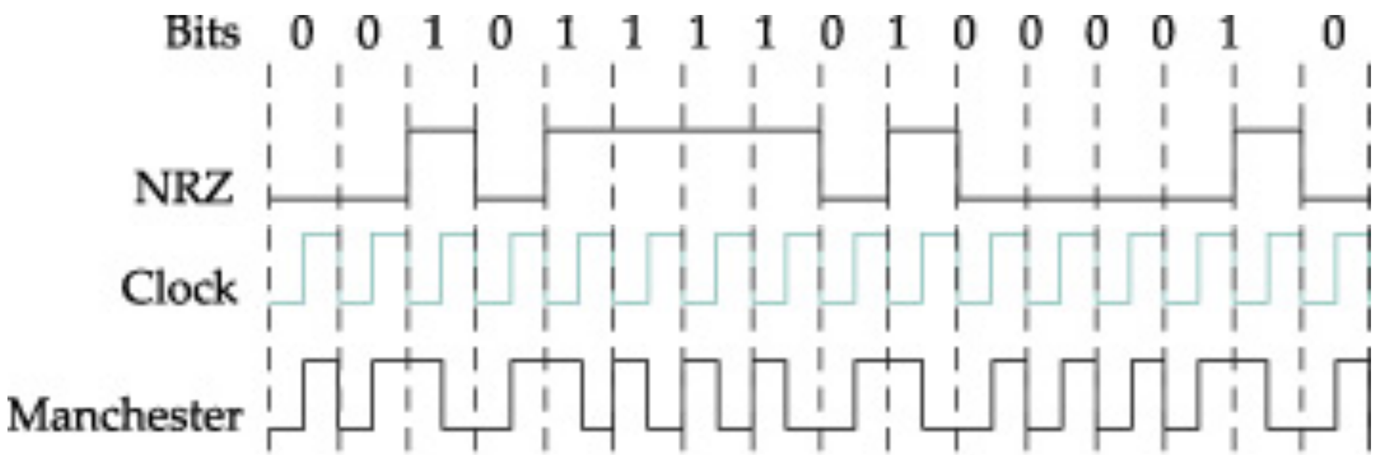
- **Goal:** Move bits from one computer to another
  - Sender and receiver have independent clocks
  - No separate "clock signal" sent on the Ethernet
- **Problem:** Agree on clock tick period



- **Problem:** Agree on clock tick alignment (*phase*)



Manchester (phase) encoding

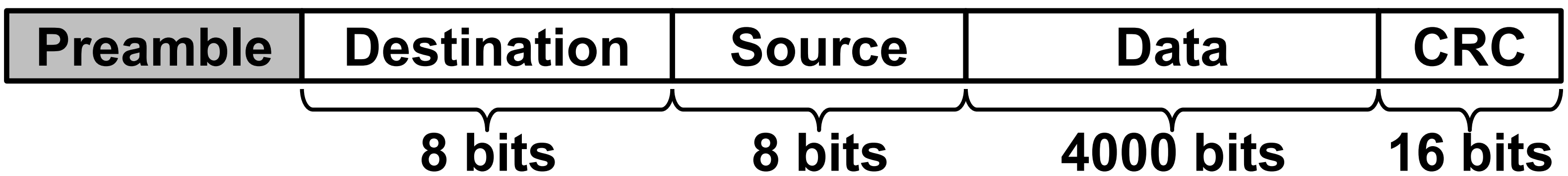


- Manchester encoding:
  - Exclusive-OR of the NRZ signal and the clock signal
  - “0” is a low-to-high transition; “1” is a high-to-low

- Transition guaranteed on every bit

- **Drawback: Halves data rate**

Ethernet framing



- Framing
  - Beginning of frame determined by presence of carrier
  - End of frame determined by absence of carrier
  - Preamble: 10101010 produces a square wave that allows receiver to frame bits
- CRC (Cyclic Redundancy Check) protects against errors on the Ether
  - Does not guard against errors introduced by the tap: rely on higher-layer checksums
- Destination address allows filtering at the link layer

Collisions on the Ethernet

- Packet of size *N* bits: *N*/*B* seconds on the wire

- From the perspective of a receiver (**B**):
  - Overlapping packets at **B** means **signals sum**
  - Not time-synchronized: result is **bit errors** at **B**

- **No fate-sharing:** **C** **receives OK** in this example

Who gets to transmit, and when?

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

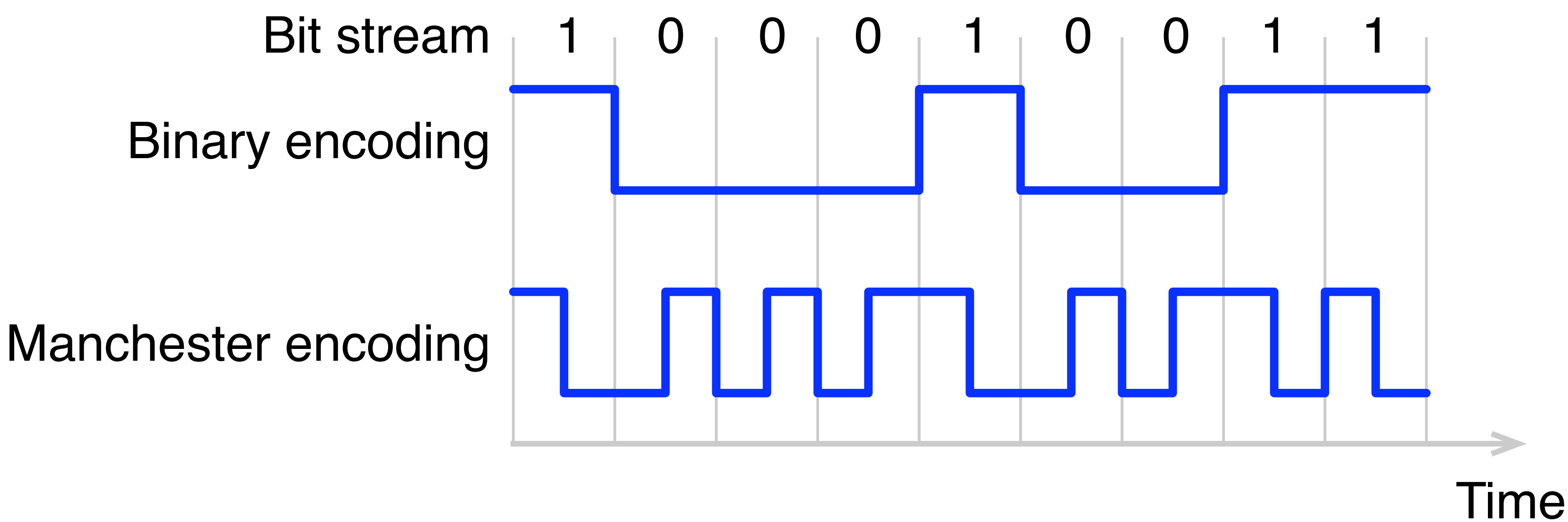
1. Begin the transmission procedure at any time

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

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

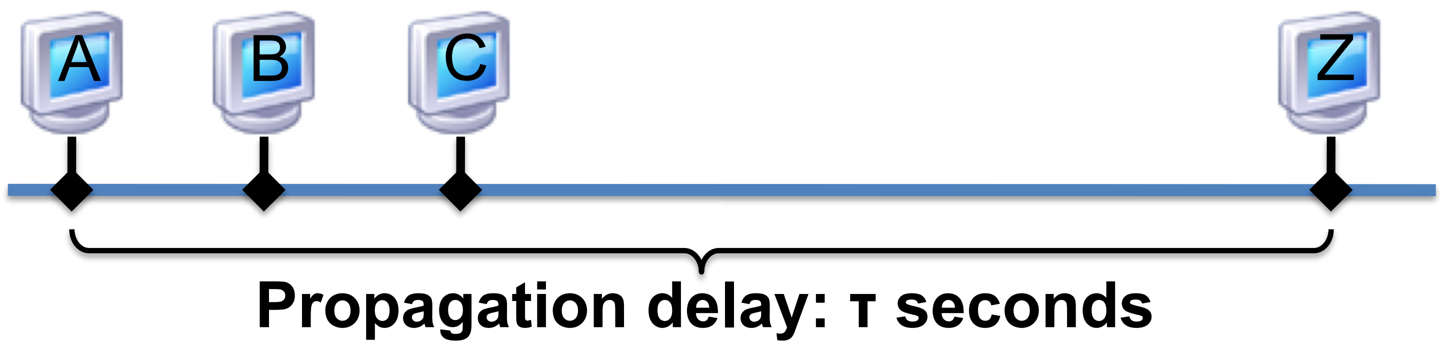
Carrier sensing

- Mechanism: measure voltage on the wire
- Binary encoding: voltage depends on the data



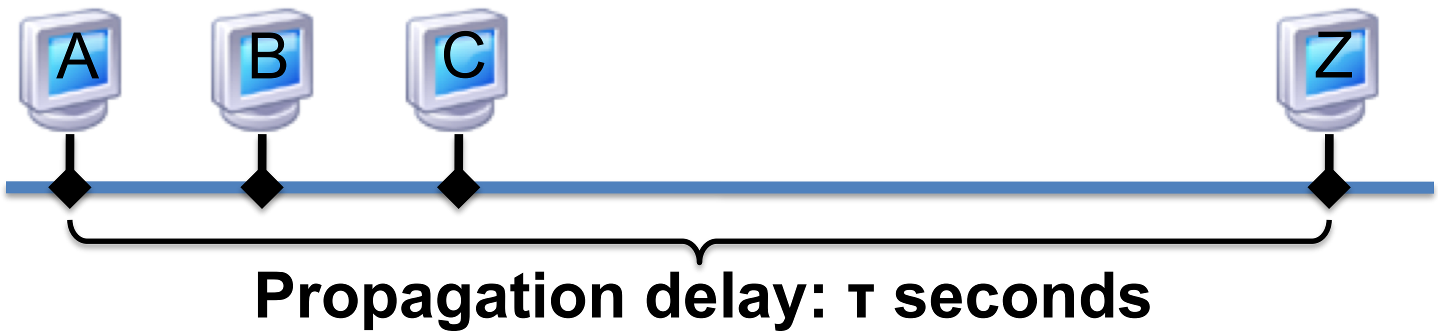
- Manchester coding: constant average voltage

Collision detection



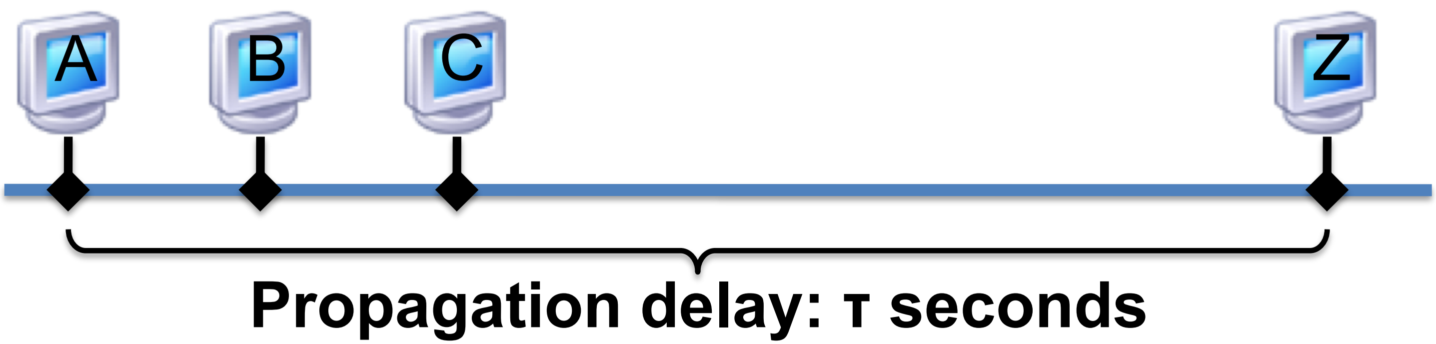
- Paper isn't clear on this point (authors did have a patent in the filing process)
- Mechanism: monitor average voltage on cable
  - Manchester encoding means your transmission will have a predictable average voltage V_0; others will increase V_0
  - Abort transmission immediately if $V_{measured} > V_0$

When might a collision happen?



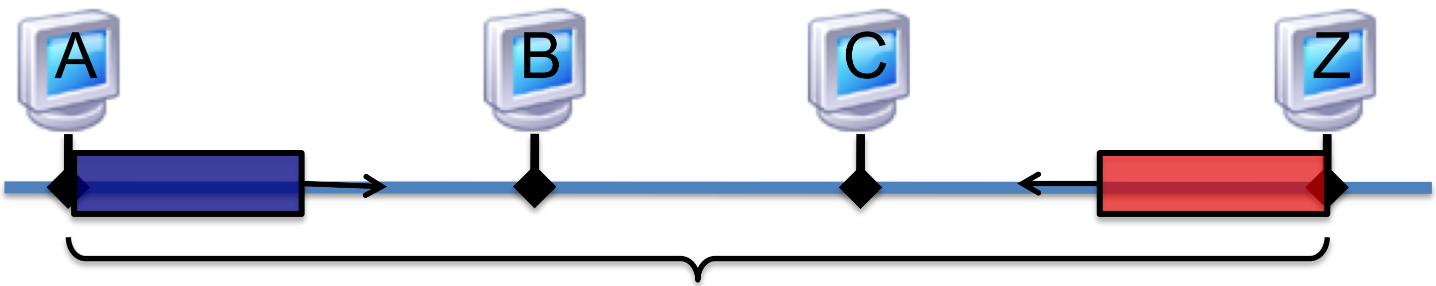
- Suppose Station A begins transmitting at time 0
- Assume that the packet lasts much longer than τ
- All stations sense transmission and defer by time τ
  - Don't begin any new transmissions

How long does a collision take to detect?



- Suppose Station A begins transmitting at time 0
- τ seconds after Z starts, A hears Z's transmission
- When does A know whether its packet collided or not?
  - **At time 2τ**

Collision detection and packet size



Propagation delay: τ seconds

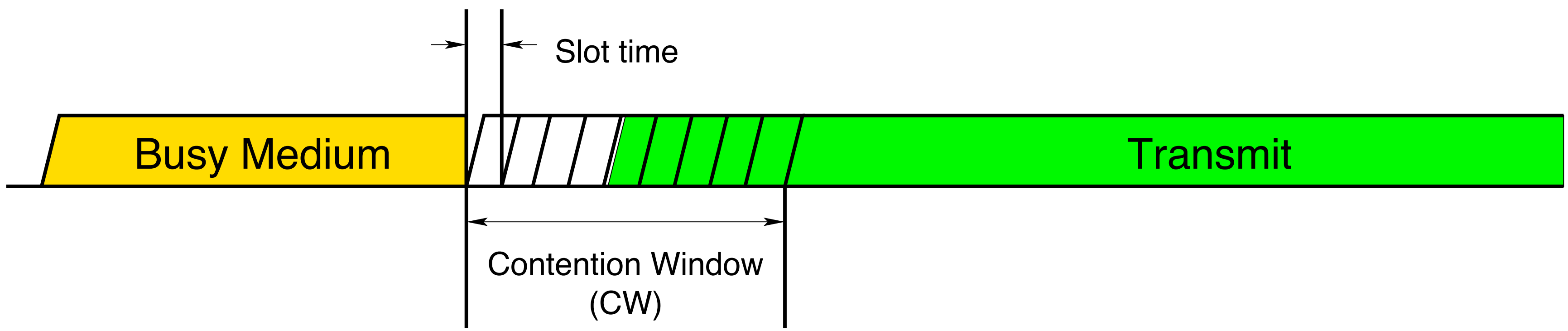
- Transmit rate *B* bits/second
- If packets take time 2τ, A will still be transmitting when Z's packet arrives at A, so A **will detect collision**
  - So **minimum packet size = 2τ*B* bits**
    - Experimental Ethernet:
      - τ = 5 μs, *B* = 3 Mbits/s → 2τ*B* = 30 bits

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)***



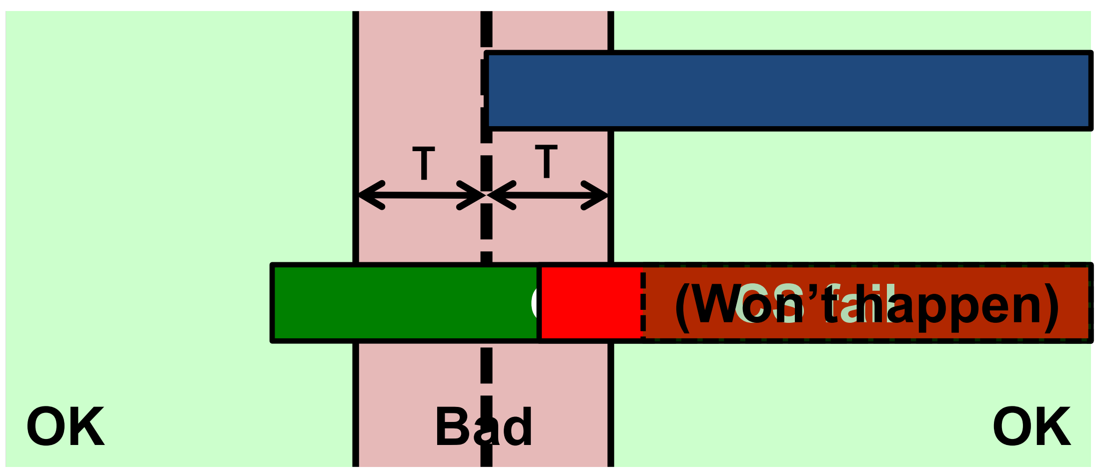
- **Goal:** Choose slot time so that **different** nodes picking **different** slots CS and defer → **don't collide**

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**

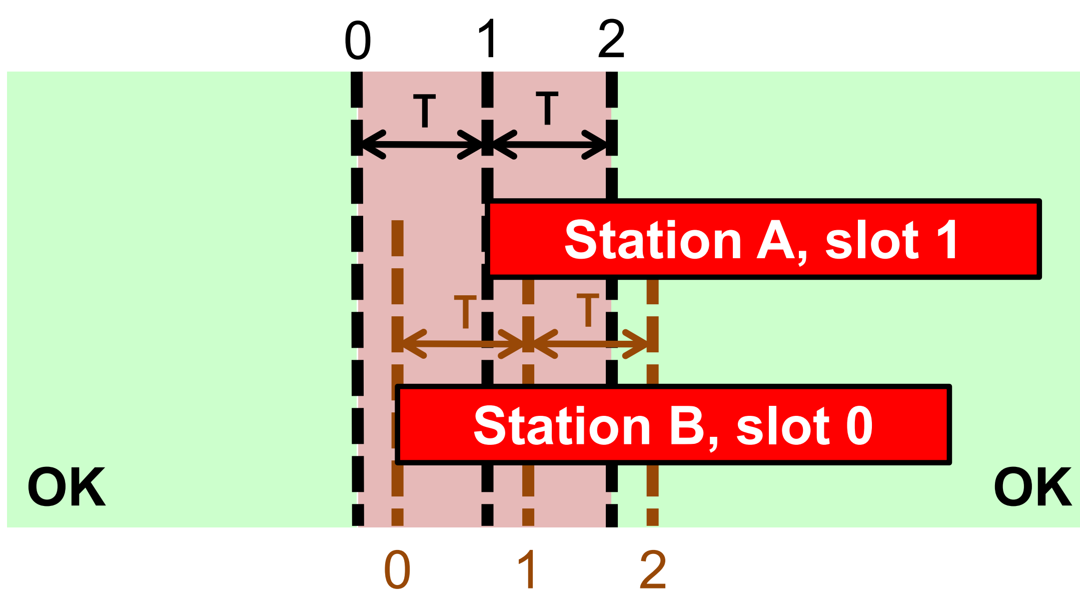
- *So should we pick a backoff slot length of τ?*



The problem of clock skew

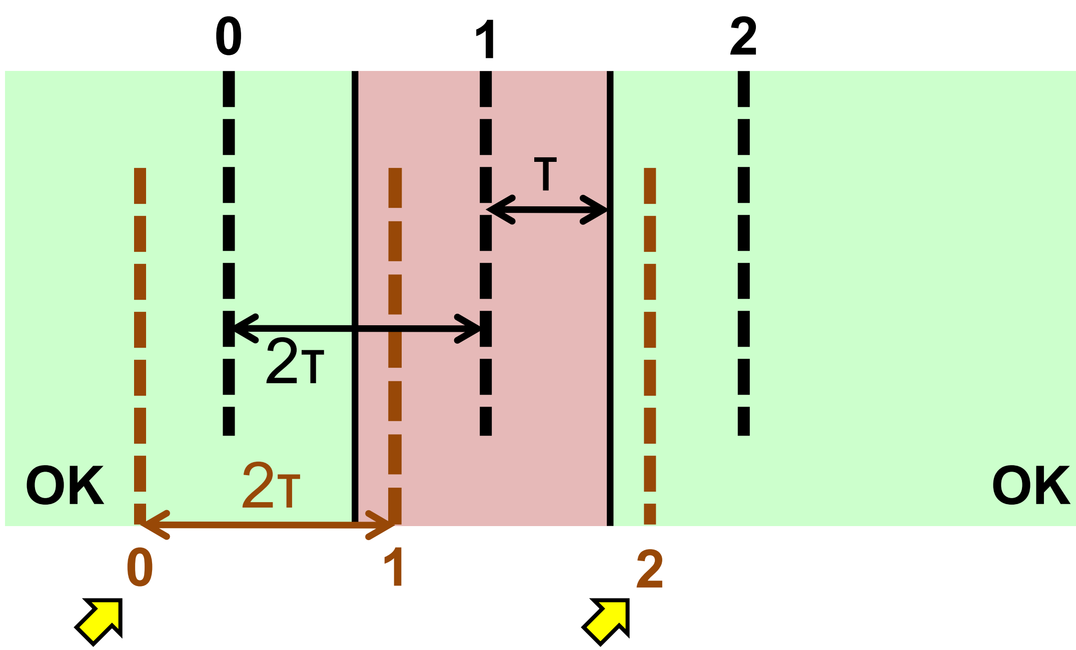
- **No!** Slots are timed off the tail-end of the last packet
  - Therefore, **stations' clocks differ by at most τ**

- 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 $\tau + \tau = 2\tau$

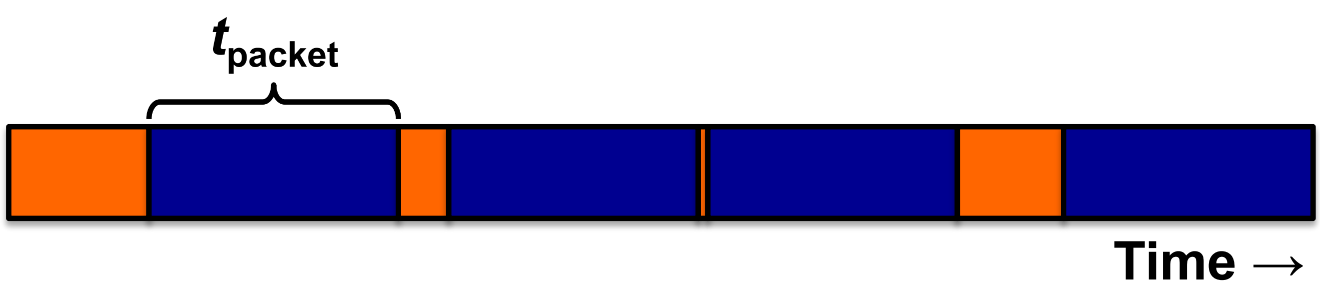


Binary Exponential Backoff

- 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

Ethernet performance analysis

- Divide time into:
  - Variable-sized **contention intervals**,
  - Fixed size **transmission intervals** (duration t_{packet})



Efficiency:

$$\frac{t_{\text{packet}}}{t_{\text{packet}} + \underbrace{(2\tau)}_{\text{slot time}} W}$$

Number of slots to acquire the Ether

Ethernet performance: Acquisition

- What's the probability that **one station acquires** the medium **without a collision?**

- Suppose there are *Q* stations waiting to send

- Assume stations know *Q* and send with probability 1/*Q* (**BEB approximates this**)

- **Slotted ALOHA → 37% probability** of successful acquisition

Ethernet performance: Waiting time

***W* =** number of slots in a contention window before acquisition of the Ether

- Probability of no wait: $p_{acquire}$
- Probability wait one slot: $(1 - p_{acquire})p_{acquire}$
- Probability wait two slots: $(1 - p_{acquire})^2 p_{acquire}$
- E[slots to wait] $= E[W] = (1 - p_{acquire})/p_{acquire}$
  $= e - 1$

Comparing CDMA and 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**

Friday Precept

Introduction to Lab 1

Tuesday Topic:

Link Layer II: MACA and MACAW