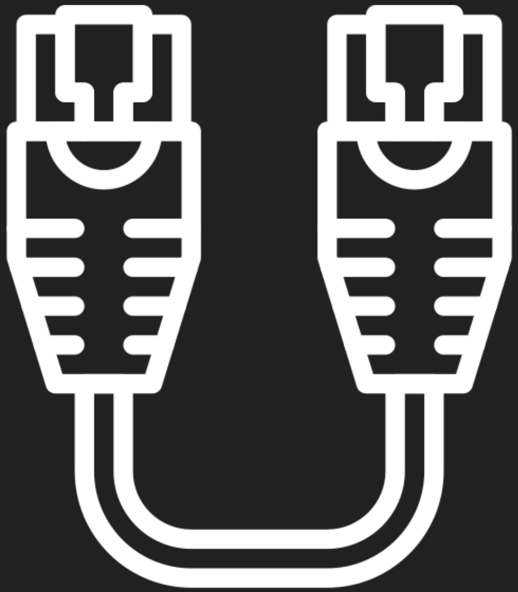
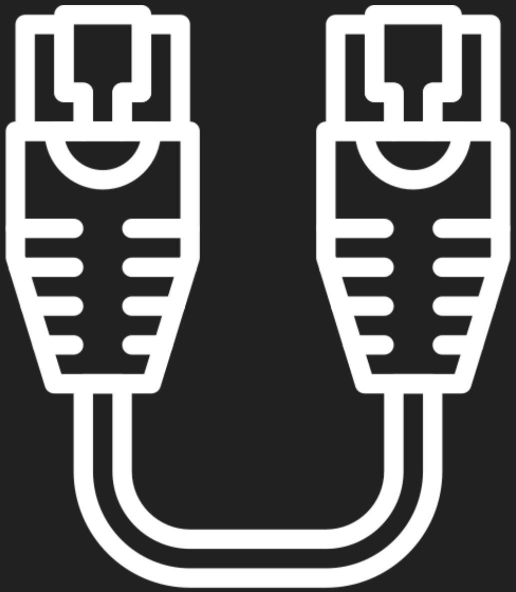
CS484/504
Computer Networks I

Joshua Reynolds

Fall 2022

Beginning of Semester Giveaway!*

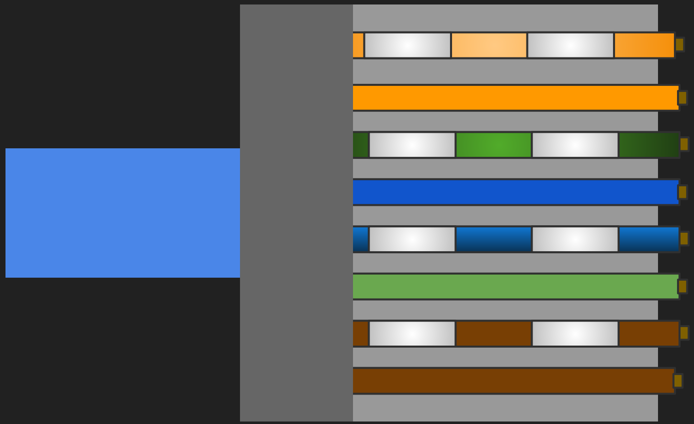




*Some assembly required

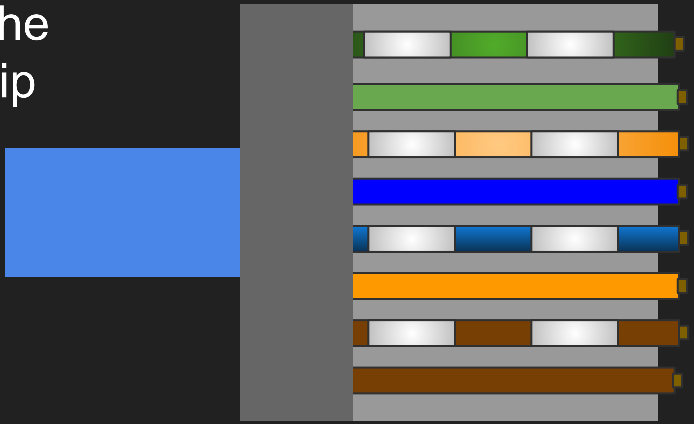
Twisted Pair Ethernet Cables with RJ45 terminators

T568B



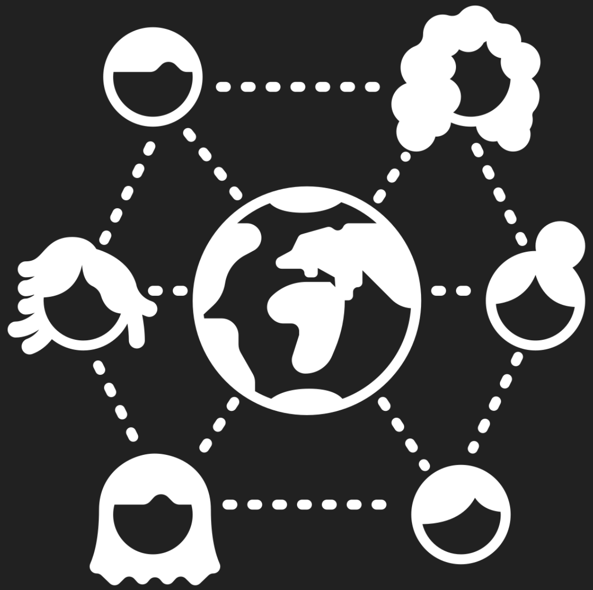
Looking from side opposite the retainer clip

T568A

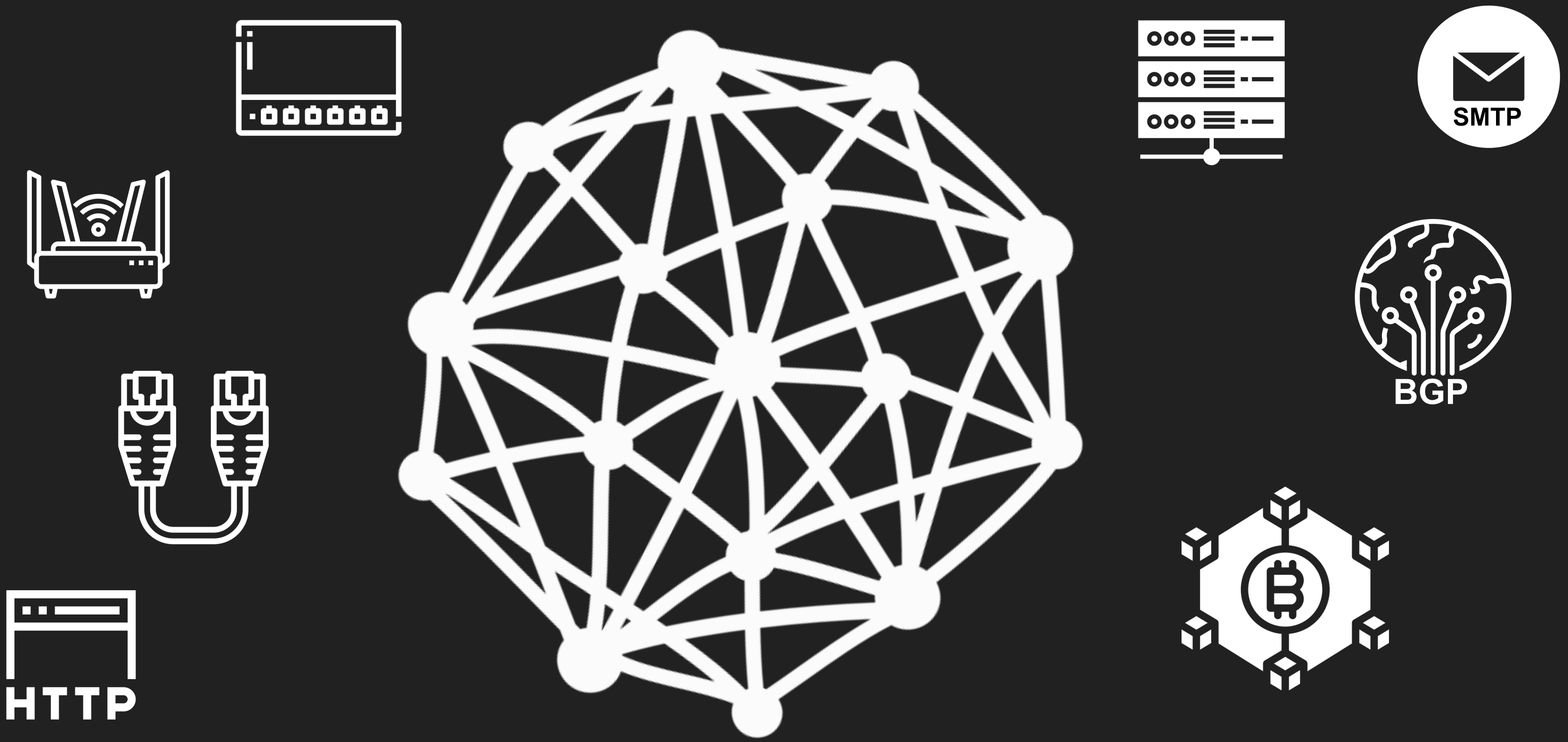


If both ends terminate the same, it is a **straight-through** or **patch** cable
If each end terminates differently, it is a **crossover** cable

The most complex system ever built by humans to be used by everyone



Computer Scientists Must Understand Networks



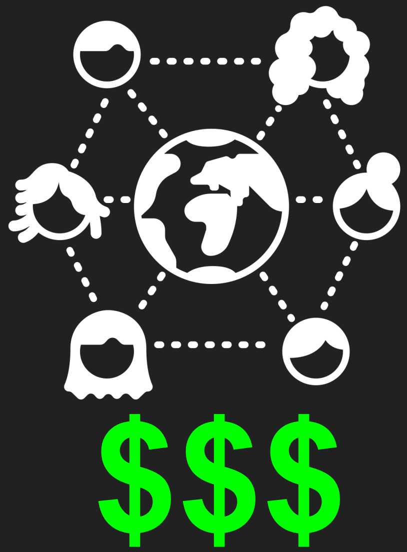
How big is the Internet?

$$$

How big is the Internet?

- 10.2% of US GDP ($2.1T)

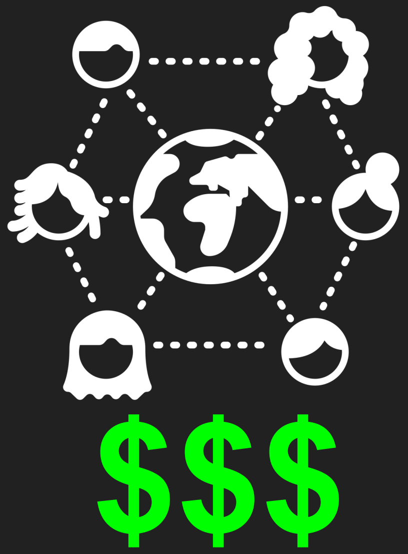
- U.S. Bureau of Economic Analysis, 2020



How big is the Internet?

- 10.2% of US GDP ($2.1T)
- Employs 7.8M people earning $1.1T in 2020

- U.S. Bureau of Economic Analysis, 2020



How can CS graduates build networking careers?

Blockchain Engineer

Network Engineer

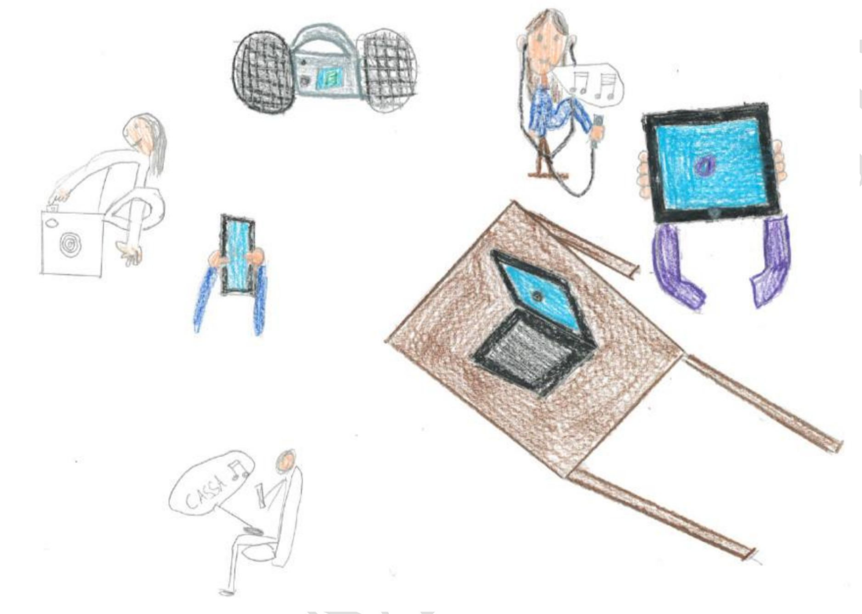
Network Security Engineer

System Administrator

Datacenter Engineer

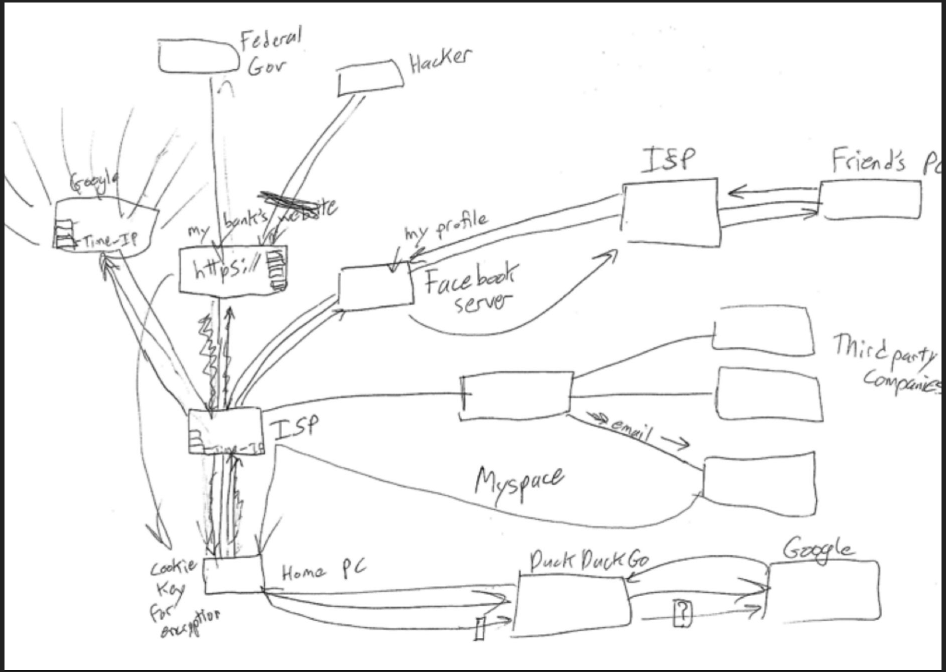
IT Network Specialist

Draw the Internet



“Draw the Internet.” A visual exploration of how children think an everyday technology, Luca Botturi, Scuola universitaria professionale della Svizzera italiana, Switzerland

Draw the Internet



Kang, Ruogu, et al. ““My Data Just Goes Everywhere:” User Mental Models of the Internet and Implications for Privacy and Security." Eleventh Symposium on Usable Privacy and Security (SOUPS 2015). 2015.

“Layers” Break Down Network Complexity

- Physical
- Data Link
- Network
- Transport
- Session
- Presentation
- Application

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	
Network	
Transport	
Session	
Presentation	
Application	

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	Sending and receiving protocols for physical connections
Network	
Transport	
Session	
Presentation	
Application	

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	Sending and receiving protocols for physical connections
Network	Efficiently routing data over many connections
Transport	
Session	
Presentation	
Application	

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	Sending and receiving protocols for physical connections
Network	Efficiently routing data over many connections
Transport	Handling lost, delayed, or out-of-order messages
Session	
Presentation	
Application	

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	Sending and receiving protocols for physical connections
Network	Efficiently routing data over many connections
Transport	Handling lost, delayed, or out-of-order messages
Session	Coordinating long-term conversations between programs
Presentation	
Application	

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	Sending and receiving protocols for physical connections
Network	Efficiently routing data over many connections
Transport	Handling lost, delayed, or out-of-order messages
Session	Coordinating long-term conversations between programs
Presentation	Hiding all the previous complexity to simplify use
Application	

“Layers” Break Down Network Complexity

Physical	Sending signals on wires, radios, fiber optics, etc.
Data Link	Sending and receiving protocols for physical connections
Network	Efficiently routing data over many connections
Transport	Handling lost, delayed, or out-of-order messages
Session	Coordinating long-term conversations between programs
Presentation	Hiding all the previous complexity to simplify use
Application	Build something that interacts with the real world

In this course you will learn how each layer works and why

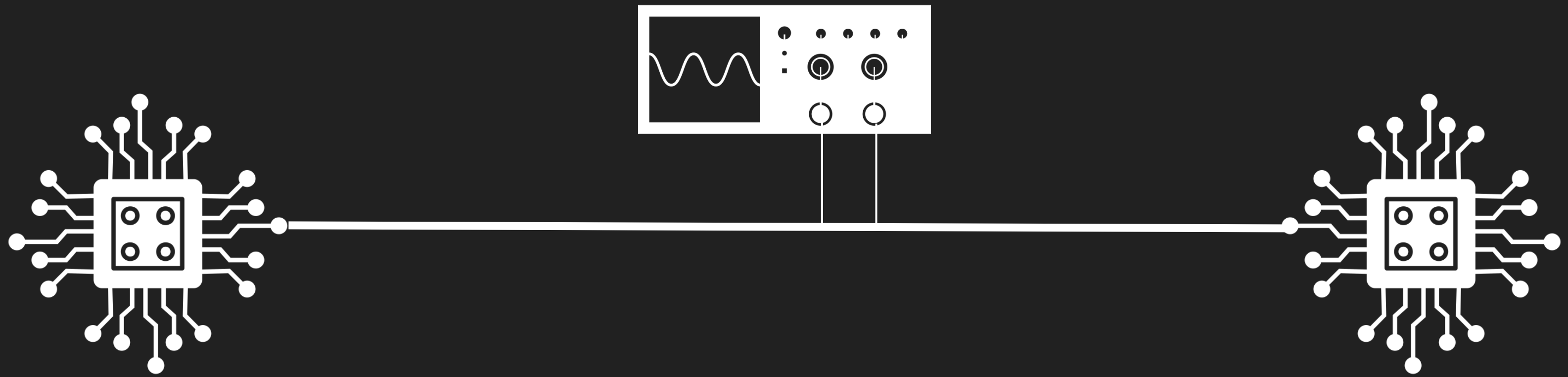
28 Lectures	(10%)
3 Projects	(10% each)
4 Homeworks	(2.5% each)
1 Midterm	(20%)
1 Final	(30%)

We will use this textbook:

Computer Networking: A Top-Down Approach
7th edition

By:
James F. Kurose and Keith W. Ross

Physical Layer - Sending Signals on Wires



“Bits” of Information

What is the simplest possible message?

“Bits” of Information

What is the simplest possible message?

Yes or No

True or False

0 or 1

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

2 options:	yes/no	$\log_2(2) = 1$ bit
256 options:	1 byte	$\log_2(256) = 8$ bits
65536 options	2 bytes	$\log_2(65536) = 16$ bits

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

- 2 options: yes/no $\log_2(2) = 1$ bit
- 256 options: 1 byte $\log_2(256) = 8$ bits
- 65536 options 2 bytes $\log_2(65536) = 16$ bits
- 26 options letter

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

- 2 options: yes/no $\log_2(2) = 1$ bit
- 256 options: 1 byte $\log_2(256) = 8$ bits
- 65536 options 2 bytes $\log_2(65536) = 16$ bits
- 26 options letter $\log_2(26) = 4.7$ bits

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

- 2 options: yes/no $\log_2(2) = 1$ bit
- 256 options: 1 byte $\log_2(256) = 8$ bits
- 65536 options 2 bytes $\log_2(65536) = 16$ bits
- 26 options letter $\log_2(26) = 4.7$ bits

What does it mean to carry 4.7 bits of information?

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

2 options:	yes/no	$\log_2(2) = 1$ bit
256 options:	1 byte	$\log_2(256) = 8$ bits
65536 options	2 bytes	$\log_2(65536) = 16$ bits
26 options	letter	$\log_2(26) = 4.7$ bits

What does it mean to carry 4.7 bits of information?

Practically, you would need 5 bits to carry a single letter. But…

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

- 2 options: yes/no $\log_2(2) = 1$ bit
- 256 options: 1 byte $\log_2(256) = 8$ bits
- 65536 options 2 bytes $\log_2(65536) = 16$ bits
- 26 options letter $\log_2(26) = 4.7$ bits

What does it mean to carry 4.7 bits of information?

Practically, you would need 5 bits to carry a single letter. But…

You could create a system to send 4 letters in only __ bytes instead of 20.

Information Theory

Distinguishing between *n* possible messages requires $\log_2(n)$ bits.

2 options:	yes/no	$\log_2(2) = 1$ bit
256 options:	1 byte	$\log_2(256) = 8$ bits
65536 options	2 bytes	$\log_2(65536) = 16$ bits
26 options	letter	$\log_2(26) = 4.7$ bits

What does it mean to carry 4.7 bits of information?

Practically, you would need 5 bits to carry a single letter. But…

You could create a system to send 4 letters in only **19** bytes instead of 20.

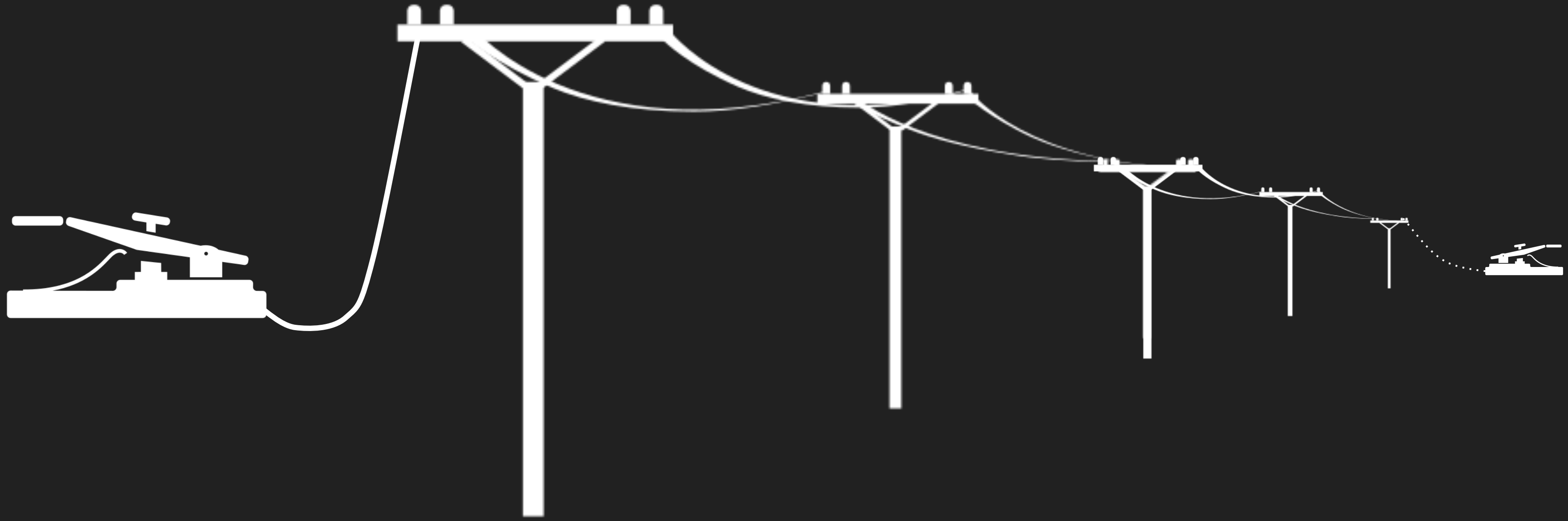
Rope Signaling Demo

Sending Signals - Pulse Amplitude Modulation

A sends a message to B using wire coils to cause voltage pulses in the wire. Each pulse has a specific amplitude.

B receives the message from A using its wire coils to detect voltage pulses from A and measure their amplitude.

Line Codes - interpreting voltage pulses as information



PAM Dimensionality

Each voltage pulse has a specific amplitude.

A and B agree ahead of time how many different pulse amplitudes to use.

If there are 3 amplitudes, it is called PAM3

If there are 5 amplitudes, it is PAM5

When multiple wires are used in parallel, each one may send a different amplitude.

If there are 16 amplitudes on 4 wires, we call it 4D-PAM16

If there are 4 amplitudes on 2 wires, we call it 2D-PAM4

Transmission Symbols

Consider a 4D-PAM5 signal. **How many wires? How many amplitudes?**

Transmission Symbols

Consider a 4D-PAM5 signal. 4 wires, 5 amplitudes.

Each time we get a pulse, it will have any of 5 values across 4 wires.

How many possible combinations?

Transmission Symbols

Consider a 4D-PAM5 signal. 4 wires, 5 amplitudes.

Each time we get a pulse, it will have any of 5 values across 4 wires.

5^4 = 625 different possible configurations

We call the specific configuration we see a “symbol”

Transmission Symbols

Consider a 4D-PAM5 signal. 4 wires, 5 amplitudes.

Each time we get a pulse, it will have any of 5 values across 4 wires.

$5^4 = 625$ different possible configurations

We call the specific configuration we see a “symbol”

How many bits of information is carried when we distinguish between 625 different possibilities?

Transmission Symbols

Consider a 4D-PAM5 signal. 4 wires, 5 amplitudes.

Each time we get a pulse, it will have any of 5 values across 4 wires.

5^4 = 625 different possible configurations

We call the specific configuration we see a “symbol”

How many bits of information is carried when we distinguish between 625 different possibilities?

$\log_2(625)$ = 9.29 bits of information carried in each symbol

Measures of transmission speed

Baud rate: number of transmission symbols per second (bd)

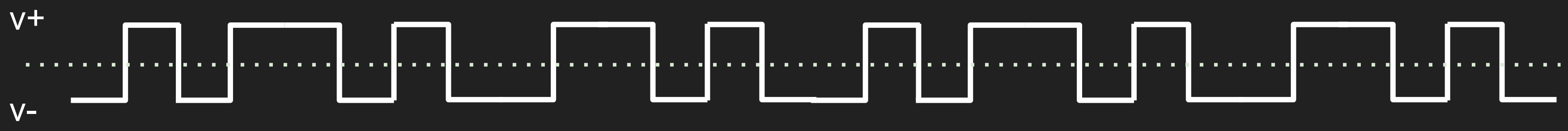
Bit rate: number of **b**its **p**er **s**econd (bps)

Bit rate = **Baud rate** *x* [the number of useful bits carried per symbol]

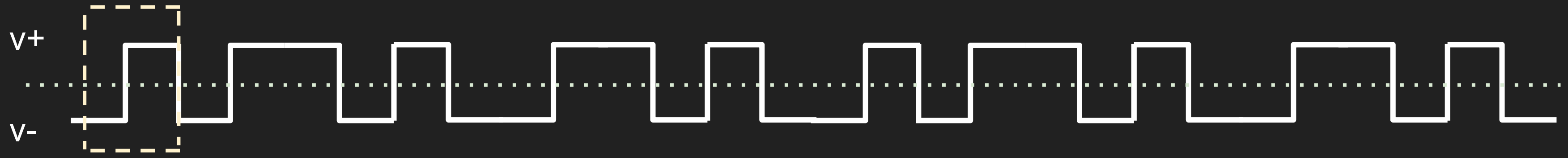
When each symbol carries one bit, the bit rate and baud rate are equal

If each symbol carries 2.4 useful bits, the bit rate is 2.4 times greater than the baud rate

Manchester Encoding - Ethernet 10BaseT

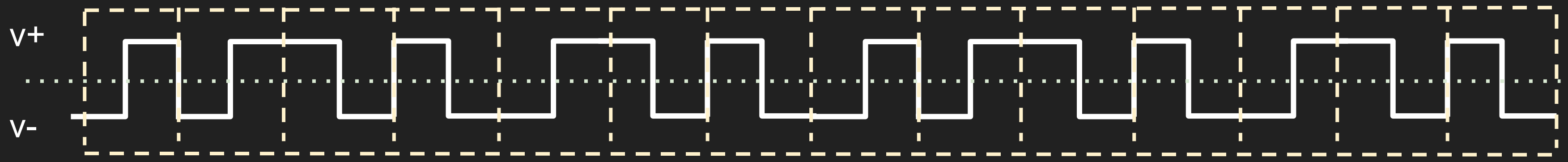


Manchester Encoding - Ethernet 10BaseT



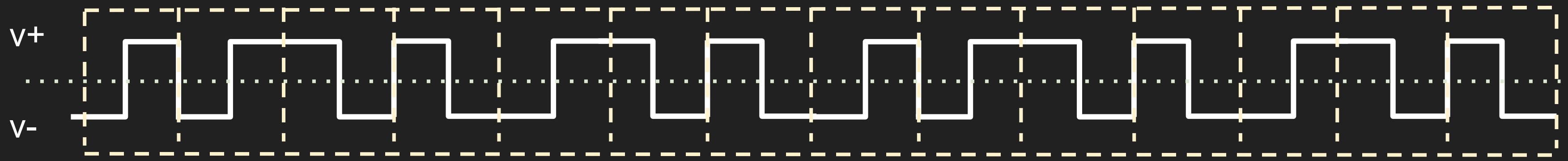
A symbol is sent every clock period. The symbol is a voltage transition.

Manchester Encoding - Ethernet 10BaseT



A symbol is sent every clock period. The symbol is a voltage transition.

Manchester Encoding - Ethernet 10BaseT

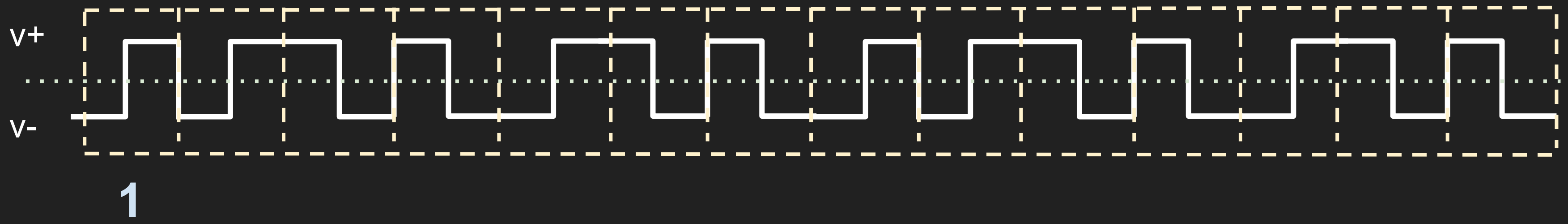


A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

Negative to positive voltage change = 1

Manchester Encoding - Ethernet 10BaseT

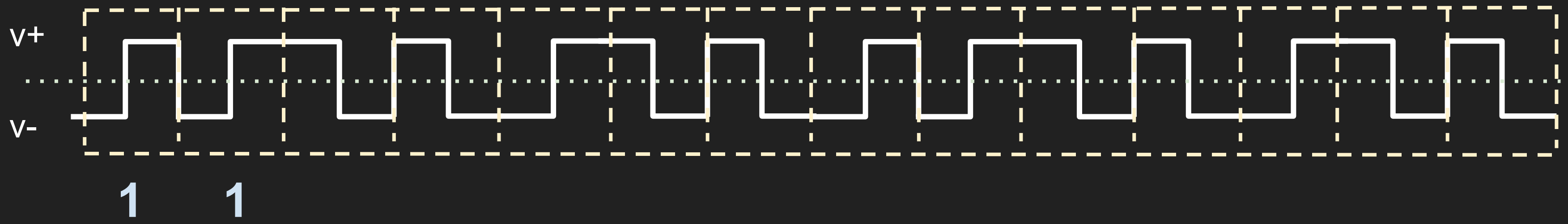


A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

Negative to positive voltage change = 1

Manchester Encoding - Ethernet 10BaseT

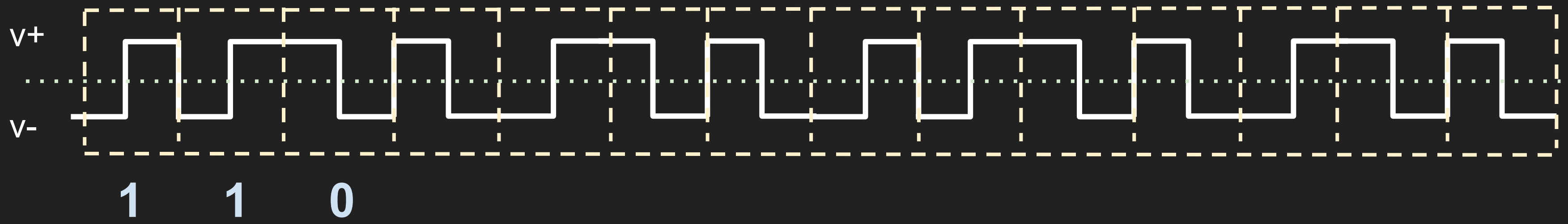


A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

Negative to positive voltage change = 1

Manchester Encoding - Ethernet 10BaseT

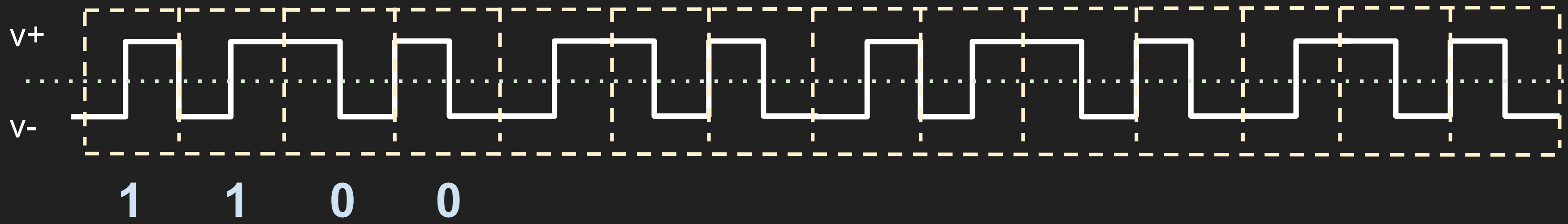


A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

Negative to positive voltage change = 1

Manchester Encoding - Ethernet 10BaseT

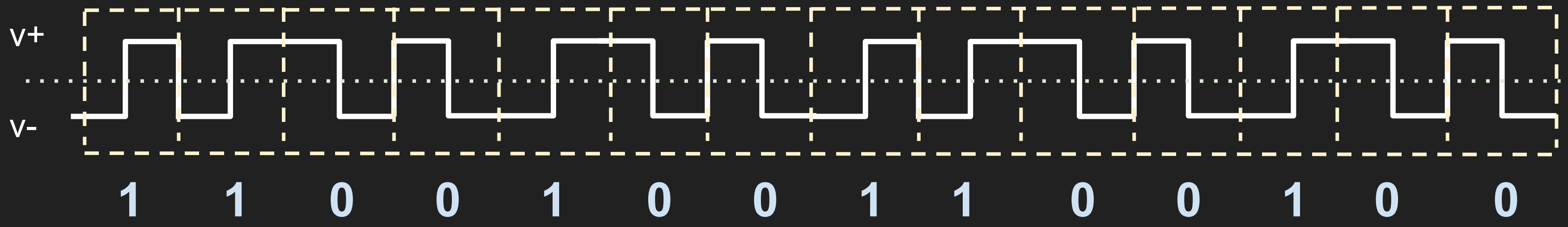


A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

Negative to positive voltage change = 1

Manchester Encoding - Ethernet 10BaseT



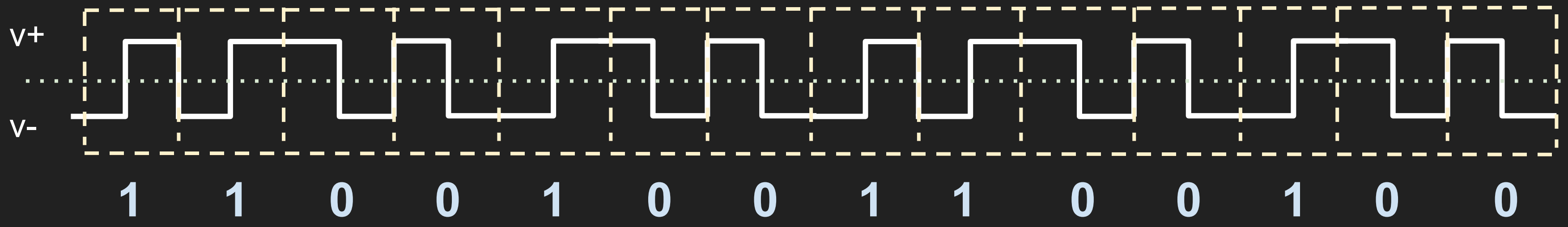
A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

Negative to positive voltage change = 1

Bit rate: 10 million bits per second. 1 bit per symbol

Manchester Encoding - Ethernet 10BaseT



A symbol is sent every clock period. The symbol is a voltage transition.

Positive to negative voltage change = 0

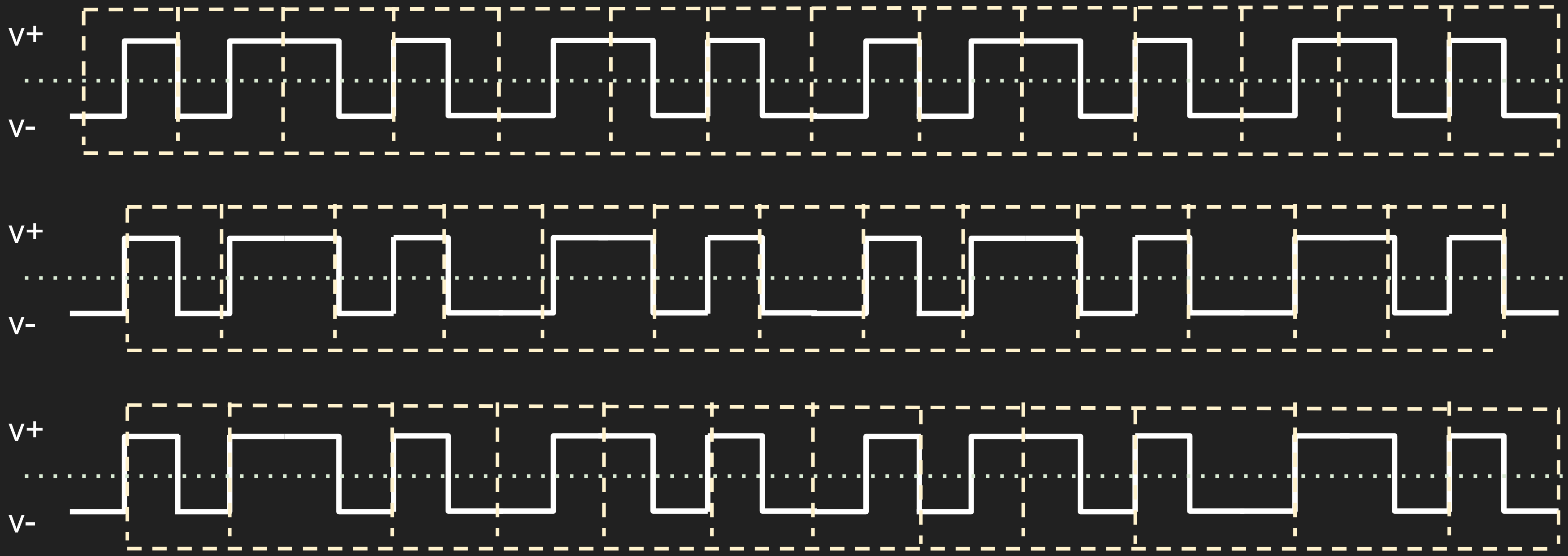
Negative to positive voltage change = 1

Bit rate: 10Mbps. 1 bit per symbol. Baud Rate: 10Mbd

https://www.youtube.com/watch?v=i8CmibhvZ0c

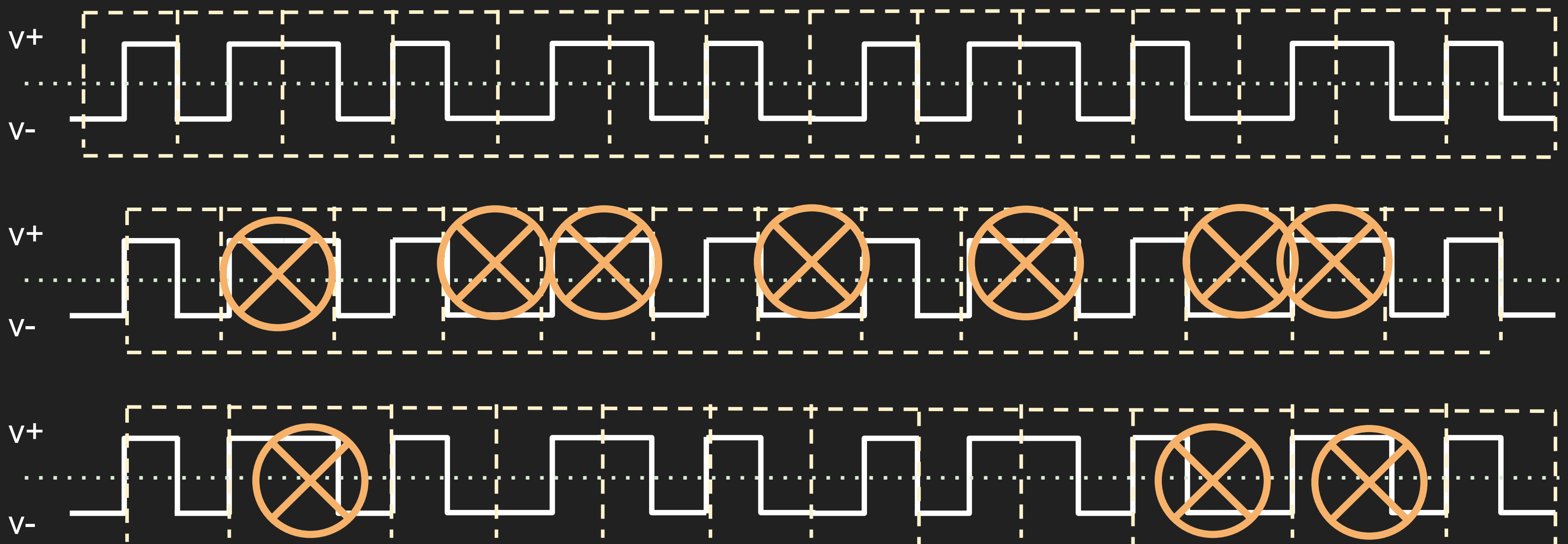
Manchester Encoding is "Self-Clocking"

You don't need to know the clock time period of the transmitter to receive.
Which clock division is correct?

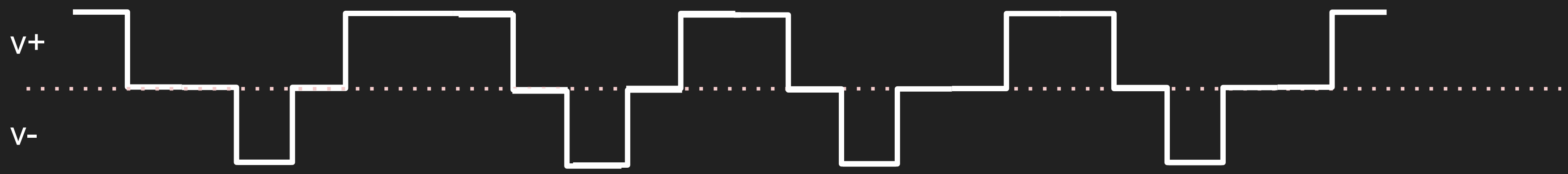


Manchester Encoding is "Self-Clocking"

You don't need to know the clock time period of the transmitter to receive.
Which clock division is correct?



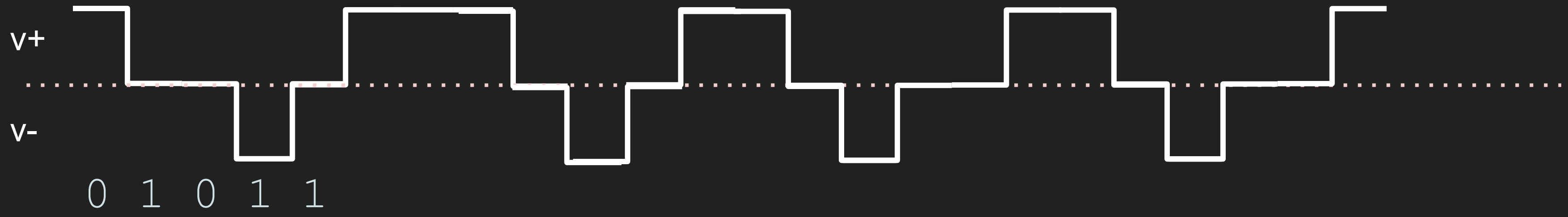
Multi-Level Transmit (MLT) for 100BaseTX



Every transition is a 1

Every clock period without a transition is 0

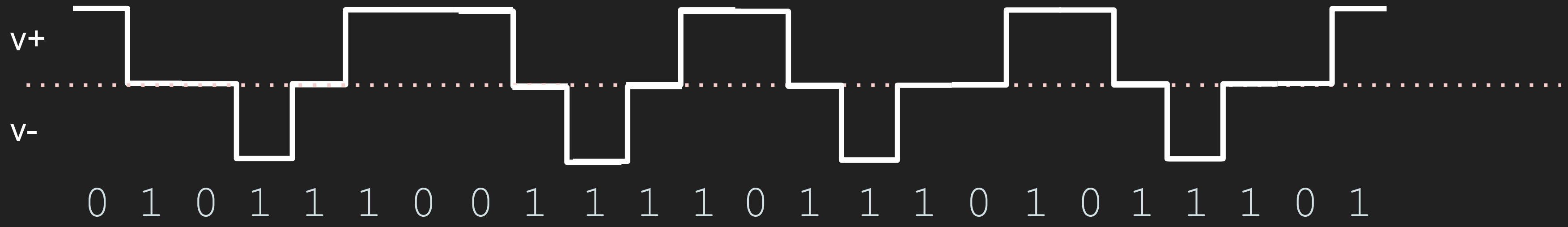
Multi-Level Transmit (MLT) for 100BaseTX



Every transition is a 1

Every clock period without a transition is 0

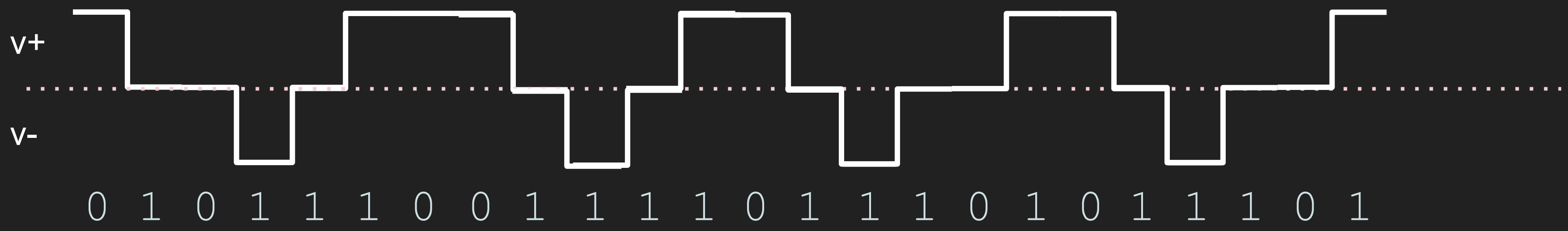
Multi-Level Transmit (MLT) for 100BaseTX



Every transition is a 1

Every clock period without a transition is 0

Multi-Level Transmit (MLT) for 100BaseTX

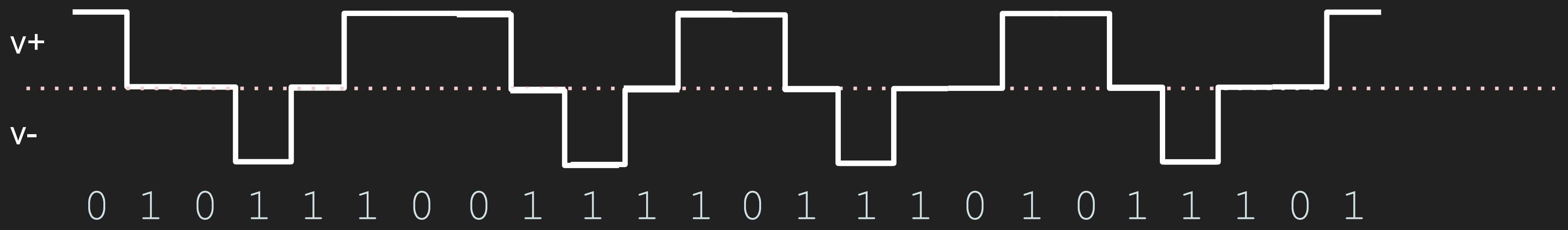


Every transition is a 1

Every clock period without a transition is 0

Is it self-clocking?

Multi-Level Transmit (MLT) for 100BaseTX



Every transition is a 1

Every clock period without a transition is 0

Is it self-clocking? Yes, except…

Self-Clocking Issues in MLT

What if I want to send one million zeros using MLT?

Do we lose track of the clock if there are no transitions?

We need to require ones frequently, but still allow sending all zeros

4B5T - Forcing enough ones to be self-clocking

For every 4 bits we want to send, we actually send 5 bits according to the table of 4B5T rules.

Example rule:

When you want to send “0000”, actually send “11110”

When you see “11110”, understand that as “0000”

4B5T - Forcing enough ones to be self-clocking

For every 4 bits we want to send, we actually send 5 bits according to the table of 4B5T rules.

Example rule:

> When you want to send “0000”, actually send “11110”
>
> When you see “11110”, understand that as “0000”

There are never more than three 0s in a row under the 4B5T rules.

4B5T - Forcing enough ones to be self-clocking

For every 4 bits we want to send, we actually send 5 bits according to the table of 4B5T rules.

Example rule:

> When you want to send “0000”, actually send “11110”
>
> When you see “11110”, understand that as “0000”

There are never more than three 0s in a row under the 4B5T rules.

What is the extra cost to do this?

4B5T - Forcing enough ones to be self-clocking

For every 4 bits we want to send, we actually send 5 bits according to the table of 4B5T rules.

Example rule:

When you want to send “0000”, actually send “11110”

When you see “11110”, understand that as “0000”

There are never more than three 0s in a row under the 4B5T rules.

What is the extra cost to do this? **4 bits becomes 5, so 25% extra cost**

4B5T - Forcing enough ones to be self-clocking

For every 4 bits we want to send, we actually send 5 bits according to the table of 4B5T rules.

Example rule:

When you want to send “0000”, actually send “11110”

When you see “11110”, understand that as “0000”

There are never more than three 0s in a row under the 4B5T rules.

What is the extra cost to do this? **4 bits becomes 5, so 25% extra cost**

Bit rate: 100Mbps. 0.8 bits per symbol. Baud rate: ???

4B5T - Forcing enough ones to be self-clocking

For every 4 bits we want to send, we actually send 5 bits according to the table of 4B5T rules.

Example rule:

When you want to send “0000”, actually send “11110”

When you see “11110”, understand that as “0000”

There are never more than three 0s in a row under the 4B5T rules.

What is the extra cost to do this? **4 bits becomes 5, so 25% extra cost**

Bit rate: 100Mbps. 0.8 bits per symbol. Baud rate: 125Mbd

1000BaseT Gigabit Ethernet

Can send and receive data on same wires at the same time

Uses all 4 wire pairs - 4D-PAM5

4 instances of 5 voltage levels carries $log_2(625) = 9.29$ bits of information per symbol.

1000BaseT Gigabit Ethernet

Can send and receive data on same wires at the same time

Uses all 4 wire pairs - 4D-PAM5

4 instances of 5 voltage levels carries $log_2(625) = 9.29$ bits of information per symbol.

8 bits are usable, the other 1.29 are used for error correction

Bit rate: 1Gbps. 8 bits per symbol. **Baud rate:** ???

1000BaseT Gigabit Ethernet

Can send and receive data on same wires at the same time

Uses all 4 wire pairs - 4D-PAM5

4 instances of 5 voltage levels carries $\log_2(625) = 9.29$ bits of information per symbol.

8 bits are usable, the other 1.29 are used for error correction

Bit rate: 1Gbps. 8 bits per symbol. **Baud rate:** 125Mbd

>1Gbps

10 Gigabit per second is the fastest widely-available using copper cables

Projects exists to push up to 200Gbps and beyond on copper!

Use Cat 6, Cat 7, etc. cables that are shielded from interference

Limited to shorter lengths of cable

Want to learn more? Take *EE395*

Signal vs Noise

EM fields exist all around us

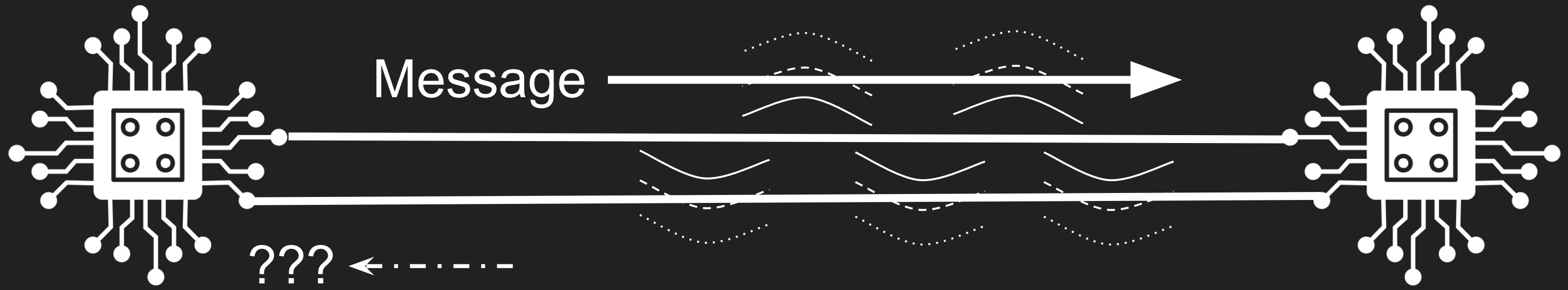
Waves that are not part of the message we call “Noise”

Which ones are signals meant for us, vs just noise coming from another pieces of electronics, the earth, the sun?

As we measure smaller and smaller signals faster and faster, the harder it is to tell the difference between signal and noise

If there is too much “noise”, we can’t hear the signal

Crosstalk - Signals jumping between wires



Twisted Pairs

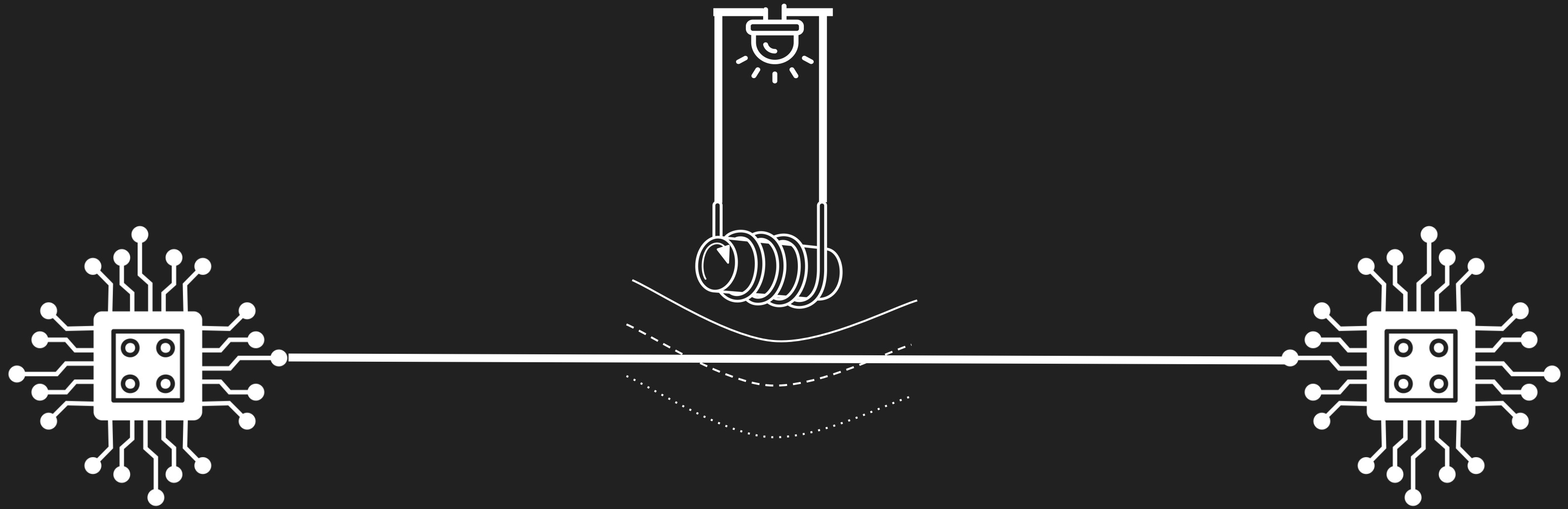
Transmit exact opposite waves on two wires and twist them together

The EM fields cancel out most of the cross-talk

Less energy lost over distance

Untwisted Pair Demo

Electro-Magnetic Interference



Serial Cable Demo

Shielding

EM fields flow around the outside of conductors

Faraday Cage:

https://www.youtube.com/watch?v=x7uCAvEhP1E

Cat 6, 7, 8 cables are shielded to block external “noise”

Cat 7 demo

Why do we care about the physical layer?

Thinking in layers means we can ignore lower levels that “just work”

We call this idea “abstraction” or “indirection” and we use it all over CS to make our life easier

But what happens when it stops “just working”?

Knowing how the physical layer helps you troubleshoot

What physical layer problems could a network have?

Image Attribution

“internet” by Richard from the Noun Project

Network by IconsGhost from Noun Project

“career” by Phạm Thanh Lộc from Noun Project

Network Switch by monkik from Noun Project

Router by Studio 365 from Noun Project

Ethernet to Ethernet by Ben Davis from Noun Project

Server by Creative Mania from Noun Project

photonics by WEBTECHOPS LLP from Noun Project

Blockchain by iconixar from Noun Project

http by Bearfruitidea from Noun Project

Email by i cons from Noun Project

circuit by Vectors Point from Noun Project
Oscilloscope by Devin Goodall from Noun Project
telegraph by Luke Anthony Firth from Noun Project
power line by Hayden Kerrisk from Noun Project
Battery by Anusha Narvekar from Noun Project
LED by Adrien Coquet from Noun Project
coil by Oleksandr Panasovskyi from Noun Project
coil by Nikita Kozin from Noun Project

Shannon, Claude — Author: Jacobs, Konrad — Source: Konrad Jacobs, Erlangen — Copyright CC BY-SA 2.0 de