CS 3640: Introduction to Networks and Their Applications

Fall 2023, Lecture 6: Error handling in the link layer

Instructor: Rishab Nithyanand
Teaching Assistant: Manisha Keim

Coming up

- Assignment 2 releasing on Thursday.

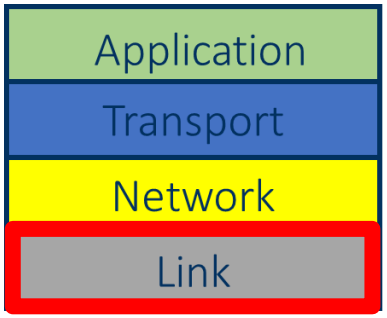
Today's class

1. Recap: Role of the link layer

2. Error handling in the link layer

3. Overview of medium access control protocols

Recap: Overview of the link layer

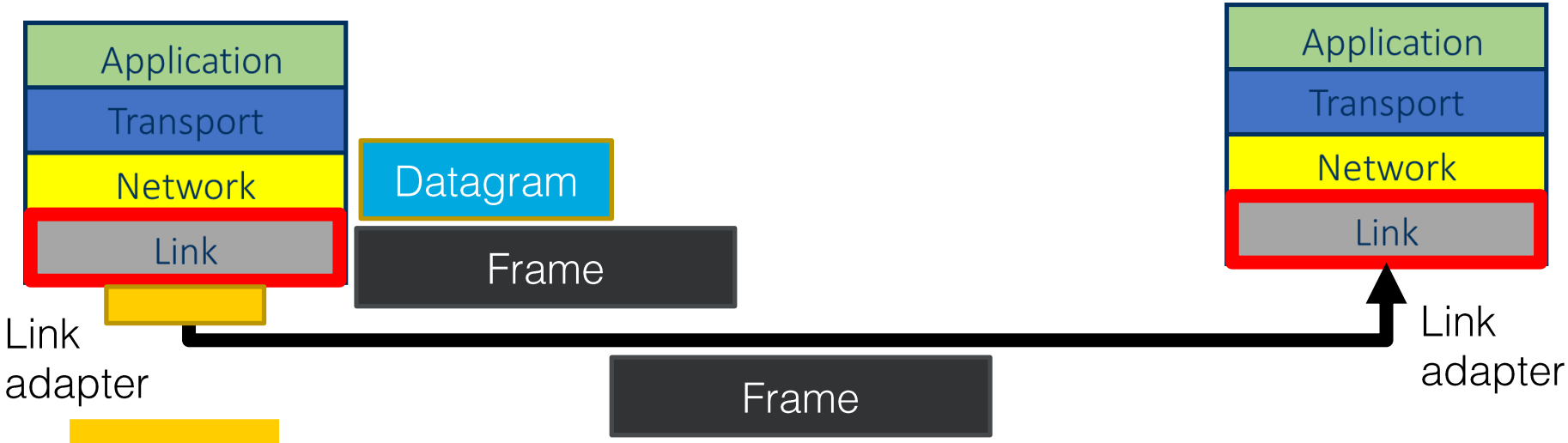


- **What are the functions of the link layer?**
  - Write bits to and read bits from the network interface.
    - How does the link layer perform error detection and correction?
    - How do multiple devices share a link?

- **The link layer protocol can be different things on different devices.**
  - Can be Ethernet on one hop, WiFi on the next.
  - They are usually implemented on an “adapter” or Network Interface Card (NIC).

Recap: Overview of the link layer

- **How do adapters communicate?**
  - At the sender's side:
    - Link layer gets a datagram from Network layer.
    - Figures out which interface to use for the network layer destination.
    - Encapsulates datagram with link layer header. (Frame)
    - Adds error checking (or, correction) data to header so the receiver may identify if errors occurred.
    - Writes frame to link.



Recap: Overview of the link layer

- **How do adapters communicate?**
  - At the receiver's side:
    - Link layer gets a frame from the link.
    - Looks for errors that may have occurred during transmission (i.e., only errors introduced by the link layer) and corrects them if required/possible.
    - Strips link layer headers and passes the datagram up to the network layer.

Elements in a link-layer header

- **MAC source and MAC destination**: These are basically link-layer “addresses” for devices connected to the same medium. Every NIC has a unique 48-bit MAC address. [more next week]

- **Payload length**: The size of the data being transported in the frame (includes headers from above layers). Different link-layer protocols have different restrictions on max size of frame.

- **Error correction data**: Usually a 32-bit sequence aimed at identifying when errors may have occurred in the payload. [more in this lecture]

Today's class

1. Recap: Role of the link layer

2. Error handling in the link layer

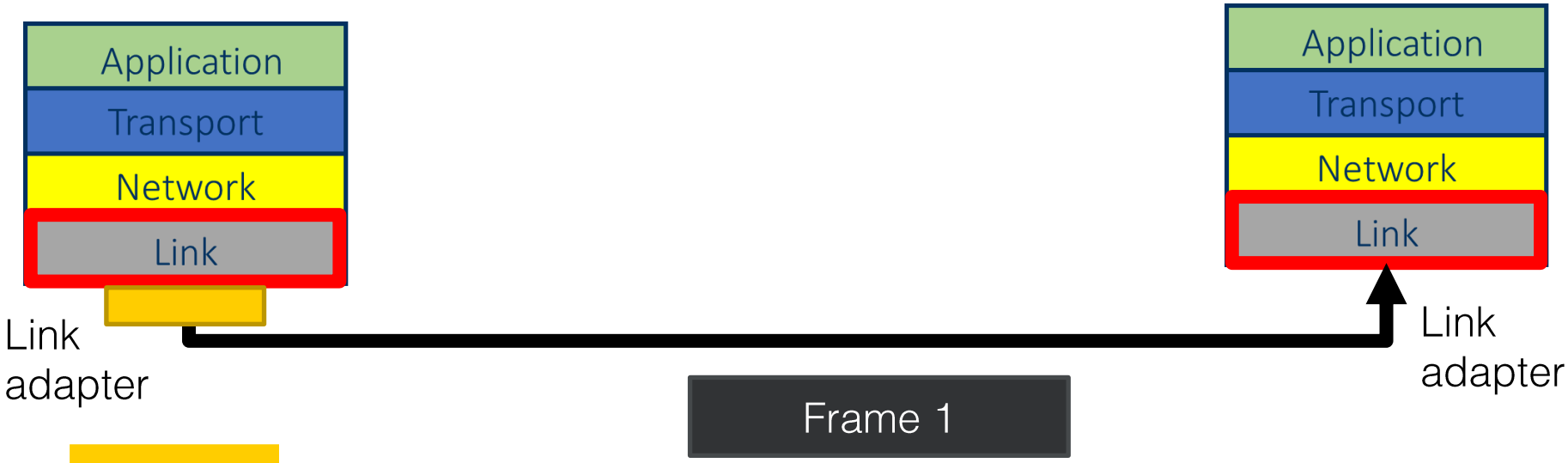
3. Overview of medium access control protocols

Error detection and correction

- The link layer may perform error detection and correction on frames when possible and if asked to do so.
- Why?
  - The physical world is inherently noisy. E.g., interference from radio transmissions and microwaves.
  - **Discuss: Is this a violation of the end-to-end principle? When is it OK?**

Error detection and correction

- **Discuss: How would you detect errors in frames?**
  - Hint: Redundancy might help.



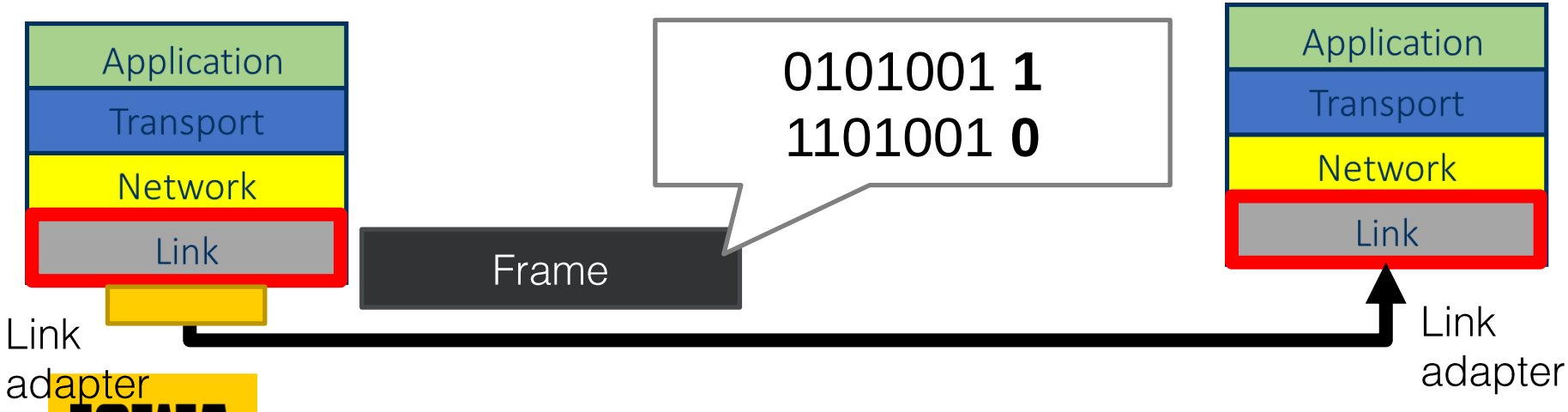
Error detection and correction

- **A simple solution:** Send multiple copies of each frame. An error has occurred if copy 1 != copy 2 != … != copy n.

- **Problems**
  - Overhead!
  - n transmissions for each frame.

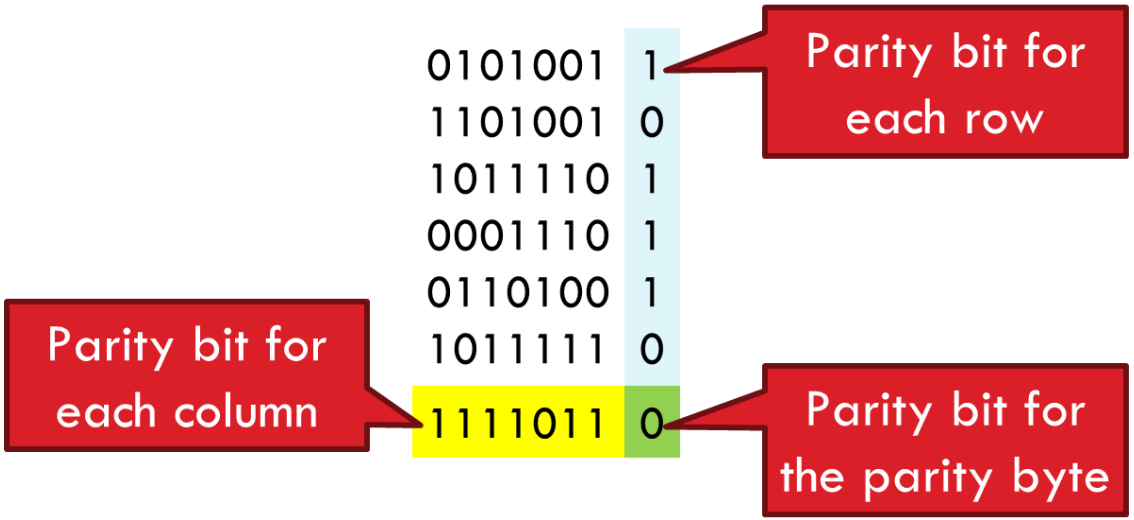
Error detection and correction

- **A better idea: Parity bits**
  - Add extra bits to make sure number of 1s is even.
  - Example: 7-bit ASCII chars. Make 8th bit the parity bit.
  - If number of 1s in any 8bit sequence is odd, error occurred.
  - Overhead: 1 parity bit for every 7 frame bits (14%).
- **Problem?**
  - **Discuss:** What's the minimum number of bits that need to change for this error detection mechanism to fail? Can it be used for error correction?



Error detection and correction

- **A better idea: Two-dimensional parity bits**
  - Imagine your frame as a two-dimensional array of bits.
  - Add one parity bit for each row and for each column.
  - Can detect **most** errors with only marginally higher overhead
    - **Discuss:** What's the minimum number of bits that need to change for this error detection mechanism to fail? Can it be used for error correction?



Error detection and correction

- **An even better idea: Checksums**
  - What if you summed up all the bytes in your frame.
  - Included the sum in the frame.
  - Requires only 16 bits overhead for the whole frame! (~1% overhead)

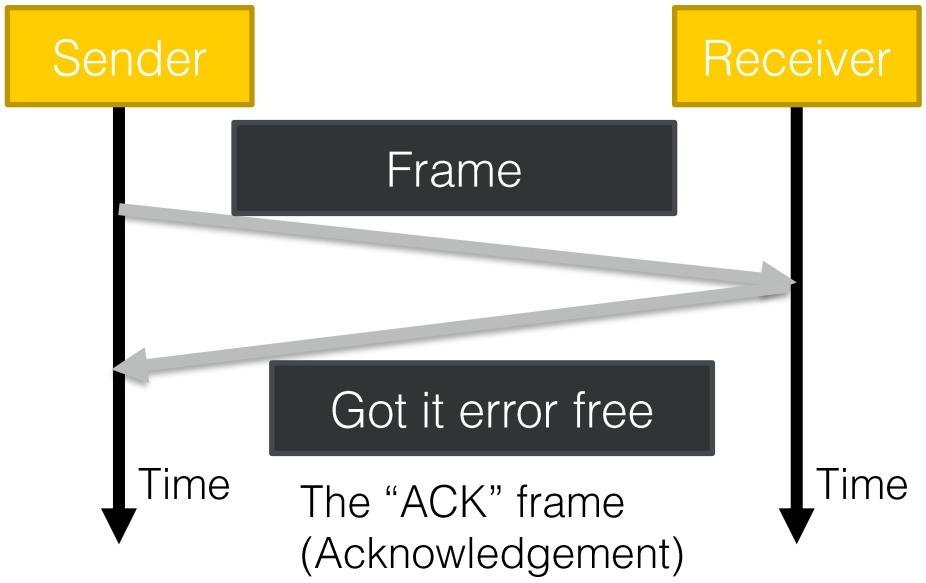
  - This is used in the transport layer and IP layer.
  - **Problem**: Still can have errors because of sum collisions.
    - Example (data): 10000000 + 01111111 = 11111111
    - Error: 11000000 + 00111111 = 11111111

Error detection and correction

- **What we actually use in the link layer: Cyclic Redundancy Check (CRC)**
  - A more complicated version of checksum.
  - **Idea**: Perform a series of mathematical operations on the frames data bits to get a semi-unique value. E.g., compute a cryptographic hash of all the data bits.
  - Typically, 32 bit overhead per frame.
    - Chance of error = chance of hash collisions for a given frame ~ $1/2^{32}$
    - Very cheap to implement in hardware.
    - Very quick.

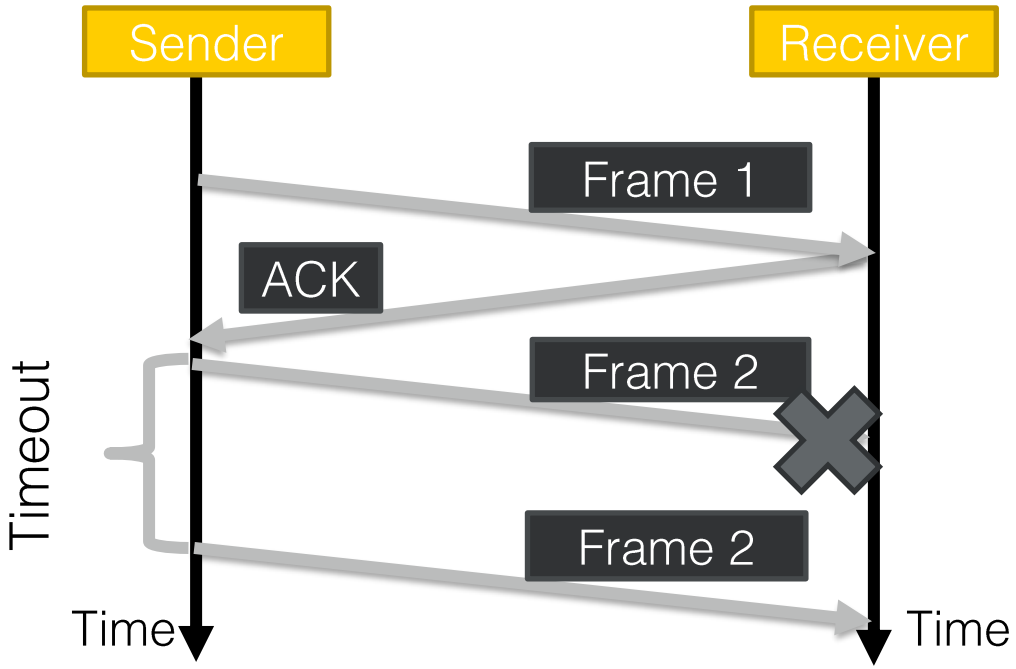
Signaling errors to the sender

- **How does a sender know if a frame was:**
  - Received error free?
  - Received at all?
  - **Discuss:** How do you do this IRL?



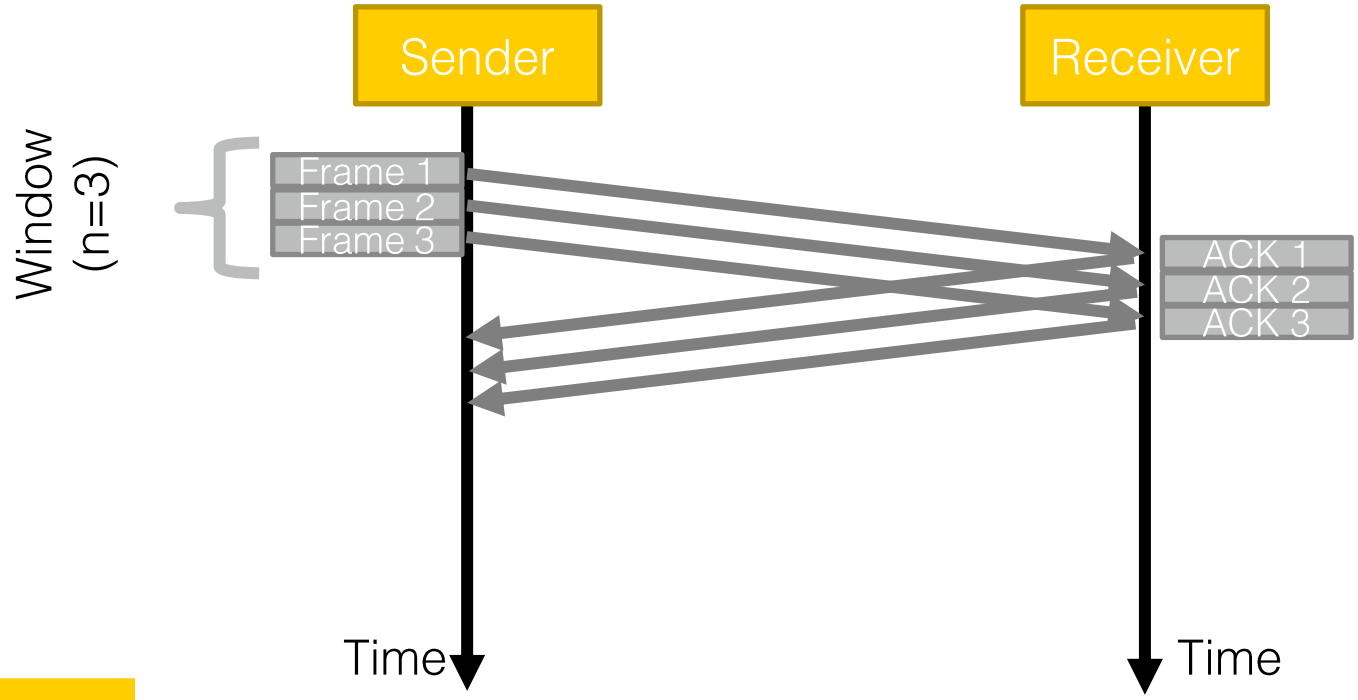
Signaling errors to the sender

- **The Stop and Wait protocol**
  - I'll wait for an ACK before I send the next frame. If I don't get an ACK in x ms, I'll resend the frame.
    - Used by Bluetooth and other low power/bandwidth protocols.
  - Problem: Poor efficiency. At best, only one frame every RTT seconds.
  - **Discuss**: How would you improve the efficiency of this protocol.



Signaling errors to the sender

- **The Sliding Window Protocol**
  - We can allow multiple frames to be unacknowledged.
    - This is the “window”.
  - Better efficiency: Can allow n frames per RTT.



Today's class

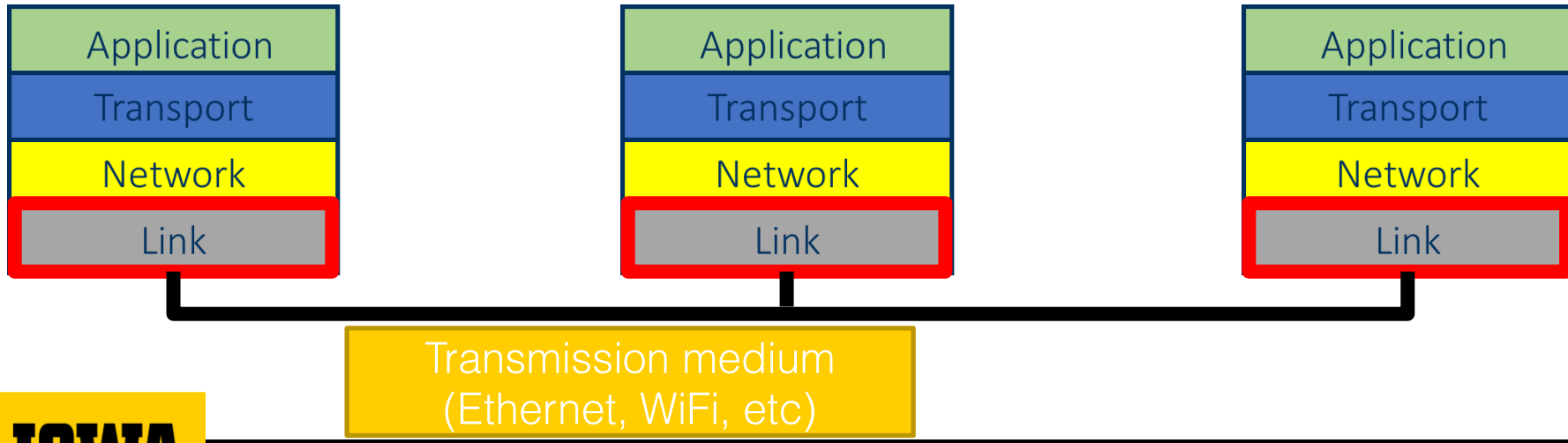
1. Recap: Role of the link layer

2. Error handling in the link layer

3. Overview of medium access control protocols

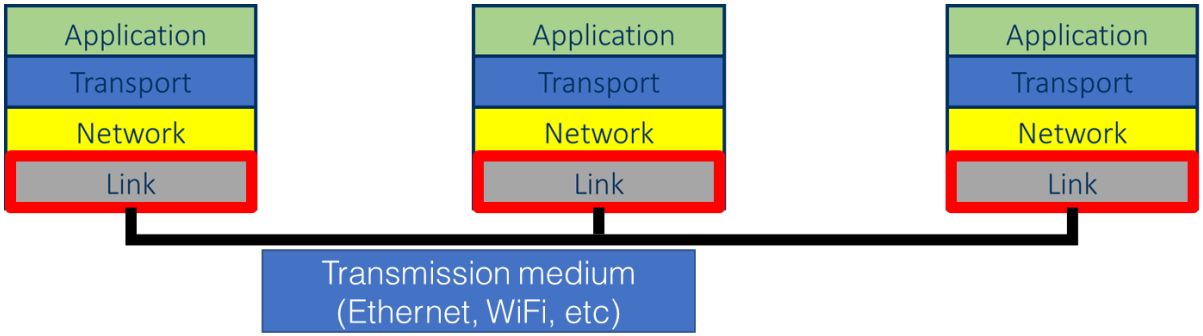
Medium Access Control (MAC) protocols

- **What is a MAC protocol?**
  - A transmission medium can be shared by many devices.
  - If everyone talks at the same time, we have “collisions” and unintelligible data.
  - MAC: Rules for sharing a common transmission medium.



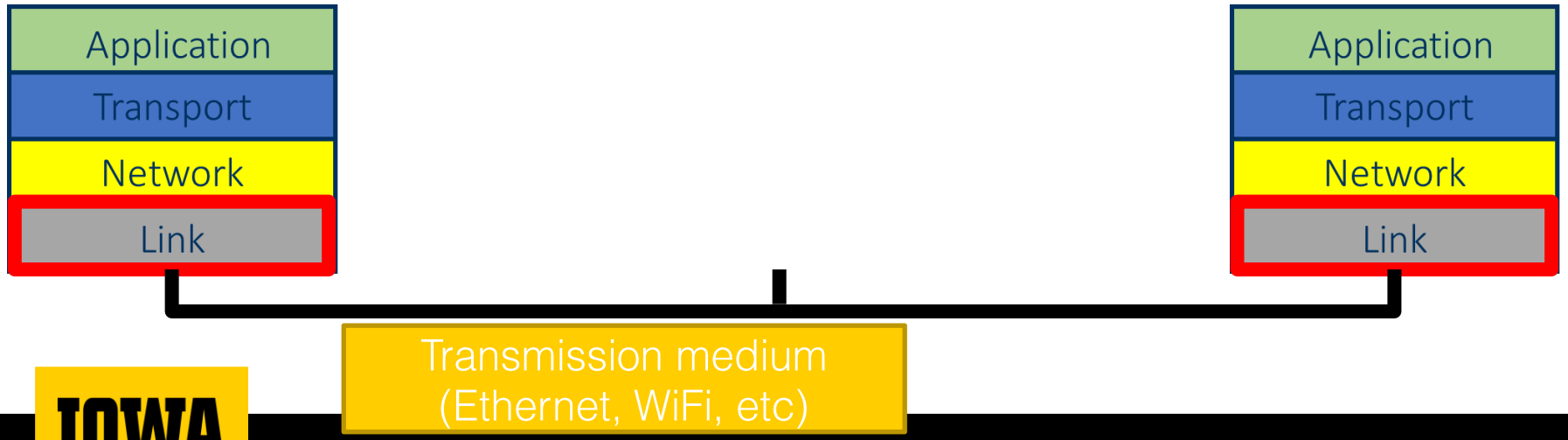
Medium Access Control (MAC) protocols

- **General strategies for MAC protocols**
  - Idea 1: Partition the transmission channel so each host has its share.
    - We briefly saw this – Time and Frequency division. Each host has a fixed share (time or frequency band) in the medium.
    - What if a host has nothing to send?
  - Idea 2: Pass a "transmit now" token to hosts.
    - Like me cold-calling someone to answer a question. (I'm giving you the token).
    - What if multiple people really want to give an answer?
  - **Problem**: Transmission channel utilization isn't great.



Medium Access Control (MAC) protocols

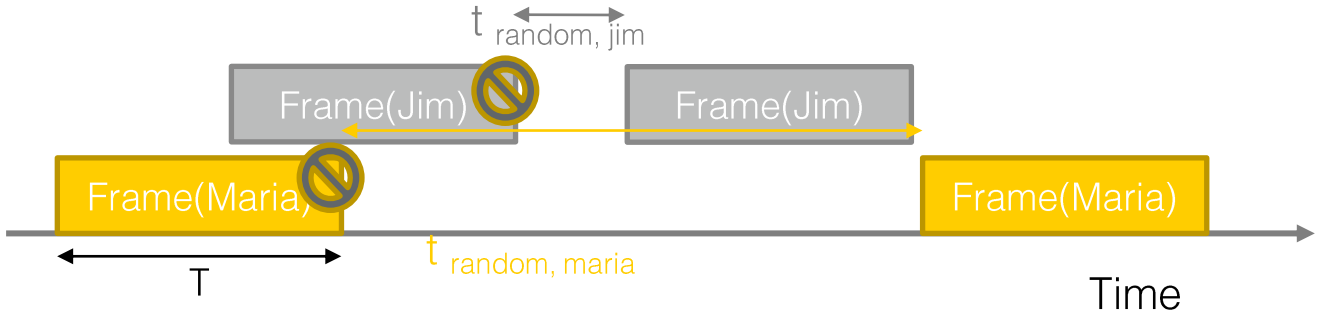
- **General strategies for MAC protocols**
  - Idea 3: Allow collisions, we'll figure out how to recover data.
    - Allows much higher utilization.
    - Now we have new problems:
      - How to identify when a collision has occurred.
      - How to recover from a collision.
    - This strategy is called "Random access MAC" or "Contention-based MAC".
    - Used by Ethernet, mobile transmission protocols, and others.



The ALOHA MAC protocol (circa 1970)

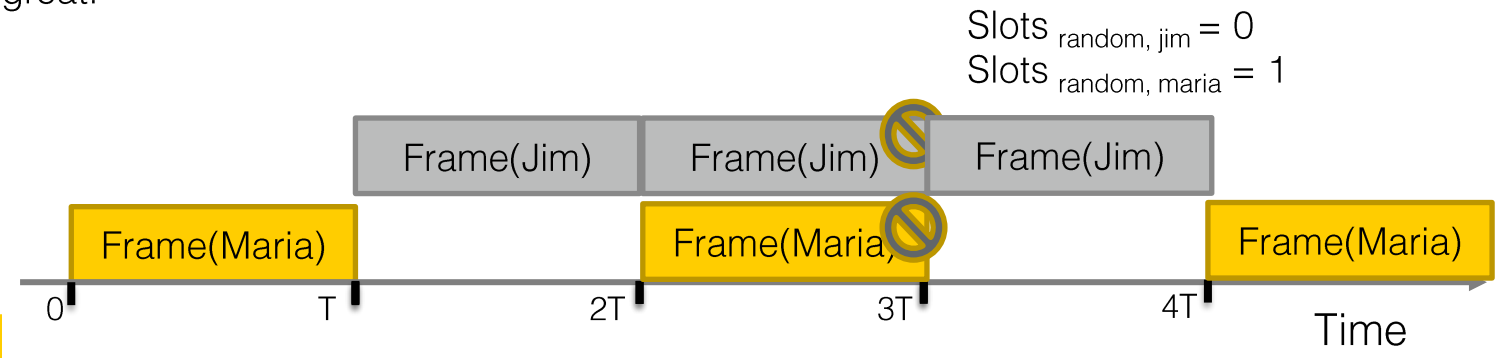
ALOHA: Additive Links On-line Hawaii Area (developed at UHawaii)

- **Idea: Send a frame as soon as you need to.**
  - If you receive a frame while transmitting, a collision has occurred.
    - Remember everyone using the same medium will receive all transmissions on that medium.
  - Wait for some random time and transmit again.
  - Eventually, a frame will get transmitted without collisions.
- **Assumptions:**
  - All frames are equally sized.
  - Errors (collisions) are detectable.
- **Problem**: For successful transmissions, no other frame from any other host should start within T time before or after you.
  - Sensitive transmission period: 2T for each frame.
  - Scales terribly. (Theoretical maximum throughput for large number of hosts with random transmission times: 18%).



The Slotted ALOHA MAC protocol (circa 1970)

- **Idea: If we allow transmissions only at certain time points, the "sensitive" period for transmissions reduces.**
  - Transmissions on the channel can occur only every T seconds.
    - The channel is divided into slots, but anyone can send in any slot.
  - Sensitive transmission period: T for each frame.
  - If a collision occurs, wait some random number of time slots and retransmit.
- **Assumptions:**
  - All frames are equally sized.
  - All host clocks are synchronized (!!!)
  - Errors (collisions) are detectable.
- Sensitive period reduced from 2T to T.
  - Theoretical maximum throughput is doubled to 36%.
  - Still not great.



What you should remember from this lecture

- **What is the role of the link layer? Where is it implemented?**
- **How is error detection/correction done at the link layer?**
  - What are the different approaches?
    - General ideas and considerations for selection of an approach.
  - When is it not violating the end-to-end principle?
- **Why is stop-and-wait less efficient than the sliding window?**
- **What function do MAC protocols provide?**
  - Why is slotted ALOHA better than ALOHA?
- **There is a general theme of: “We can take care of the worst-case later”.**
  - The Internet is a best-effort service.