DTU

Embedded Systems for Vehicles

# Week 6: Wired Networked Embedded Systems

**Xenofon (Fontas) Fafoutis**
Associate Professor
xefa@dtu.dk
www.compute.dtu.dk/~xefa

# From Intra-Board Communication to Local Networks

- Abstraction layers: The OSI reference model

| Layer | | Data Unit | Function |
|---|---|---|---|
| 7 | Application | Data | Distributed application protocols |
| 6 | Presentation | Data | Translation of data (encryption/decryption, compression, etc) |
| 5 | Session | Data | Managing communication sessions |
| 4 | Transport | Segment | Transmission of segments to processes |
| 3 | Network | Packet | Transmission of packets to nodes over multiple networks |
| 2 | Data Link | Frame | Transmission of frames between physically-connected nodes |
| 1 | Physical | Bit | Transmission and reception of bits over a physical medium |

# Physical Layer (PHY)

- The Physical Layer defines means for transmitting raw bits over a physical medium
- It is typically implemented in hardware (PHY chip)
- It abstracts the physical aspects of communication
  - Modulation, demodulation, bit synchronisation, line coding, multiplexing, filtering…
- All these procedures are abstracted to:
  - Bits transmitted on one end and received to the other end


RTL8201 Ethernet PHY chip

Image source: Wikipedia

# Line Coding

- Defines how the logical ‘0’ and ‘1’ are physically represented on the line

- NRZ (Non-Return-to-Zero): signal does not return to zero at the middle of the bit
  - NRZ-L: ‘0’ is zero voltage, ‘1’ is positive voltage
  - NRZ-M or NRZ-I: ‘0’ is same voltage, ‘1’ is a transition
  - NRZ-S: ‘1’ is same voltage, ‘0’ is a transition

- RZ (Return-to-Zero)
  - ‘0’ is zero voltage, ‘1’ is positive voltage returns to zero at the middle of the bit

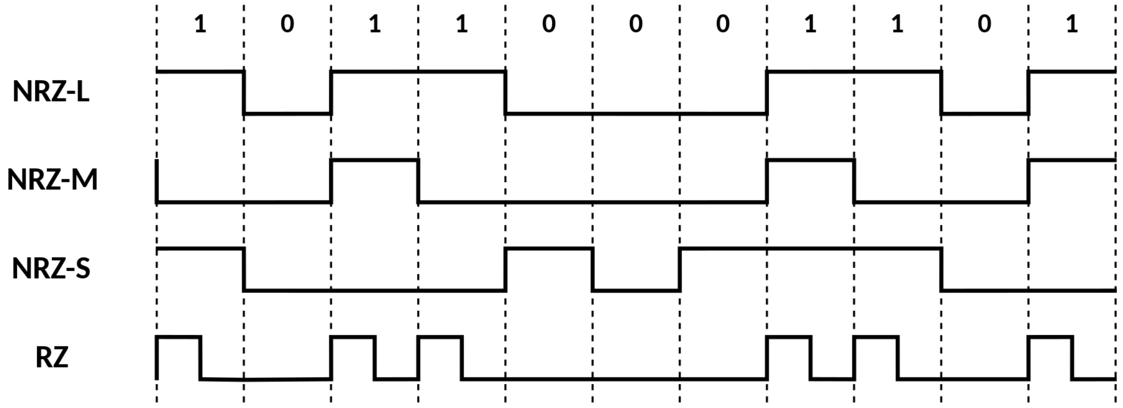



Image source: Wikipedia

# Manchester Coding

- Encodes logical value as a transition in the middle of the bit
- Original:
  - ‘0’ is a low-to-high transition
  - ’1’ is a high-to-low transition
- IEEE 802.3 (Ethernet)
  - ‘0’ is a high-to-low transition
  - ’1’ is a low-to-high transition
- Self-clocking: Digital inputs can detect transitions without the need of a clock
- Requires double the bandwidth

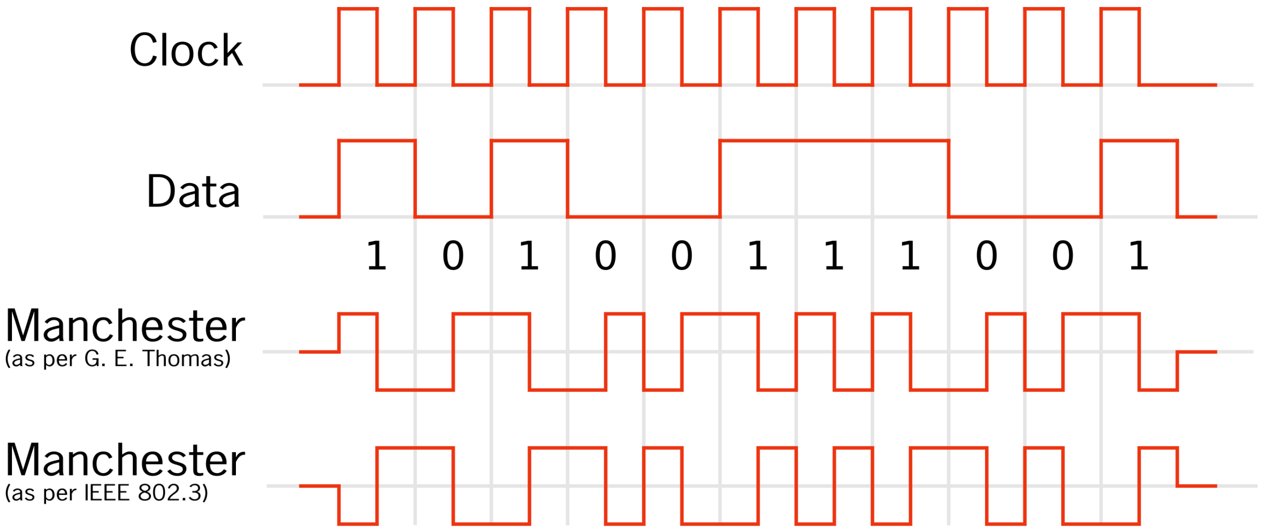
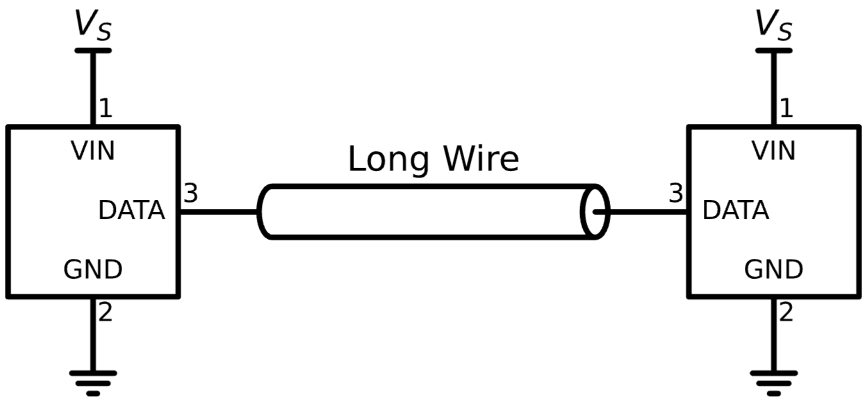
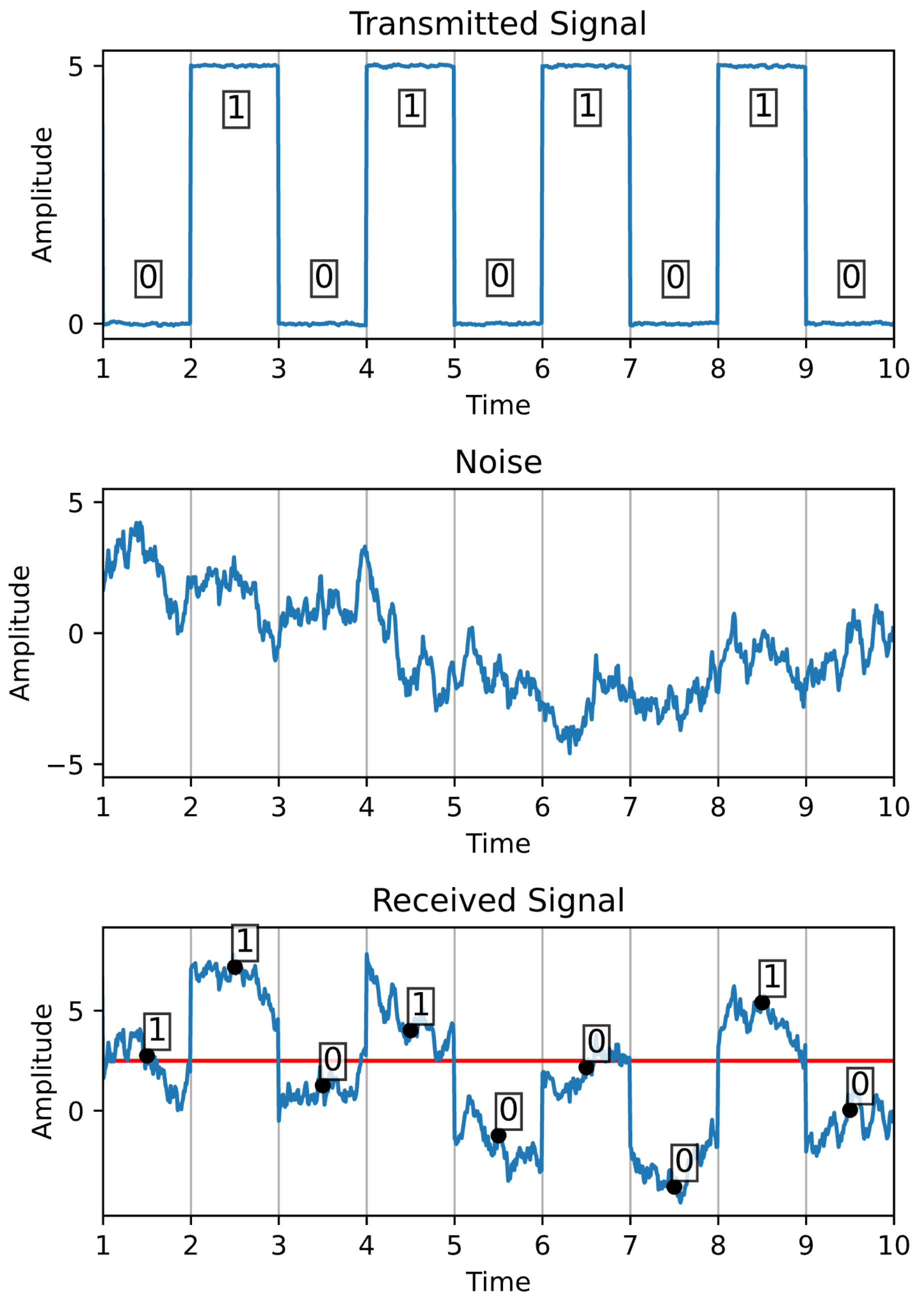



Image source: Wikipedia

# Transmission over Long Wires

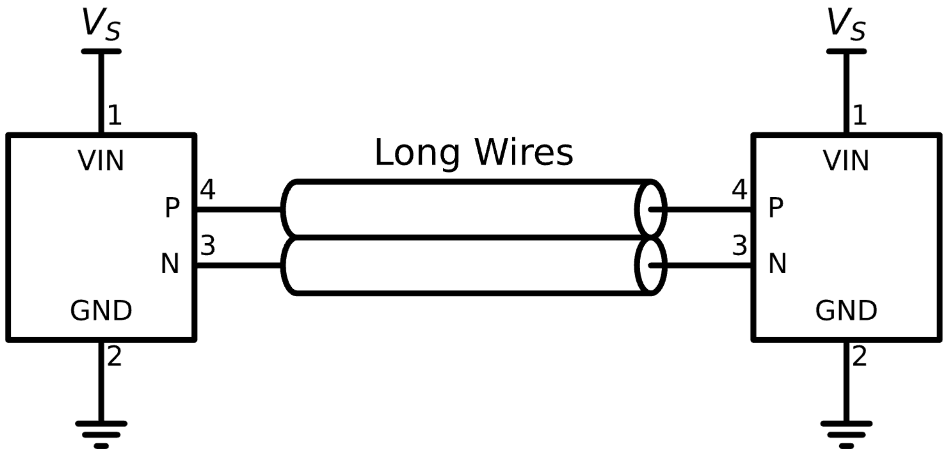
- Wires are imperfect conductors
  - They have some small resistance (mOhm/m)
  - The have a small capacitance
- These imperfections are negligible in short wires but have side-effects when the wires are longer
  - Wires pick up electromagnetic interference (EMI)
  - Ground level at both sides not the same
  - High frequencies get attenuated
- Result: **Bit Errors**

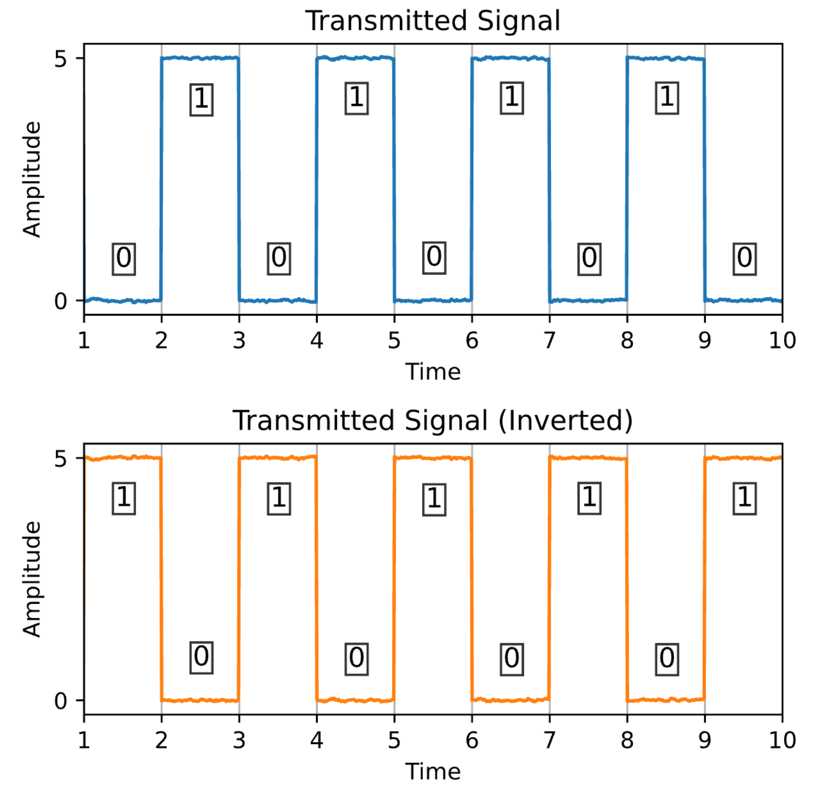


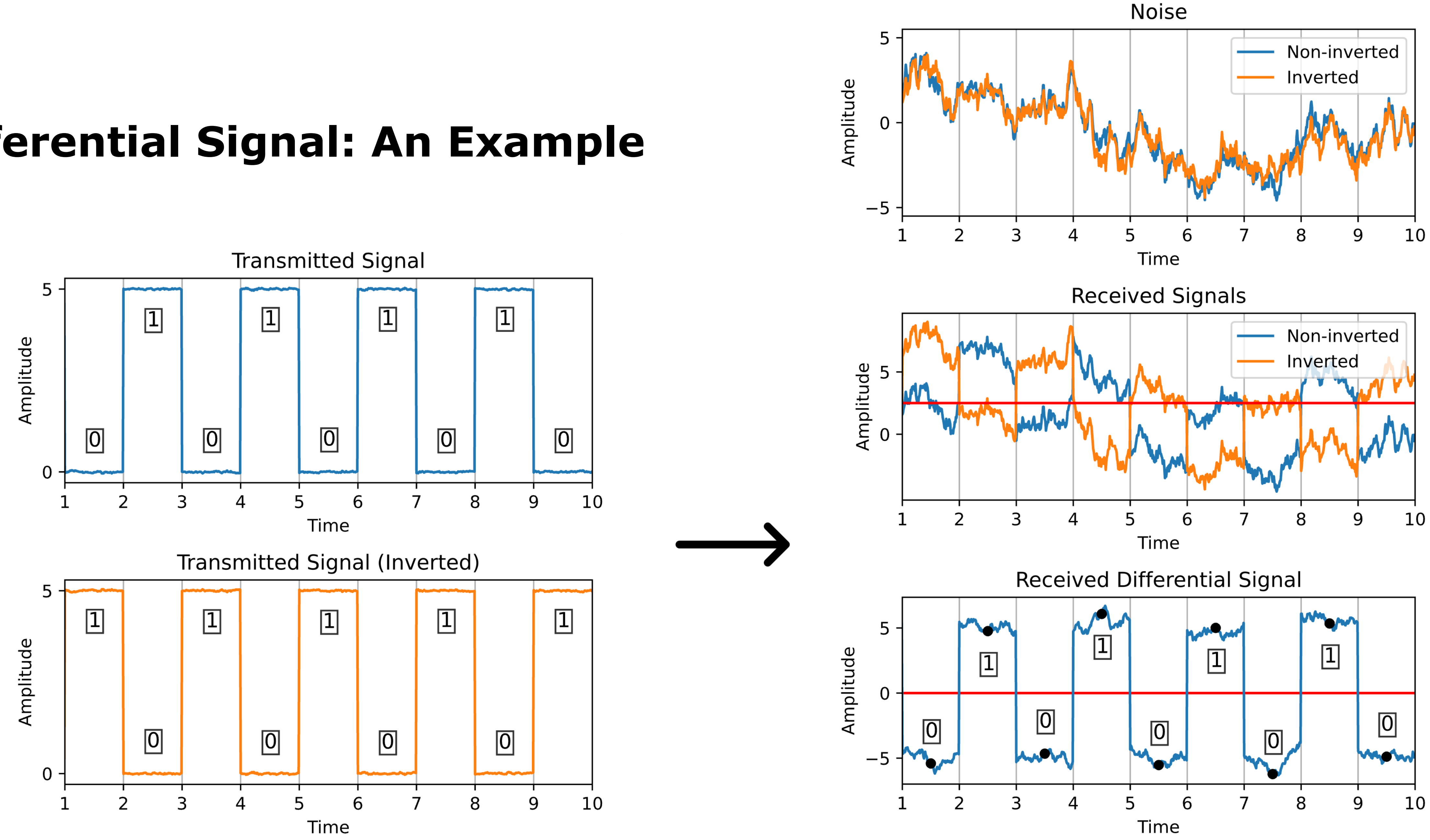
# Differential Signalling with Balanced Lines

- We connect the two devices with two wires of the same type and length (balanced lines)
- We transmit the signal and the inverted data signal
  - Non-inverted Signal (P): '0' is GND, '1' is $V_S$
  - Inverted Signal (N): '0' is $V_S$, '1' is GND
- Similar to a differential ADC, the receiver measures the difference of the two (P-N)
  - A positive difference means '1' and vice versa
- Advantages
  - Noise cancellation due to balanced lines
  - Additional noise resilience due to double signal voltage at the receiver
  - Enables low-voltage operation (lower consumption, higher speeds)



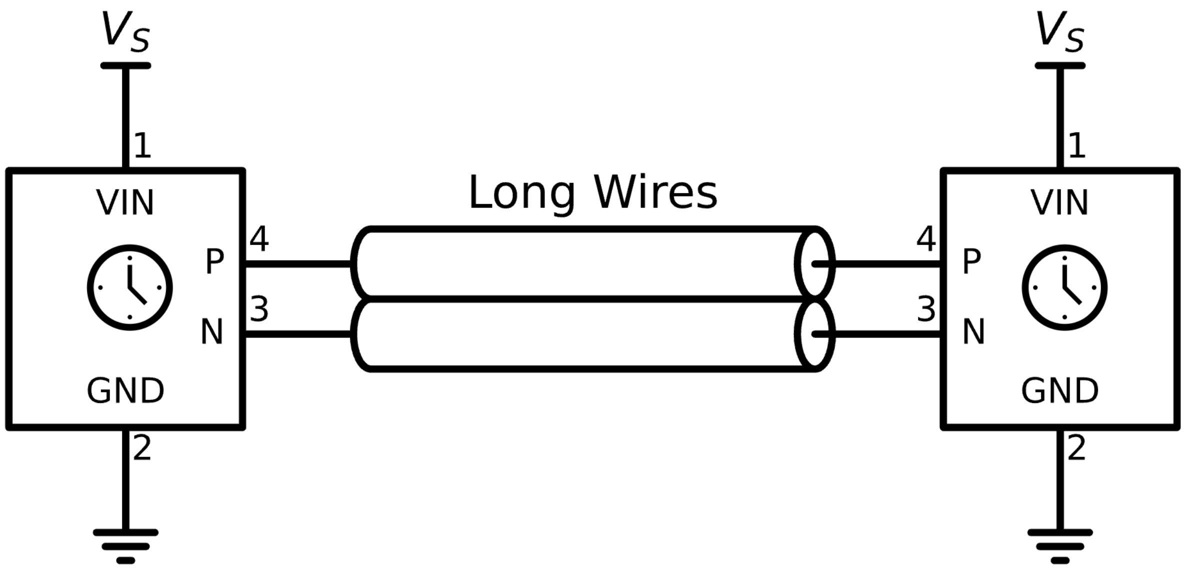



# Differential Signal: An Example

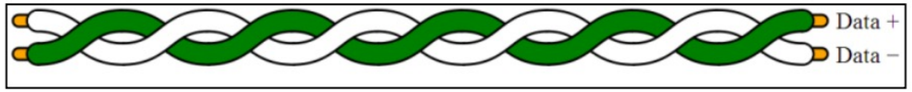


# Synchronisation at longer distances

- Devices operate asynchronously using their own clocks (no clock line)
  - Clocks that drift relative to each other
- Transmission rates are pre-agreed and defined by the protocol
- Changes on the line (high-to-low, low-to-high) in predetermined patterns are used for synchronisation
- Long periods without a change on the line can cause de-synchronisation
  - Bit stuffing: adding bits to force a transition



# USB 2.0 (Universal Serial Bus)




- Uses differential signalling with balanced lines
  - Lines are twisted to reduce noise

- Differential '1': D+ high, D- low
- Differential '0': D+ low, D- high

- Encodes data in NRZ-I with bit stuffing
  - Logic '1': no change in voltage level
  - Logic '0': change in voltage level
  - To maintain synchronisation, after 6 consecutive '1's, adds a '0' to ensure a transition (red in figure)

| Name | Colour | Function |
|---|---|---|
| VBUS | Red | 5V power supply |
| D- | White | Data (-), 3.3V |
| D+ | Green | Data (+), 3.3V |
| GND | Black | Ground |

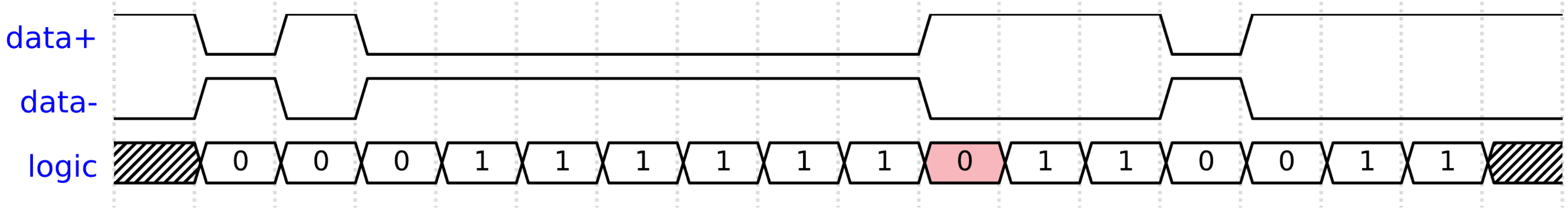
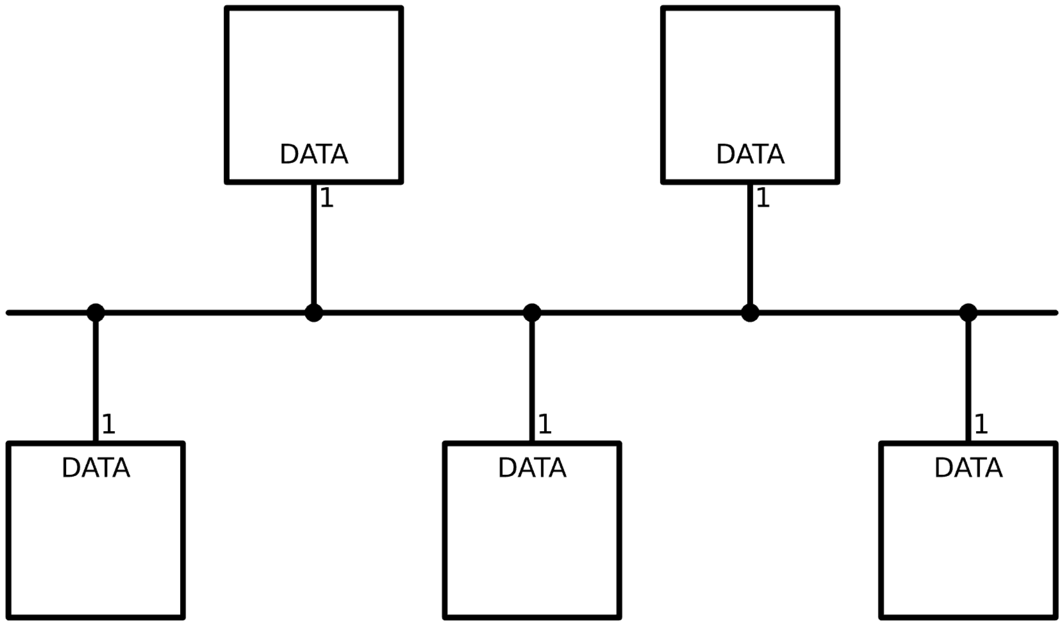



Image source: AN57294 by Cypress Semiconductor

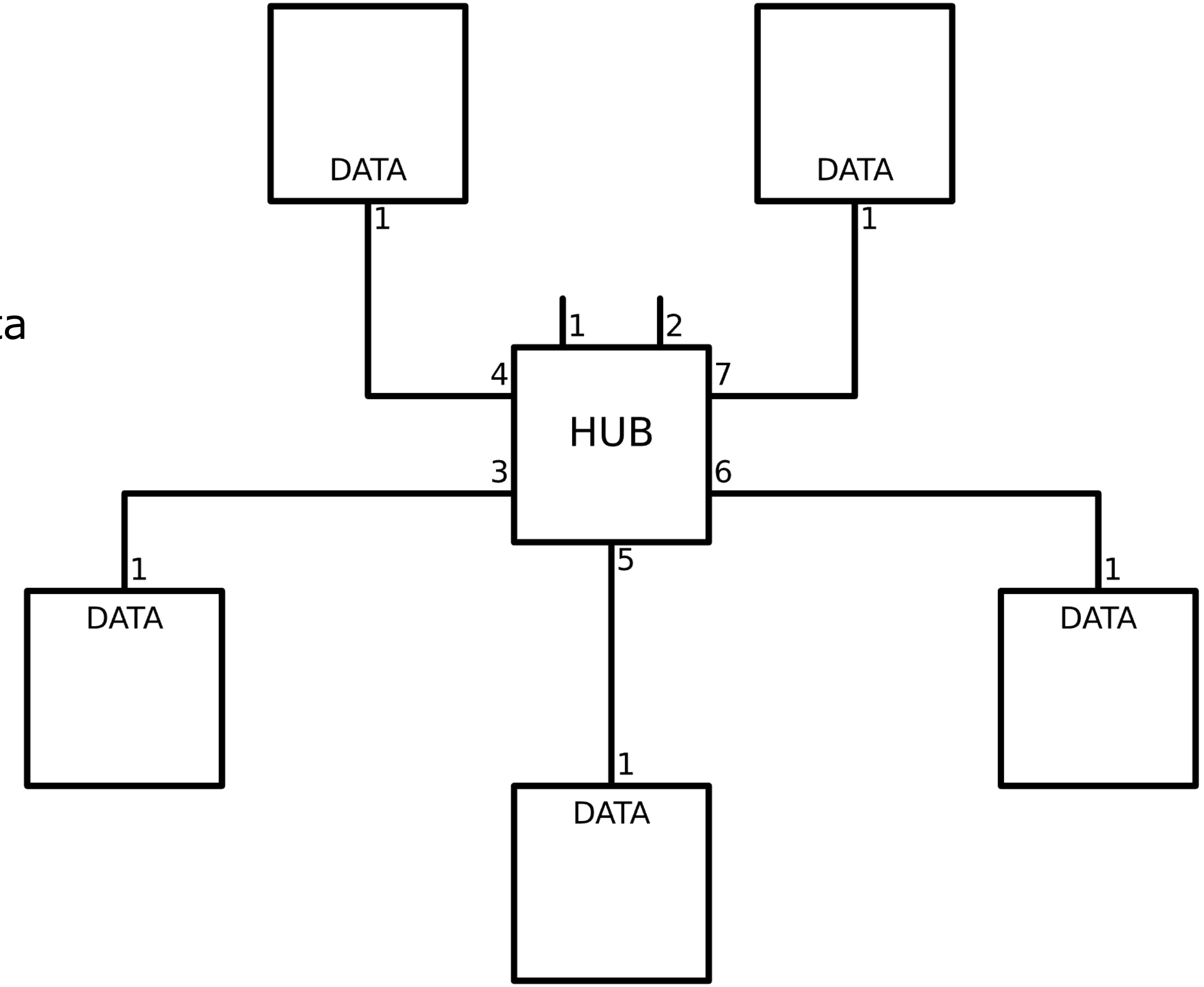
# Bus Topology

- Devices tap on a shared cable
- Good fit for environments that are long
  - E.g. elevators, assembly lines, cars
- Easy to add devices
  - Anywhere on the cable
- Vulnerable to cable damage
  - Break in the cable splits bus in two
- Commonly found in embedded controller networks



## Star Topology

- Devices connect to a hub with a dedicated cable
- In its simplest form, the hub retransmits the data received from one port to all others
  - Emulates a bus
- Hub is a centralised point that can implement more intelligent features
- Easy to detect and isolate failures
  - A broken wire affects only one device
- Hub is a single point of failure



# Data Link Layer

- Responsible for the transmission of data frames between physically-connected nodes (local networks)

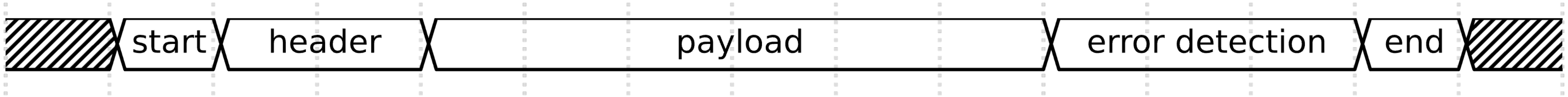
- Logical Link Layer:
  - Common interface to protocols above
  - Error control (retransmissions, acks, etc)
  - Flow control

- Medium Access Control (MAC) Layer
  - Frame format and synchronisation
  - Physical addressing
  - Error detection
  - Coordinates access to a shared medium

# Frames

- Defining the meaning of sequences of bits

- Typical fields
  - Start: Designates the beginning of frame, used by receiver to synchronise
  - Header: Protocol-specific information, such as configurations, addresses, etc
  - Payload: Higher-level data
  - Error Detection: A code used to detect errors (bit flips)
  - End: Designates the end of the frame



# Error Detection

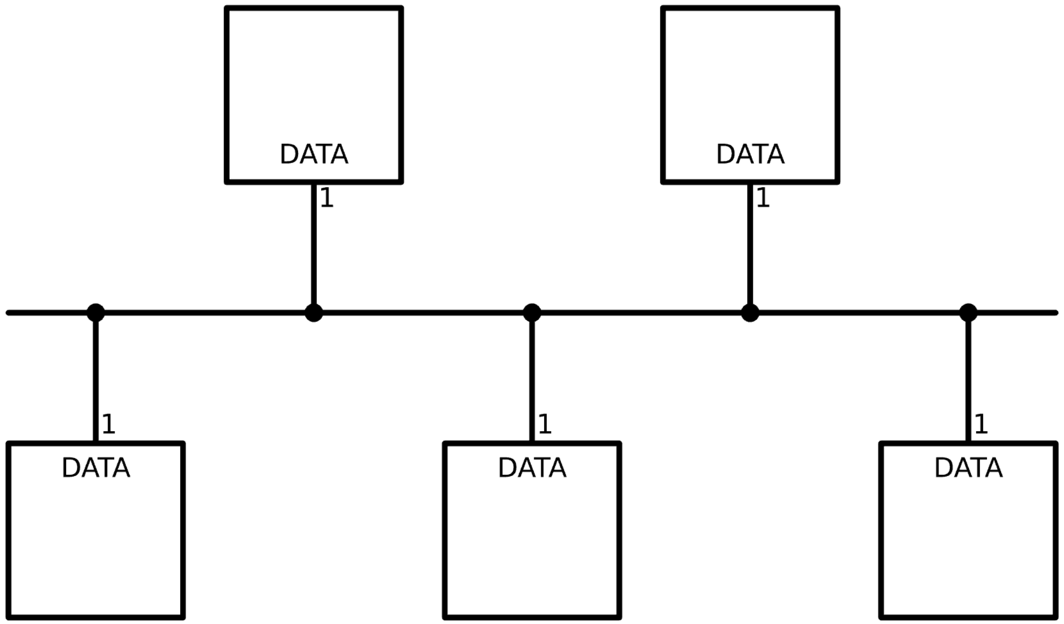
- Main idea:
  - Add a code to the frame that is generated from the header/payload of the frame
  - Receiver repeats the process and if it generates a different code, the frame is discarded

- Parity check
  - Adds one bit to ensure that the total number of ‘1’ is odd or even as pre-agreed
  - Detects 1-bit errors

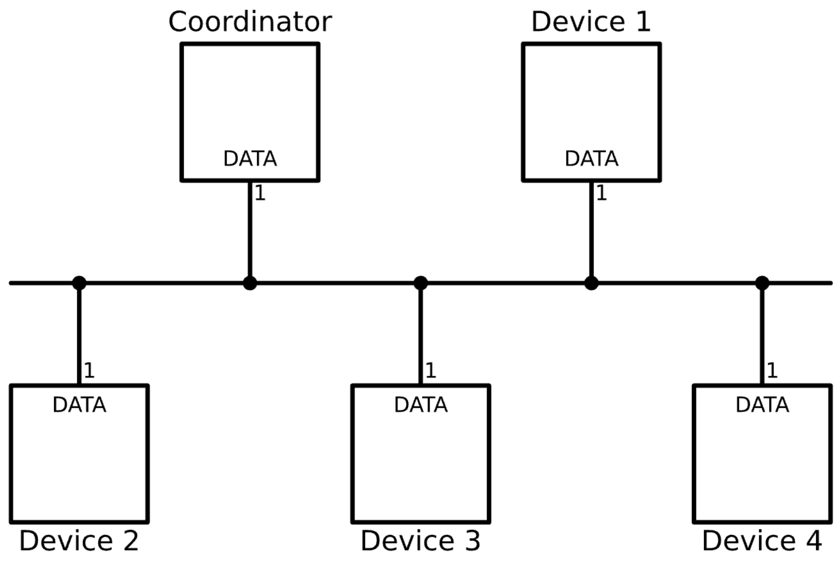
- Cyclic Redundancy Checks (CRC)
  - Adds a CRC code of N bits (e.g. CRC-15 adds 15 bits)
  - Detects more sophisticated errors

# Sharing the medium

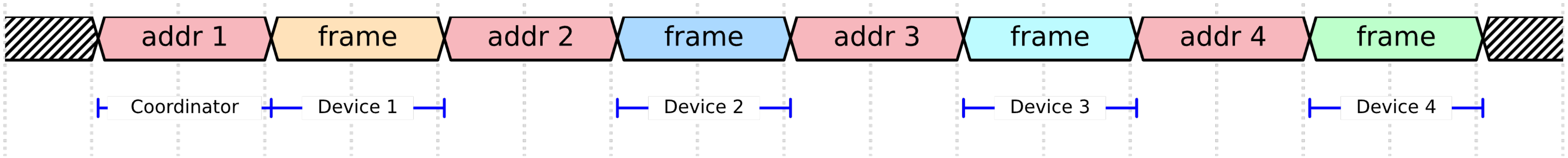
- Medium Access Control (MAC)
- Two devices cannot transmit at the same time on a shared bus
- How we can avoid such collisions?
- Who gets priority?



# Polling



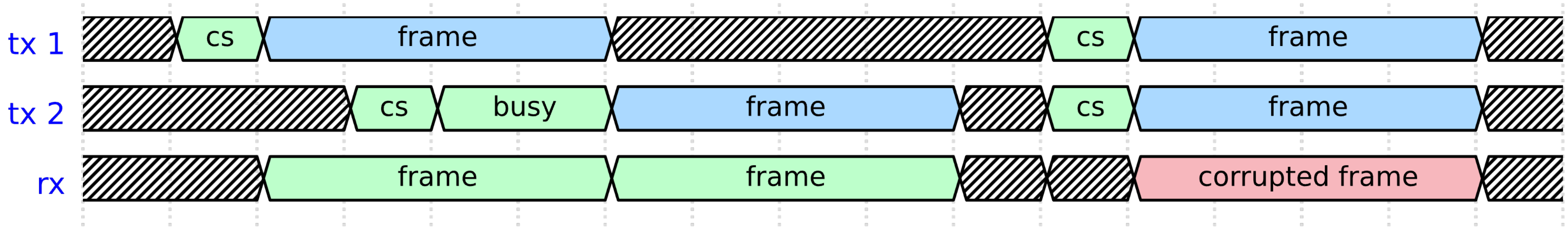

- Coordinator polls the the other devices
  - Device does not transmit unless polled
- All communication goes through the coordinator
  - Device-to-device communication not possible
- Coordinator can give more transmission opportunities to devices that need
  - E.g. Device 1 -> Device 2 -> Device 3 -> Device 3 -> Device 4 -> Device 1 …
- Advantages: simple and bounded latency
- Disadvantages: polling overhead, device-to-device overhead, not easily extendable, single point of failure



# CSMA (Carrier Sense Multiple Access)

- Listen-before-talk
- The transmitter monitors the wire for transmissions (carrier sense, cs)
  - If idle, it proceeds with the transmission
  - If busy, it waits and transmits when it becomes idle
- Collisions can happen if two devices transmit (almost) at the same time
  - Transmitters are not aware that collision happened

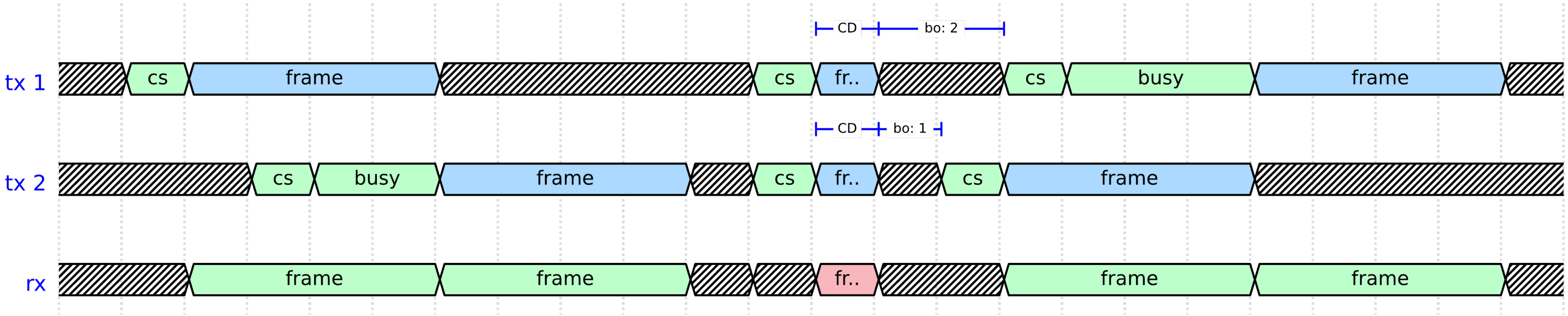
Blue: transmit
Green: receive
Red: receive with errors



# CSMA/CD (Collision Detection)

- Collision Detection
  - While transmitting also sense medium for other transmissions
  - If collision detected, wait some time randomly and try again layer
- Advantages:
  - Collisions are stopped early without wasting a lot of bandwidth
  - Retransmissions

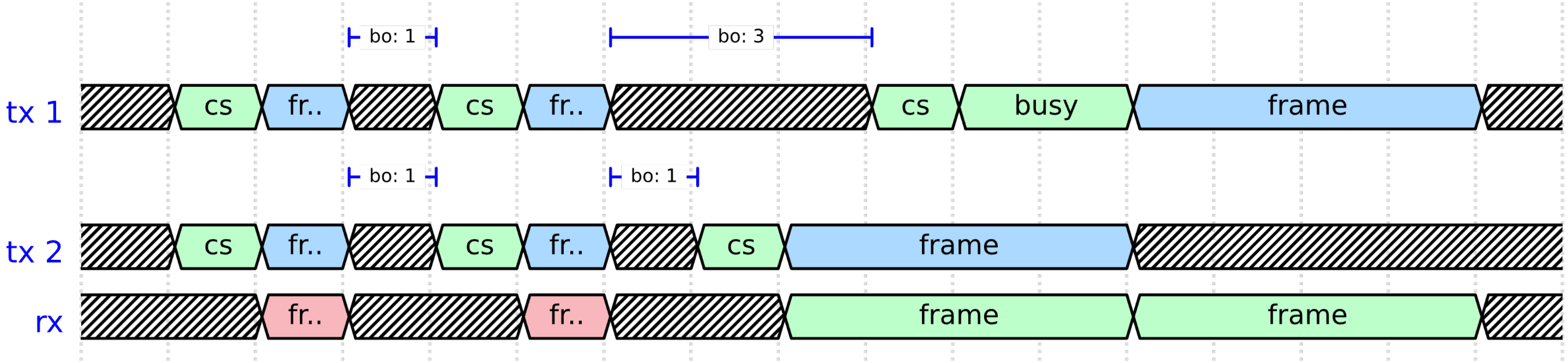
Blue: transmit and listen
Green: receive
Red: receive with errors



# Binary Exponential Backoff

- After collision choose random number *bo* in [0, CW] and back off for *bo* time units
- How large shall the contention window be?
  - Too small: high probability for another collision in high traffic
  - Too big: unnecessarily large idle time in low traffic
- Exponential backoff (initially CW=1)
  - After each collision CW doubles
  - After a successful transmission it resets to CW=1

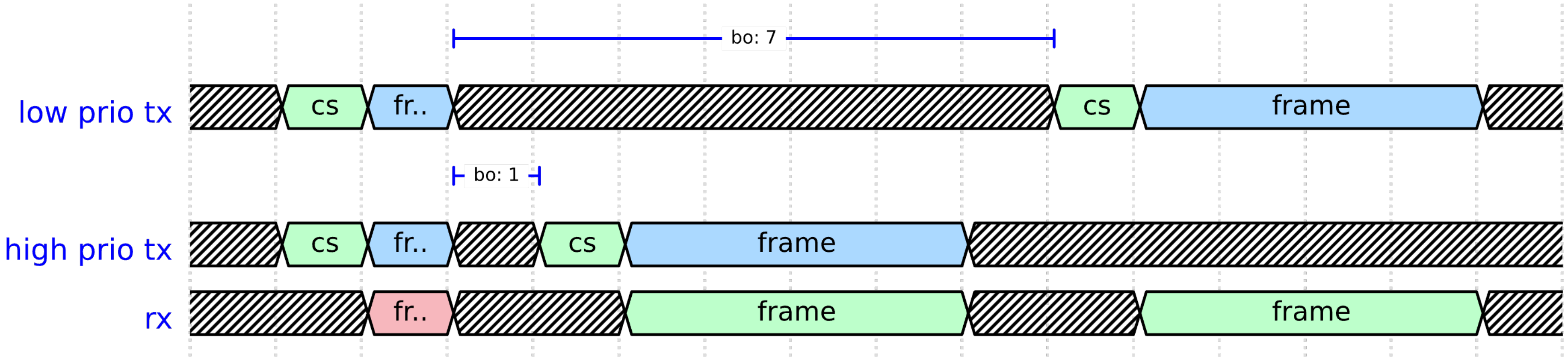
Blue: transmit and listen
Green: receive
Red: receive with errors



# Probabilistic Prioritisation

- Device prioritisation via the Binary Exponential Backoff
- Low priority devices start with a large contention window (e.g. CW=8)
- High priority devices start with a small contention window (e.g. CW=1)
- Low priority devices can transmit before high priority devices if lucky!

Blue: transmit and listen
Green: receive
Red: receive with errors

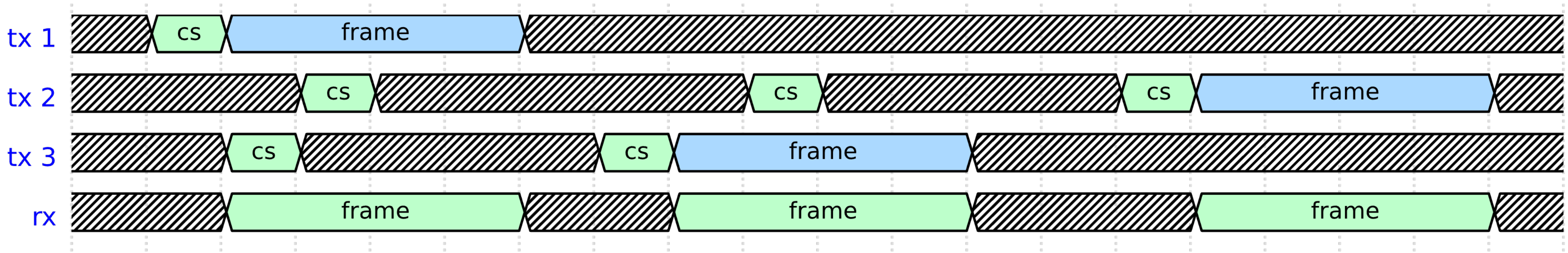


# CSMA/CA (Collision Avoidance)

- Collision Avoidance
  - When not physically possible to sense medium for other transmissions while transmitting
    - Collisions are more costly as they cannot be stopped early
  - If channel busy, wait some time randomly before attempting first transmission
  - Binary exponential backoff

- Collisions are less likely to occur in the first place
  - But costs bandwidth

Blue: transmit
Green: receive

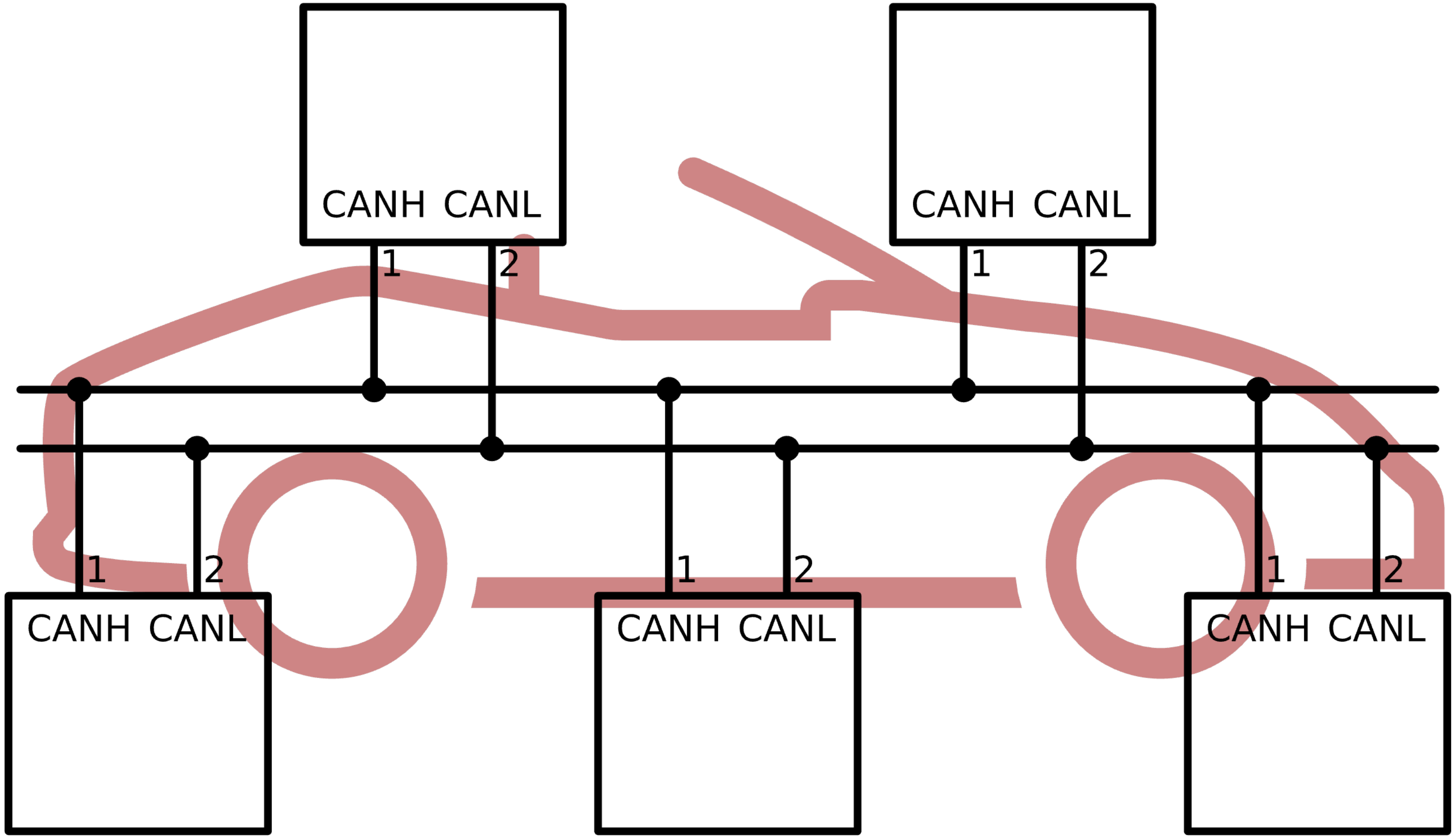


# CSMA Trade-offs

- Pros
  - Low latency in low traffic conditions (no waiting for your turn)
  - Flexible to add/remove devices from the network
  - Probabilistic prioritisation possible (smaller CW for high priority devices)

- Cons
  - Unbounded latency for individual frames (can in theory wait forever for an idle medium)
  - Poor efficiency under heavy traffic conditions (collisions very likely)

# CAN Bus


CANH CANL
1
2
CANH CANL
1
2
1
2
CANH CANL
1
2
CANH CANL
1
2
CANH CANL

# CAN Bus

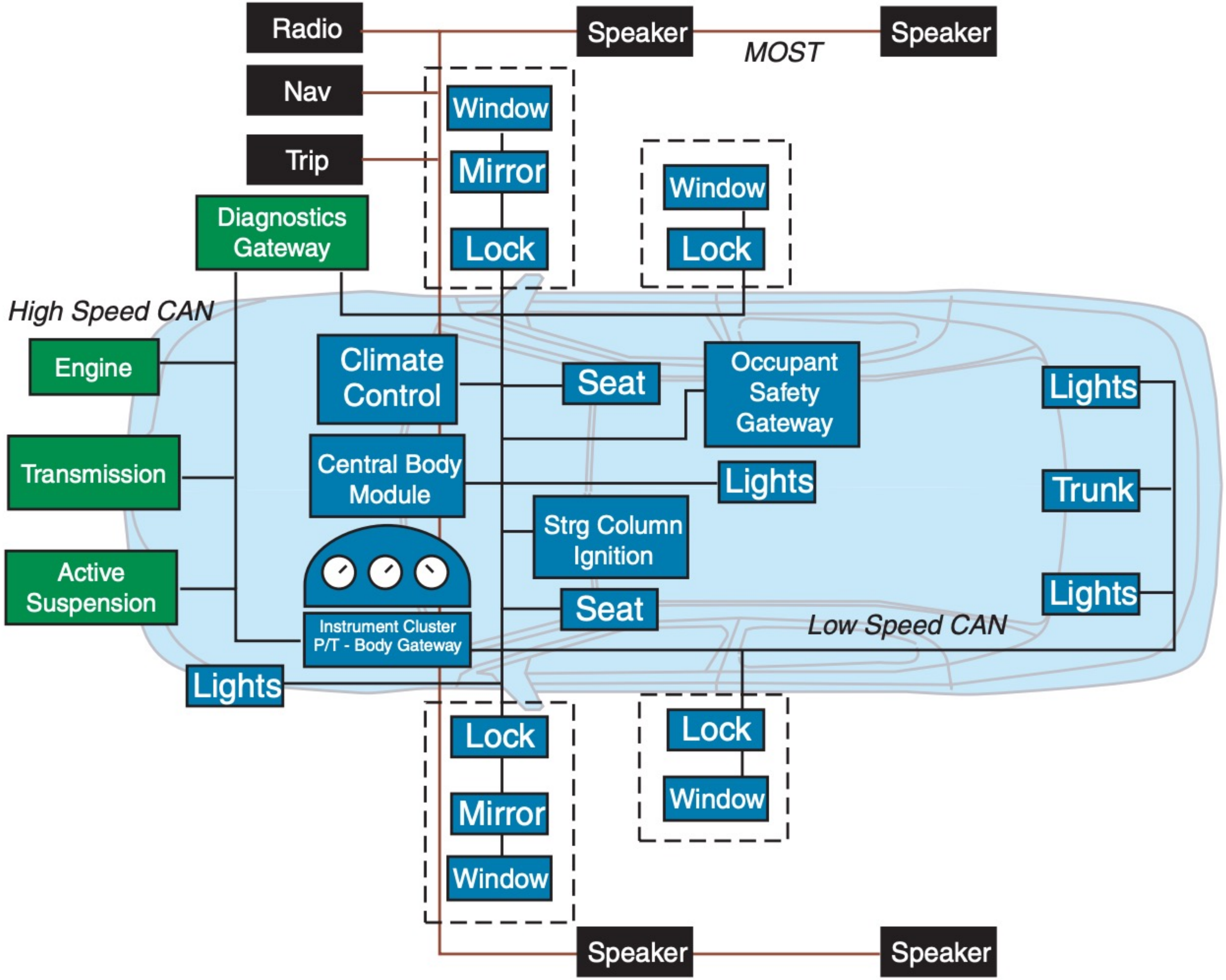


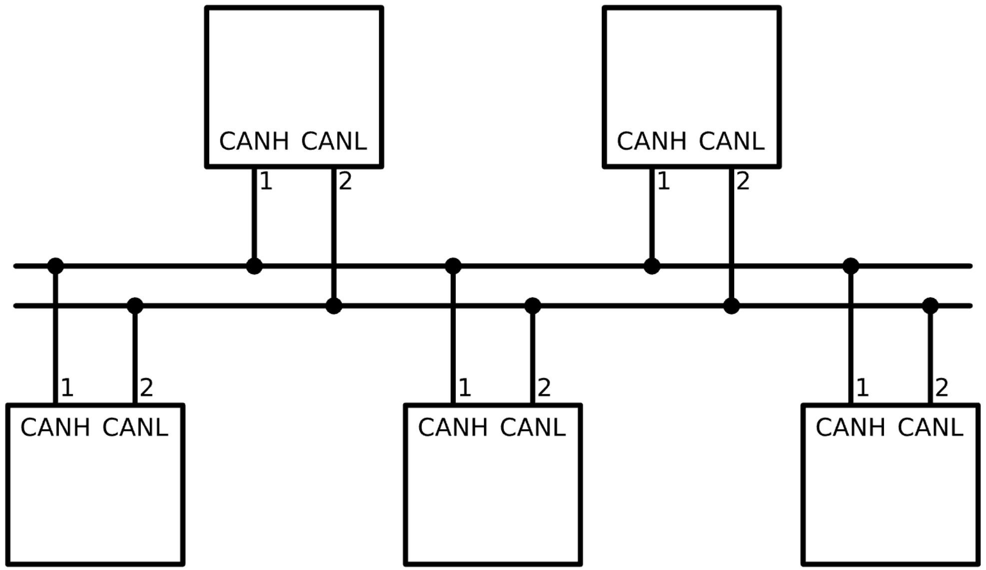

Image source: https://www.eecs.umich.edu/courses/eecs461/doc/CAN_notes.pdf

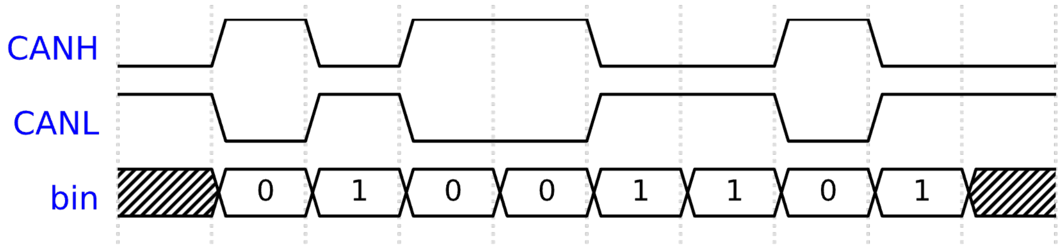
# CAN Bus Signal

- Uses differential signalling with two balanced lines (CANH and CANL)
  - ‘1’ is transmitted passively (recessive state)
  - ‘0’ is actively driven (dominant state)

- Low-Speed CAN (up to 125 Kbps)
  - To transmit ‘1’: Resistors pull CANH to GND and CANL to 5V
  - To transmit ‘0’: CANH is actively driven to 5V, CANL is actively driven to GND

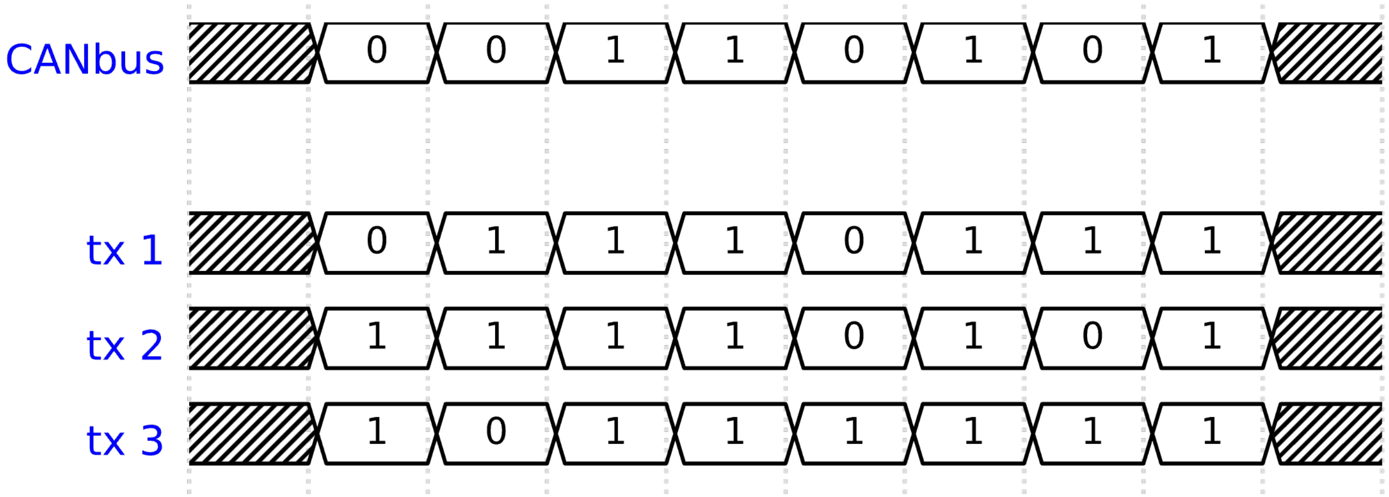
- High-Speed CAN (up to 1 Mbps)
  - To transmit ‘1’: Pullup resistors pull CANH and CANL to 2.5V
  - To transmit ‘0’: CANH is actively driven to 3.5V, CANL is actively driven to 1.5V



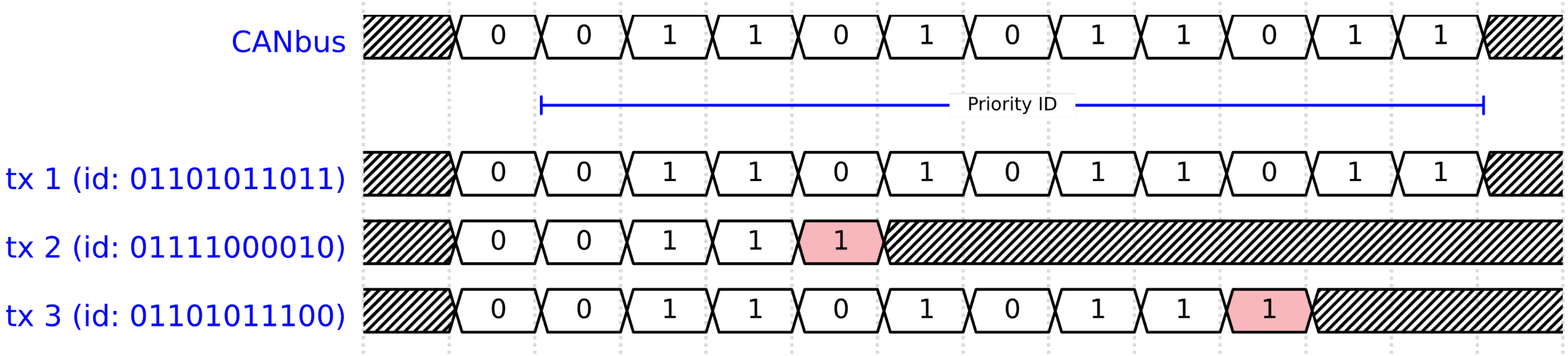
# Dominant state wins!

- If **one** device transmits ‘0’ and others transmit ‘1’ at the same time, the CAN bus state is ‘0’
  - Because ‘0’ is actively driven (dominant), while ‘1’ is passive via pullup resistors (recessive)
  - Only if **all** nodes transmit ‘1’, the bus is ‘1’
  - ‘0’ always wins



# CSMA/CD with Arbitration

- Each node gets a **unique** 11-bit ID
- The frame starts with a SOF (Start Of Frame) dominant bit '0' for synchronisation
- The first field after the SOF is an 11-bit node ID
  - The highest the priority of the node, the lower the ID
- Transmitters read the bus and if an inconsistency is observed (i.e. they transmit '1' but the bus is '0') the stop the transmission and try later
  - Collision is non-destructive, so the highest priority message goes through without errors

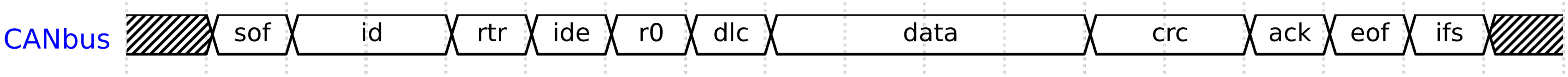


# CAN Bus Frame Types

- Communication is sender-initiated and broadcast-based
  - All nodes in the bus receive all frames and higher-level protocols determine if they should be ignored or not
- Bit Stuffing: to maintain synchronisation, a bit of opposite polarity is added after 5 consecutive bits of the same polarity
- Two frame types:
  - Standard: 11-bit IDs (2048 unique identifiers)
  - Extended: 29-bit IDs (537 million unique identifiers)
- Four frame types:
  - Data frame: a frame with data by an ID
  - Remote frame: a frame requesting data from an ID
  - Error frame: a frame transmitted by any node that detects an error
  - Overload frame: a frame to inject a delay

# Standard Data Frame

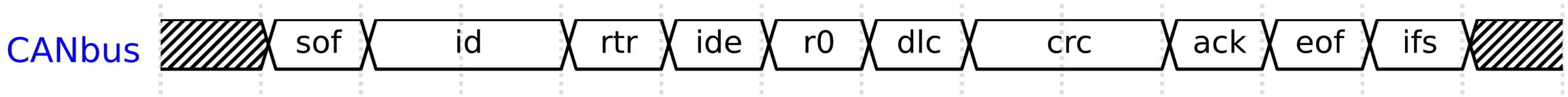
- SOF (start of frame, 1 bit): denotes the beginning of a frame, must be dominant ‘0’
- ID (identifier, 11 bits): unique ID of sender, also representing message priority
- RTR (remote transmission request, 1 bit): must be dominant ‘0’ for data frames
- IDE (identifier extension bit, 1 bit): must be dominant ‘0’ for standard frames
- R0 (reserved bit,  1 bit): reserved for future extensions of CAN
- DLC (data length code, 4 bits): number of bytes the follow (0 to 8 bytes)
- DATA (data, 0-64 bits): data to be transmitted, length defined by DLC
- CRC (cyclic redundancy code, 16 bits): 15-bit error detection code, last bit recessive ‘1’ as delimiter
- ACK (acknowledgement, 2 bits): transmitter transmits recessive ‘1’, the receiver who received it without errors overwrites with a dominant ‘0’, last bit recessive ‘1’ as delimiter
- EOF (end of frame, 7 bits): denotes the end of frame, must all be recessive ‘1’
- IFS (inter-frame spacing, 3 bits): space between messages, must all be recessive ‘1’



DTU

# Standard Remote Frame

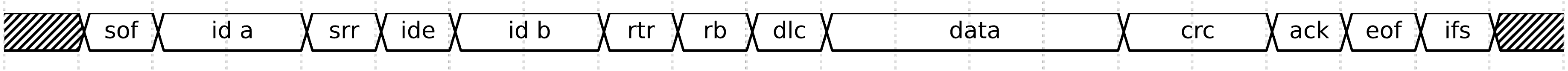
- SOF (start of frame, 1 bit): denotes the beginning of a frame, must be dominant ‘0’
- ID (identifier, 11 bits): unique ID of receiver, also representing message priority
- RTR (remote transmission request, 1 bit): must be dominant ‘1’ for remote frames
- IDE (identifier extension bit, 1 bit): must be dominant ‘0’ for standard frames
- R0 (reserved bit,  1 bit): reserved for future extensions of CAN
- DLC (data length code, 4 bits): number of expected bytes for requested message (0 to 8 bytes)
- CRC (cyclic redundancy code, 16 bits): 15-bit error detection code, last bit recessive ‘1’ as delimiter
- ACK (acknowledgement, 2 bits): transmitter transmits recessive ‘1’, the receiver who received it without errors overwrites with a dominant ‘0’, last bit recessive ‘1’ as delimiter
- EOF (end of frame, 7 bits): denotes the end of frame, must all be recessive ‘1’
- IFS (inter-frame spacing, 3 bits): space between messages, must all be recessive ‘1’



# Extended Data Frame

- SOF (start of frame, 1 bit): denotes the beginning of a frame, must be dominant ‘0’
- ID A (identifier, 11 bits): most significant part of unique ID of sender
- SRR (substitute remote request, 1 bit): must be recessive ‘1’
- IDE (identifier extension bit, 1 bit): must be recessive ‘1’ for extended frames
- ID B (identifier, 18 bits): least significant part of unique ID of sender
- RTR (remote transmission request, 1 bit): must be dominant ‘0’ for data frames
- RB (reserved bits,  2 bit): reserved for future extensions of CAN
- DLC (data length code, 4 bits): number of bytes the follow (0 to 8 bytes)
- DATA (data, 0-64 bits): data to be transmitted, length defined by DLC
- CRC (cyclic redundancy code, 16 bits): 15-bit error detection code, last bit recessive ‘1’ as delimiter
- ACK (acknowledgement, 2 bits): transmitter transmits recessive ‘1’, the receiver who received it without errors overwrites with a dominant ‘0’, last bit recessive ‘1’ as delimiter
- EOF (end of frame, 7 bits): denotes the end of frame, must all be recessive ‘1’
- IFS (inter-frame spacing, 3 bits): space between messages, must all be recessive ‘1’

CANbus



# CAN Bus: Advantages and Disadvantages

- Advantages
  - Low latency in light traffic
  - Strong prioritisation for vital traffic
  - Robustness (abundant error checking procedures)
    - CRC and ACK, SOF, EOF, delimiters, IFS at message level
    - Bits transmitted are also read for inconsistencies
    - Bit stuffing rule of no more than 5 consecutive bits of the same logic level

- Disadvantages
  - Unfair to low-priority nodes, starvation in high traffic
  - Poor latency for low-priority nodes

# IEEE 802.3: Ethernet

- A standard for wired local area networks

- A family of standards that evolves since 1983
  - Originally by Xerox in 1976 at 2.94 Mbps
  - 10BASE5: first standard (10 Mbps)
  - Now up to 400 Gbps (data centres)

- Up to 1500 Bytes of data per frame

# Ethernet: MAC Addresses

- A unique 48-bit identifier (6 bytes) for each Ethernet interface
  - Typically written in hex, for example AA:BB:CC:DD:EE:FF
- It is composed of two 24-bit parts
  - Organizationally Unique Identifier (OUI): identifies the manufacturer (assigned by IEEE)
  - Hardware identifier: identifies the specific hardware (assigned by the owner of the OUI)
- A MAC address of FF:FF:FF:FF:FF:FF is a broadcast address

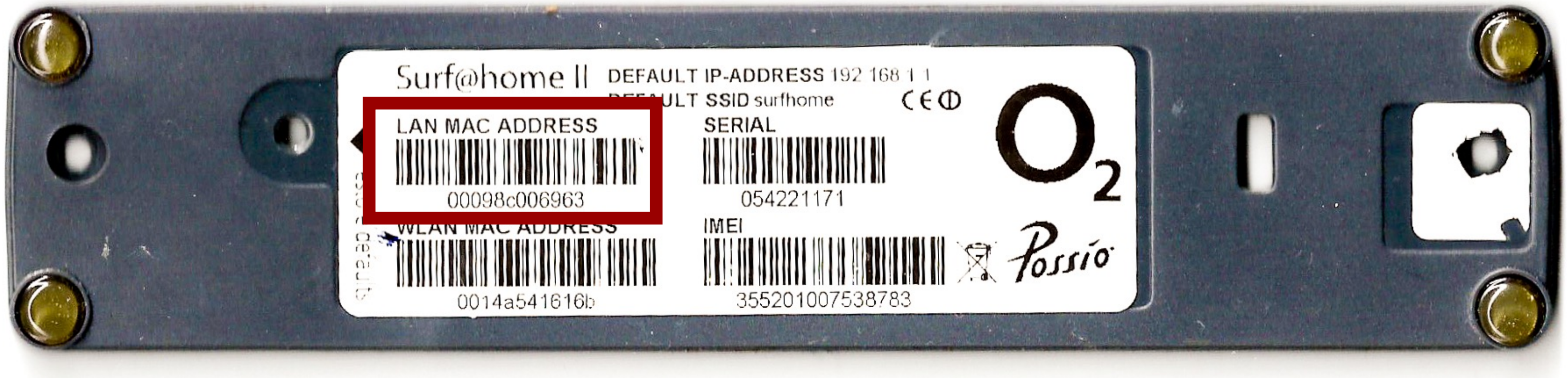
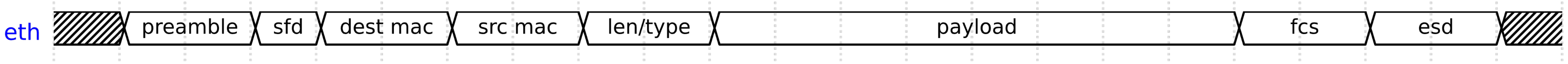



Image source: Wikipedia

# Ethernet Frame

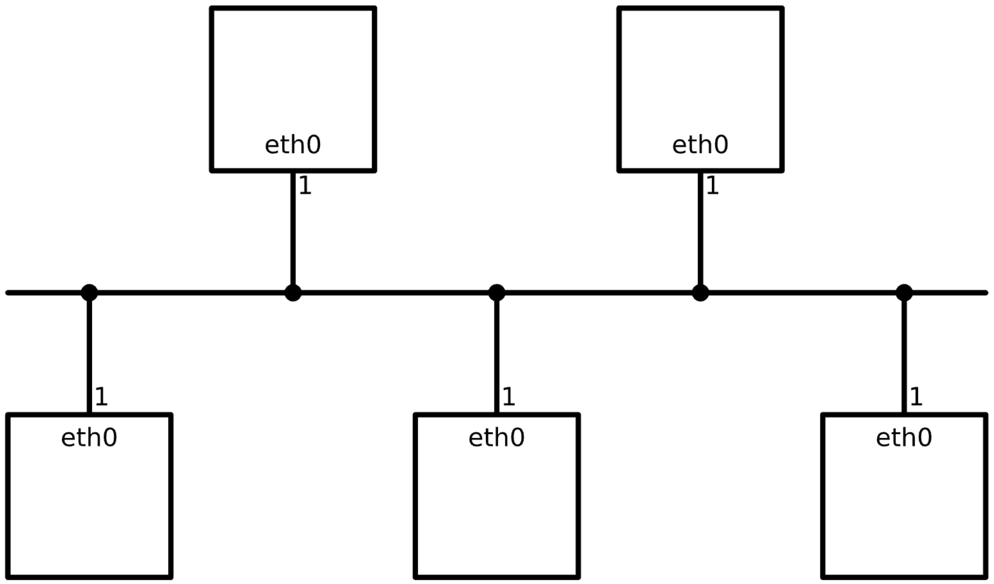
- Preamble (7 bytes): Seven bytes of 0b10101010 for synchronisation
- SOF (start of frame delimiter, 1 byte): 0b10101011, marks the beginning of the frame
- Dest MAC (6 bytes): MAC address of the destination
- Src MAC (6 bytes): MAC address of the source
- Length/Type (2 bytes): Values <1500 indicate the payload length, values >=1536 represent the type of packet inside the payload
- Payload (46-1500 bytes): Data, padding is added to reach the minimum 46 bytes
- FCS (frame check sequence, 4 bytes): Error detection code using CRC-32
- ESD (end of stream delimiter, 4 bytes): Marks the end of the frame

- Note: The first information in the packet is the destination address so receivers can quickly determine if the frame is for them (to process it) or not (to ignore it)



# The Original Ethernet

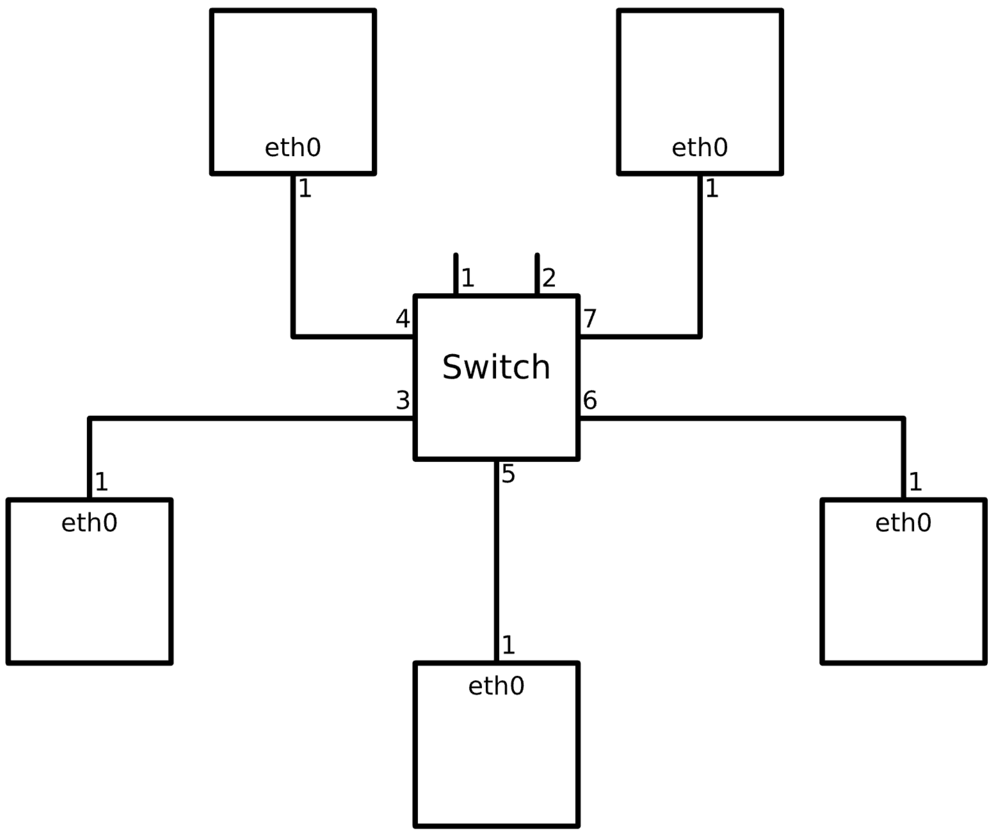
- 10BASE2 and 10BASE5 (10Mbps)
  - Original implementations over a single coaxial cable (half-duplex)
  - Manchester encoding
    - ‘0’ is a high-to-low transition
    - ’1’ is a low-to-high transition
  - Bus topology and repeater hubs
  - CSMA/CD



# Ethernet Switch

- It learns where each device is connected
  - Maintains a table that maps MAC addresses to ports
  - Entries at the table have an expiration date
  - If destination is in the table,
    frame is forwarded to the correct port
  - Frame is repeated to all ports otherwise

- Unlike repeater hub that emulates a bus

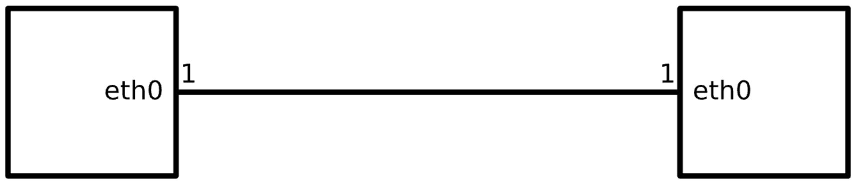


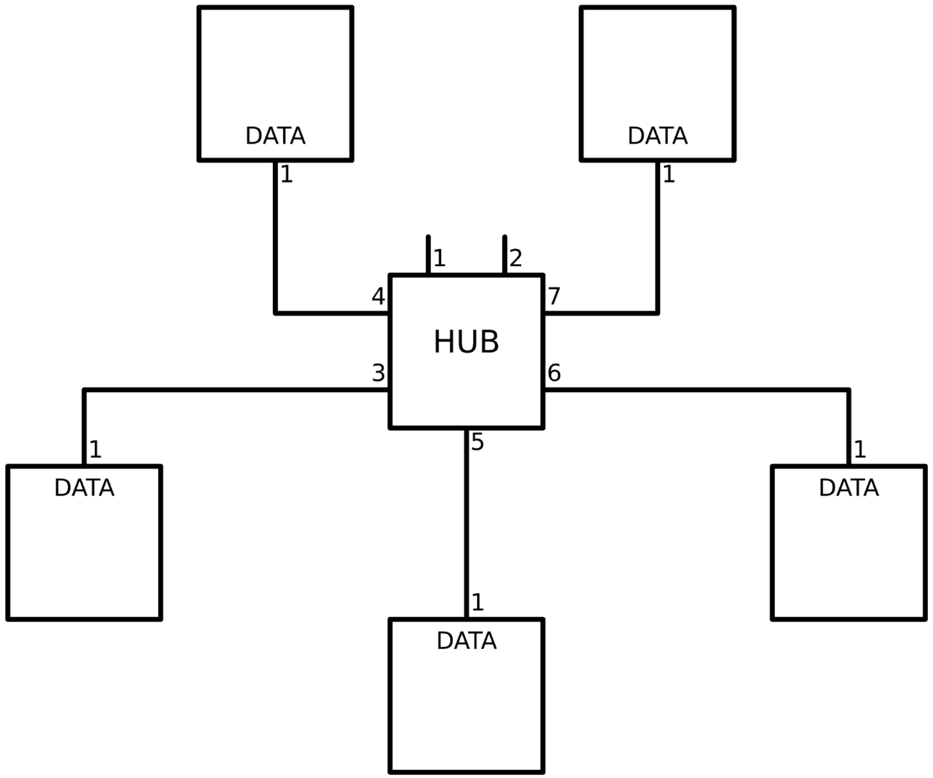
# The Common Ethernet

- 10BASE-T (10 Mbps) and 100BASE-TX (100 Mbps)
  - Peer-to-peer and switched topologies
  - Full-duplex: differential signalling over 2 twisted pair lines (TX+, TX-, RX+, RX-)
  - No collisions, throughput is doubled

- 10BASE-T keeps Manchester encoding

- 100BASE-TX uses:
  - Multi-Level Transition 3 (MLT3)
  - 4B/5B Encoding



# Multi-Level Transition 3 (MLT3)

- Three voltage levels are defined: -1, 0, +1
- A transition order is defined: -1, 0, +1, 0, -1, …
- Line Coding
  - A ‘0’ is denoted as no transition
  - A ‘1’ is denoted as a transition

| Data | 1 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |
|---|---|---|---|---|---|---|---|---|
| Code | -1 | 0 | 1 | 1 | 0 | -1 | -1 | 0 |

- Advantage
  - Small transitions allow higher speeds
- Disadvantage
  - Many continuous zeros lead to no transitions and, eventually, loss of synchronisation

# 4B/5B Encoding

- Encodes 4 bits of data into 5 bits
  - 5 bits are then transmitted using MLT-3
- 5 bits allow 32 values
  - 16 values of data (4 bits)
  - Signal code
- Ensures frequent transitions by avoiding 5 bit values that have less than two ‘1’s
- Needs 125 Mbps transmission rate to achieve the 100 Mbps

**TABLE 3: 4B/5B ENCODING**

| Code | Value | Definition |
|---|---|---|
| 0 | 11110 | Data 0 |
| 1 | 01001 | Data 1 |
| 2 | 10100 | Data 2 |
| 3 | 10101 | Data 3 |
| 4 | 01010 | Data 4 |
| 5 | 01011 | Data 5 |
| 6 | 01110 | Data 6 |
| 7 | 01111 | Data 7 |
| 8 | 10010 | Data 8 |
| 9 | 10011 | Data 9 |
| A | 10110 | Data A |
| B | 10111 | Data B |
| C | 11010 | Data C |
| D | 11011 | Data D |
| E | 11100 | Data E |
| F | 11101 | Data F |
| I | 11111 | Idle |
| J | 11000 | SSD (Part 1) |
| K | 10001 | SSD (Part 2) |
| T | 01101 | ESD (Part 1) |
| R | 00111 | ESD (Part 2) |
| H | 00100 | Transmit Error |

Image source: AN1120 by Microchip

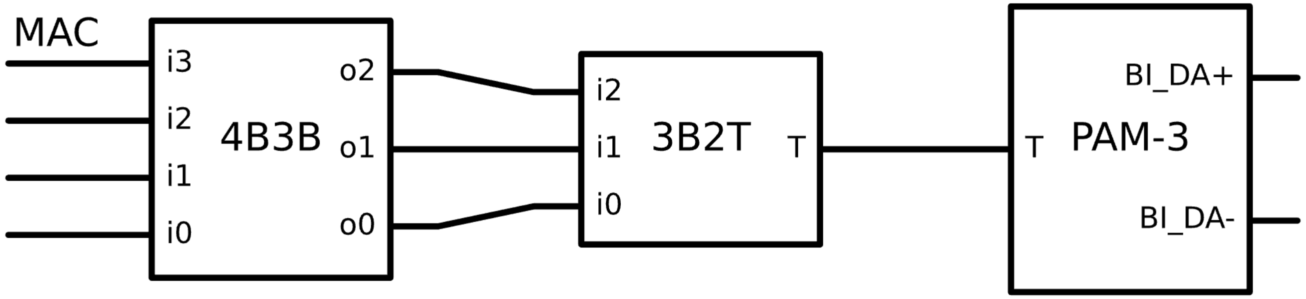
# The Automotive Ethernet

- Requirements
  - Infotainment and new sensors require higher data rates in vehicles (than CAN)
  - Need for reduction of cable weight, cost, energy footprint, EMI
  - No need to support large cable lengths, 15m is enough
- 100BASE-T1
  - Uses only 1 bi-directional full-duplex twisted pair cable (echo cancellation)
  - 4B3B to 3B2T with PAM3 encoding
- Transparent to MAC layer

# 4B3B 3B2T PAM-3 Modulation

- 4B3B
  - Receives 4-bit blocks in parallel at 25 MHz (100 Mbps total)
  - Outputs 3-bit blocks in parallel at 33.33 MHz (100 Mbps total)
- 3B2T
  - Receives 3-bit blocks in parallel at 33.33 MHz (100 Mbps total)
  - Outputs 2 ternary bits, TA and TB, (i.e. 3-state bits) in sequence at 66.66 MHz (100 Mbps total)
- PAM-3
  - Modulates a ternary bit in 3 voltage levels: -1 V, 0 V, +1 V

| 3-bit data | TA | TB |
|---|---|---|
| 000 | -1 | -1 |
| 001 | -1 | 0 |
| 010 | -1 | 1 |
| 011 | 0 | -1 |
| SSD/ESD | 0 | 0 |
| 100 | 0 | 1 |
| 101 | 1 | -1 |
| 110 | 1 | 0 |
| 111 | 1 | 1 |



# Echo Cancellation

- The transmitted signal is removed from the received signal
- Enables parallel transmissions (full-duplex) over a single twisted pair cable

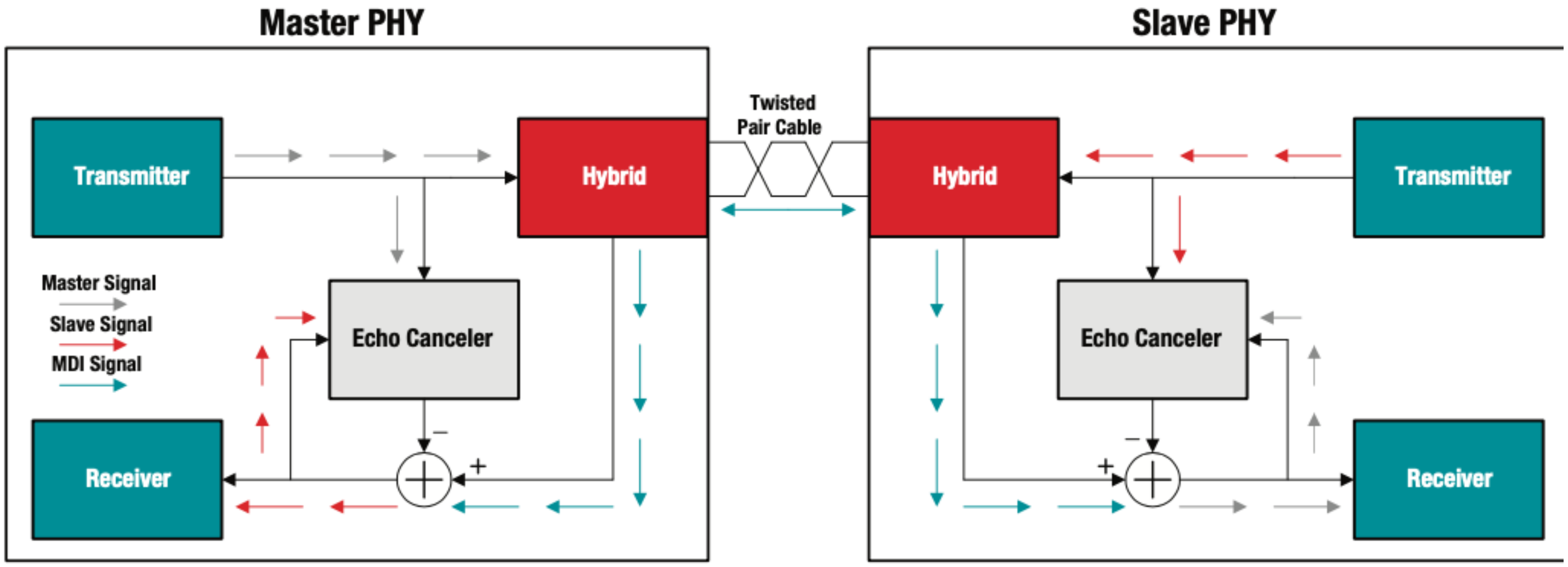



Image source: 100BASE-T1 Ethernet: the evolution of automotive networking by Texas Instruments

# Comparison

| Property | Classic Ethernet (100BASE-TX) | Automotive Ethernet (100BASE-T1) |
|---|---|---|
| Data rate | 100 Mbps | 100 Mbps |
| Baud rate / clock | 125 MHz | 66.66 MHz |
| No. Twisted Pairs | 2 | 1 |
| Line Code | 4B5B MLT-3 | 4B3B PAM-3 |
| Max Distance | 100 m | 15 m |

# Gigabit Ethernet

- 1000BASE-T (1000 Gbps)
  - Differential signalling over 4 bi-directional full-duplex twisted pair lines (echo cancelation)
    - Each operates at 125 MHz
  - 4D-PAM-5 Encoding
    - Encodes 2-bits:-2, -1, 1, 2
    - ‘0’ for error correction
  - 8 bits are transmitted in parallel (4 lines, 2 bits per symbol, symbol rate 125 MHz)
- Gigabit Automotive Ethernet (1000Base-T1)