Contention-Based Sharing (Ethernet)

Wireless Networks L07

Dr. Ahmed Mahdi Jubair

Today

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

2. **Contention-based sharing**
   - 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

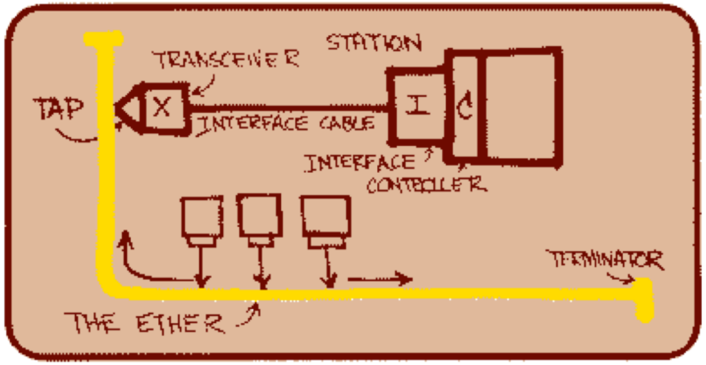
Today

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

2. **Contention-based sharing**
   - Unslotted ALOHA, SlottedALOHA
   - **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 ALOHApacket radio

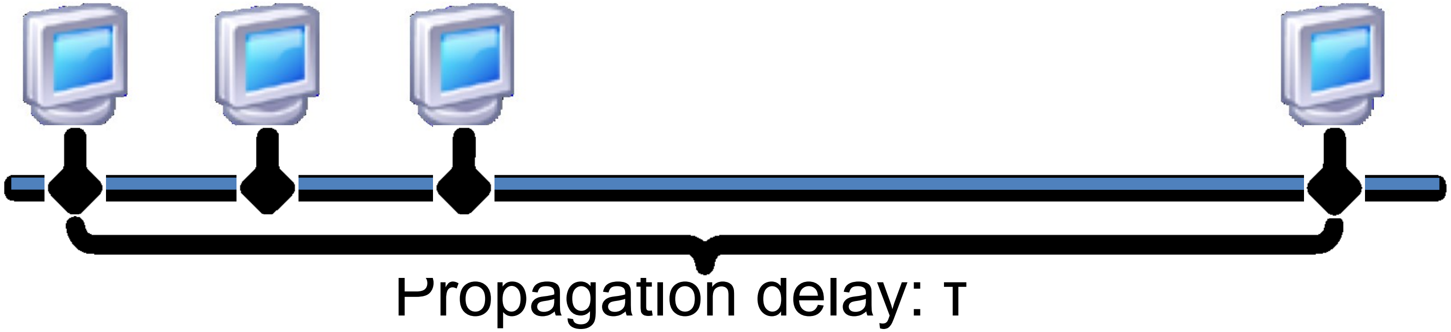
- Metcalfe later founds *3Com*, acquired by HP in April ’10 for USD $2.7 bn

The Ethernet: Physical design

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

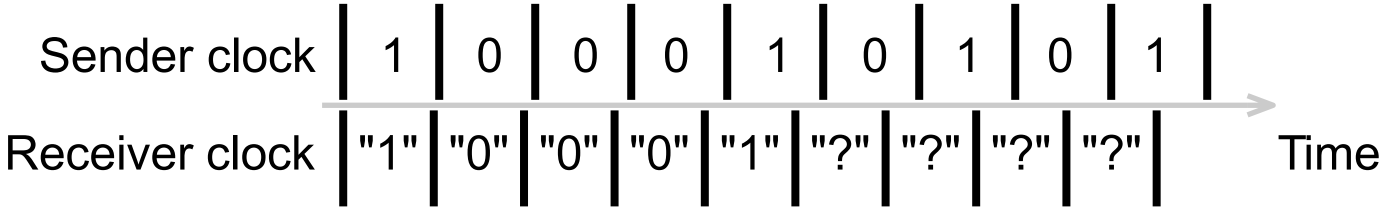
- Experimental Ethernet
  - Data rate: *B* = 3 Mbits/s
  - Maximum length: 1000 m

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



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

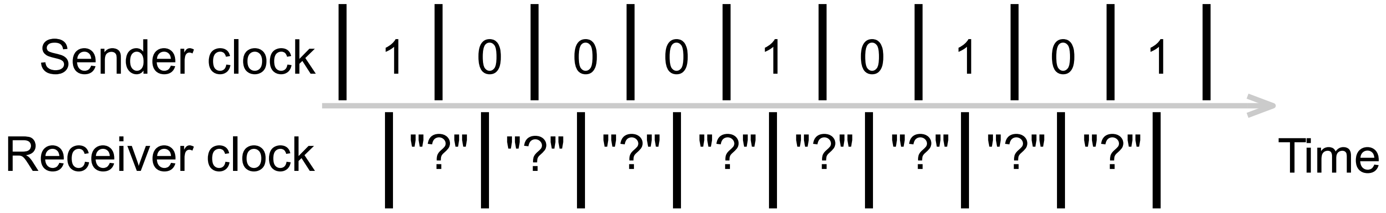


Sender clock | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

Receiver clock |"1"|"0"|"0"|"0"|"1"|"?"|"?"|"?"|"?"|

Time

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

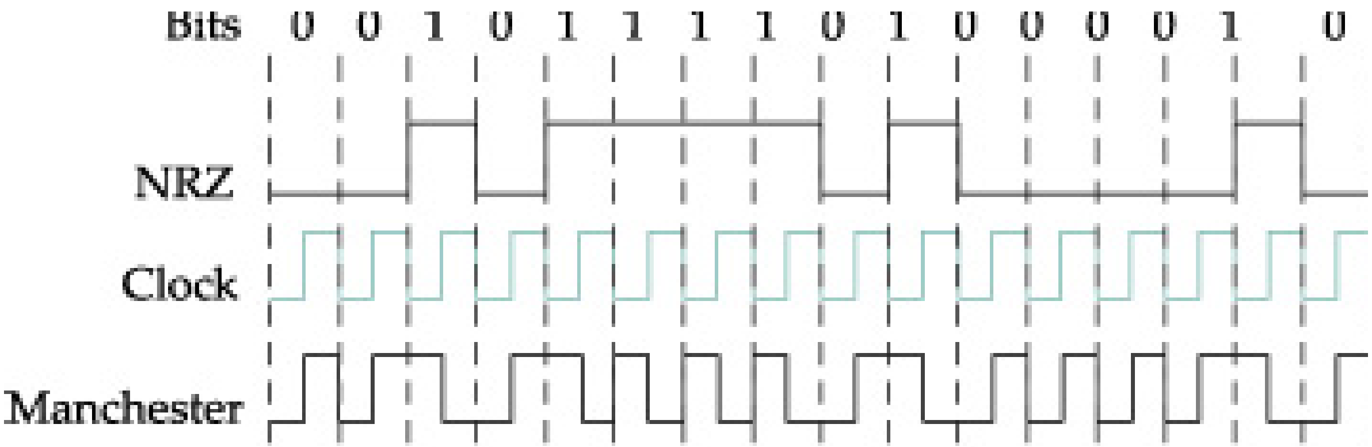


Sender clock | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 0 | 1 |

Receiver clock |"?"|"?"|"?"|"?"|"?"|"?"|"?"|"?"|

Time

Manchester (phase) encoding

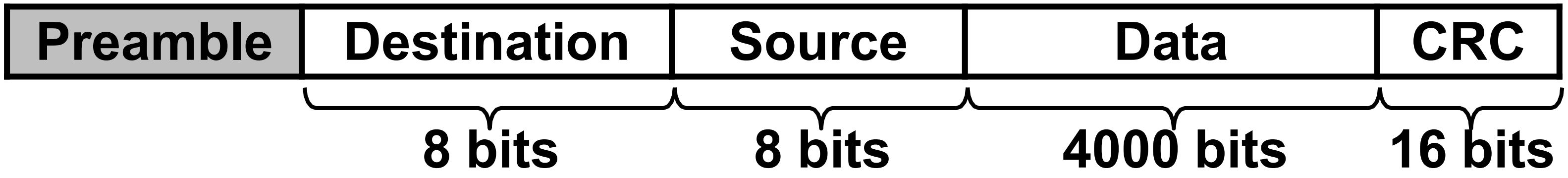


- Manchester encoding:
  - Exclusive-OR of the NRZ signal and the clock signal
  - “0” is a low-to-high transition; “1” is a high-to-low

- Transition guaranteed on every bit

- **Drawback: Halves data rate**

Ethernet framing

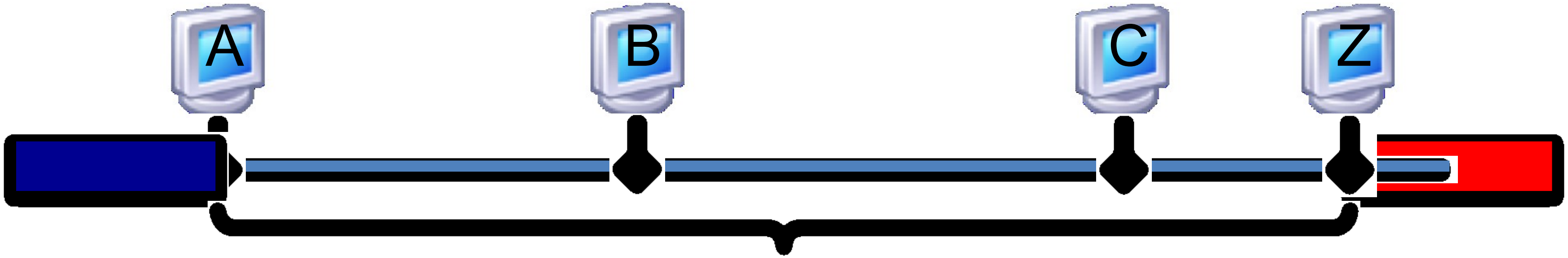


Preamble | Destination | Source | Data | CRC

8 bits | 8 bits | 4000 bits | 16 bits

- 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



Propagation delay: τ seconds

- 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

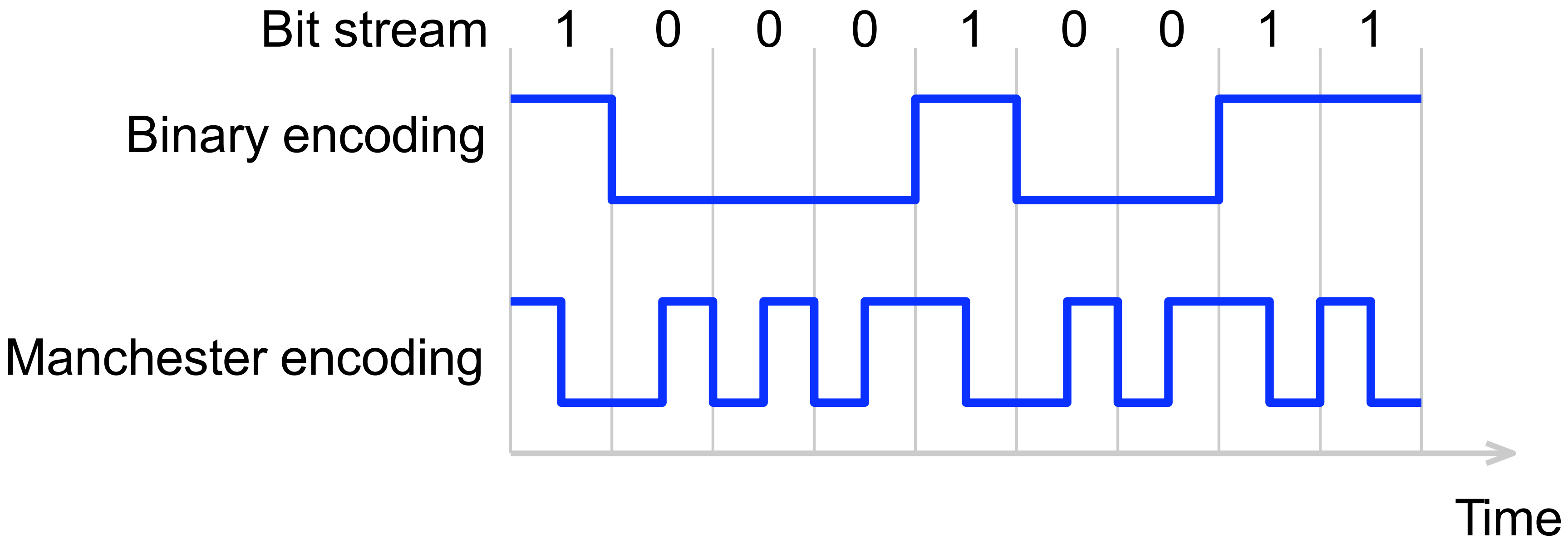
Who gets to transmit, and when?

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

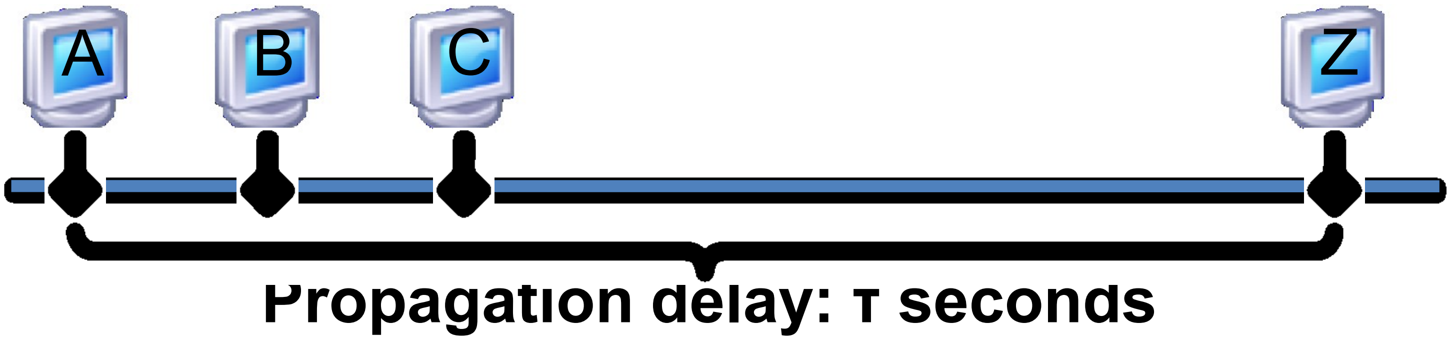
Carrier sensing

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



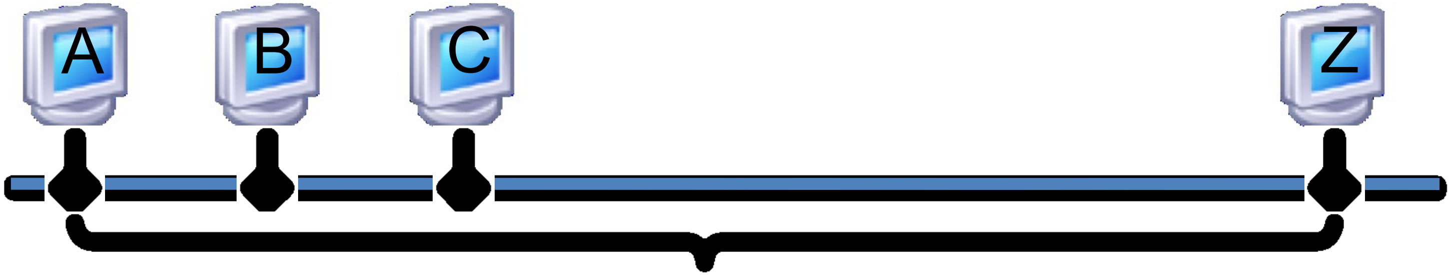
- Manchester coding: constant average voltage

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 $V_{\text{measured}} > V_0$

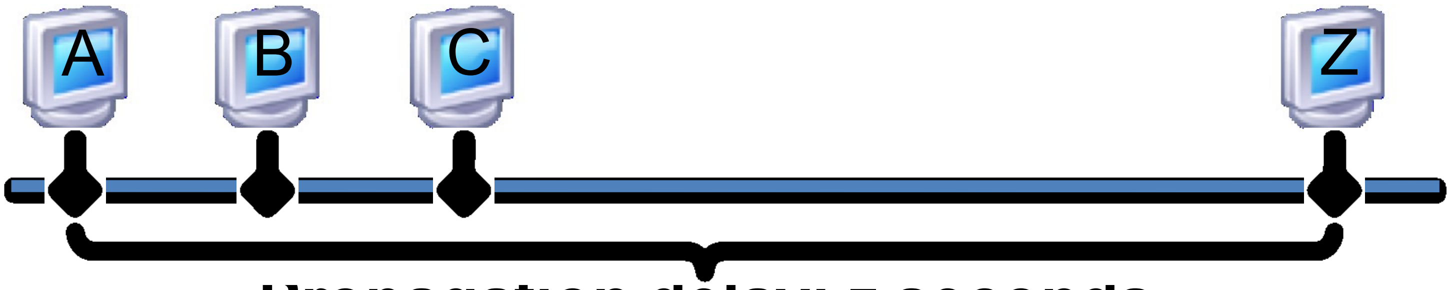
When might a collision happen?



Propagation delay: τ seconds

- 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

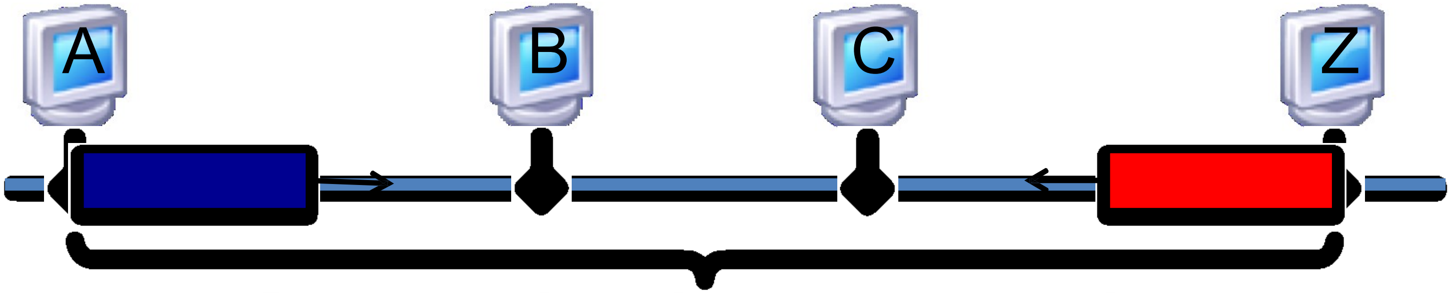
How long does a collision take to detect?



Propagation delay: τ seconds

- 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

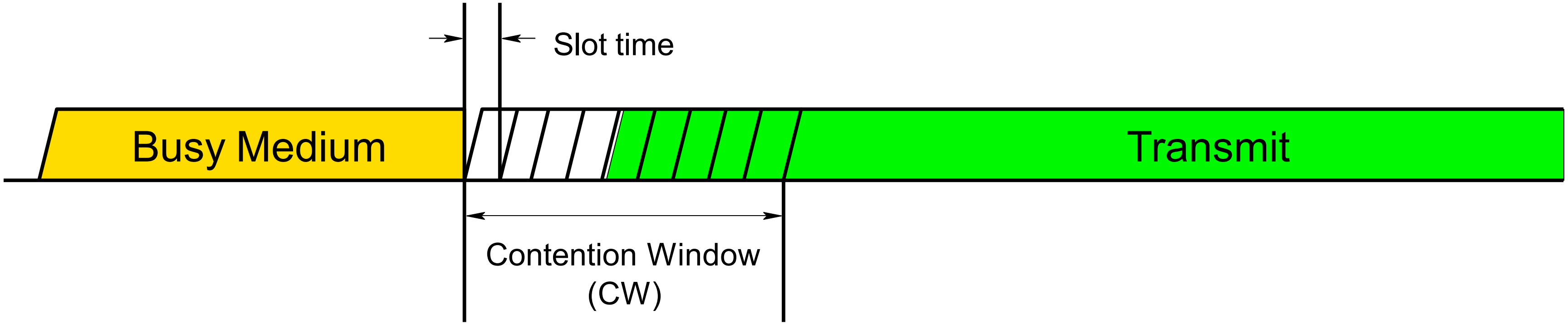
- Transmit rate *B* bits/second
- If packets take time 2τ, A will still be transmitting when Z's packet arrives at A, so A **will detect collision**
  - So **minimum packet size = 2τ*B* bits**
    - Experimental Ethernet:
      - τ = 5 μs, *B* = 3 Mbits/s → 2τ*B* = 30 bits

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)***



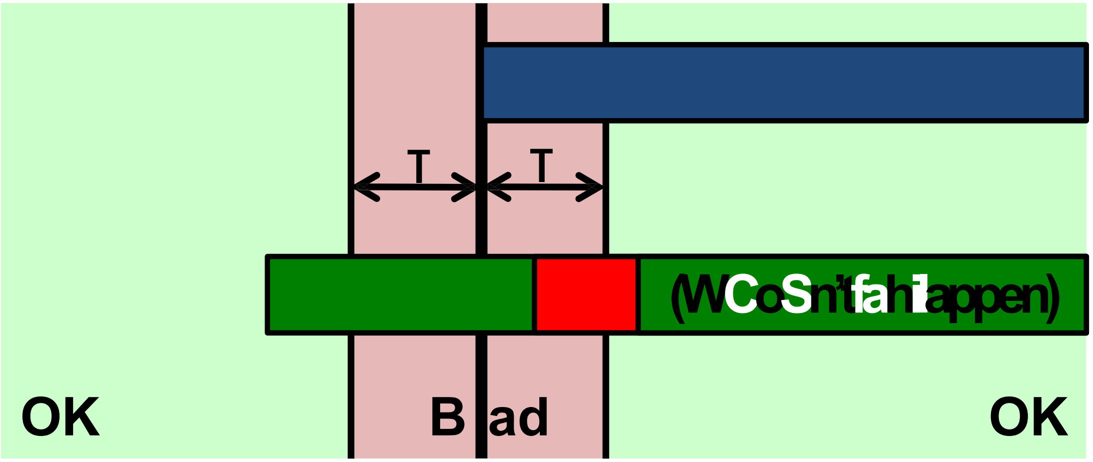
- **Goal:** Choose slot time so that **different** nodes picking **different** slots CS and defer → **don't collide**

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**

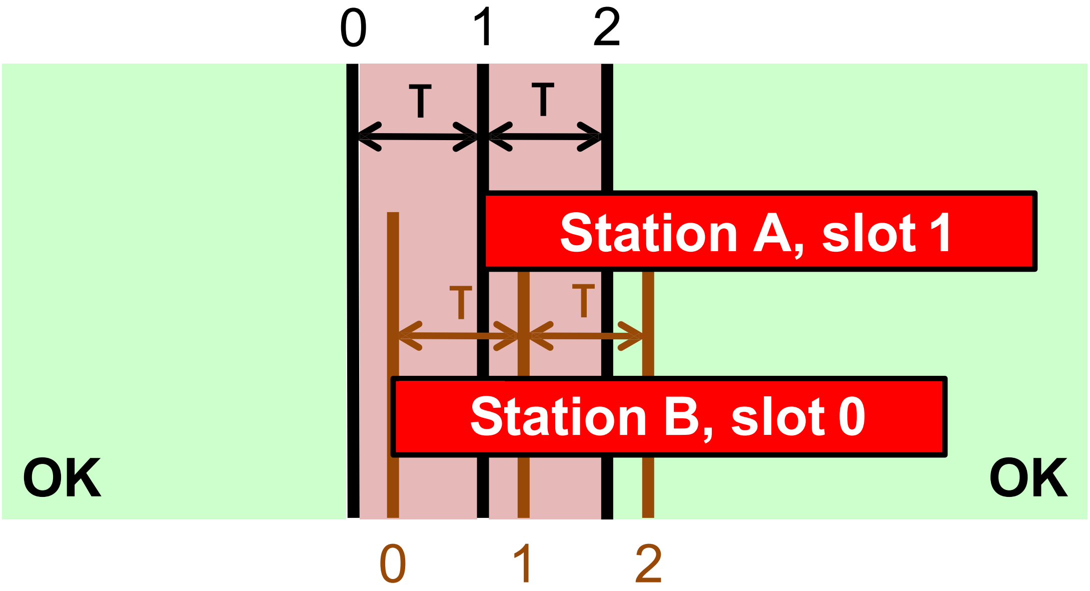
- *So should we pick a backoff slot length of τ?*



The problem of clock skew

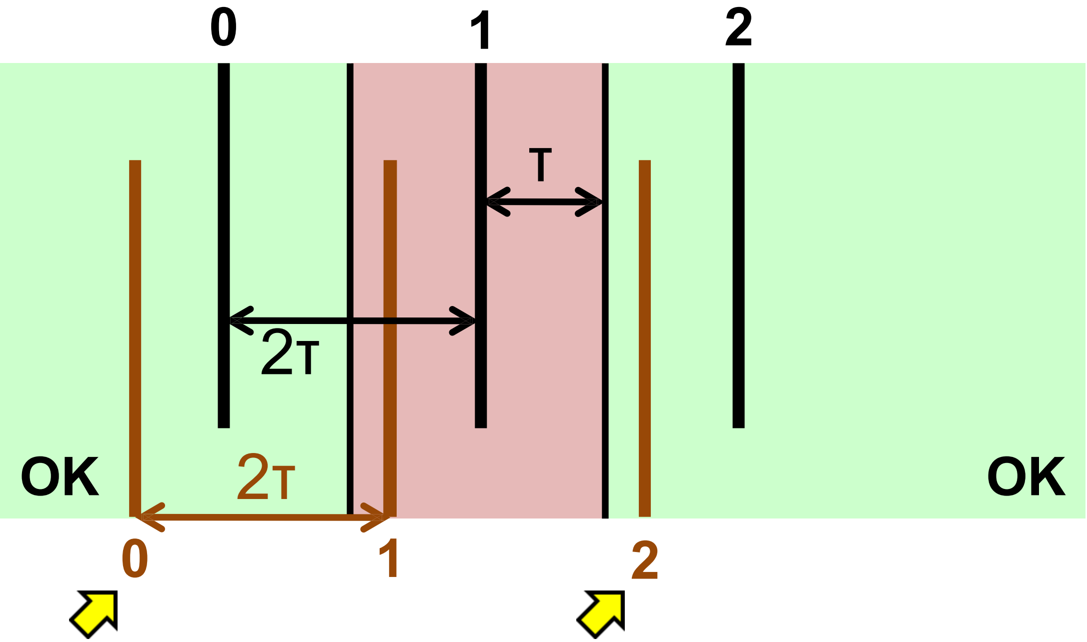
- **No!** Slots are timed off the tail-end of the last packet
  - Therefore, **stations' clocks differ by at most τ**

- 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

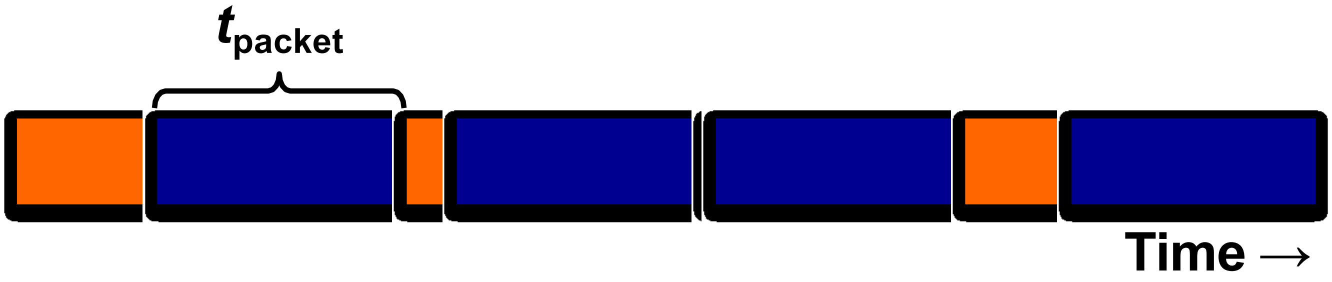
- Binary exponential backoff (**BEB**): double CW size on each consecutive collision

- 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: $(1 - p_{\text{acquire}})p_{\text{acquire}}$
- Probability wait two slots: $(1 - p_{\text{acquire}})^2 p_{\text{acquire}}$
- E[slots to wait] = E[W] = $(1 - p_{\text{acquire}})/p_{\text{acquire}}$

  $= e - 1$

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**