Introduction to Computer Networks

CS640

Error Detection and Ethernet

https://pages.cs.wisc.edu/~mgliu/CS640/F21/

Ming Liu
mgliu@cs.wisc.edu

Today

Last lecture

- Encoding and Framing

Today

- Error detection
- Ethernet

Announcements

- Lab1 due on 09/30/2021 11:59PM

Error Coding

Transmission may introduce errors into a message

- Single bit errors v.s. burst errors

Error Detection

- Require a check that messages are invalid
- Require redundant check bits/bytes
- **Goal: maximize the probability of detecting errors using only a small number of redundant bits**

Error Correction

- Forward error correction: many related code words map to the same data word
- Detect errors and retry transmission

Parity

Even parity

- Append parity bit to 7 bits of data to make an even number of 1's
- Odd parity accordingly defined

1010100	1
1010101	0

Parity

Even parity

- Append parity bit to 7 bits of data to make an even number of 1's
- Odd parity accordingly defined

1010100	1
1010101	0

1 in 8 bits of overhead

Parity

Even parity

- Append parity bit to 7 bits of data to make an even number of 1's
- Odd parity accordingly defined

Data	Parity
1010100	1
1010101	0

1 in 8 bits of overhead

Can detect a single error

Data	Parity
1010101	1
1000010	0

But nothing beyond that

2-D Parity

Make each byte even parity

Finally, a parity byte for all bytes of the packet

Example: five 7-bit character packet, even parity

0110100
1011010
0010110
1110101
1001011

2-D Parity

Make each byte even parity

Finally, a parity byte for all bytes of the packet

Example: five 7-bit character packet, even parity

0110100	1
1011010	0
0010110	1
1110101	1
1001011	0

2-D Parity

Make each byte even parity

Finally, a parity byte for all bytes of the packet

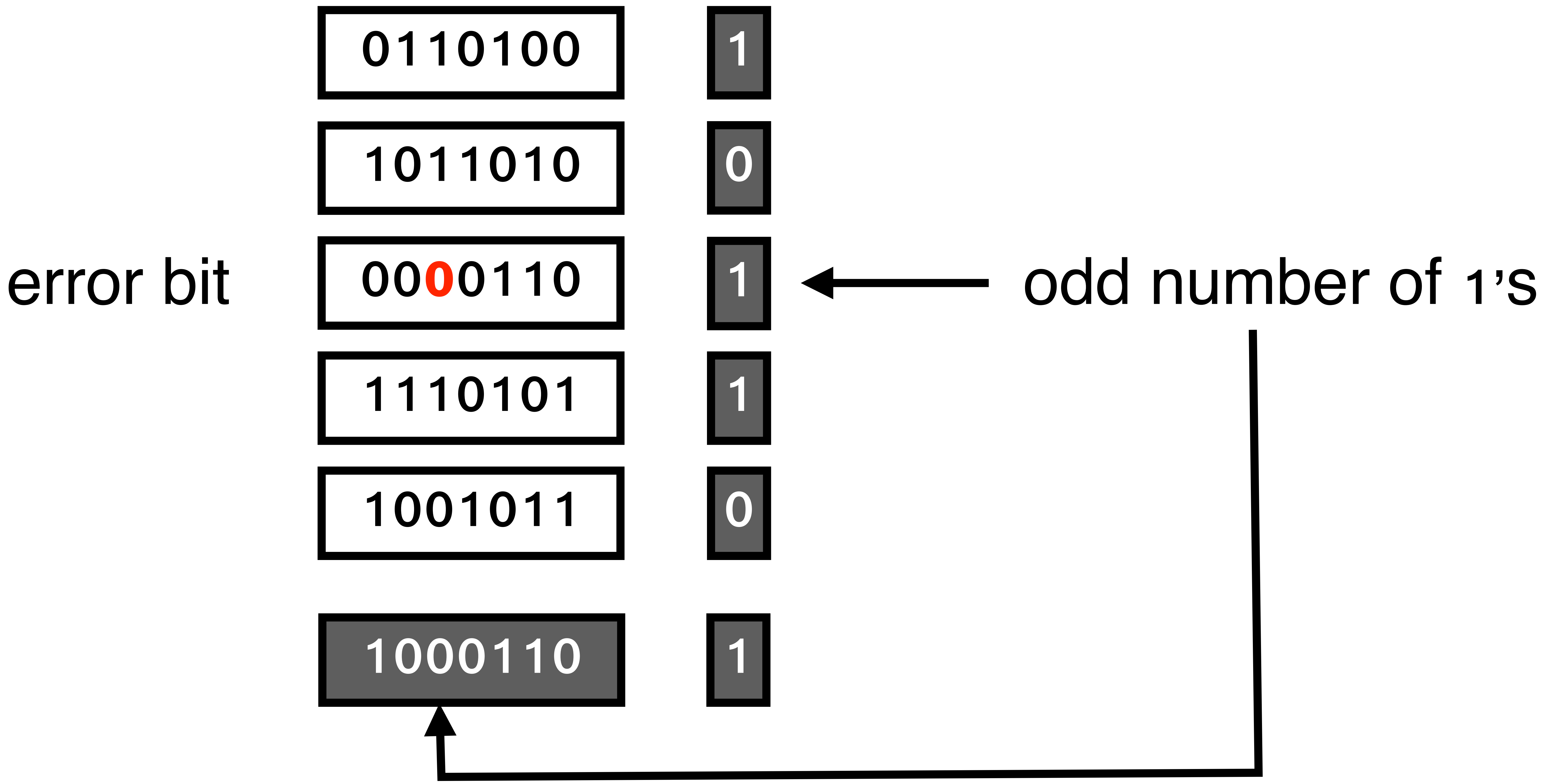
Example: five 7-bit character packet, even parity

0110100	1
1011010	0
0010110	1
1110101	1
1001011	0
1000110	1

Effectiveness of 2-D Parity

1-bit errors can be detected and corrected

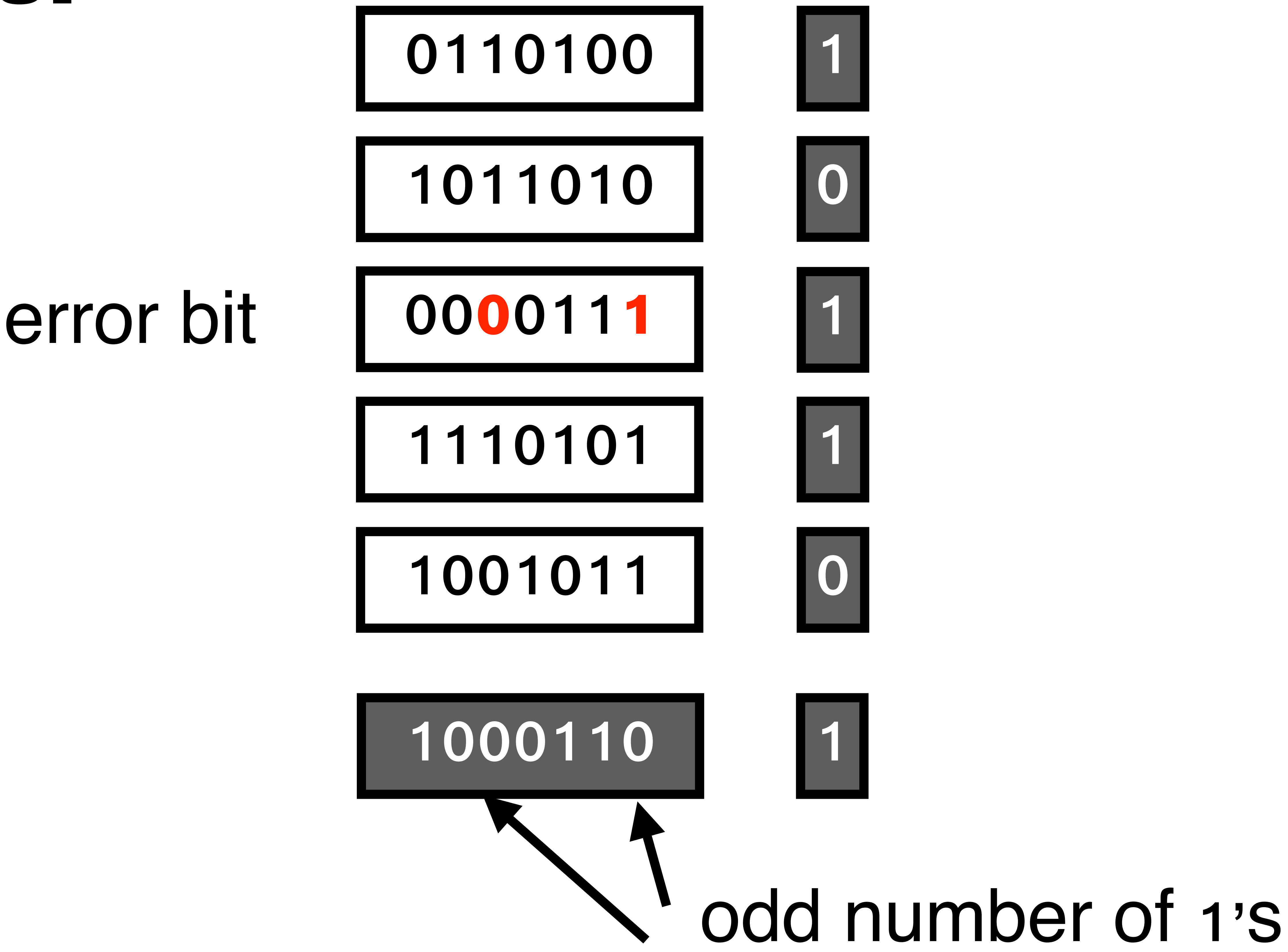
Example with even parity per byte:



Effectiveness of 2-D Parity (cont'd)

2-bit errors can also be detected

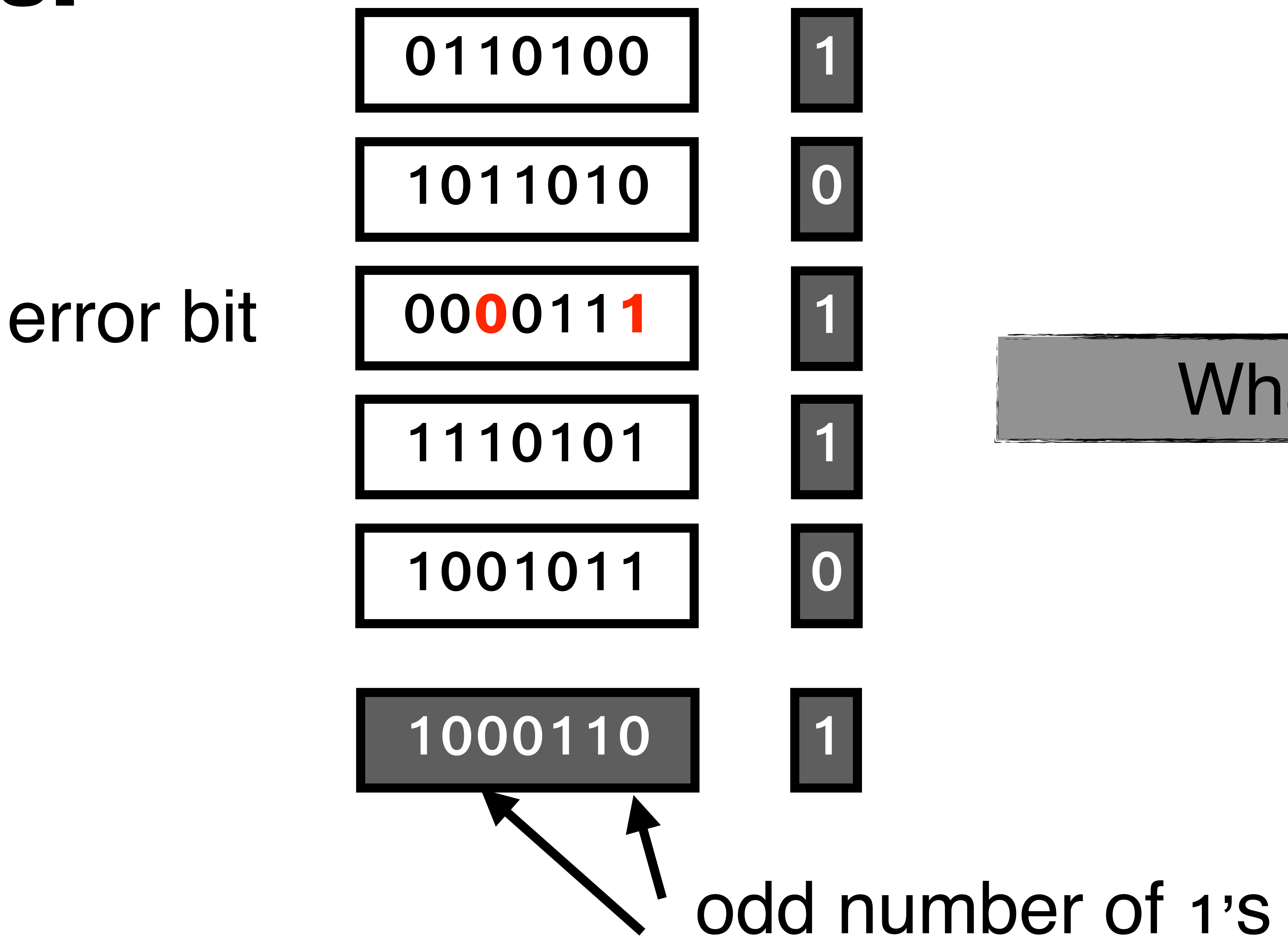
Example:



Effectiveness of 2-D Parity (cont'd)

2-bit errors can also be detected

Example:



What about 3+ bit errors?

Internet Checksum

Checksum = add up all the words that are transmitted

- Receiver performs the same calculation on the recovered data
- E.g., ones complement arithmetic

Simple but not robust

- 16 redundant bits for the whole message
- Easy to be implemented in the software
- Concurrent errors without hurting the sum cannot be detected => low probability

Cyclic Redundancy Codes (CRC)

Commonly used codes that have good error detection properties

- Ethernet frame check sequence (CRC-32)

Based on division of polynomials

- Errors can be viewed as adding terms to the polynomial
- Should be unlikely that the division will still work

Example

- CRC-32: Ethernet
- CRC-8, CRC-10, CRC-32: ATM
- CRC-5: USB token packets

CRC More

Send and receiver agree on the same divisor polynomial

- Hard to select

Can be implemented very efficient in hardware

- k-bit shift register and XOR gates

Can also be used to correct a small number of errors

CRC More

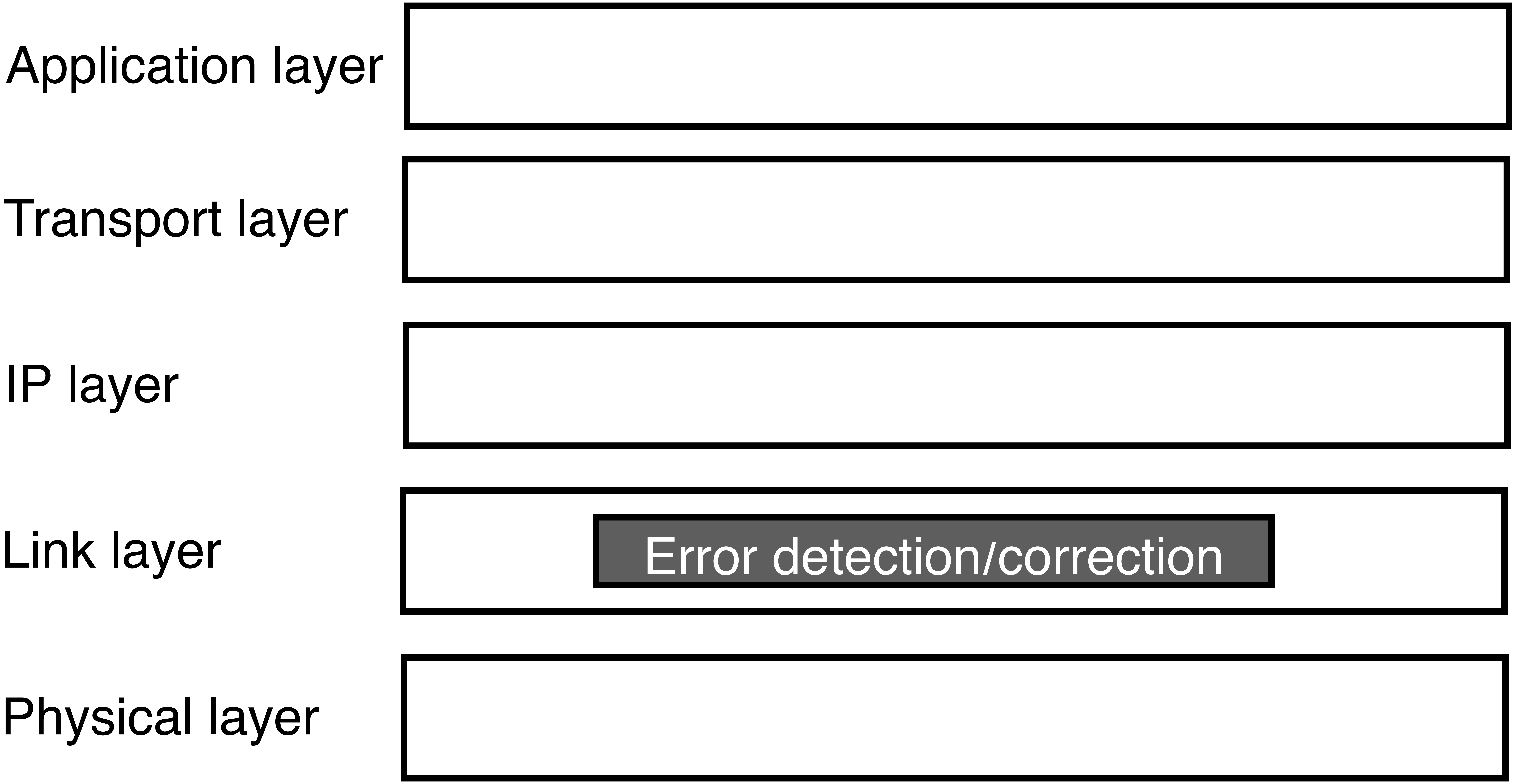
Send and receiver agree on the same divisor polynomial

- Hard to select

ECE/MATH 641 — Introduction to Error-Correcting Codes

Can also be used to correct a small number of errors

Error Detection in Layering



Error Detection in Layering

Application layer — Error detection/correction

Transport layer — Error detection/correction

IP layer — Error detection/correction

Link layer — Er

Physical layer — Er

Failure location

Protocol processing efficiency

......

Multiple Access Protocols

Prevent two or more nodes from transmitting at the same time over a broadcast channel

- If they do, we have a collision, and receivers will not be able to interpret the signal

Several classes of multiple access protocols

- Partitioning the channel, e.g., frequency-division or time division multiplexing
- Taking turns, e.g., token-based, reservation-based protocols, polling based
- Contention based protocol, e.g., Ethernet

Desirable MAC Properties

Broadcast channel of capacity R bps

- 1 node -> throughput = R bps
- N nodes -> throughput = R/N bps, **on average**
- Decentralized -> scalability
- Simple and inexpensive -> cost efficiency

Ethernet

Medium access layer's service interface to higher layers:

- Arbitrate access to a shared link in a fair and efficient way
- Move packets across a switched network

Ethernet: one of the most popular LAN technologies

- Designed for multiple access over a shared medium
- Focus on medium access part and discuss switching part later

Ethernet today

- largely point to point
- Campus, enterprise, and data center network

Key Idea of Ethernet

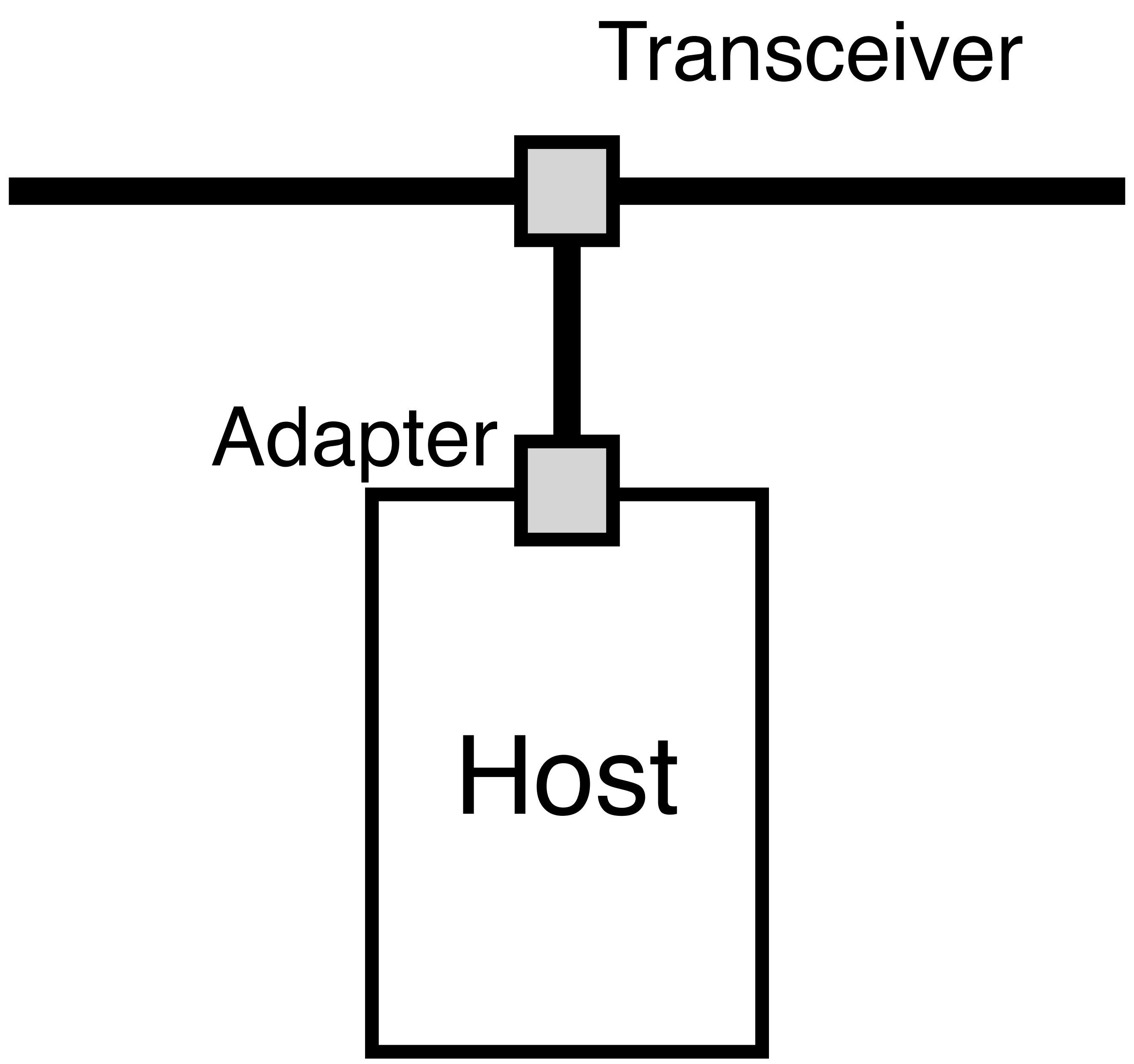
Access the channel in a “random” fashion — when collisions occur, recover

- Each node transmits at highest rate of R bps
- Collision: two or more nodes transmitting at the same time
    - Each node retransmits until collided packet gets through
- Key: don’t retransmit right away
    - Wait a random interval of time first

Ethernet Physical Layer

10Base5 standard based on thick coax -> 500m

- Nodes are connected using thick coax cables and vampire taps
- Multiple Ethernet segments can be joined together by repeaters



Network repeater

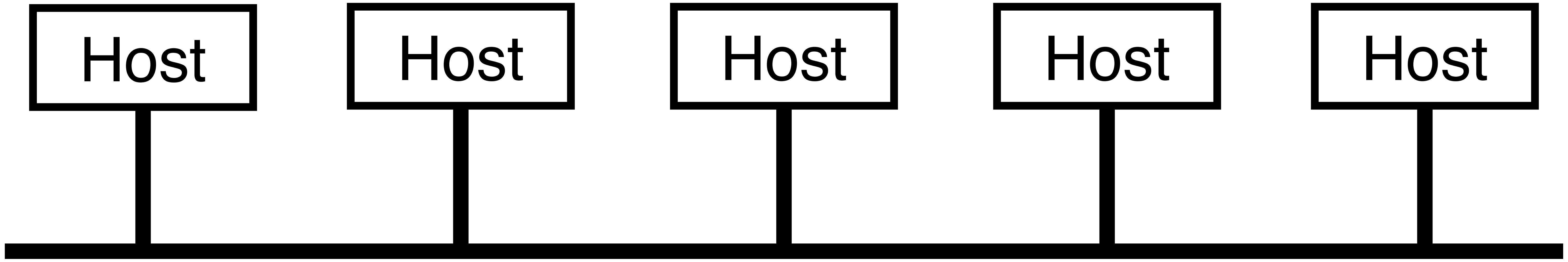
Ethernet Physical Layer

10Base5 standard based on thick coax -> 500m

- Nodes are connected using thick coax cables and vampire taps
- Multiple Ethernet segments can be joined together by repeaters

10Base2 standard based on thin coax -> 200m

- Nodes are connected using thin coax cables and BNC “T” connectors in a bus topology
- Thick coax no longer used



Ethernet Physical Layer

10Base5 standard based on thick coax -> 500m

- Nodes are connected using thick coax cables and vampire taps
- Multiple Ethernet segments can be joined together by repeaters

10Base2 standard based on thin coax -> 200m

- Nodes are connected using thin coax cables and BNC "T" connectors in a bus topology
- Thick coax no longer used

10BaseT uses twisted pair and hubs -> 100m

- Stations can be connected to each other or to hubs
- Hub (a dumb device) acts as a concentrator

Ethernet Physical Layer

10Base5 standard based on thick coax -> 500m

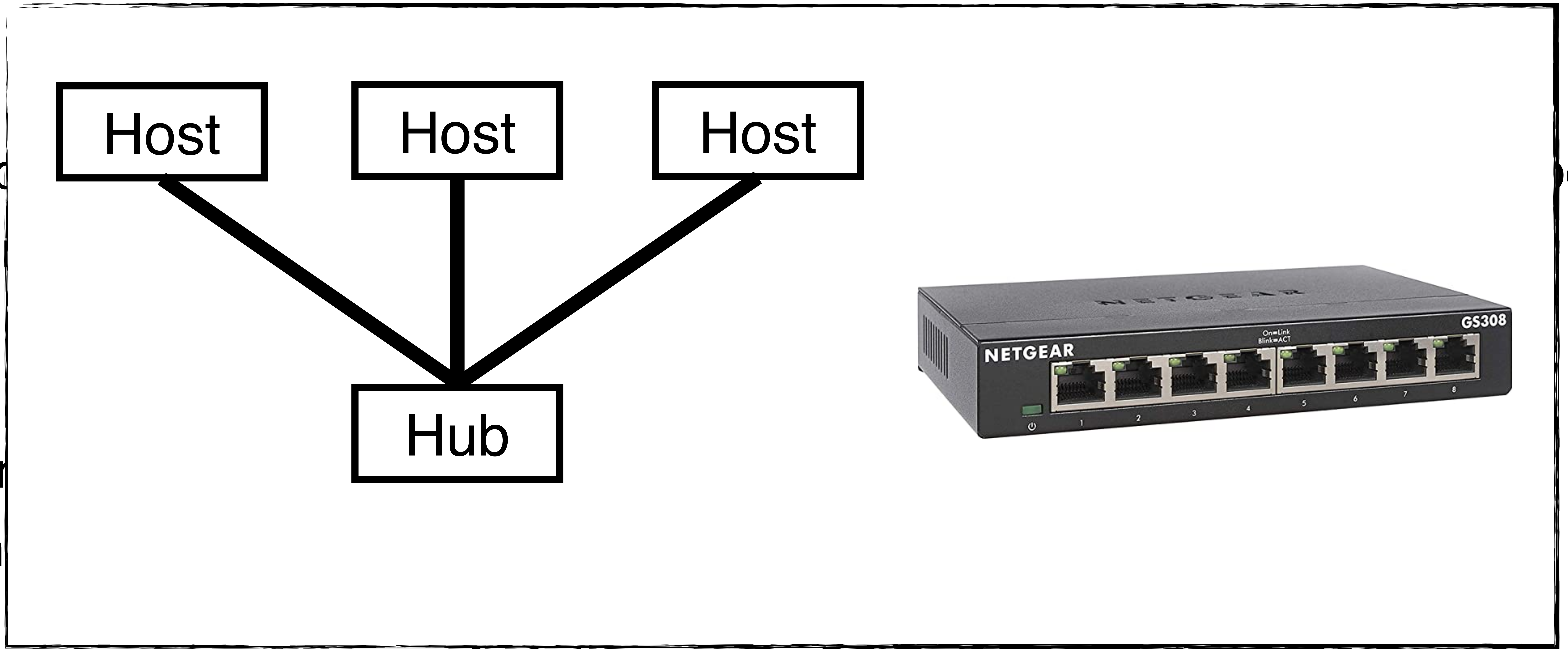
- Nodes are connected using thick coax cables and vampire taps
- Multiple Ethernet segments can be joined together by repeaters

10Base2

- Nodes are c ology
- Thick coax

10BaseT

- Stations car
- Hub (a dum



Ethernet Frame Format

8B	6B	6B	2B			4B
Preamble	Dest	Source	Type	Data	Pad	CRC

Preamble marks the beginning of the frame

- Also provides synchronization

Source and detention are 48-bit IEEE MAC addresses

- Flat address space
- Hardwired into the network interface

Type field is a demultiplexing field

- What network layer (layer 3) should receive this packet?

Ethernet Frame Format (cont'd)

8B	6B	6B	2B			4B
Preamble	Dest	Source	Type	Data	Pad	CRC

Max frame size = 1500B; min=46B

- Need padding to meet min requirement

CRC for error detection

Ethernet Host Side

Transceiver: detects when the medium is idle and transmits the signal when a host wants to send

- Connect to “Ethernet adaptor”
- Sits on the host

Any host signal broadcast to everybody

- But transceiver accepts frames addressed to itself
- Also frames sent to broadcast address
- All frames, if in promiscuous mode

Ethernet Host Side (cont’d)

When transmitting, all hosts on the same segment, or connected to the same hub, compete for medium

- Said to “share same collision domain”
- Bad for efficiency!

Sender-side: MAC Protocol

Carrier-sense multiple access with collision detection (CSMA/CD)

- MA = multiple access
- CS = carrier sense
- CD = collision detection

CSMA/CD Algorithm Overview

Sense for carrier

- "Medium idle" ?

If medium busy, wait until idle

- Sending would force a collision and waste time

Send packet and sense for collision

If no collision detected, consider packet delivered

CSMA/CD Algorithm Overview (cont'd)

Otherwise, abort immediately, perform exponential back off and send packet again

- Start to send after a random time picked from an internal
- Length of the interval increases with every collision, retransmission attempt
- At the i-th try, the send waits for a period chosen from {0, 1 * 51.2us, …, 2^(i-1) * 51.2} and tris again
- 51.2us is the RTT of a maximally configured Ethernet with 2500m

Collision Detection

Why do we need this?

- Because there is no centralized control and two senders can send simultaneously

When a sender detects a collision

- Transmit a jamming sequence of 32 bits
- Thus a sender will minimal transmits 96bits = preamble + jamming

Collision Detection: Implications

The worst case to detect collision

- Say maximum propagation delay across the segment is **D**
- Say on sender A starts sending a frame at **t**
- First byte of this frame reaches the other end at **t + D**
- Say just prior to this, a different sender B at the other end transmits a signal, detects a collision and transmits a jamming frame
- Unfortunately, the first sender A will not know of this until **t + 2D**

Ethernet frames needs to be long enough (span 2D) so that collisions can be detected

- d depends on max length of ethernet cable

Minimum Packet Size

Considering a maximally configured Ethernet is 2500m

- with up to four repeaters
- RTT is 51.2us
- The speed is 10Mbps

Minimum Packet Size

Considering a maximally configured Ethernet is 2500m

- with up to four repeaters
- RTT is 51.2us
- The speed is 10Mbps

BDP = 10Mbps * 51.2us = 512 bits = 64B

- Ethernet header = DST + SRC + TYPE + CRC = 6B + 6B + 2B + 4B = 18B
- Minimal padding = 64B - 18B = 46B
- We need padding

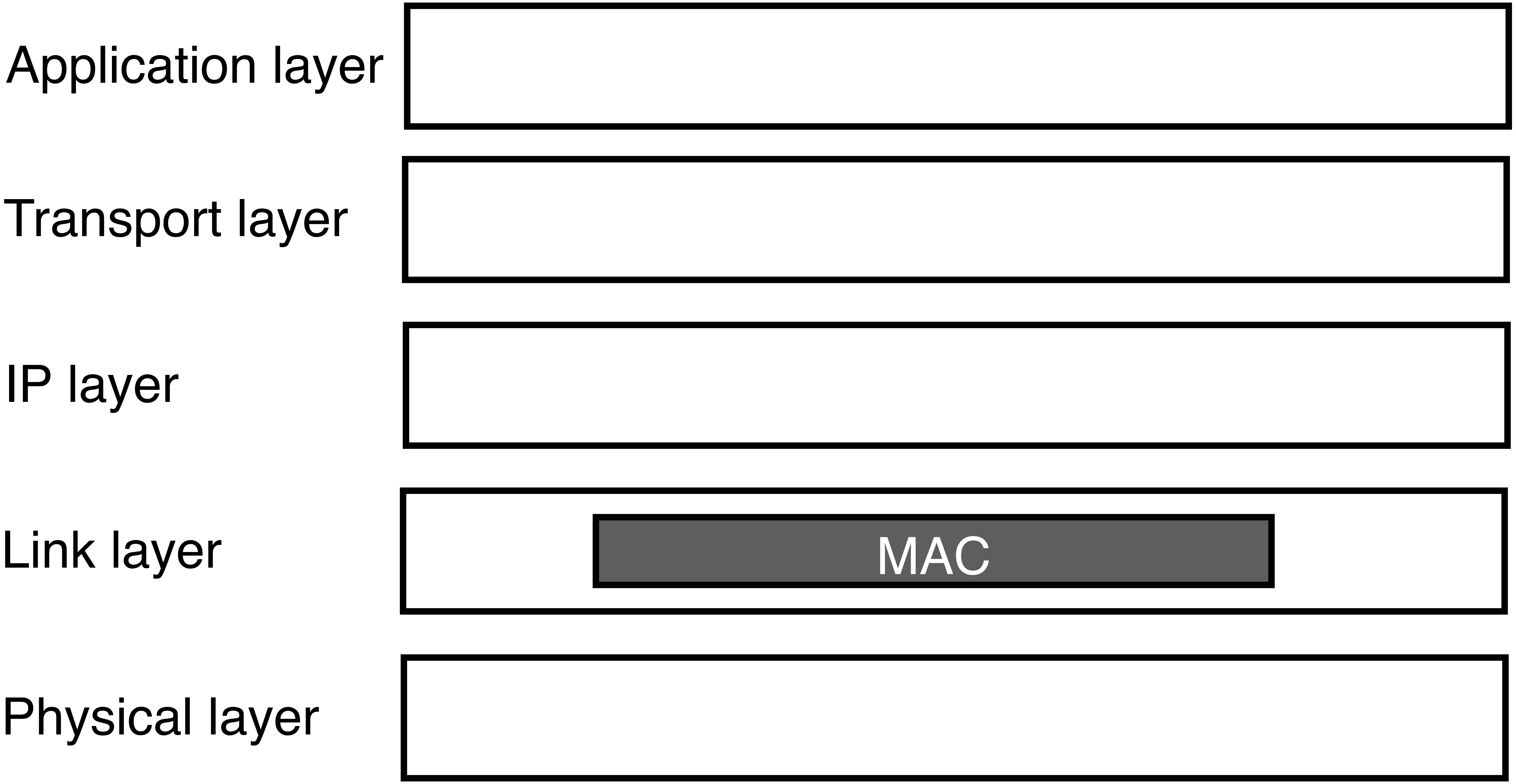
Why maximum packet size is 1500B?

Why maximum packet size is 1500B?

- **Hardware limits — reduce the memory requirement of a NIC**
- **Efficiency — transmission latency and packet header overhead**

[1]. https://blog.benjojo.co.uk/asset/1hhfq2UR8P
[2]. https://blog.benjojo.co.uk/post/why-is-ethernet-mtu-1500
[3]. Ethernet: Distributed Packet Switching for local Computer Networks, 1980

MAC in Layering



Summary

Today

- Error detection
- Ethernet

Next lecture

- Wireless basics