Introduction to Computer Networks

CS640

Encoding

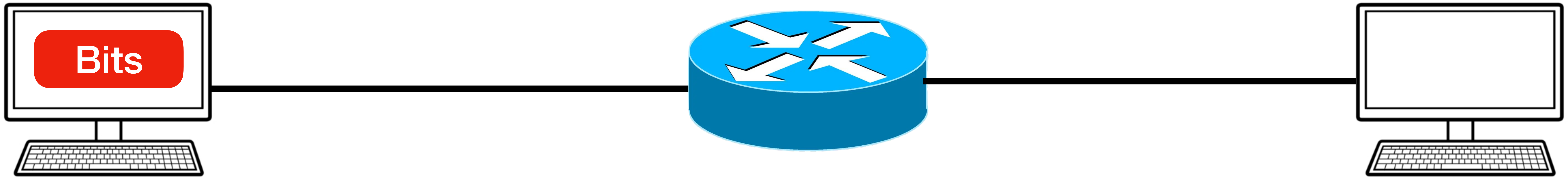
https://pages.cs.wisc.edu/~mgliu/CS640/S25/index.html

Ming Liu
mgliu@cs.wisc.edu

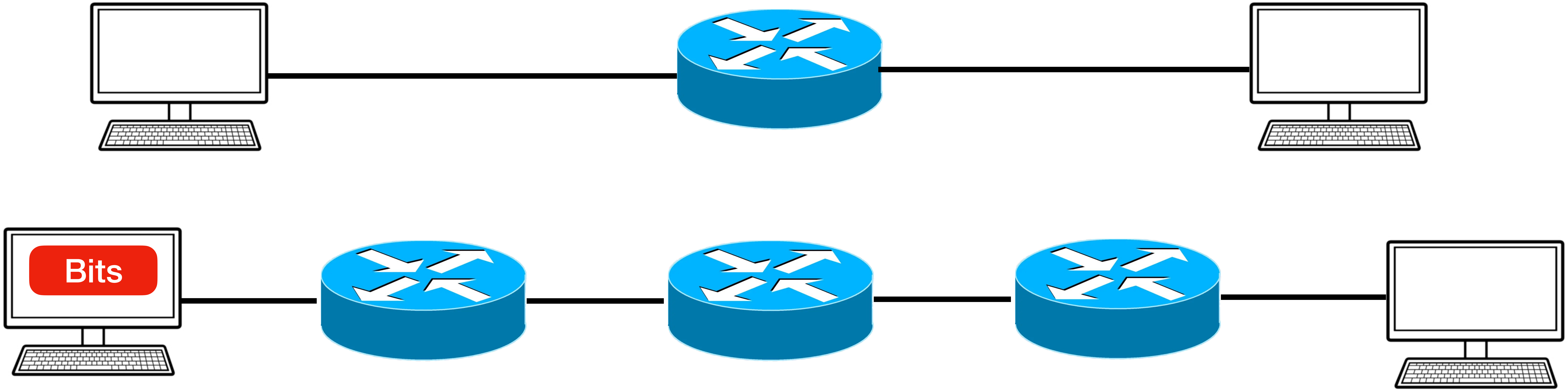
Outline

- Last
  - Computer networks: performance analysis
- Today
  - Encoding
- Announcements
  - Lab1 due on Feb 11th 11:59pm

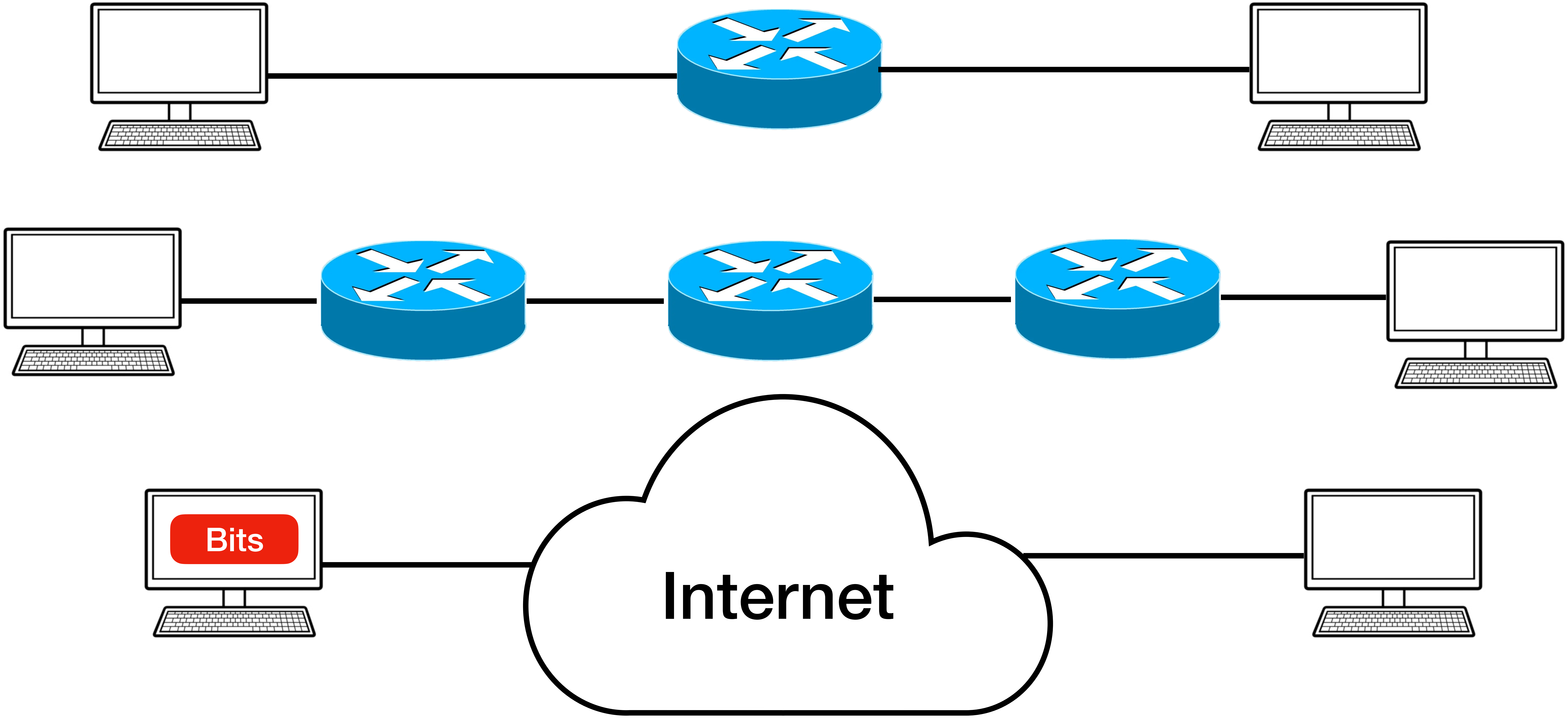
Data Transmission in Computer Networks



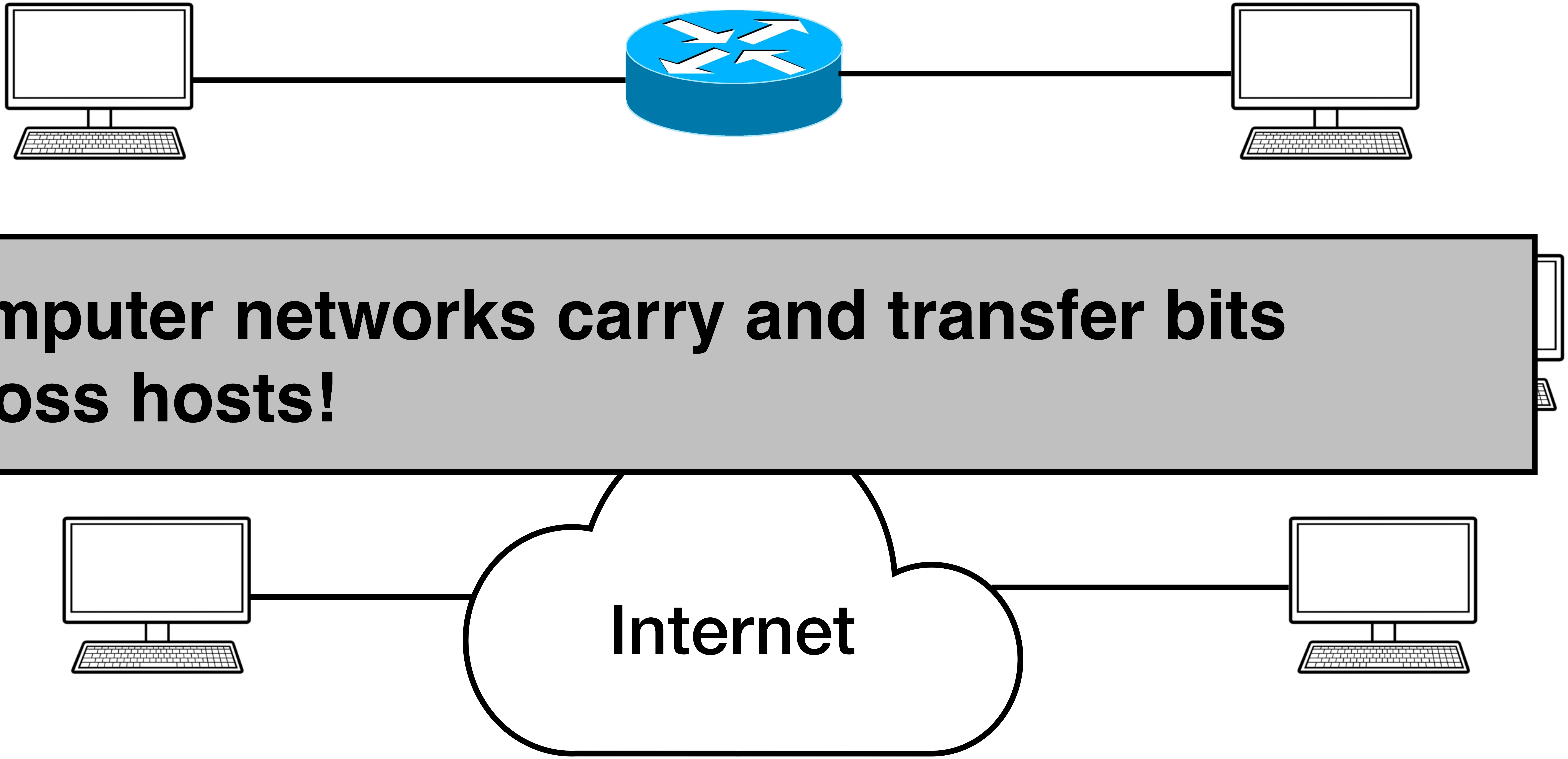
Data Transmission in Computer Networks



Data Transmission in Computer Networks



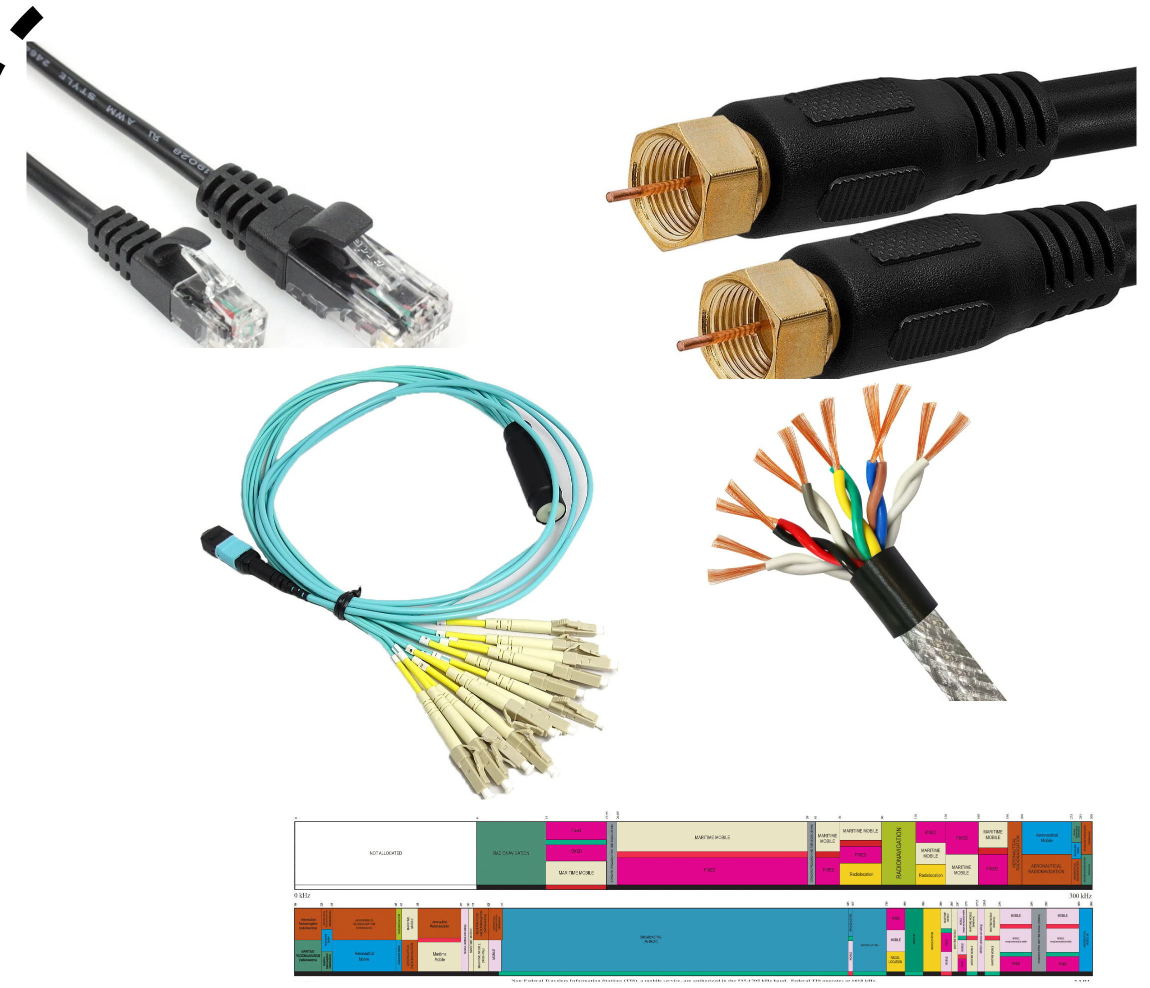
Data Transmission in Computer Networks



Computer networks carry and transfer bits across hosts!

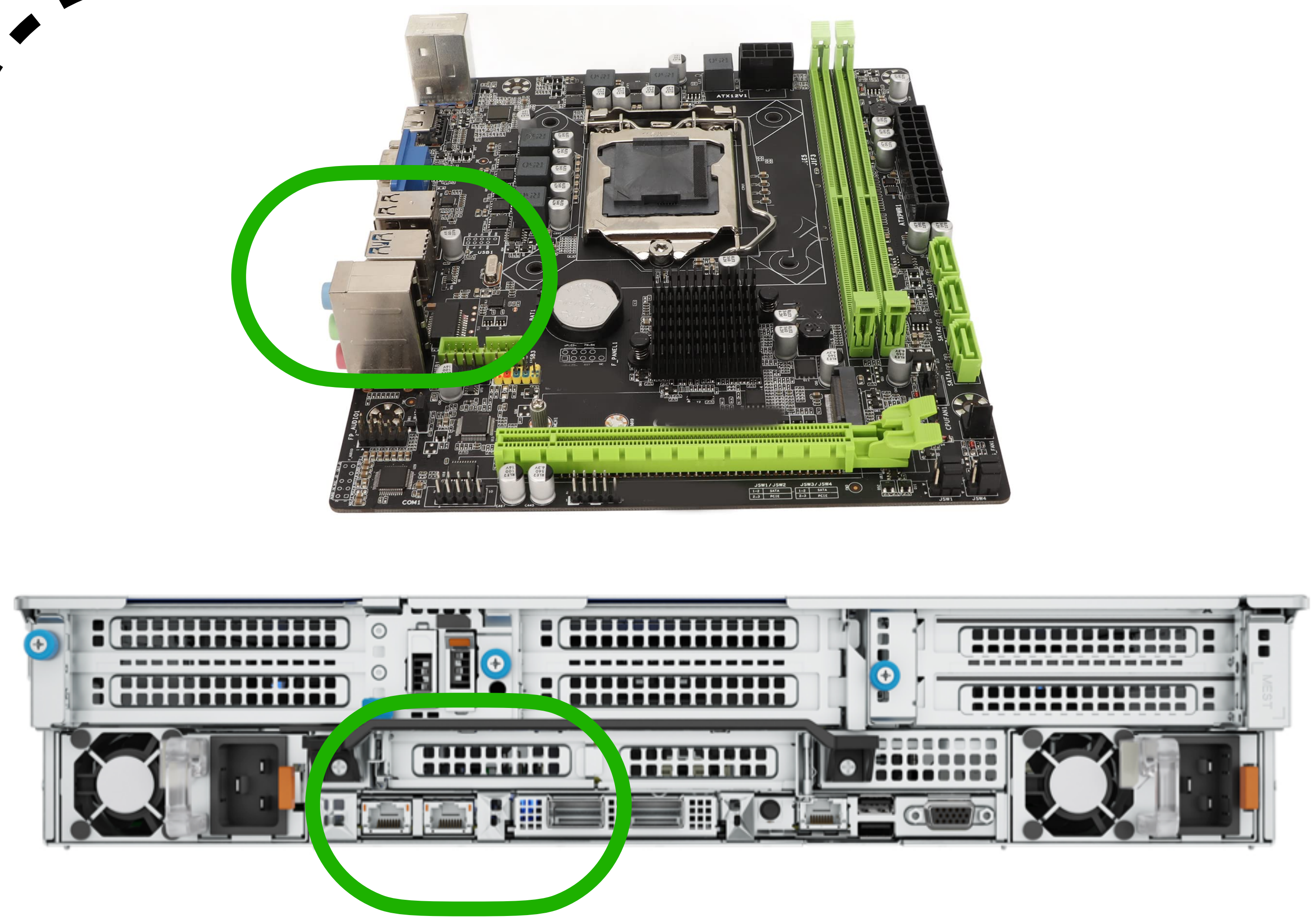
Recap: Networking Hardware

- Three types of hardware
  - Communication links
  - Multi-port routers and switches
  - Hosts with communication ports



Communication Links

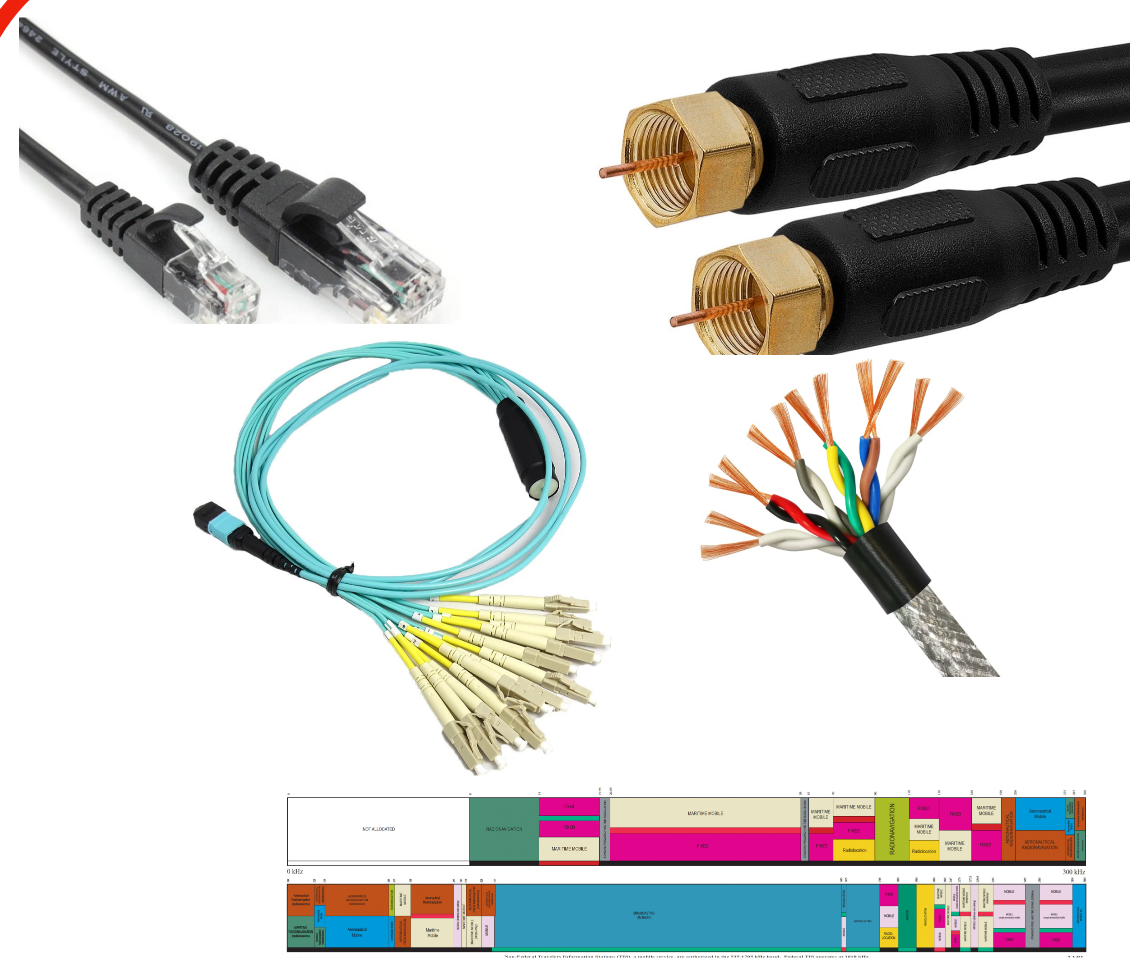
Routers and Switches



Hosts w/ Communication Ports

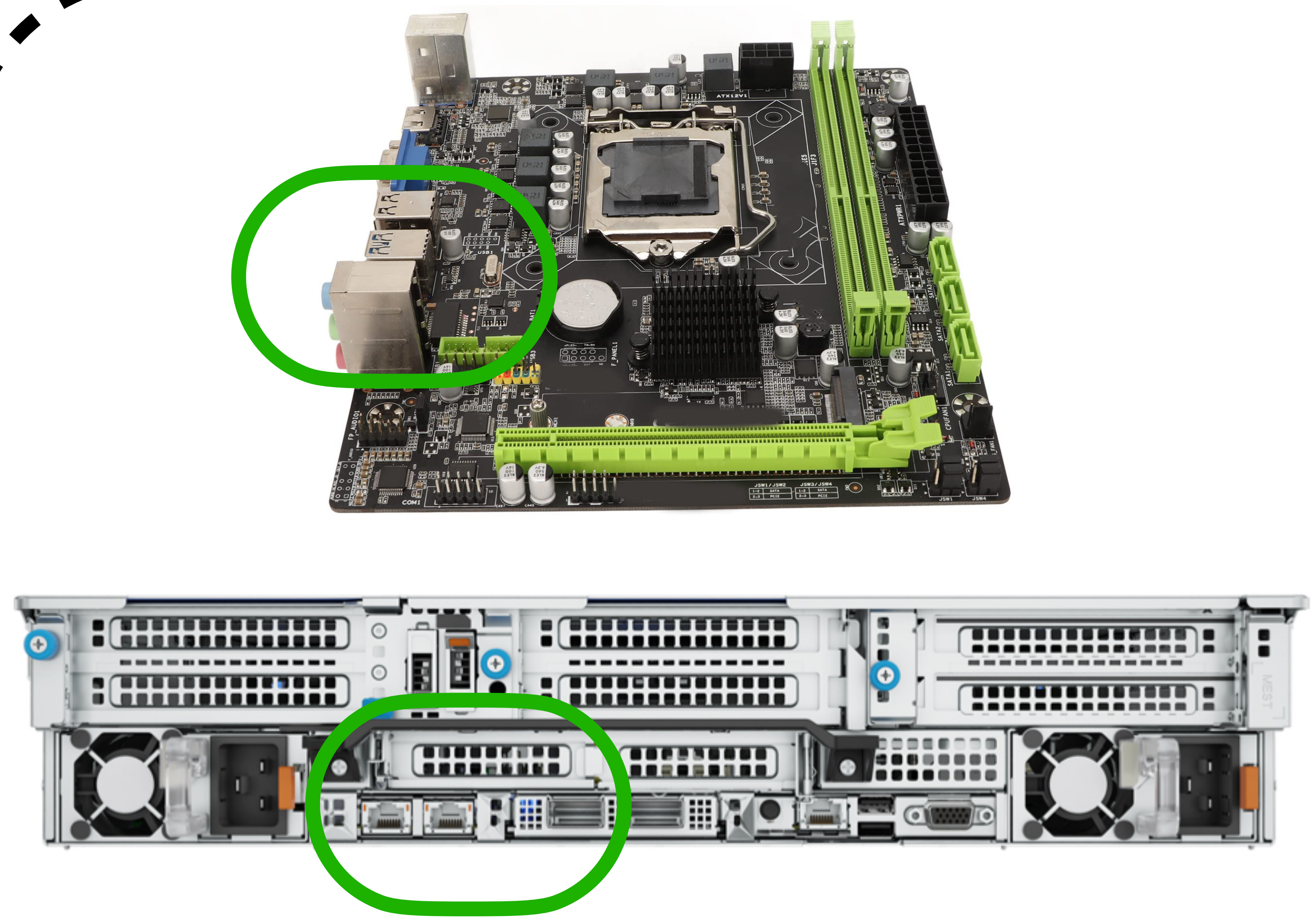
Recap: Networking Hardware

- Three types of hardware
  - Communication links
  - Multi-port routers and switches
  - Hosts with communication ports



Communication Links

Routers and Switches

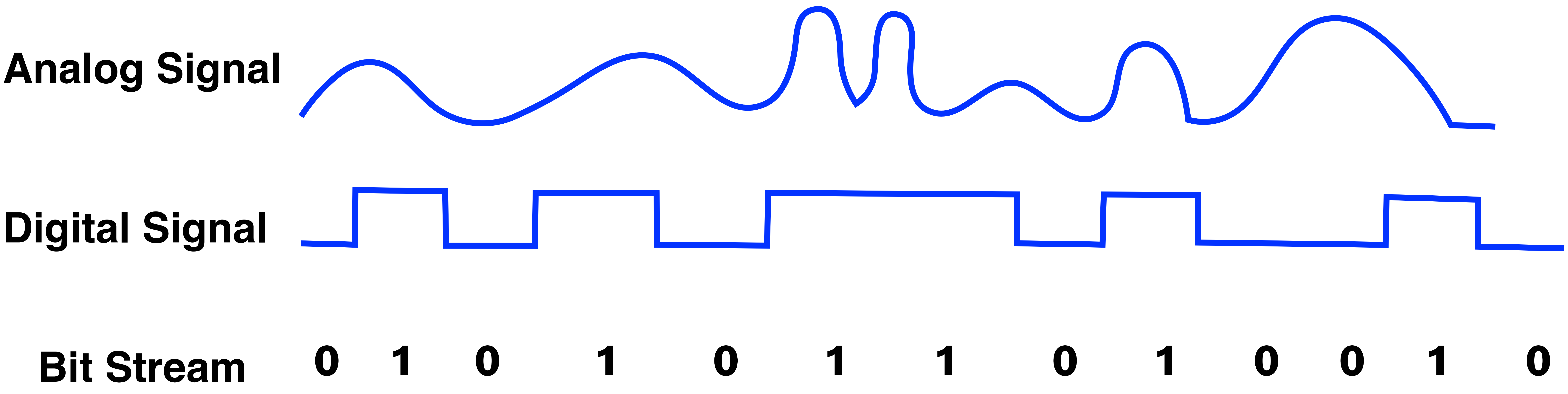


Hosts w/ Communication Ports

How can we represent bits on the link?

Bit Representation

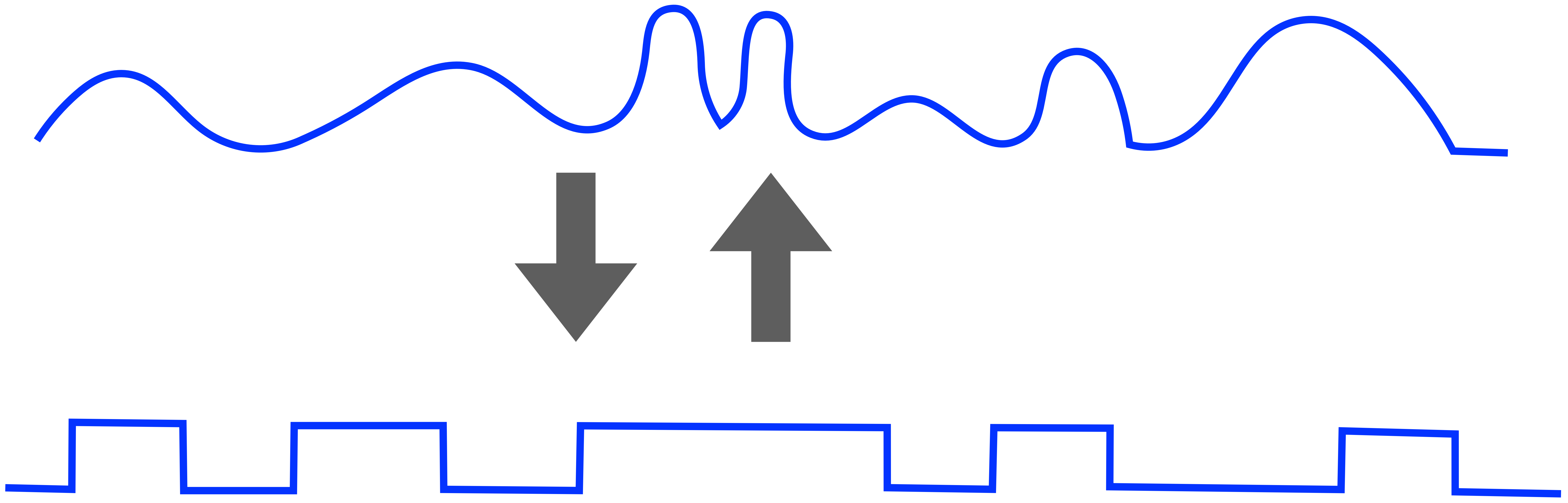
- Two discrete signals based on the communication media
  - Low signal
  - High signal



How can we reliably propagate bits across the link?

Invariant: Bits (send) = Bits (receive)

- Two parts
  - Encoding process: Binary data —> Signals
  - Decoding process: Signals —> Binary data



Why is it hard?

Why is it hard?

- #1: Signal attenuation
  - Signal strength decays when traversing a medium

Why is it hard?

- #1: Signal attenuation
  - Signal strength decays when traversing a medium
- #2: Transmission is synchronous
  - We are implicitly using the clock to sample the signal
  - Two perspectives: when it starts/ends and how long a signal is

Why is it hard?

- #1: Signal attenuation
  - Signal strength decays when traversing a medium

- #2: Transmission is synchronous
  - We are implicitly using the clock to sample the signal
  - Two perspectives: when it starts/ends and how long a signal is

- #3: Fault tolerance
  - Error detection: get rid of some illegal data bits
  - Error correction: self-correct some error bits without retransmission

Why is it hard?

- #1: Signal attenuation
  - Signal strength decays when traversing a medium

→ **Modulation Scheme**

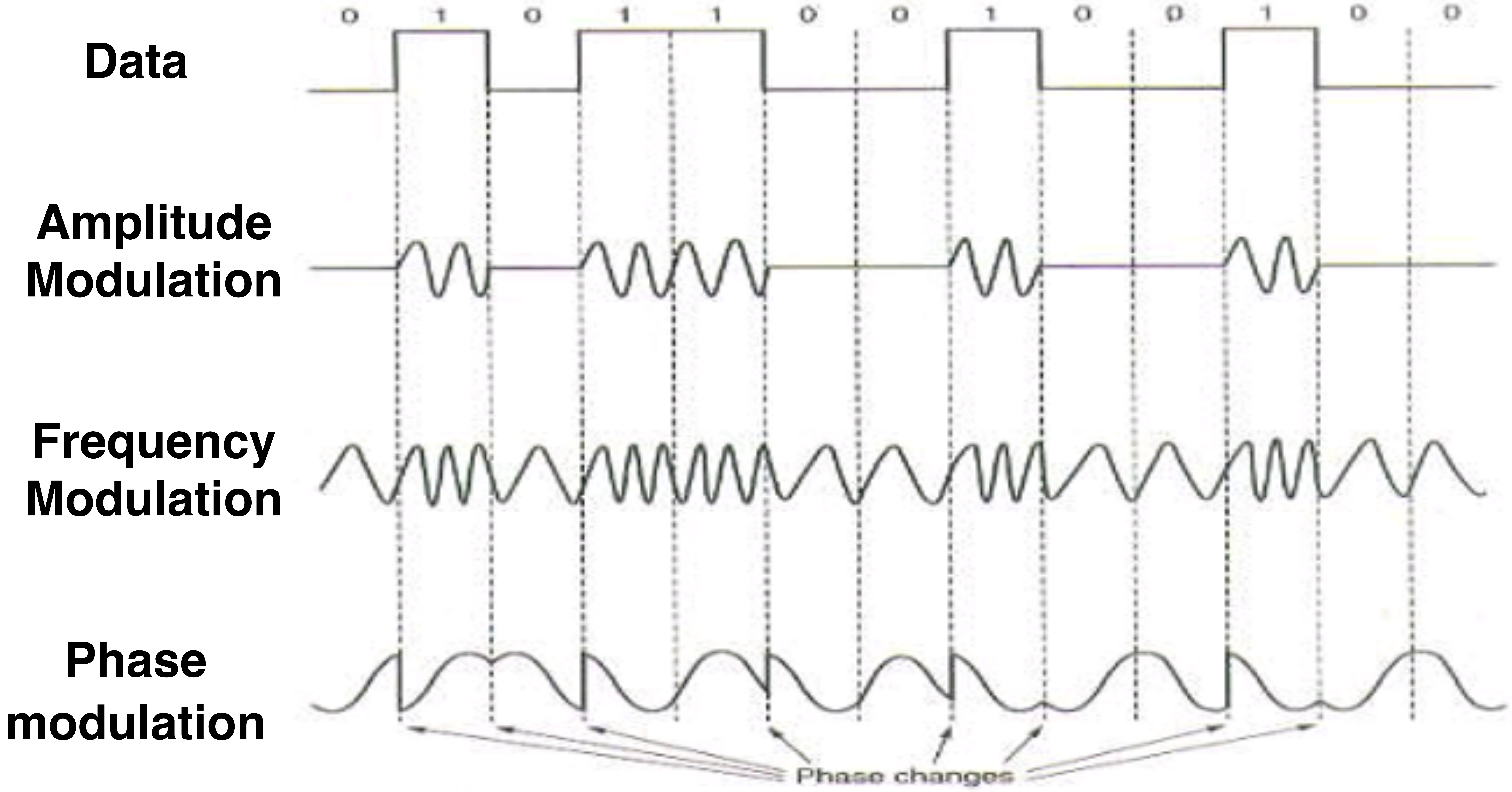
- #2: Transmission is synchronous
  - We are implicitly using the clock to sample the signal
  - Two perspectives: when it starts/ends and how long a signal is
- #3: Fault tolerance
  - Error detection: get rid of some illegal data bits
  - Error correction: self-correct some error bits without retransmission

Modulation

- Change signal attributes for effective data transmission
  - Demodulation performs the reverse process

Modulation

- Change signal attributes for effective data transmission
  - Demodulation performs the reverse process



Why is it hard?

- #1: Signal attenuation
  - Signal strength decays when traversing a medium

→ **Modulation Scheme**

- #2: Transmission is synchronous
  - We are implicitly using the clock to sample the signal
  - Two perspectives: when it starts/ends and how long a signal is
- #3: Fault tolerance
  - Error detection: get rid of some illegal data bits
  - Error correction: self-correct some error bits without retransmission

} **Encoding Scheme**

Encoding

- Define the *bit—>signal* transformation rules
  - Meet certain electrical constraints
  - Create control symbols besides regular data symbols
  - Introduce error detection or correction approaches
  - Decoding performs the reverse functionalities

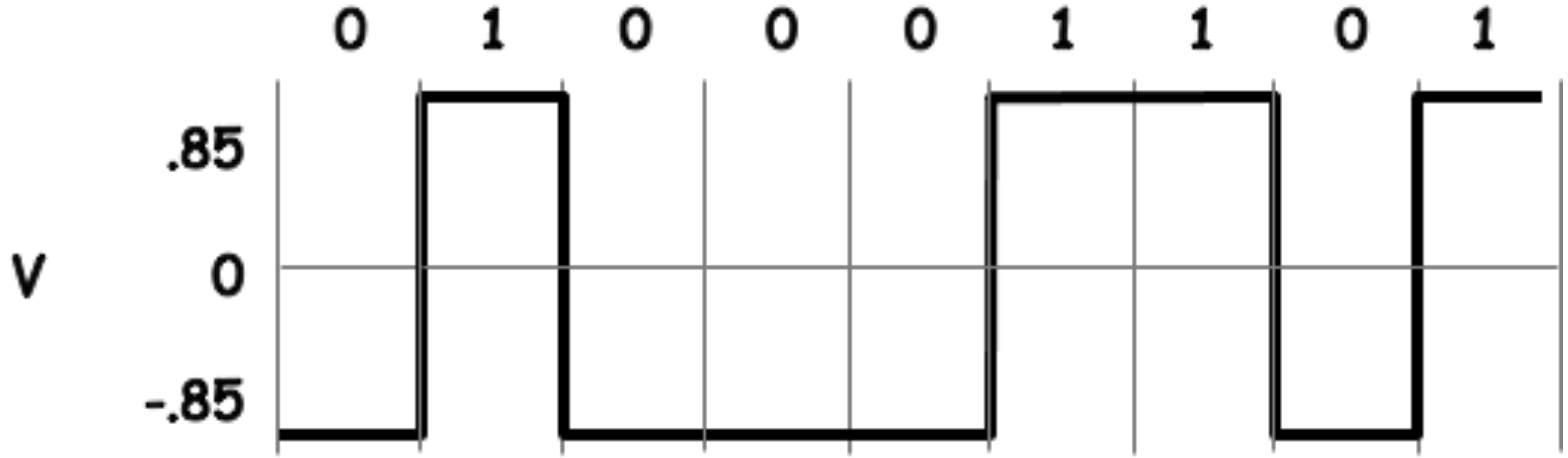
Encoding

- Define the *bit—>signal* transformation rules
  - Meet certain electrical constraints
  - Create control symbols besides regular data symbols
  - Introduce error detection or correction approaches
  - Decoding performs the reverse functionalities

- Techniques
  - Non-Return to Zero (NRZ)
  - Non-Return to Zero Inverted (NRZI)
  - Manchester Encoding
  - 4B/5B Encoding
  - ……

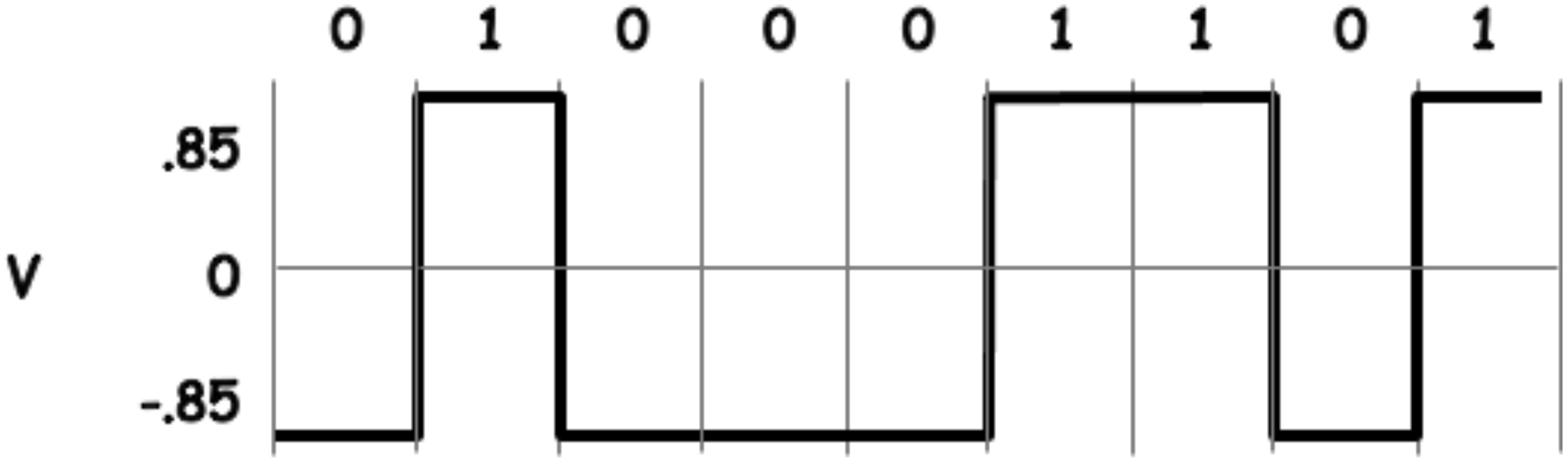
Non-Return to Zero (NRZ)

- Use signal voltage to represent bits
  - 1 —> high signal
  - 0 —> low signal



Non-Return to Zero (NRZ)

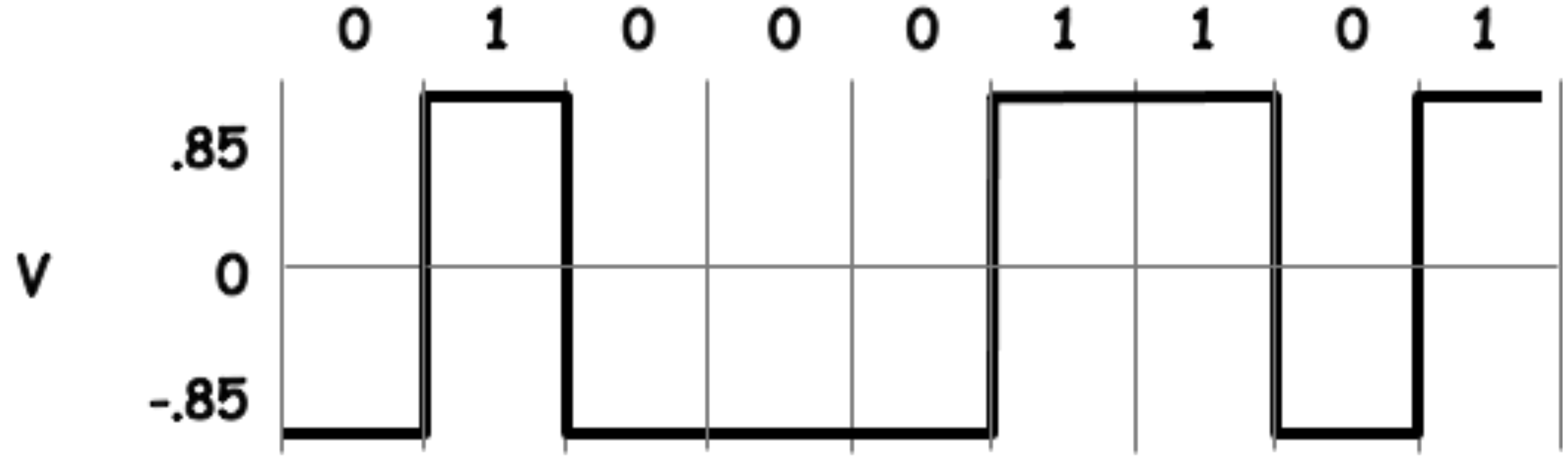
- Use signal voltage to represent bits
  - 1 —> high signal
  - 0 —> low signal



What are the issues?

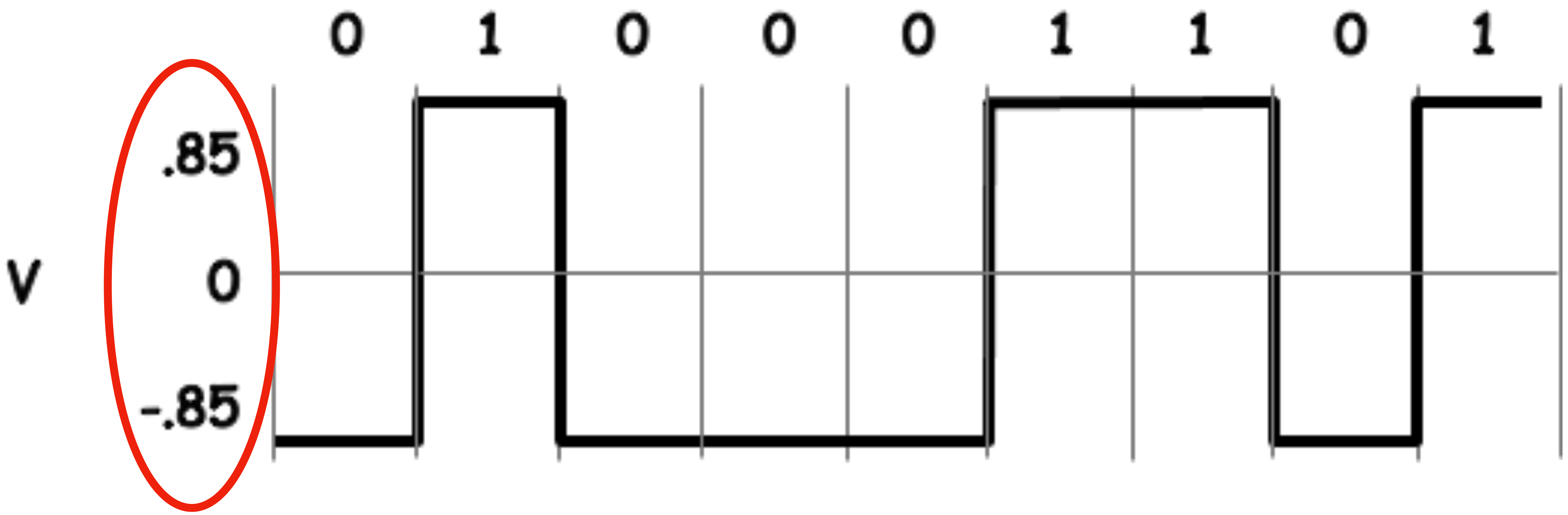
NRZ Issues

- Long sequences of 1's or 0's cause two problems
  - Baseline wander
  - Clock synchronization



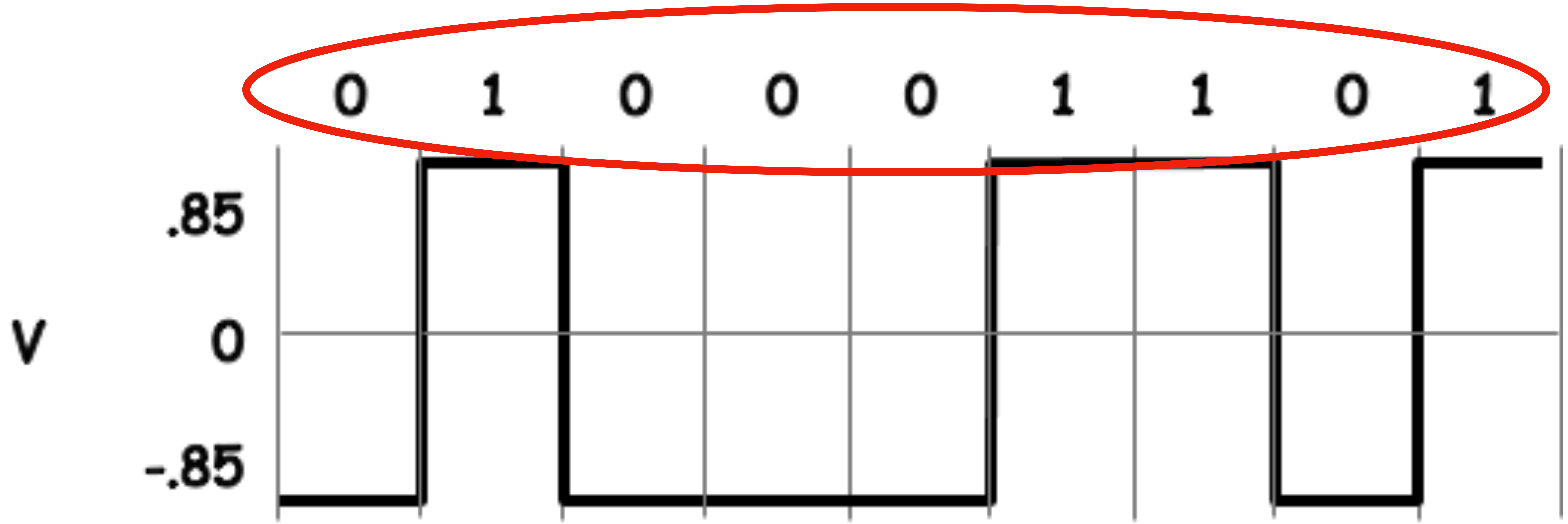
Baseline Wander

- Root cause: dynamic baseline adjustment
  - Hard to identify low (0), high (1), and noise signals



Clock Synchronization

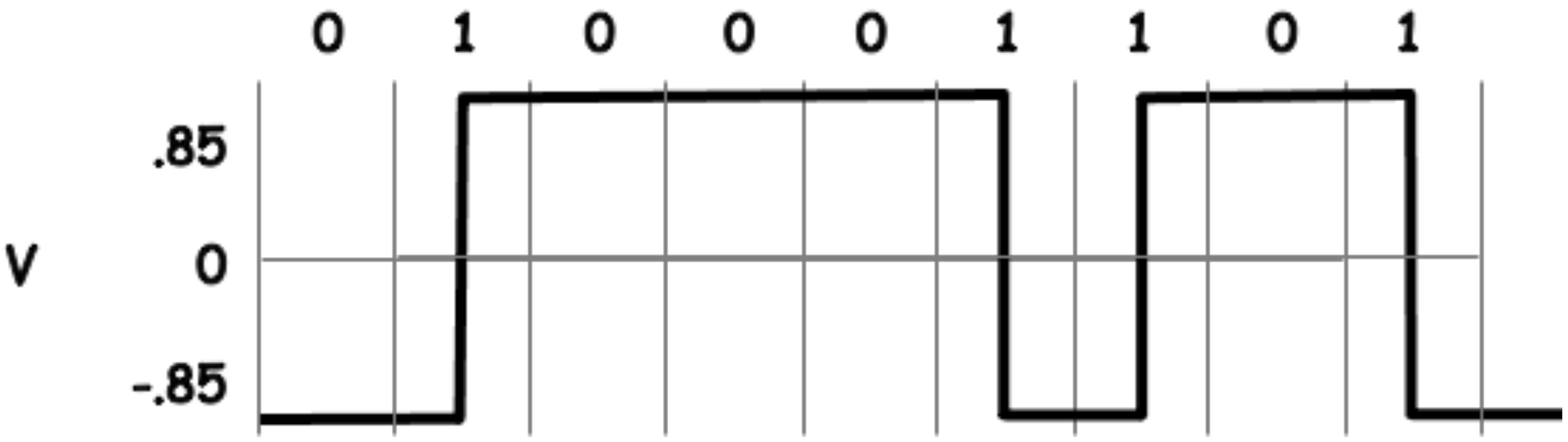
- Root cause: clock drifting cannot be avoided
  - No global clock domain



How can we get rid of the baseline?

Non-Return to Zero Inverted (NRZI)

- Use signal transition to represent bits
  - 1 —> make transition
  - 0 —> stay the same



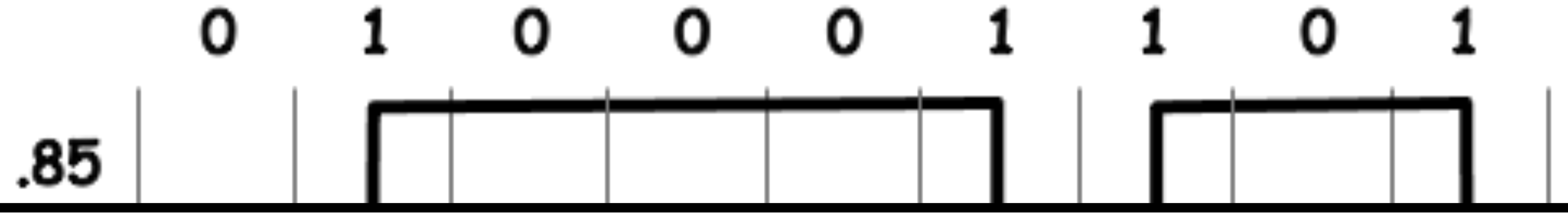
Non-Return to Zero Inverted (NRZI)

- Use signal transition to represent bits
  - 1 —> make transition
  - 0 —> stay the same

Does it address the issues of NRZ?

Non-Return to Zero Inverted (NRZI)

- Use signal transition to represent bits
  - 1 —> make transition
  - 0 —> stay the same

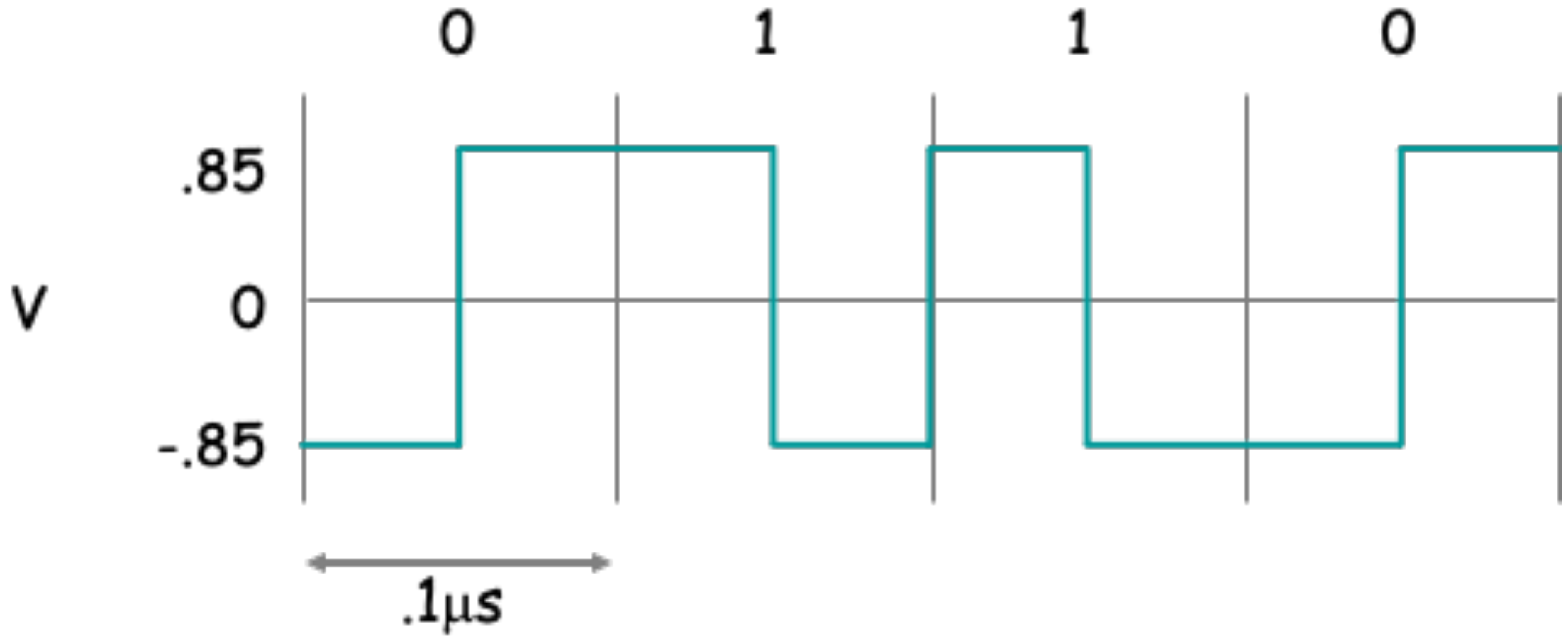


Does it address the issues of NRZ?
No, long sequences of 0’s still happen.

How can we perform frequent in-line synchronization?

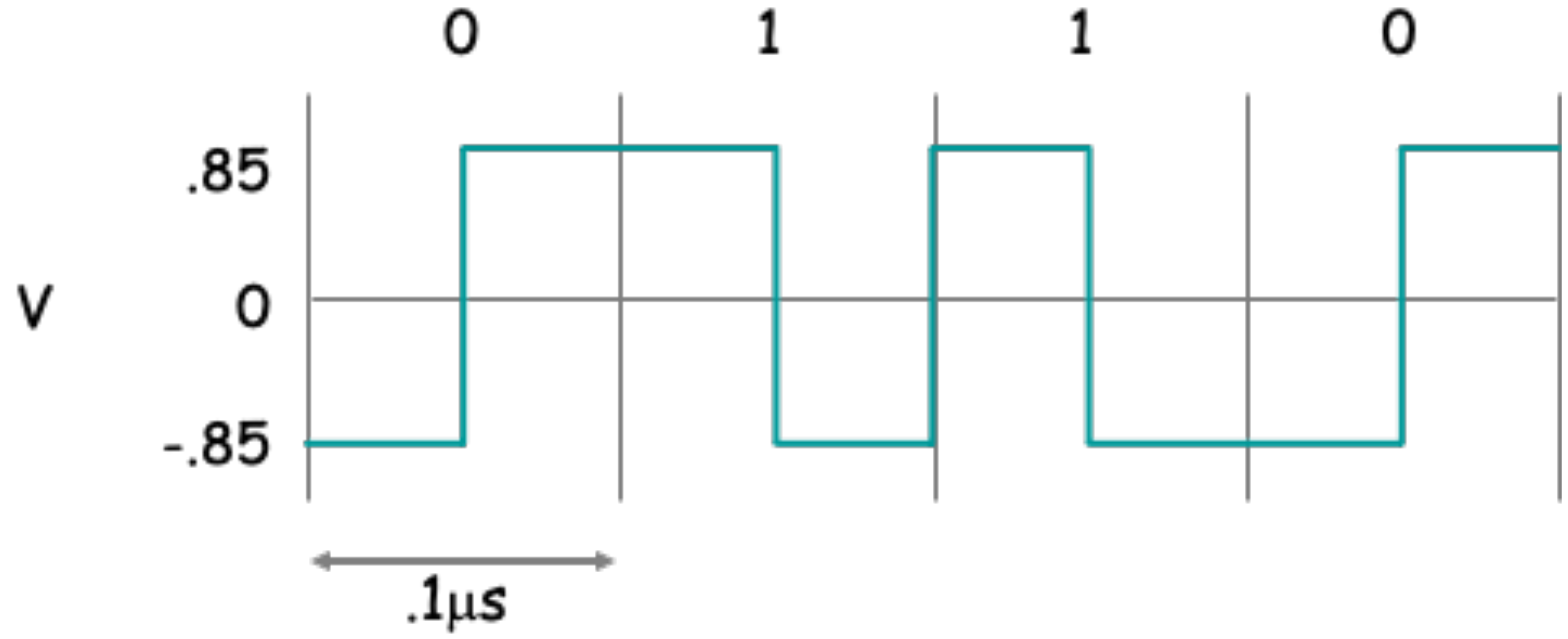
Manchester Encoding

- Use signal transition to represent bits
  - 1 —> negative transition
  - 0 —> positive transition



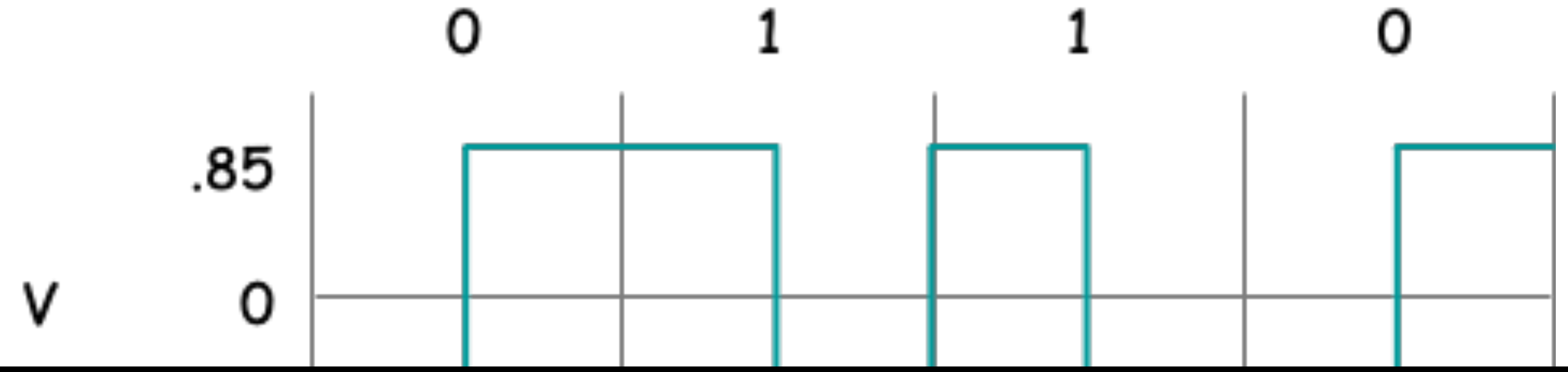
Manchester Encoding Takes Clock Implicitly

- Between two bits, there is always a signal transition



Manchester Encoding Takes Clock Implicitly

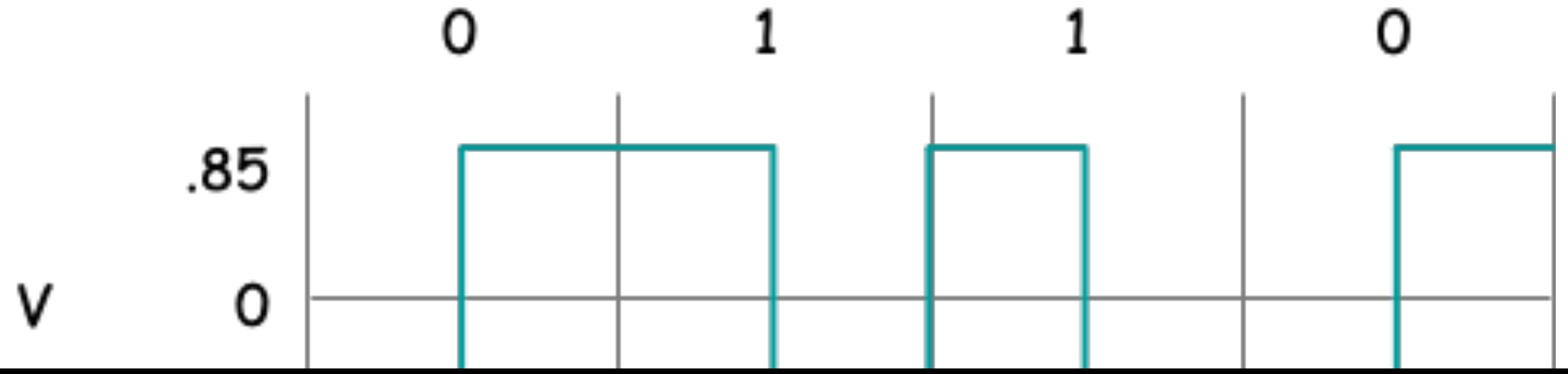
- Between two bits, there is always a signal transition



What is the downside?

Manchester Encoding Takes Clock Implicitly

- Between two bits, there is always a signal transition



What is the downside?
Low bandwidth utilization

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander			
Clock synchronization			
Complexity			
Baud rate utilization			

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander	Y	N	N
Clock synchronization			
Complexity			
Baud rate utilization			

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander	Y	N	N
Clock synchronization	Y	Y/N	N
Complexity			
Baud rate utilization			

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander	Y	N	N
Clock synchronization	Y	Y/N	N
Complexity	Low (signal monitoring)	Medium (signal monitoring + signal transition detection)	High (signal monitoring + directional signal transition detection)
Baud rate utilization			

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander	Y	N	N
Clock synchronization	Y	Y/N	N
Complexity	Low (signal monitoring)	Medium (signal monitoring + signal transition detection)	High (signal monitoring + directional signal transition detection)
Baud rate utilization	High	Medium	Low

NRZ v.s. NRZI v.s. Manchester

- Baud rate: the number of electrical state changes that can happen per second

	NRZ	NRZI	Manchester
Baseline wander	Y	N	N
Clock synchronization	Y	Y/N	N
Complexity	Low (signal monitoring)	Medium (signal monitoring + signal transition detection)	High (signal monitoring + directional signal transition detection)
Baud rate utilization	High	Medium	Low

Can we enhance NRZI with clock synchronization?

4B/5B Encoding

- Every 4 bits of data are encoded in a 5-bit code
  - Each symbol has no more than one leading zero
  - Each symbol has no more than two tailing zeros
  - Integrate in-line synchronization based on NRZI

Data	Code
0000	11110
0001	01001
0010	10100
0011	10101
0100	01010
0101	01011
0110	01110
0111	01111

Data	Code
1000	10010
1001	10011
1010	10110
1011	10111
1100	11010
1101	11011
1110	11100
1111	11101

80% Utilization

A Simple Exercise

On the sender side, how do we encode the following bit stream under NRZ, NRZI, Manchester, and 4B/5B? Suppose we represent bits 1 and 0 using high and low signals.

0010 1111 0100

Specify the clock first

Encoding Discussion

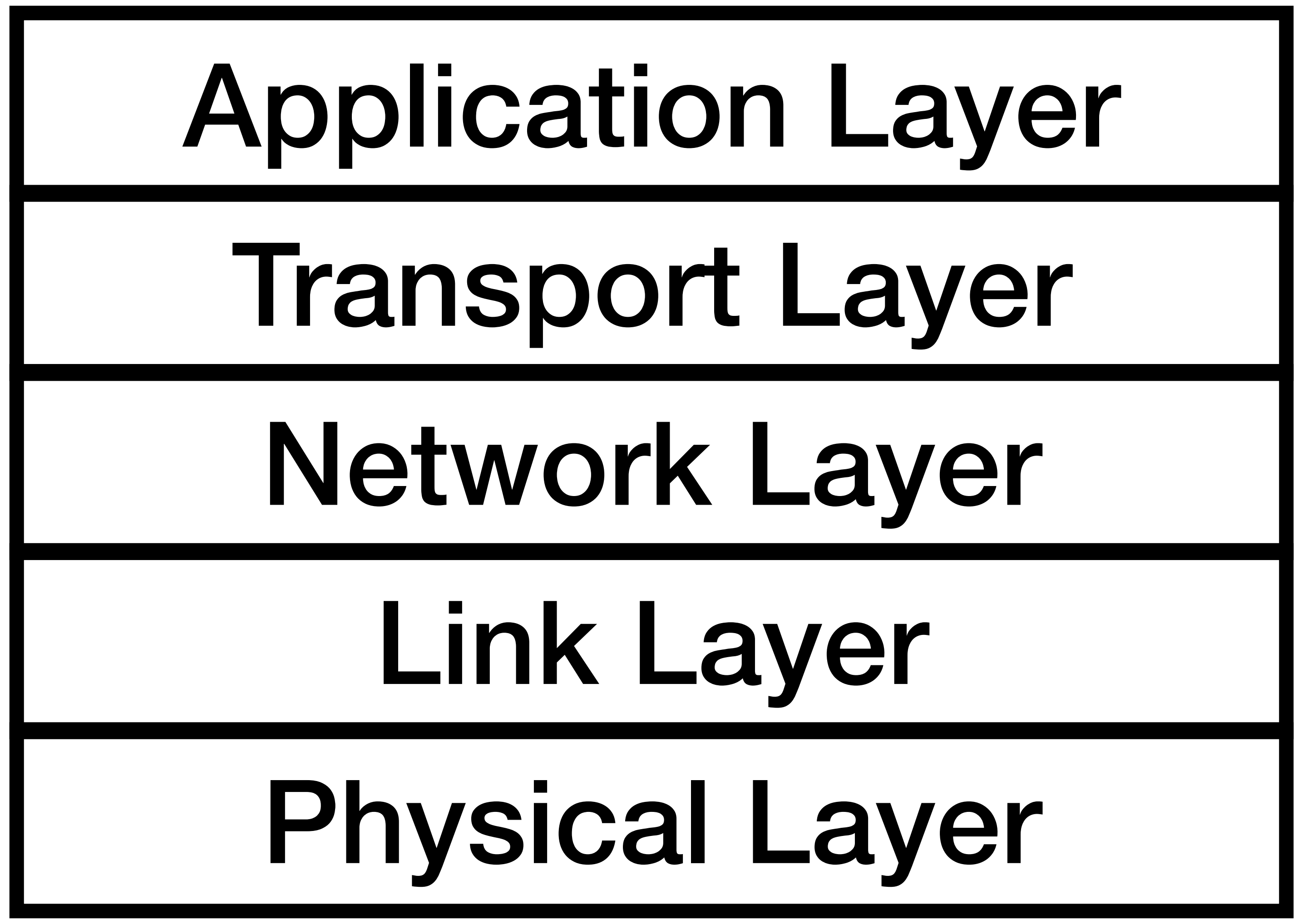
- How many bits are used to represent “0” and “1”?
  - More than “Low” and “High”

- Many other encoding schemes
  - 8b/10b: Fiber Channel and Gigabit Ethernet
  - 64b/66b: 10Gbit Ethernet
  - 128b/130b: PCIe Gen3

- Design trade-offs
  - Utilization under the clock: how many signal transitions?
  - Implementation complexity

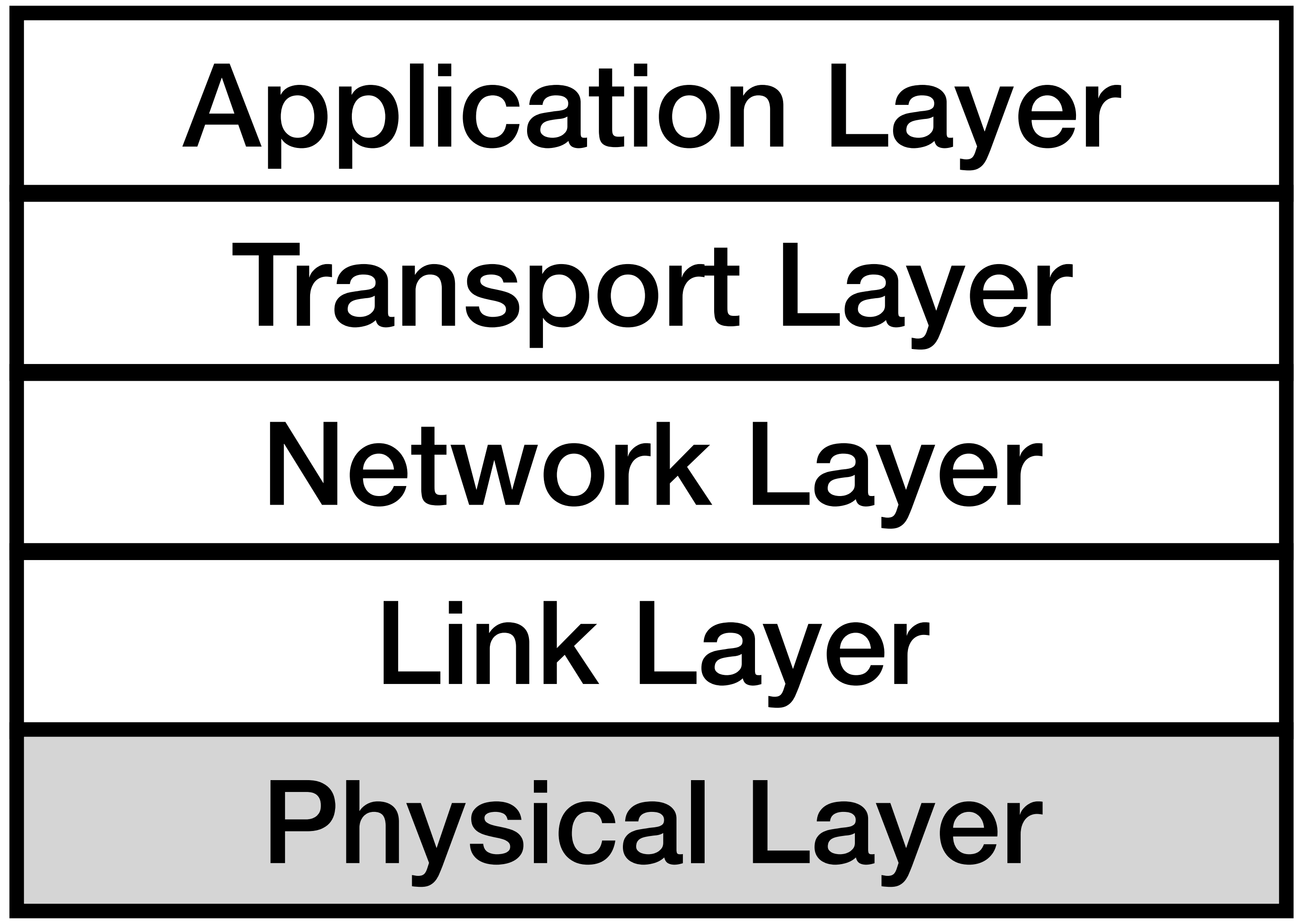
Recap: Layering

- A modular approach to building networks by abstractions
  - Introduce multiple levels of abstractions
  - Each layer focuses on different functionalities
- Two views
  - Vertical view: an interface to high-level protocols
  - Horizontal view: a peer interface to the counterpart



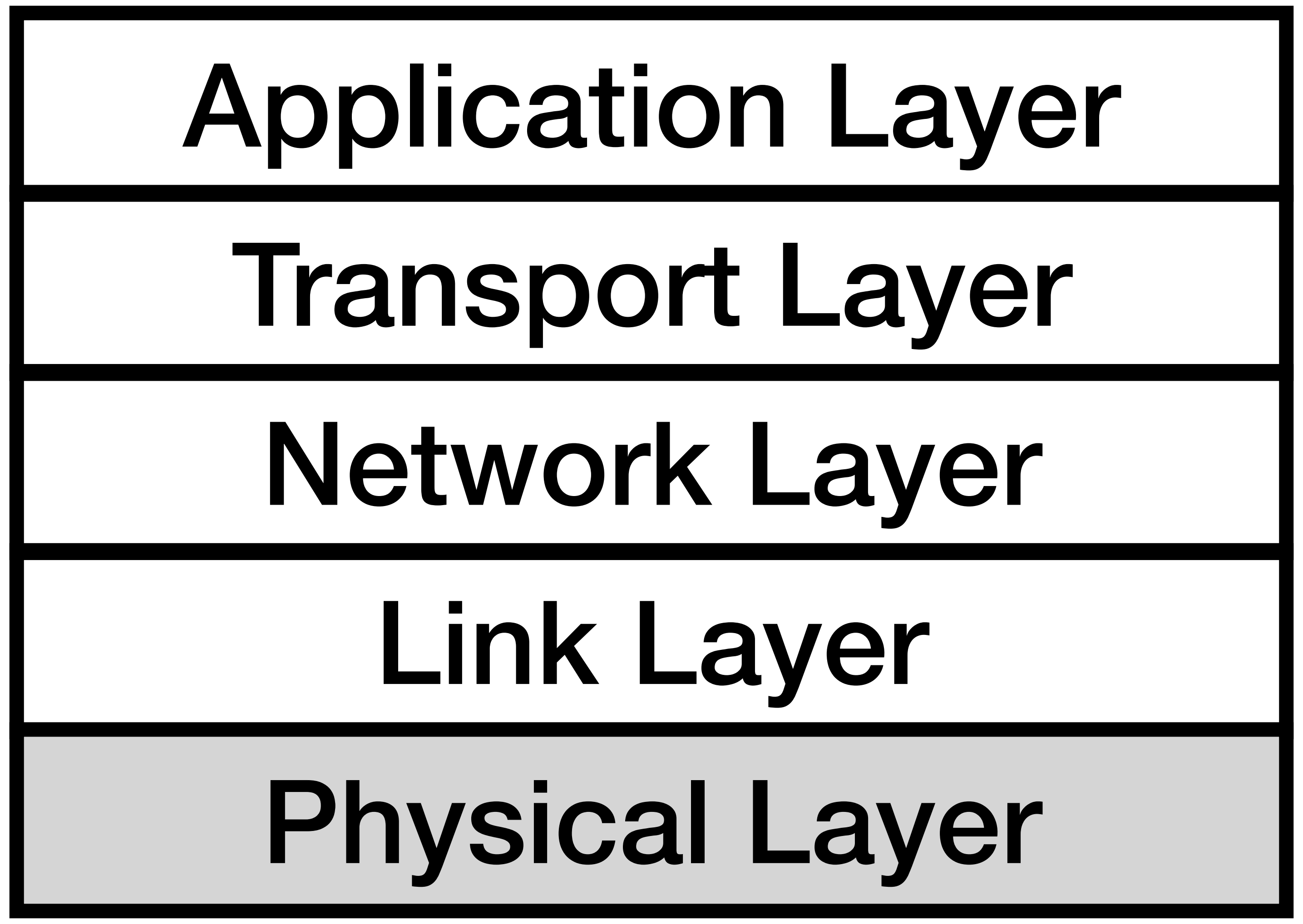
Physical Layer

- Encoding is one of its important functionalities



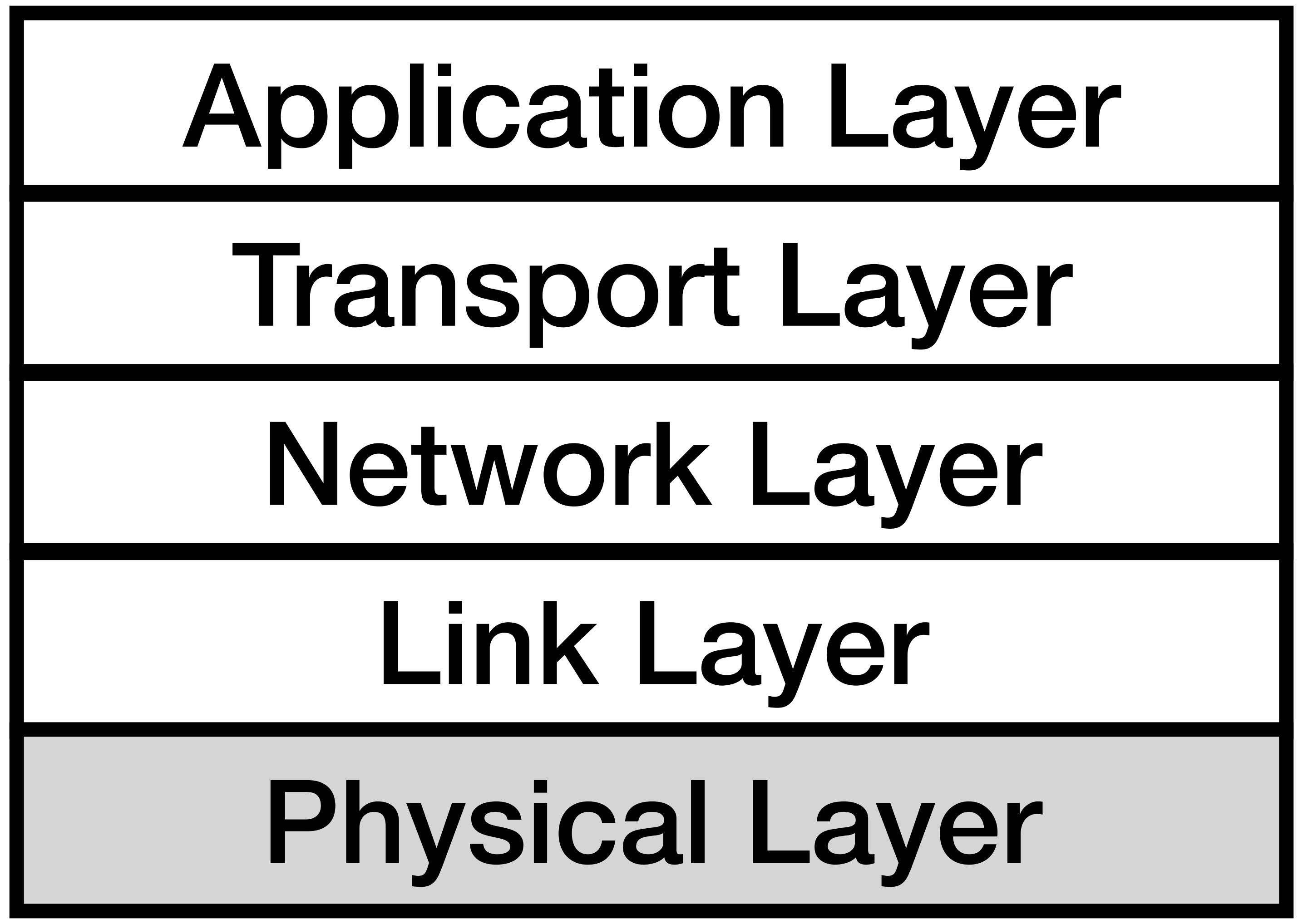
Physical Layer

- Encoding is one of its important functionalities
- Vertical view
  - A reliable bit delivery channel for a fixed-sized bitstream



Physical Layer

- Encoding is one of its important functionalities
- Vertical view
  - A reliable bit delivery channel for a fixed-sized bitstream
- Horizontal view
  - Sender: transfer bits to signals
  - Receiver: covert signals to bits



Summary

- Today
  - Encoding

- Next lecture
  - Framing and Error Handling