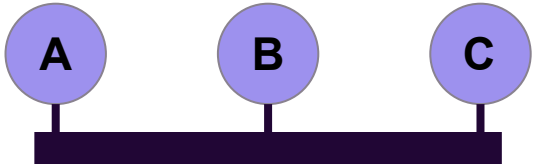
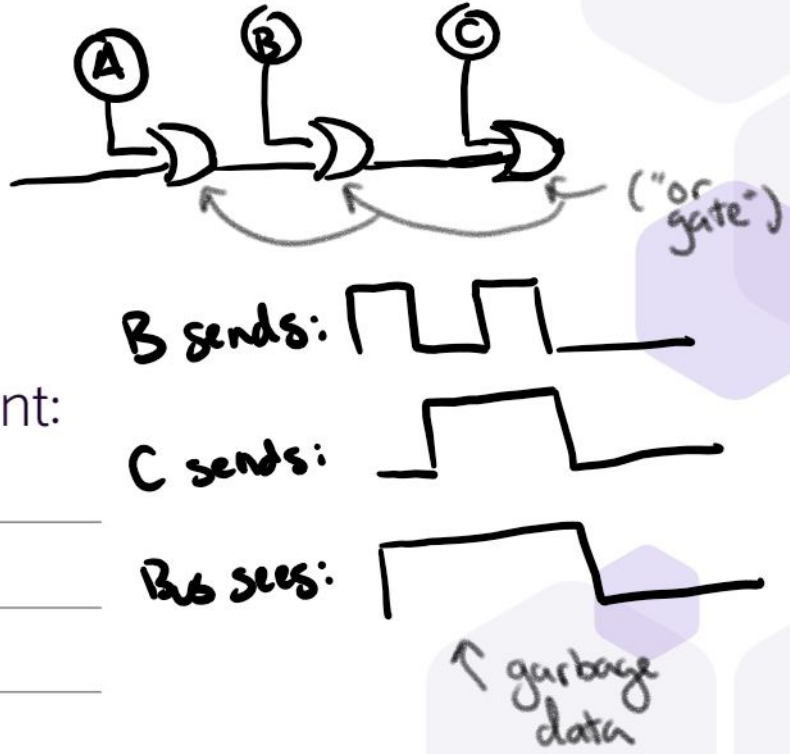
21: Communication reliability and protocols

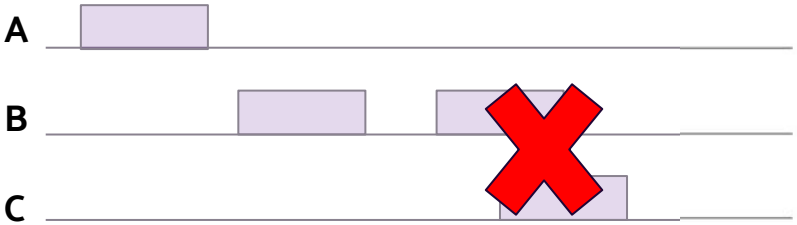
Control and data flow - Collisions

Consider a bus topology





Consider messages being sent:



How would you avoid collisions?

Coordination on centralized system

Controller tells peripherals when it is time to transmit:

Polling: controller goes around asking peripherals if they have something to transmit

TDMA (time division multiplex access): controller sends time coordination

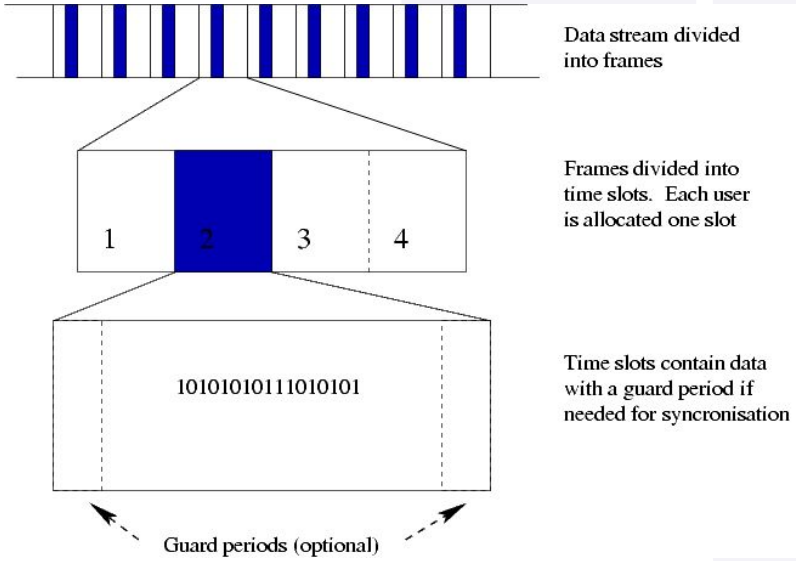


Image source

Token passing

Nodes coordinate ownership of “token” that says they can send

- Centralized system - controller passes token out
- Fully distributed system - token is passed around (e.g. round robin - “token ring”)

CSMA

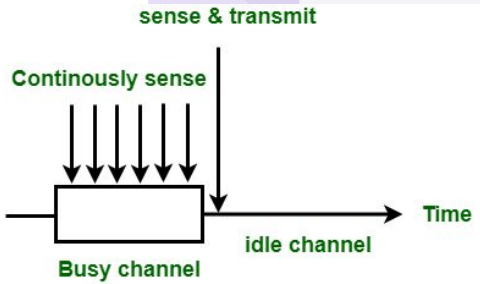
Carrier Sense Multiple Access: coordinate on fully distributed system

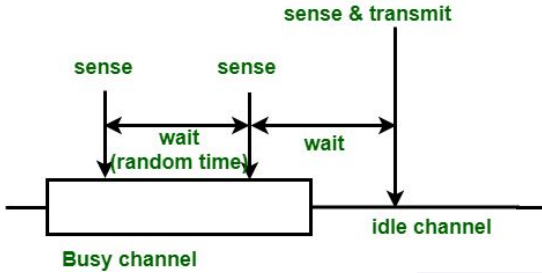
- Check if transmission line is busy before sending

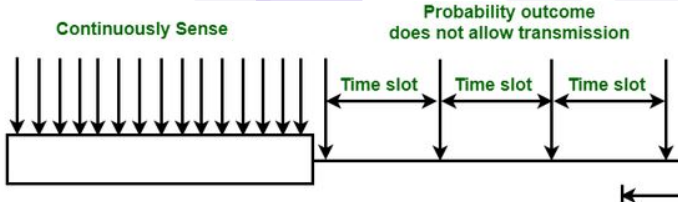
Multiple kinds:

- Check constantly (*persistent* CDMA)
- Wait before checking again (*non-persistent* CDMA)
- Transmit with probability p (*p-persistent* CDMA)

CSMA/CD (collision detection) - immediately stop transmitting when collision detected







[Image source]

Binary countdown

Each node has an arbitration ID

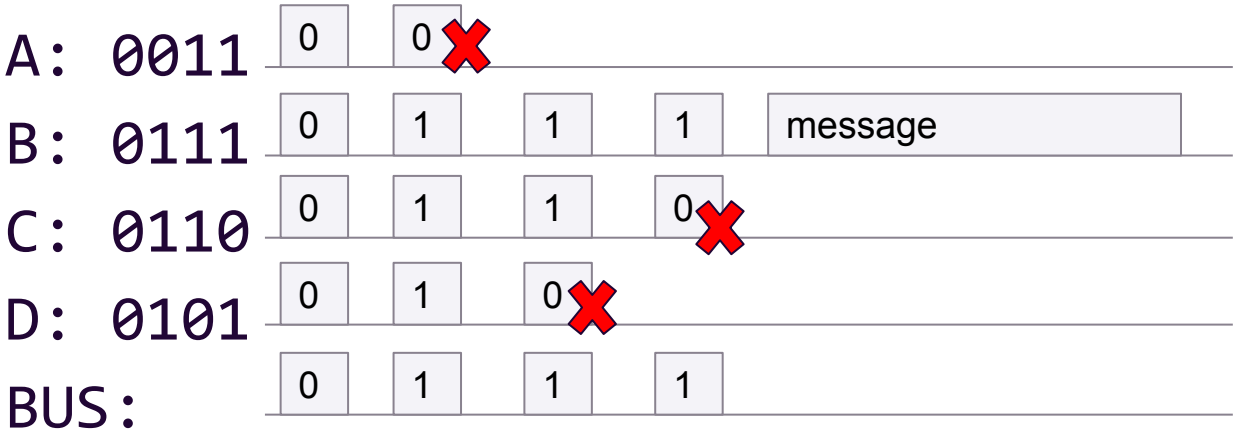
When a node wants to send, it broadcasts the first bit of the ID

If other nodes want to send at the same time, they also send their first bit

1-dominant: bus is an OR of all bits

nodes that send 0 back off

Binary countdown example



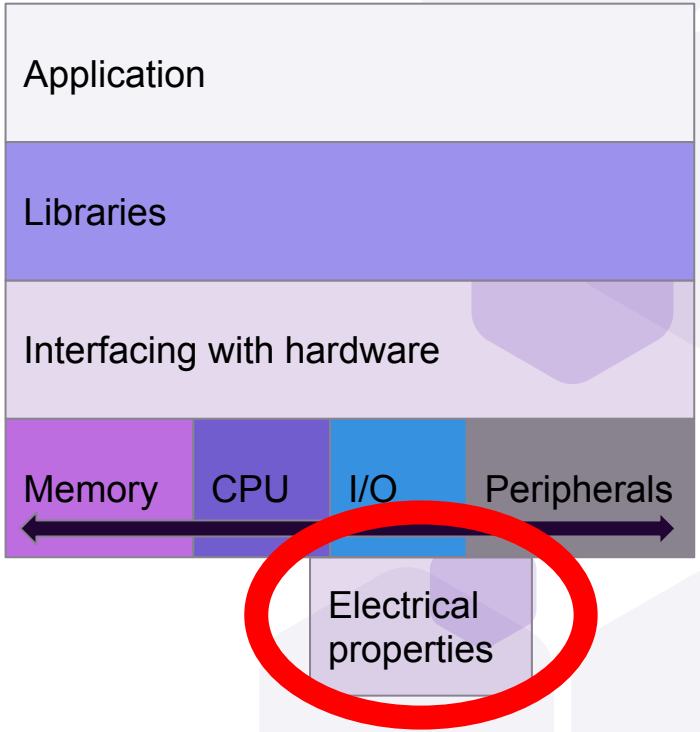
Reliability - because signals aren't perfect

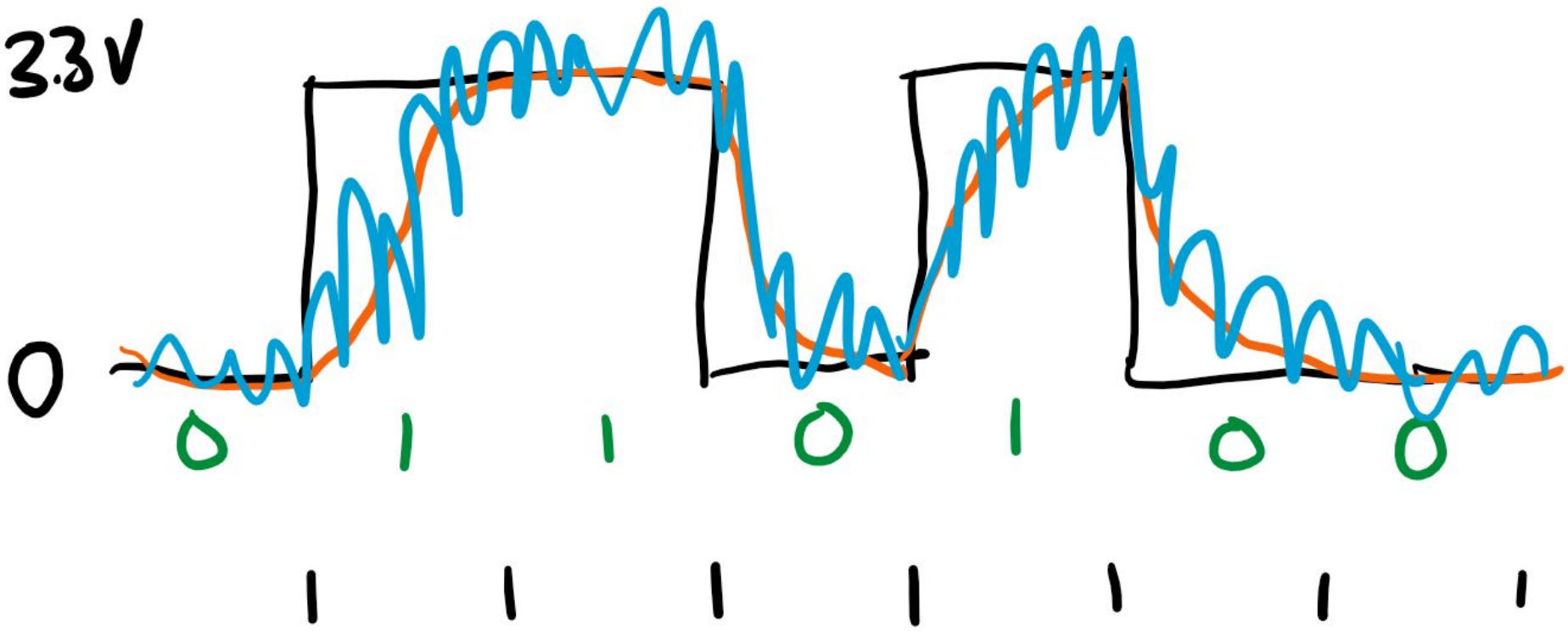
Bits are just high/low voltage signals on a wire sent w.r.t a clock

Clock may not be perfect

Wire may be noisy (electrical interference)

Wireless has even more dangers



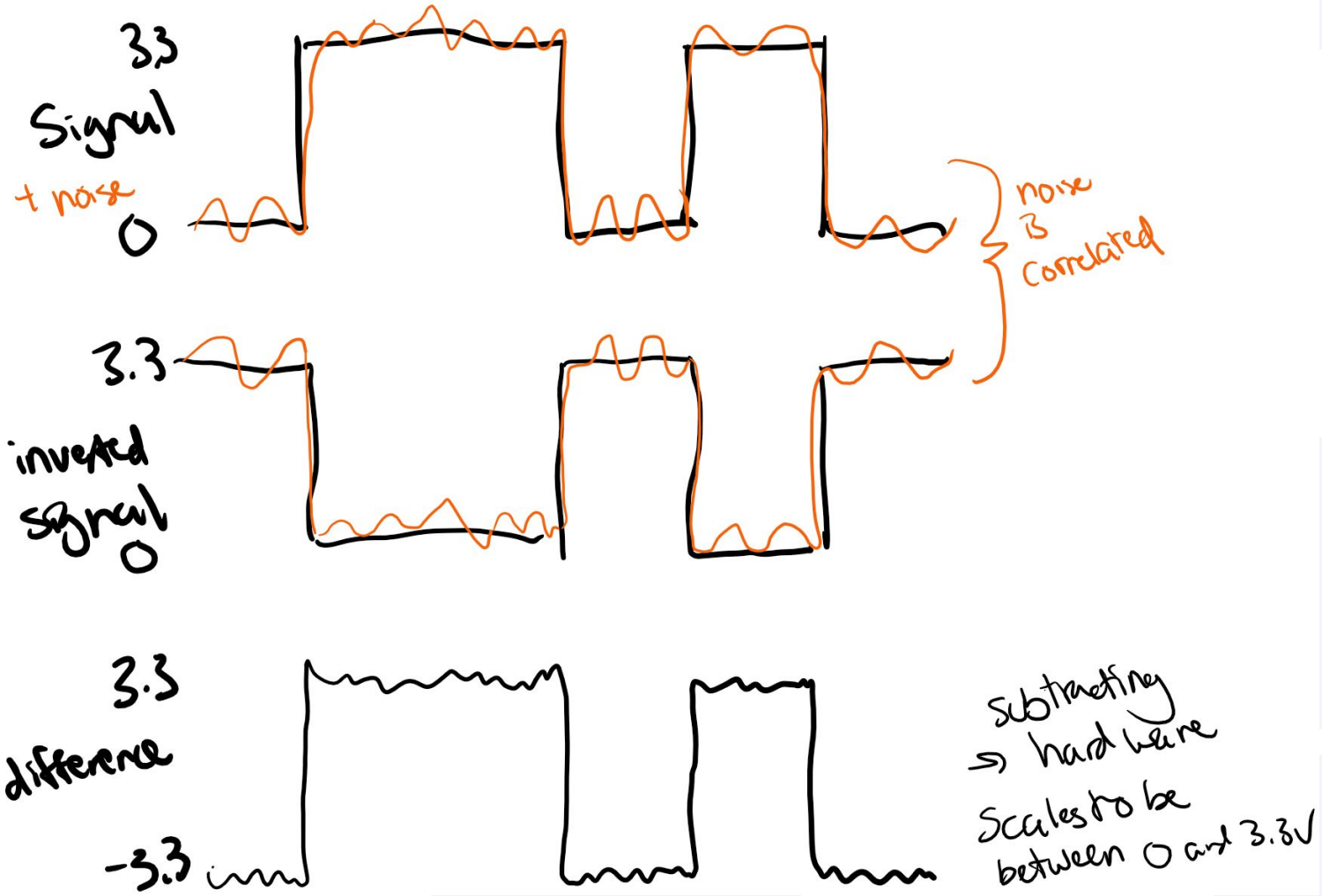

3.3V
0
0 1 1 0 1 0 0

Differential drivers for noisy signals

Assumption is that noise is somewhat correlated for two signals sent at the same time on two adjacent wires

Send signal and inverted signal

Subtract the signals to reduce noise


3.3
Signal
+ noise
0
noise is correlated
3.3
inverted signal
0
3.3
difference
-3.3
subtracting
⇒ hardware
Scales to be between 0 and 3.3V

Checksums

A computation on the data that describes something about the contents of the data

Example: parity (number of 1s odd or even)

0110 has parity 0

0111 has parity 1

Sender sends data and checksum

Receiver receives data and compares computed checksum with received checksum

Say a sender sends 7 data bits followed by a parity bit. Which of these messages will be rejected by the receiver?

A: 0000 0000

B: 1110 0000

C: 1111 0101

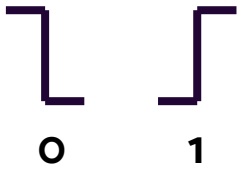
Reliability: RZ/NRZ

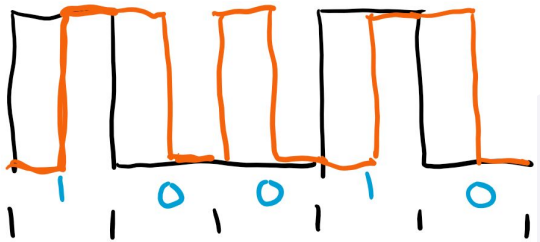
Return to zero/no return to zero

NRZ: signal can output the same value for arbitrary time

RZ: signal must have an edge every once in a while

Manchester encoding:





What are the tradeoffs between NRZ and RZ?