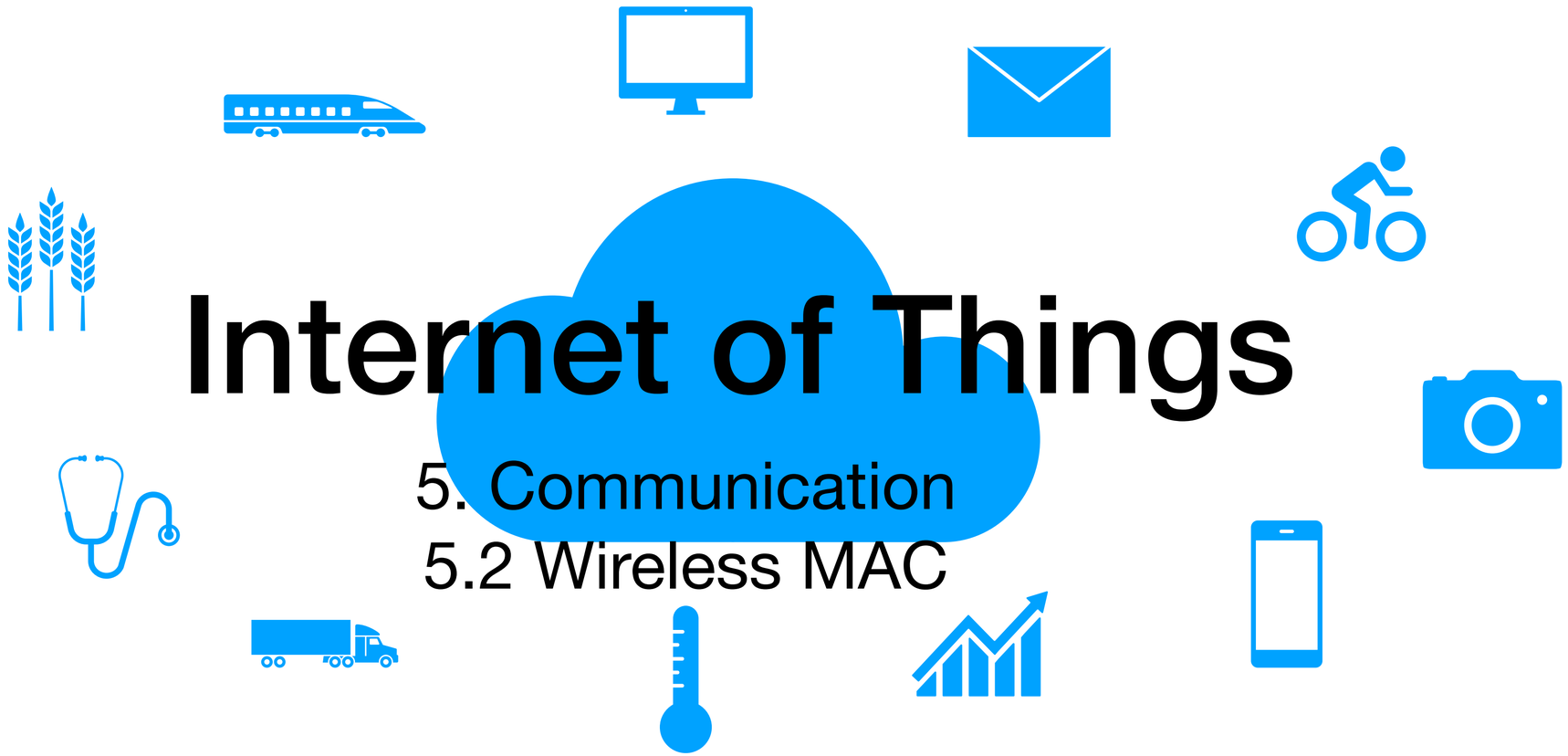


Internet of Things

5. Communication
5.2 Wireless MAC

M.Sc. Computer Science 2024-2025

Viviana Arrigoni

5.2.1 Multiple Access Techniques in Wireless Networks

- When two or more hosts transmit over the same channel at the same time, the signals experience **collisions**.
  How do we solve that?

  1. Just let transmission collide - but data gets corrupted

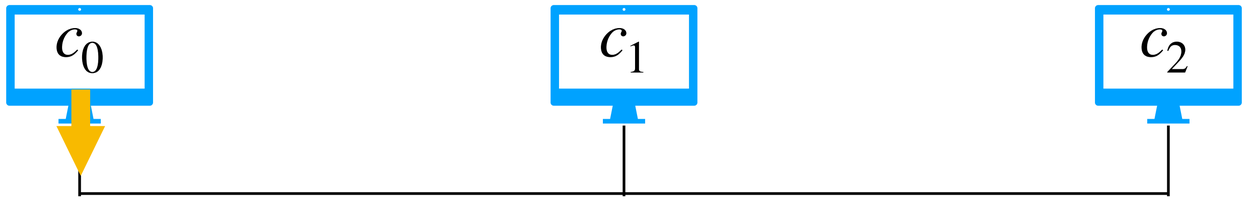
  2. Collision Detection (CD) - you let your signal collide but detect when there is a collision and the sender retransmits.

  3. Collision Avoidance (CA) - try to prevent collision (it is powerful yet hard)

Collision Detection (CD) in wired networks (1)

- Suppose we have a wired ethernet network with three computers. At time $t_0 = 0$, c_0 starts transmitting data.

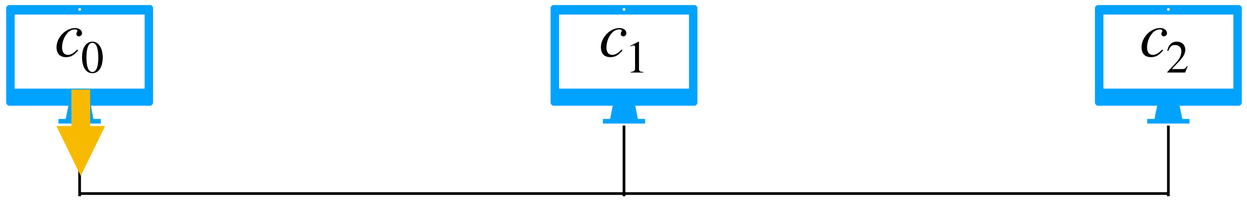
$t_0 = 0$



Collision Detection (CD) in wired networks (1)

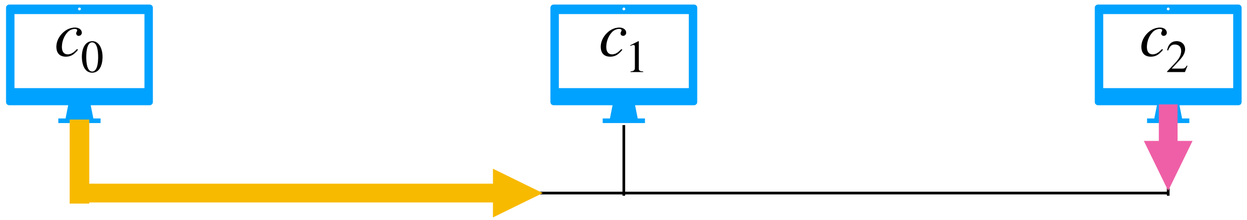
- Suppose we have a wired ethernet network with three computers. At time $t_0 = 0$, c_0 starts transmitting data.

$t_0 = 0$



- At time $t_1 = 8\mu s$, c_2 starts transmitting data.

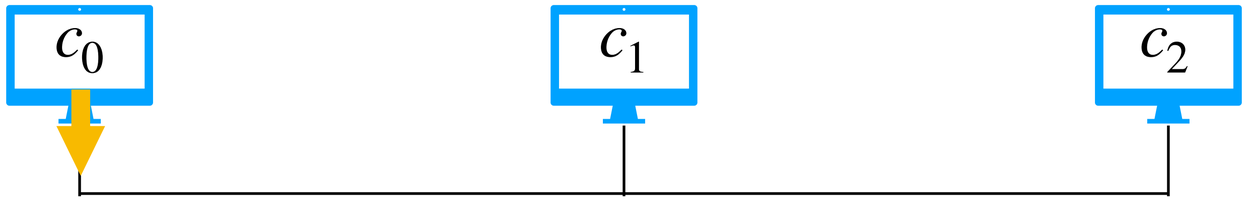
$t_1 = 8\mu s$



Collision Detection (CD) in wired networks (1)

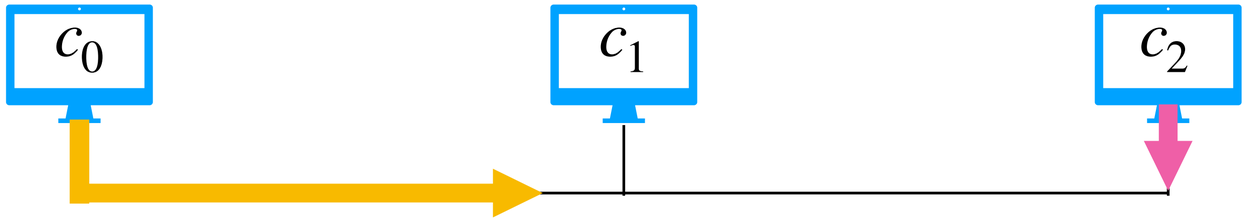
- Suppose we have a wired ethernet network with three computers. At time $t_0 = 0$, c_0 starts transmitting data.

$t_0 = 0$



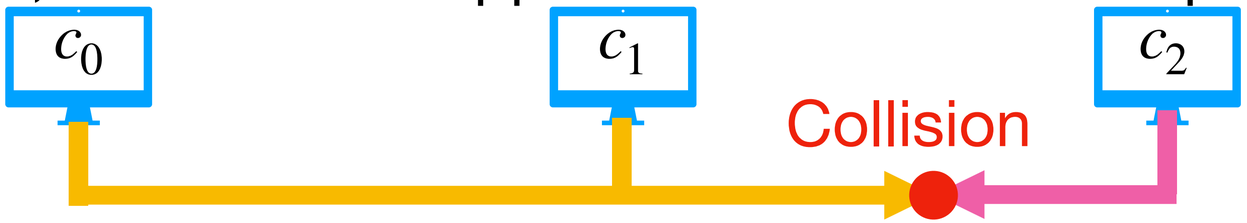
- At time $t_1 = 8\mu s$, c_2 starts transmitting data.

$t_1 = 8\mu s$



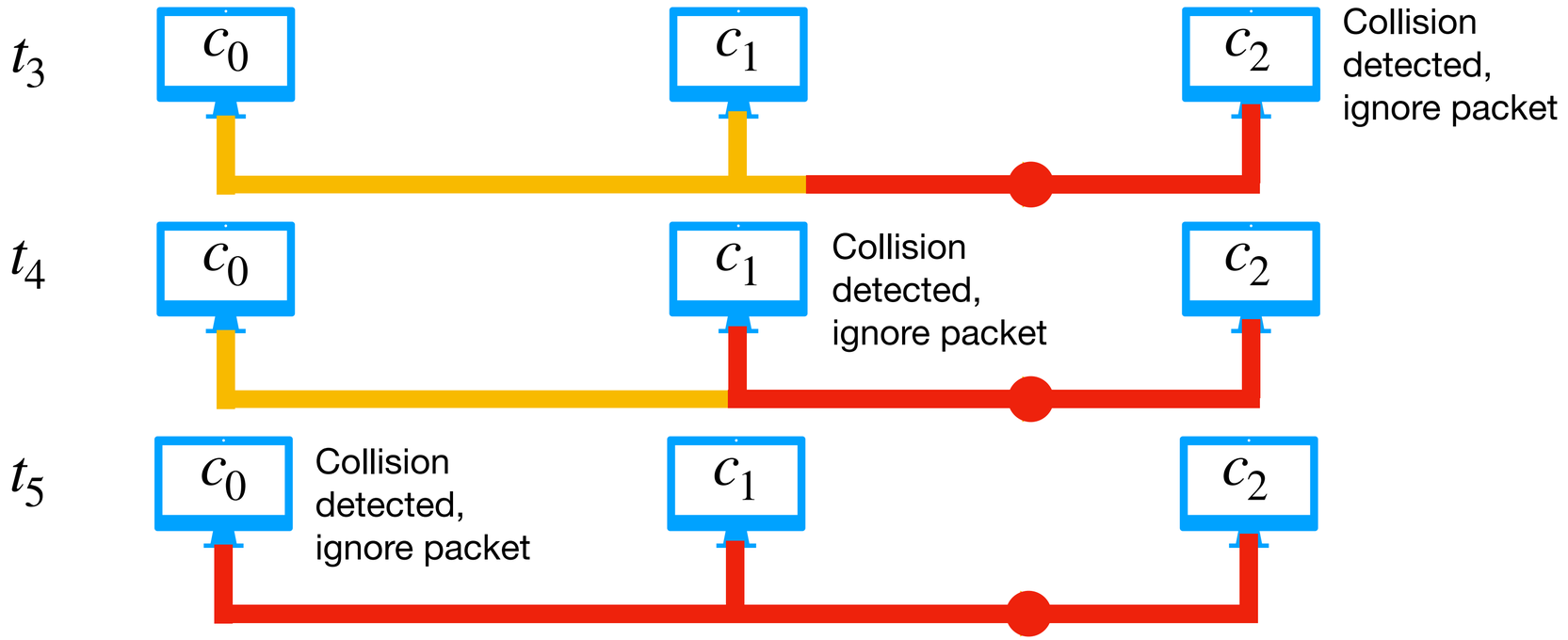
- At a certain time, collision will happen but none of the computers noticed it yet

t_2

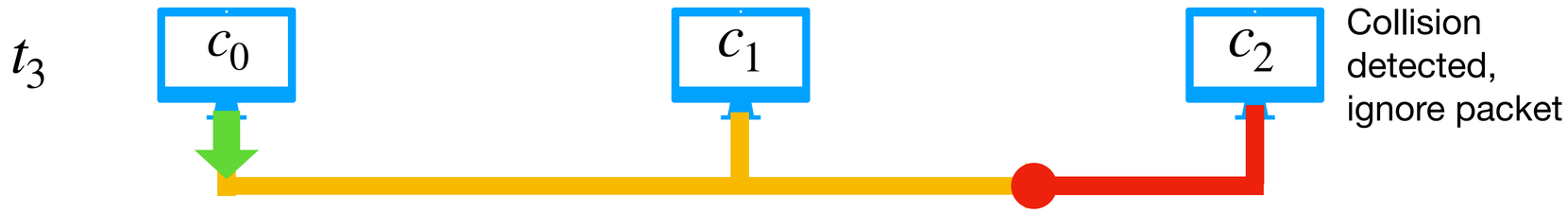


Collision Detection (CD) in wired networks (2)

- The corrupted packet propagates in the network and the computers detect the collision
  - Most network devices have hardware able to distinguish collided traffic as it usually comes with higher voltage.

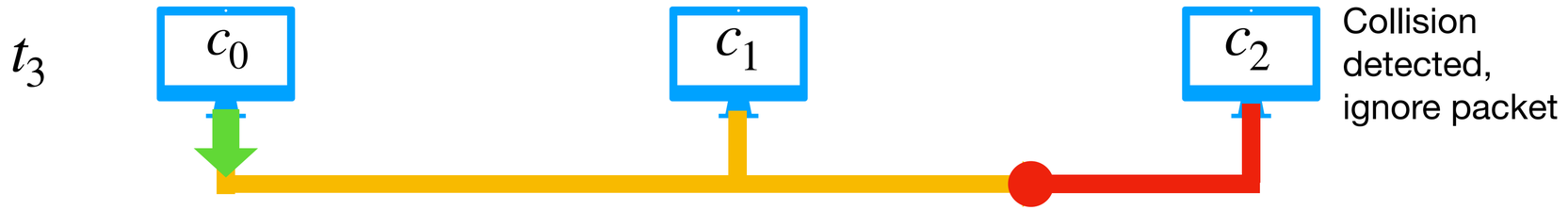


Collision Detection (CD) -small packets, wired networks (2)



Collision Detection (CD) -small packets, wired networks (2)

- If computer c_0 starts transmitting another packet after the first one but before it notices the collision, when it sees the corrupted packet coming back, it does not know whether the collision corrupted just the second packet or both.



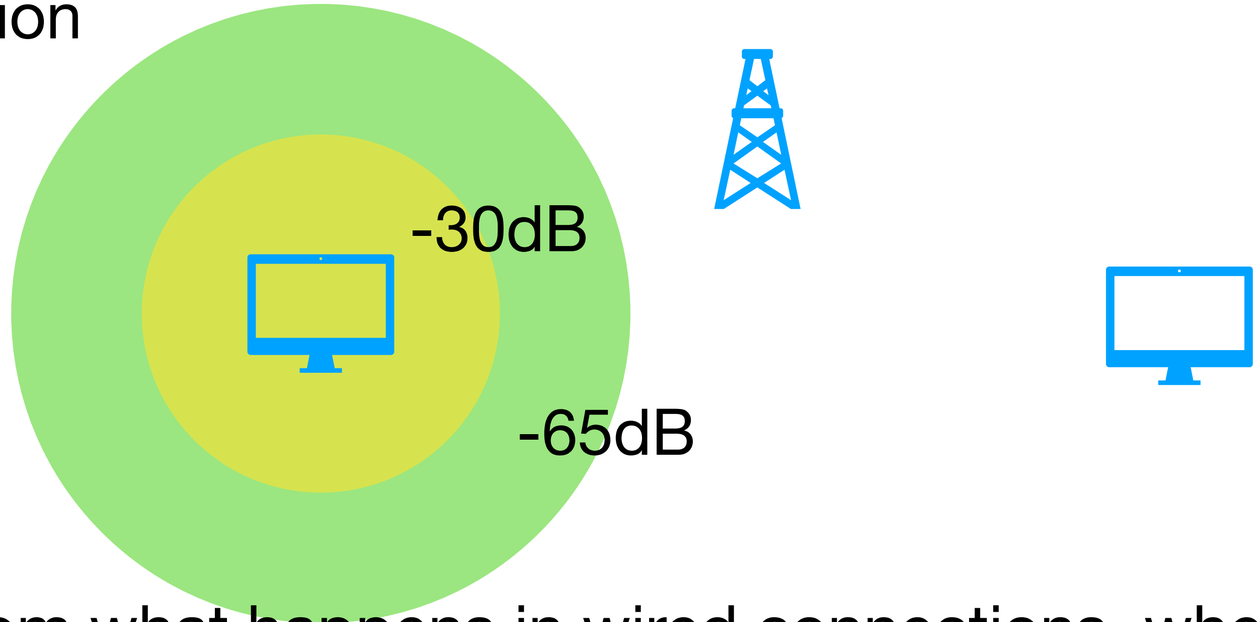
- Carrier Sense Multiple Access (CSMA) is used to manage how hosts share the same communication channel.
- CSMA/CD in wired networks sets small packet sizes to make collision detection easier.
- (CSMA/CD is actually deprecated as we now use full duplex connection with different wires for transmission and reception).

- Do we need different Media Access Control mechanisms for channel sharing in wireless networks?

- If so, why?

Collision Detection, Wireless (1)

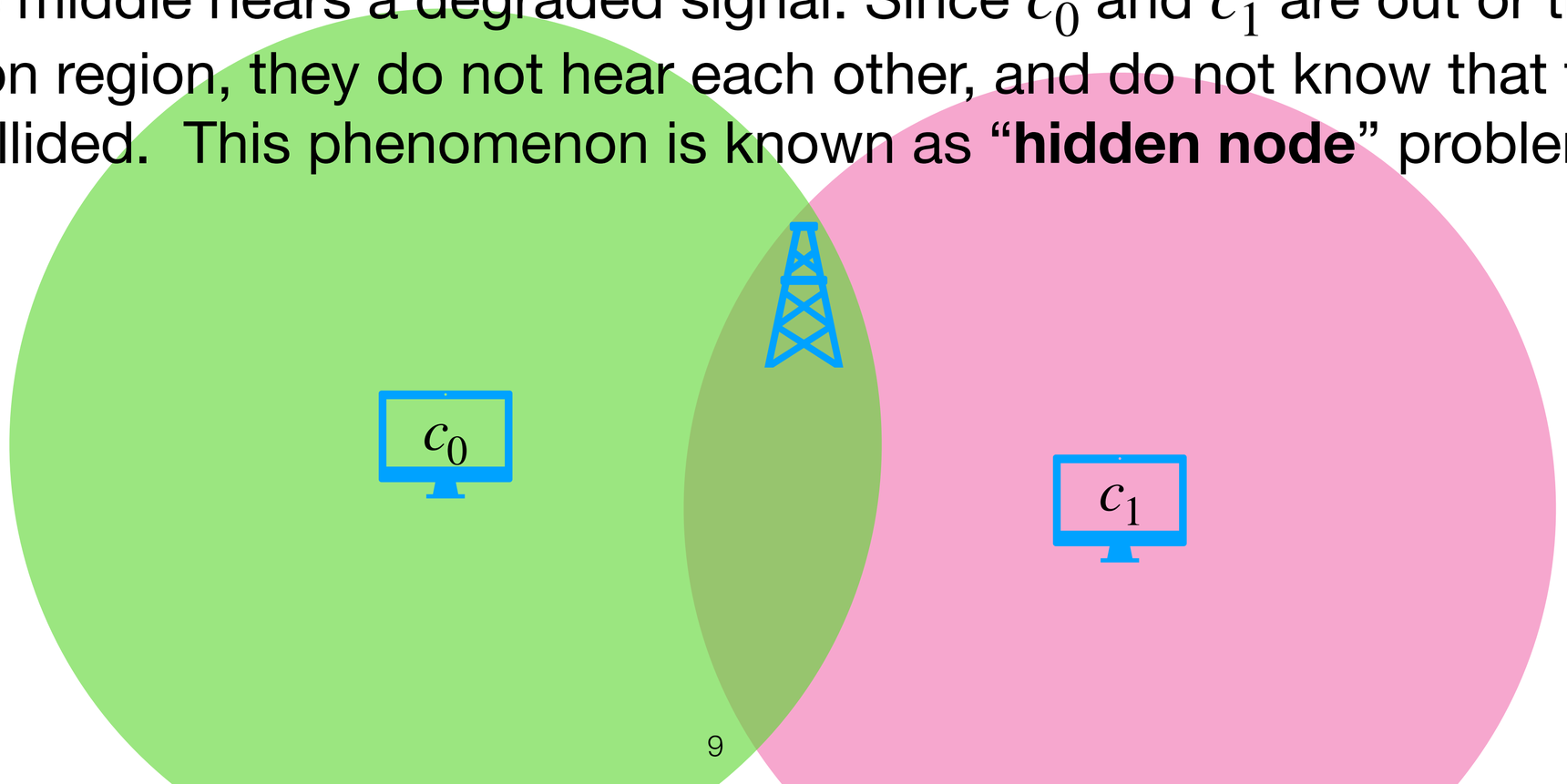
- Wireless networks are different in several aspects:
  - attenuation
  - signal degradation: after 70 dB of signal strength loss, the signal strength drops, and this has an impact not only in the performance but also in collision detection



- This is very different from what happens in wired connections, where the signal propagates through the wire and can travel long distances without attenuation

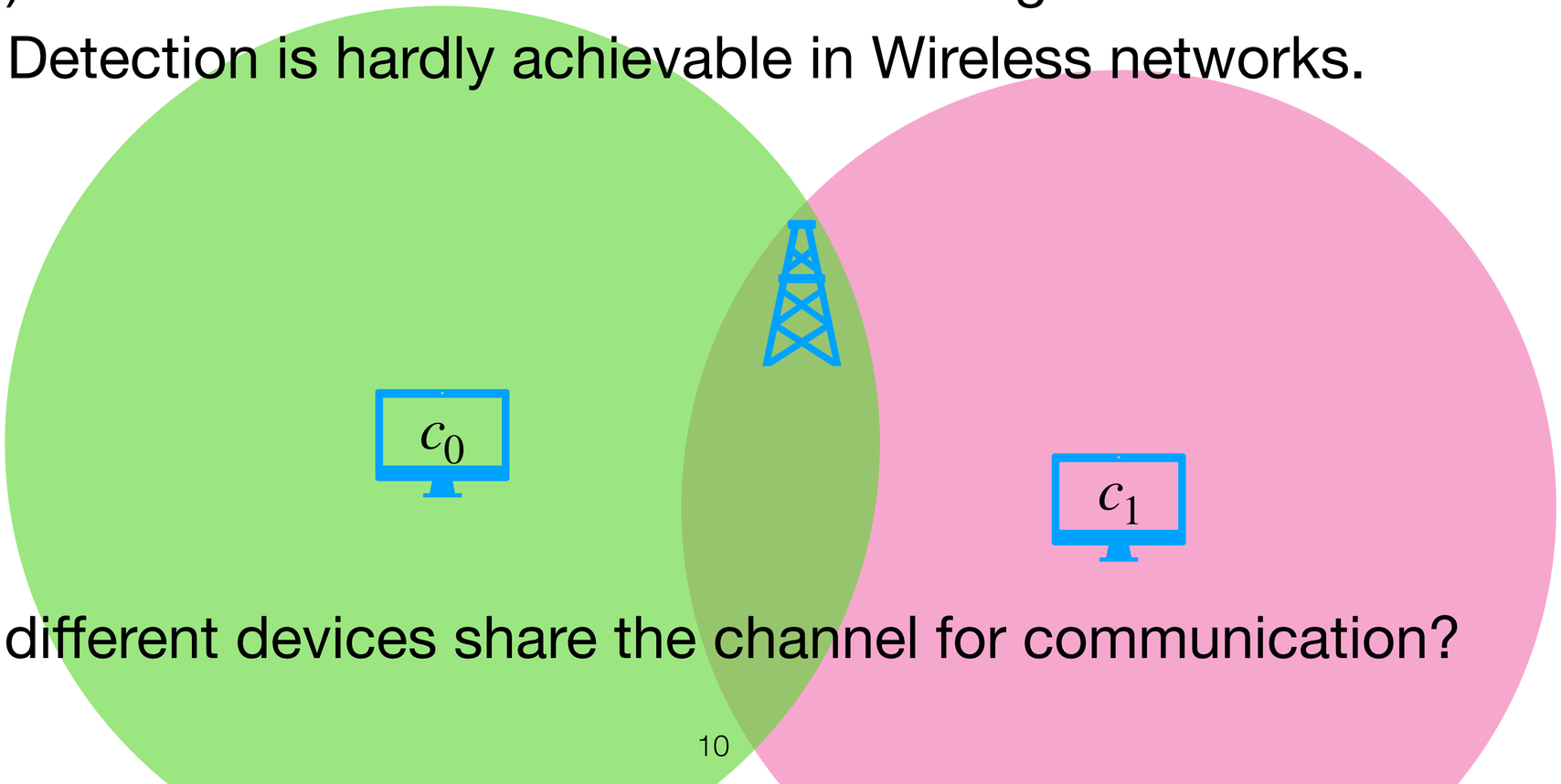
Collision Detection, Wireless (2)

Consider a similar scenario as before. At time 0, node c_0 sends a packet, and so does node c_1, $8\mu s$ later. After a while, the signals collide and the access point in the middle hears a degraded signal. Since c_0 and c_1 are out or their transmission region, they do not hear each other, and do not know that their packets collided. This phenomenon is known as "**hidden node**" problem



Collision Detection, Wireless (3)

- The hidden node problem is when a device might be visible to an access point (AP) but not to other nodes communicating with the AP.
- Collision Detection is hardly achievable in Wireless networks.



How do different devices share the channel for communication?

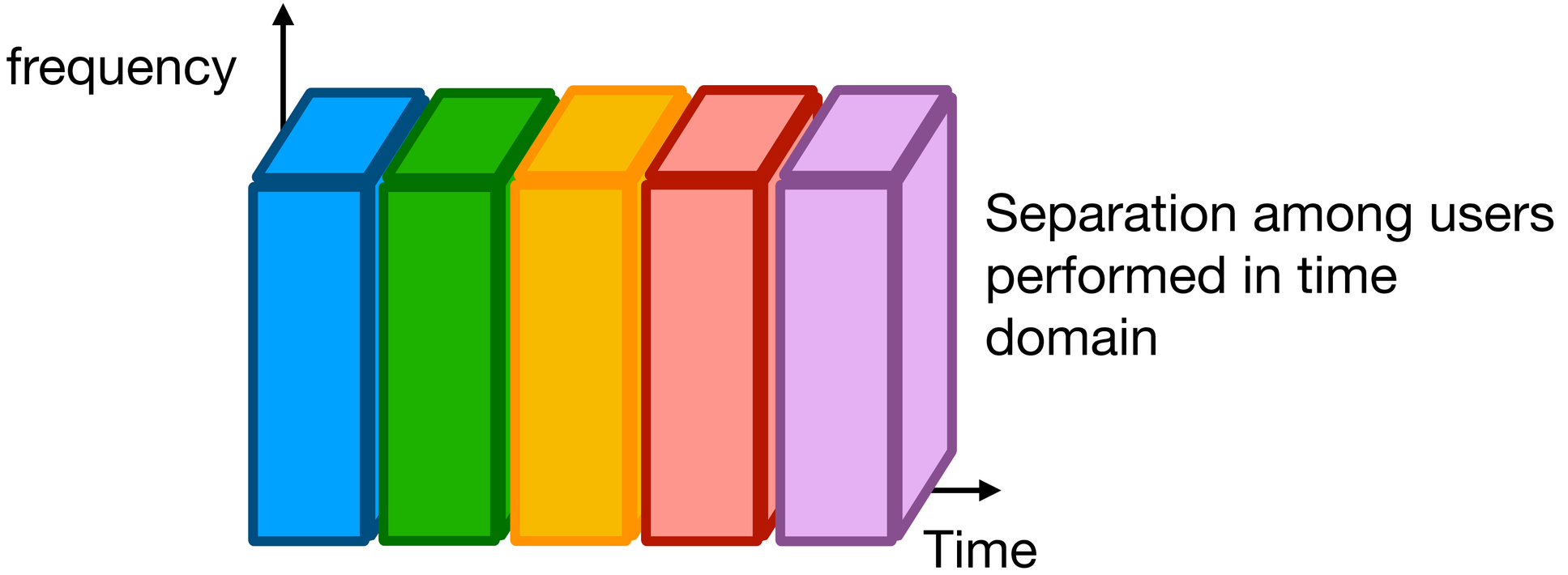
Multiple Access MAC protocols

Wireless networks

- Time Division Multiple Access (TDMA)
- Frequency Division Multiple Access (FDMA)
- ALOHA
- Multiple Access with Collision Avoidance (MACA)
- Code Division Multiple Access (CDMA)

Time Division Multiple Access (TDMA) (1)

- Uses Time Division Multiplexing.
- Each user is allowed to transmit only within specific time intervals (time slots) i.e., different users transmit in different time slots.
- When users transmit, they occupy the whole frequency band.

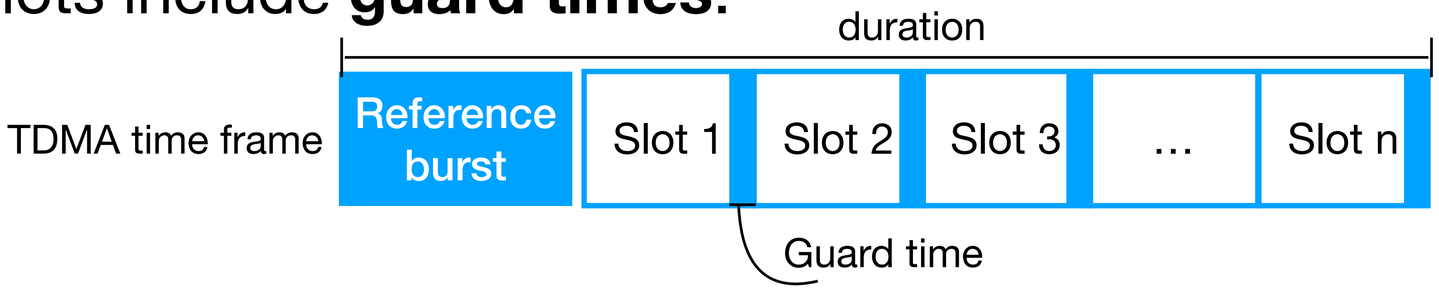


Time Division Multiple Access (TDMA) (2)

- TDMA requires a **centralised coordinator** that allocates time slots to users within a **time frame**.
- The coordinator sponsors the time frame by broadcasting a **reference burst**, containing synchronisation information and time slot allocations.
- Duration of the time frame depends on:
  - The number of users sharing the channel.
  - The length of each time slot.
  - The system's transmission rate.
- These parameters change also depending on what kind of network uses TDMA

Time Division Multiple Access (TDMA) (3)

- To prevent overlaps due to clock mismatches and propagation delay, time slots include **guard times**.

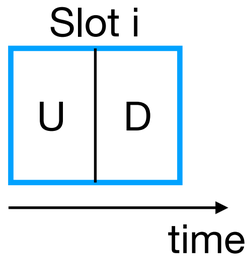


- During each time slot, the communication between a device and the coordinator can be achieved in two ways.

Half-Duplex TDMA

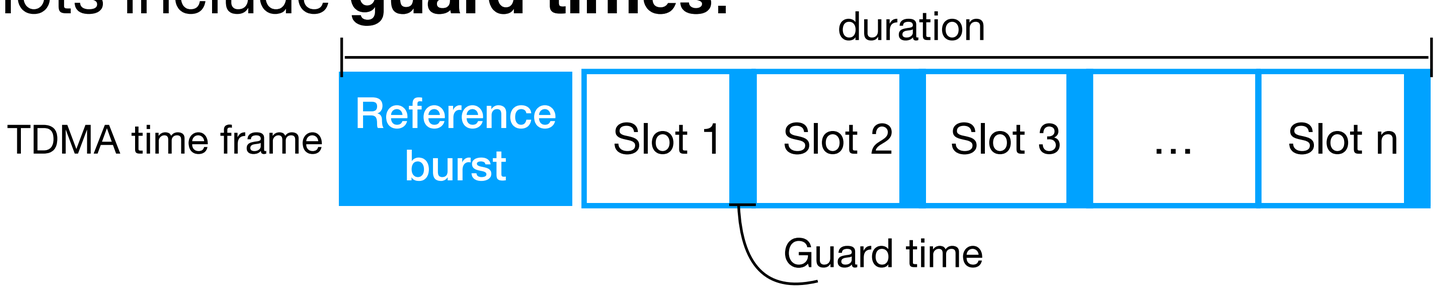
- One slot is divided into two parts, one for uplink (device to coordinator communication) and one for downlink (coordinator to device comm.).
- OR Some time slots are allocated for uplinks other for downlinks

Used in GSM



Time Division Multiple Access (TDMA) (3)

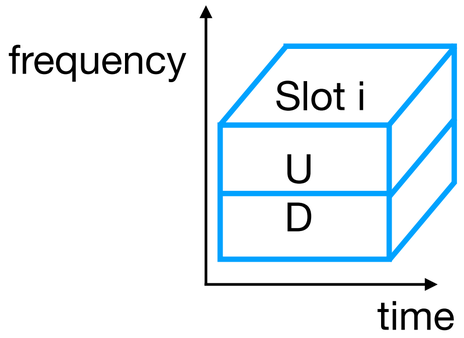
- To prevent overlaps due to clock mismatches and propagation delay, time slots include **guard times**.



- During each time slot, the communication between a device and the coordinator can be achieved in two ways.

full-Duplex TDMA + Frequency Division Duplexing

Device transmits and receives at the same time by using different frequency bands.

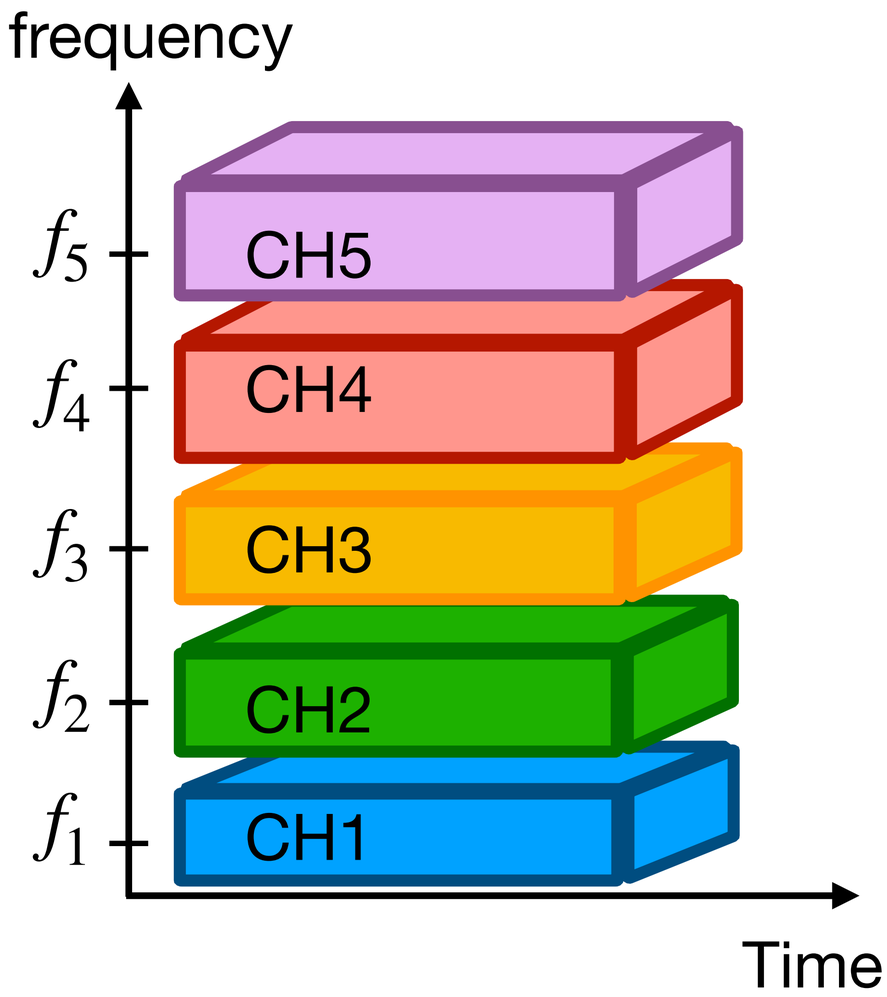


Time Division Multiple Access (TDMA) (4)

- ✅ Efficient bandwidth usage, spectrum efficiency is maximised
- ✅ Collision avoidance probability is maximised thanks to time division multiplexing.
- ✅ Low power consumption, as devices transmit only during their active time slots and can go to sleep/idle mode the remainder of the time.
- ❌ Devices need precise time synchronisation, implying overhead
- ❌ Latency issues in large networks with many users. Each user has to wait long time before transmitting.
- ❌ Inefficient for bursty traffic, as slots are allocated even when users have no data to transmit, or not enough time is allocated for users that have to transmit a lot of data
- ❌ Guard periods can introduce overhead.

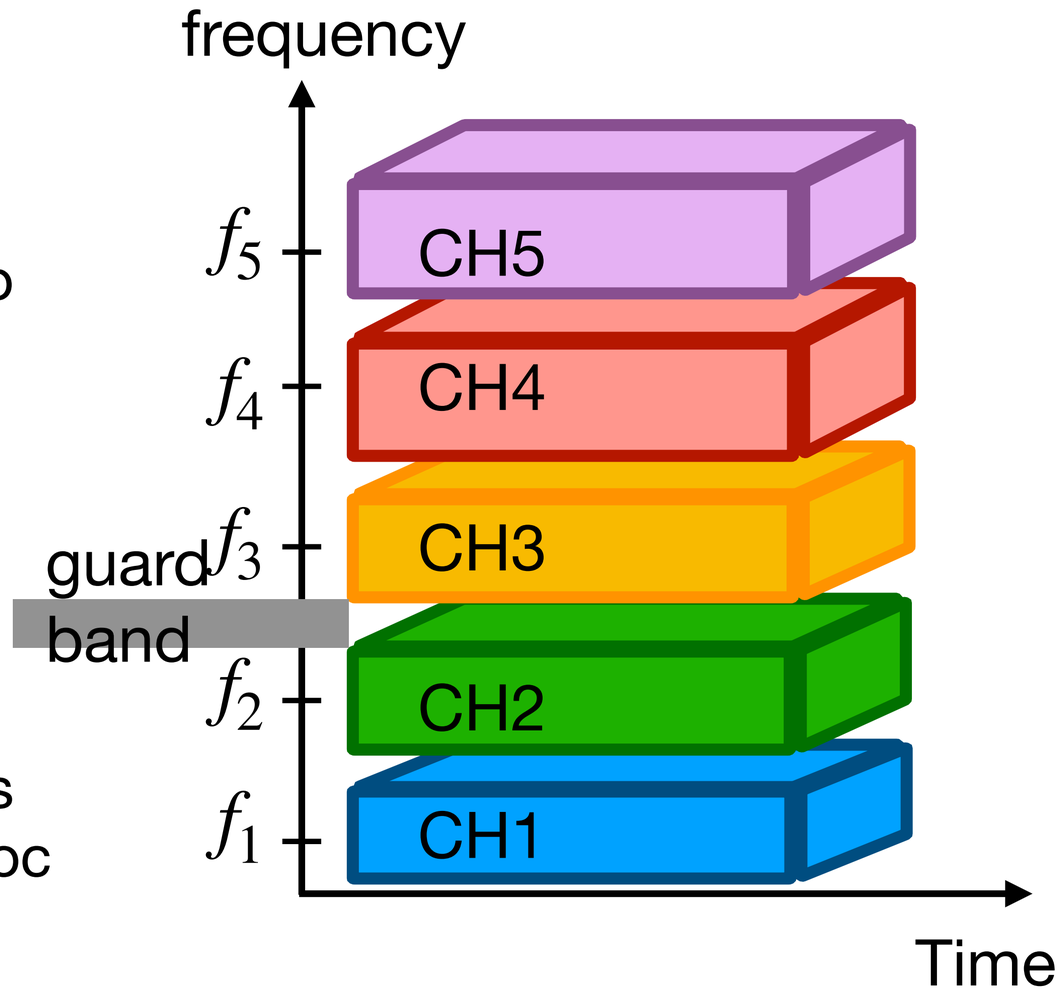
Frequency Division Multiple Access (FDMA) (1)

- The band is divided into sub-bands, each centred around the frequency that identifies the channel (Frequency Division Multiplexing).
- Each sub-band is allocated to a single user that controls it during the entire connection period.
- Different bands are used for uplink and downlink.
- No need for synchronisation among users.



Frequency Division Multiple Access (FDMA) (2)

- Guard bands are used to avoid frequency interferences.
  - Even if different frequencies are allocated for each user, imperfections in transmitters, receivers, and filters, signals can spill over into adjacent frequency bands.
- May have a coordinator or not:
  - Centralised FDMA (some cellular networks, satellite networks). Coordinator assigns frequencies to users.
  - Decentralised FDMA. Users select frequencies manually or algorithmically (Radio, some ad hoc networks)

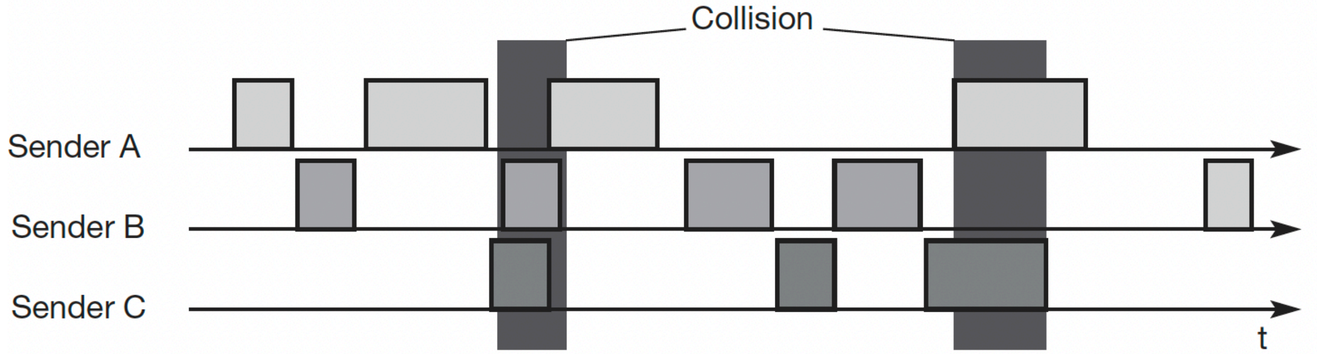


Frequency Division Multiple Access (FDMA) (3)

- ✅ Low Latency, no waiting time slots, suitable for real-time applications.
- ✅ Avoid collisions.
- ✅ Supports continuous transmission, which is good for applications needing constant data flow, like radio, TV, and cellular voice calls.
- ❌ Limited number of users. The total bandwidth is fixed, once the sub-bands are allocated, no additional users can be added.
- ❌ Guard bands waste some bandwidth
- ❌ Inefficient for bursty traffic, when users are not transmitting, they still occupy an entire channel. If a user has to transmit a lot of data, its channel might be too narrow.
- ❌ More expensive hardware than TDMA, as it requires high-quality bandpass filters and frequency synthesizers to maintain strict frequency separation

Classical ALOHA (1)

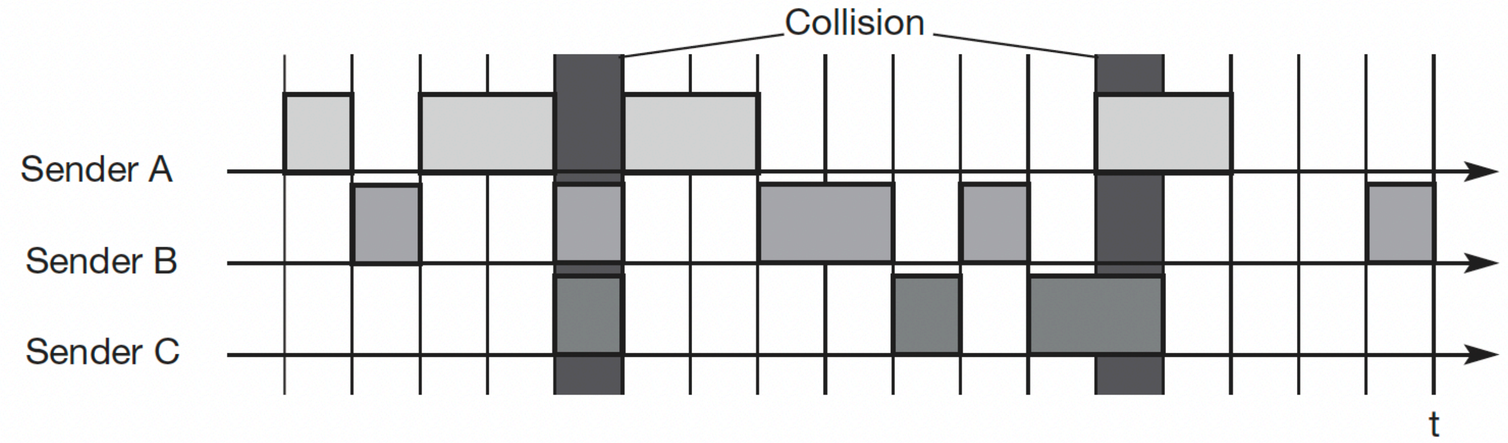
- Classical ALOHA works as a Time Division Multiple Access method without a coordinator (all senders use the entire frequency band).
- Each station can access medium at any time, following a **random access scheme.**
  - Not only there is not a central coordinator, but nodes do not coordinate among each other in a distributed fashion.



- If two or more stations access the medium at the same time, a collision occurs and the transmitted data is destroyed. Resolving this problem is left to higher layers (e.g., retransmission of data).

Slotted ALOHA

- First refinement of ALOHA
- Devices synchronise with time slots.
  - Transmission can start only at the beginning of a time slot.
  - No coordination, collisions can still occur.



ALOHA improvements (1)

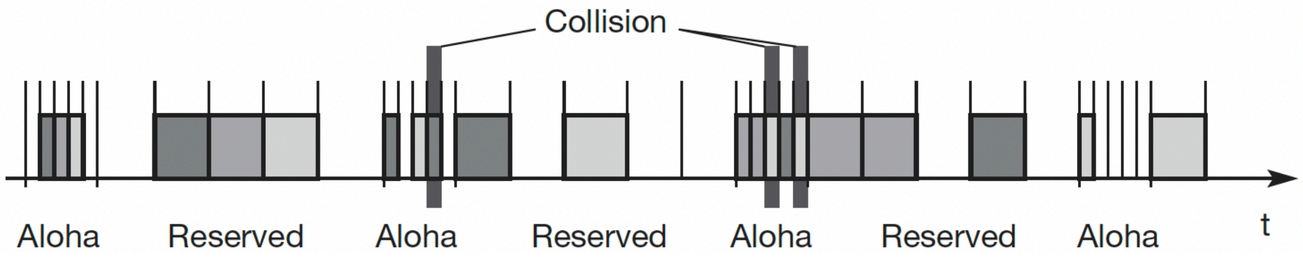
Carrier sense multiple access (CSMA) schemes

- **non-persistent CSMA.**
  Stations sense the carrier and start sending immediately if the medium is idle. If the medium is busy, the station pauses a random amount of time before sensing the medium again and repeating this pattern.
- **p-persistent CSMA.**
  Nodes also sense the medium, but only transmit with a probability of p, or defer to the next slot with the probability 1-p.
- **CSMA with collision avoidance (CSMA/CA).**
  Carrier sensing is combined with a back-off scheme in case of a busy medium to achieve some fairness among competing stations (RTS/CTS exchange, see slide 25-26). Used in Wi Fi.

ALOHA improvements (2)

Reservation Mechanisms

- **Demand assigned multiple access (DAMA or reservation ALOHA)**
  Mostly used in satellite communication. Needs a central coordinator.
  Two phases:
  - **Contention phase**
    - follows the slotted Aloha scheme.
    - Stations exchange messages in a random access way to reserve future slots during the transmission phase.
    - Collisions can occur.
    - If successful, a time slot in the future is reserved (no other station is allowed to transmit during this slot)
    - The satellite collects all successful requests and sends a r**eservation list** with access rights for future slots.



  - **Transmission phase**
    - Stations transmit data during their reserved time slots.

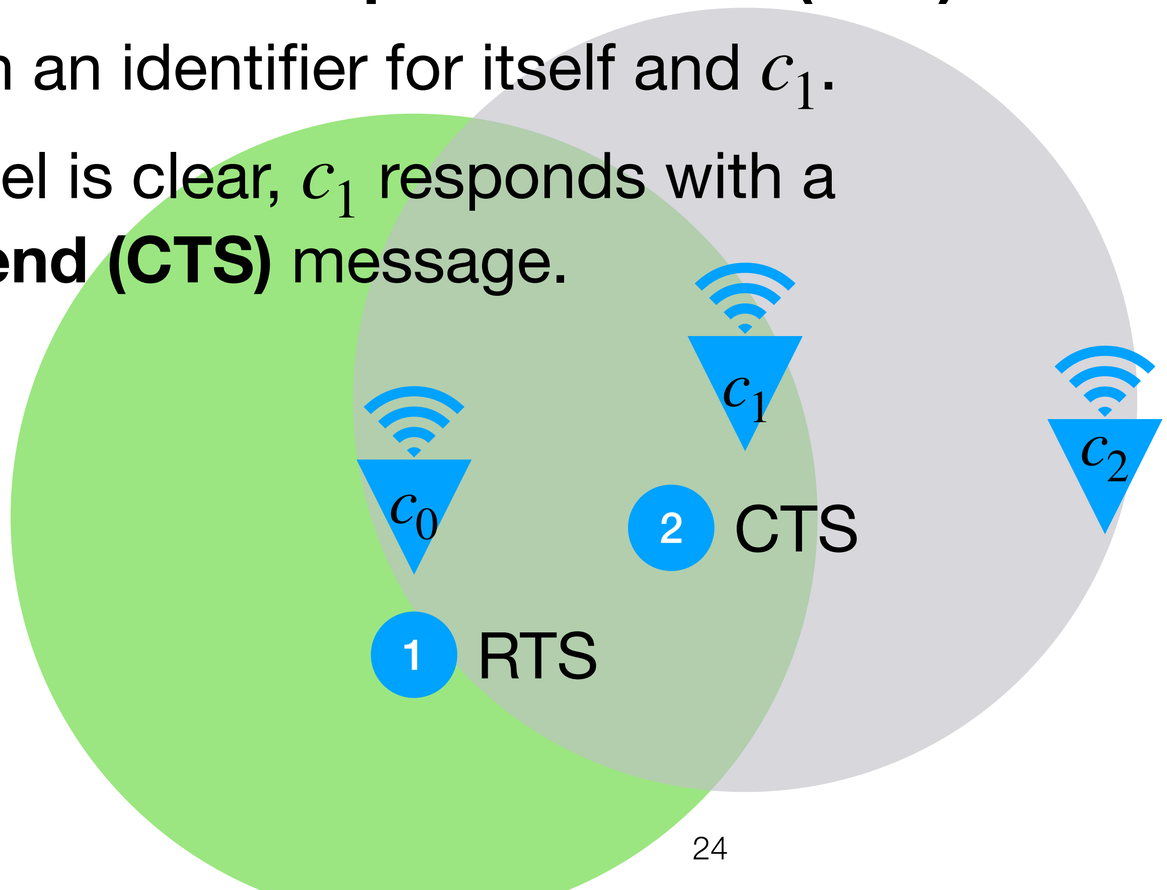
Explicit reservation mechanism

Multiple Access Collision Avoidance (MACA) (1)

- Hidden node problem can be solved by synchronisation with the coordinator, that usually is an AP or a base station.
- ALOHA and slotted ALOHA are very prone to collisions, but they are super simple and versatile.
- Carrier sensing improvement methods still suffer the hidden node problem. A device might sense the medium and believe it is free, but this is just because it cannot hear signals from other devices.
- **MACA** presents a simple scheme to avoid collisions and solve the hidden node problem.
  - Does not need a coordinator, is random access as ALOHA but uses dynamic reservation.

Multiple Access Collision Avoidance (MACA) (2)

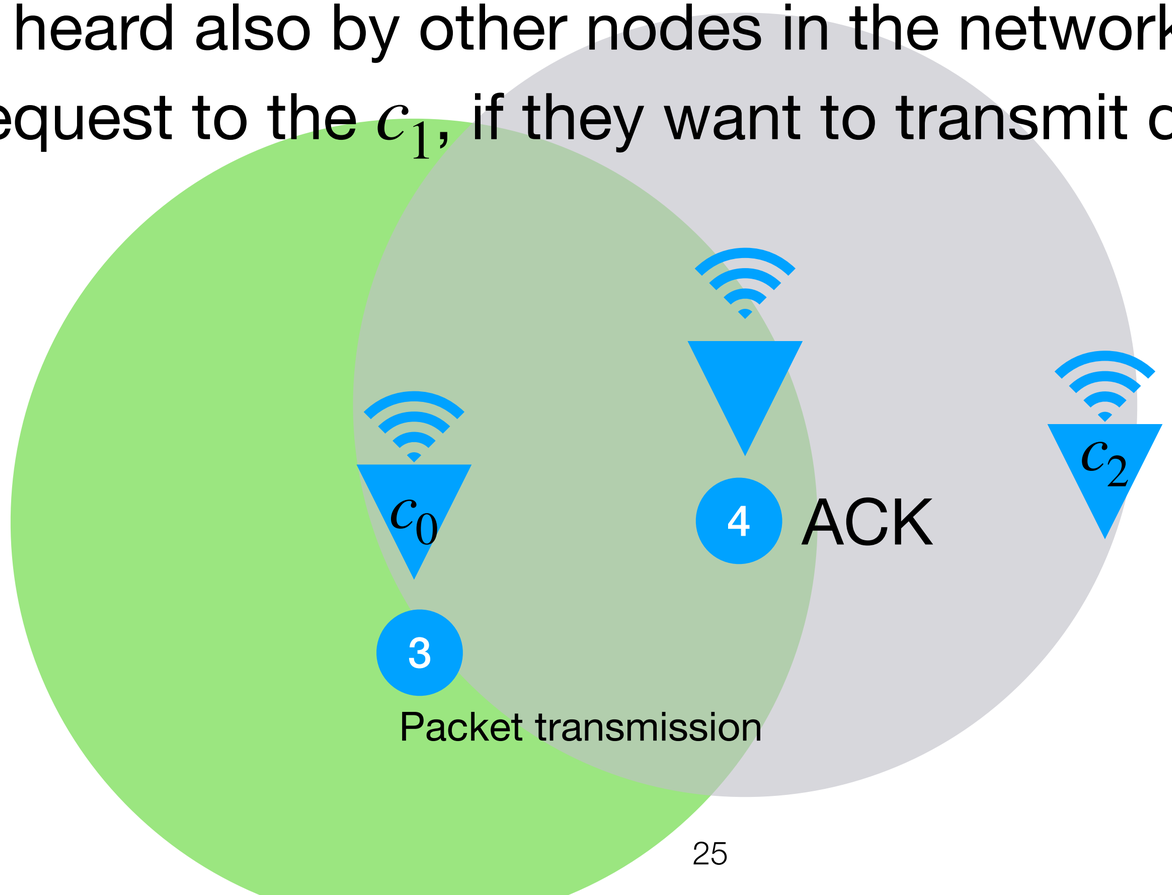
- c_0 and c_2 want to send to c_1. c_0 is the first to start the transmission and is hidden from c_2.
- c_0 first broadcasts a **Request To Send (RTS)** request with an identifier for itself and c_1.
- If the channel is clear, c_1 responds with a **Clear To Send (CTS)** message.



The CTS message is not only heard by c_1, but also by the other nodes in c_1's range (including c_2). These nodes know that someone else is starting to transmit, and therefore do not begin their own transmission, otherwise their signals might collide.

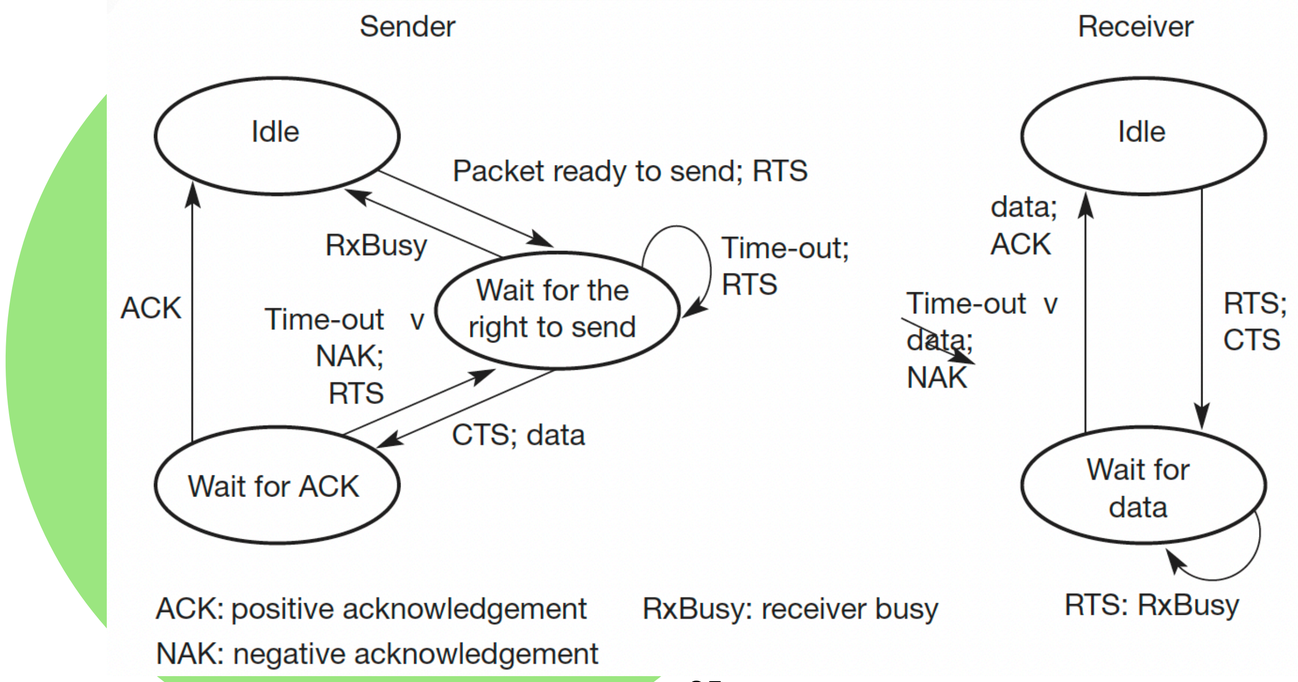
Multiple Access Collision Avoidance (MACA) (3)

- Upon receiving the CTS message from c_1, c_0 transmits its packet.
- The packet is received by c_1, which responds with an ACK.
- The ACK is heard also by other nodes in the network, which can send their RTS request to the c_1, if they want to transmit data.



Multiple Access Collision Avoidance (MACA) (3)

- Upon receiving the CTS message from c_1, c_0 transmits its packet.
- The packet is received by c_1, which responds with an ACK.
- The ACK is heard also by other nodes in the network, which can send their RTS request to the c_1, if they want to transmit data.



Multiple Access Collision Avoidance (MACA) (4)

- ✅ Mitigates the Hidden Terminal Problem thanks to RTS/CTS.
- ✅ Reduces probability of collisions, specially in high traffic conditions
- ✅ More fairness between nodes
- ❌ Increased overhead
- ❌ Scalability issues

Code Division Multiple Access (CDMA) (1)

- Uses Code Division Multiplexing for dividing the channel and assigns different codes to each.

- We have seen that in DSSS, codes should be orthogonal (their inner product should be zero).
  Why?

- To maximise their distance

- Example 1:
  A wants to send 1 and B wants to send 0. A's chip code is a = 010011, B's chip code is b = 110101.

Code Division Multiple Access (CDMA) (2)

- Example 1:
  To illustrate this example, let's assume that we encode the binary 0 as -1 and 1 as +1.

- A sends a*(+1) = +1*(-1,+1,-1,-1,+1,+1)=(-1,+1,-1,-1,+1,+1)

- B sends b*(-1) = -1*(-1,-1,+1,-1,+1,-1)=(+1,+1,-1,+1,-1,+1)

- Signals are transmitted at the same time using the same frequency, so signals superimpose. Assume they have the same strength.

- The receiver sees the signal a*(+1)+ b*(-1) = (0,2,-2,0,0,2).

Code Division Multiple Access (CDMA) (3)

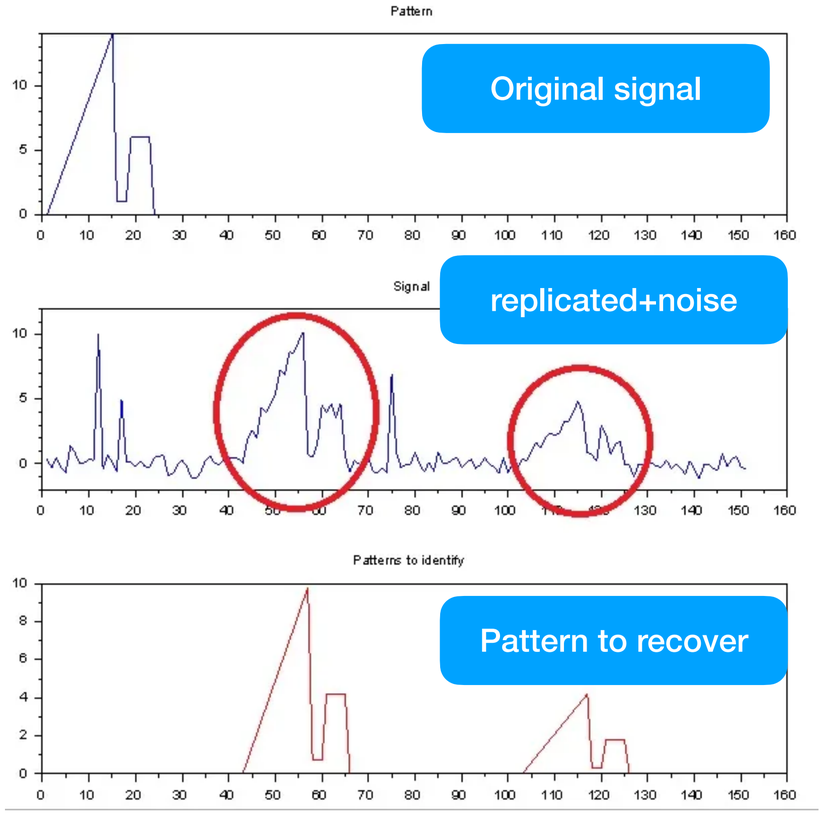
- Example 1:
  The receiver wants to read the signal from A, and therefore applies a to C
- C*a = <(0,2,-2,0,0,2), (-1,+1,-1,-1,+1,+1)> =0+2+2+0+0+2=6.
- As the result is much higher than 0, the receiver understands that A sent a +1.
- If the receiver does the same thing on B, it gets -6, which is way less than 0, hence it understands that B sent a 0.
- This example is oversimplified: codes were short, noise was neglected, the two signals were sent with the same power (see chapter 3 of “mobile communication” for more realistic examples)

Code Division Multiple Access (CDMA) (4)

- CDMA is a powerful multiple access scheme, but requires complex circuits in the transmitter and receiver antennas supporting it.

- Communicating with n devices requires programming of the receiver to be able to decode n different codes (and sending with n codes, too).

  - “Rake receivers”, which can separate multi path effects thanks to several correlators (signal processing circuits that implement correlation algorithms).



Code Division Multiple Access (CDMA) (5)

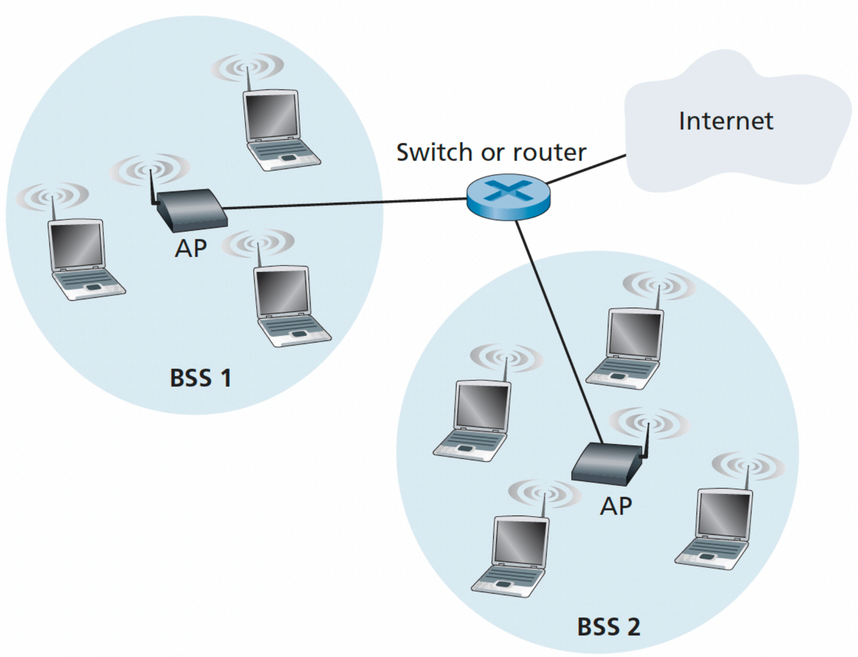
- It is mostly used for military purposes and cellular networks.
  - In cellular networks, the complexity of the antennas is mostly at the base station, since mobile devices only communicate with the base stations.
  - Implementing many different spreading codes for ad-hoc networks is not feasible.
- ✅ Efficient use of bandwidth, all users transmit over the entire frequency spectrum at the same time
- ✅ Resistant to interferences and noise
- ✅ Enhanced security and privacy, as spreading codes add a level of obfuscation in transmitted data
- ❌ Complex system design
- ❌ Code management
- ❌ Not scalable with many users

5.2.2 Wireless Networks

- Wireless Networks are very different from one another.
- The WiFi routers, our computers/laptops and mobile devices connected to the router form a wireless network, but it is not a IoT network.
- Not all wireless networks are IoT networks, while most IoT networks are wireless networks.

WiFi: IEEE 802.11 Wireless LANs

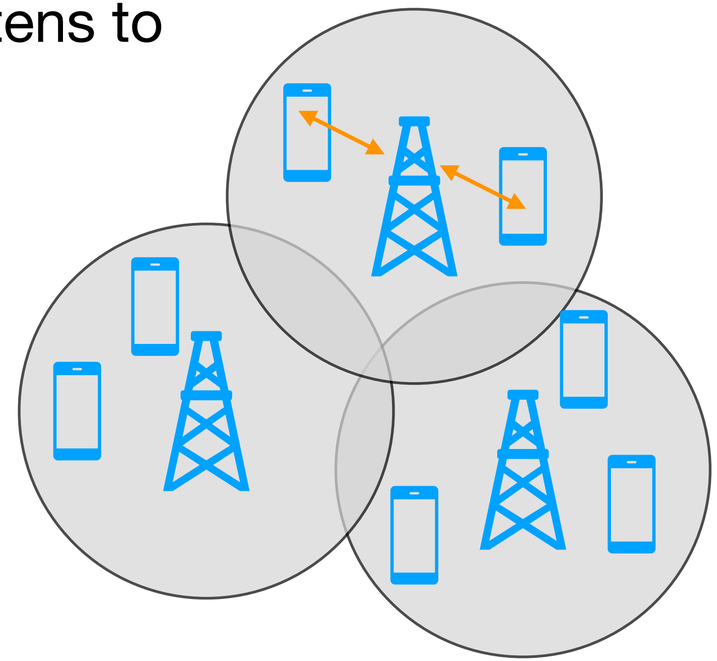
- Fundamental building block of the WiFi architecture is the **basic service set** (BSS).
- A BSS contains multiple wireless devices supporting WiFi (e.g., computers, mobile phones etc) and an Access Point (AP).
- APs are interconnected with a router through an Ethernet connection.



- Can be “infrastructure-less”, meaning without an AP. Computers can organise in an “ad-hoc" fashion. E.g., two apple devices exchanging files with AirDrop.

Cellular Networks

- Cellular Networks are wireless networks consisting of both stationary and mobile nodes.
- Stationary nodes are base stations (BS) connected by wired links, forming a fixed infrastructure.
- The number of mobile nodes is much larger than the number of BSs.
- Each BS covers a large region with little overlap and serves tens to hundreds of mobile nodes in the region.
- Each mobile node is a single hop away from its closest base station.
- BSs have sufficient power supply and the mobile users can conveniently recharge the batteries in their handsets.
  - Energy conservation is a secondary matter



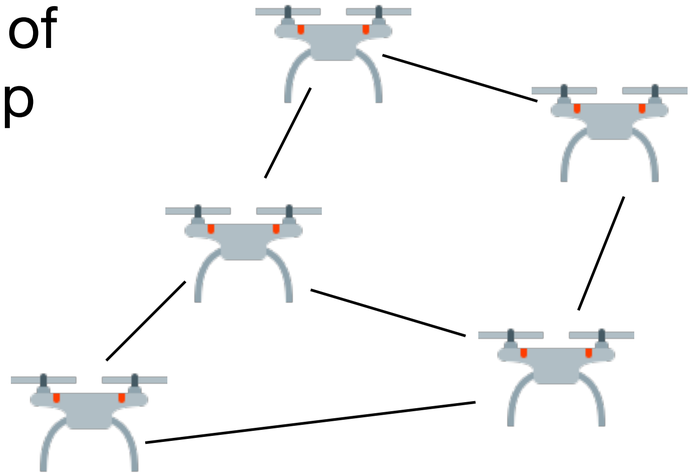
Mobile Ad-hoc Networks (MANETs)

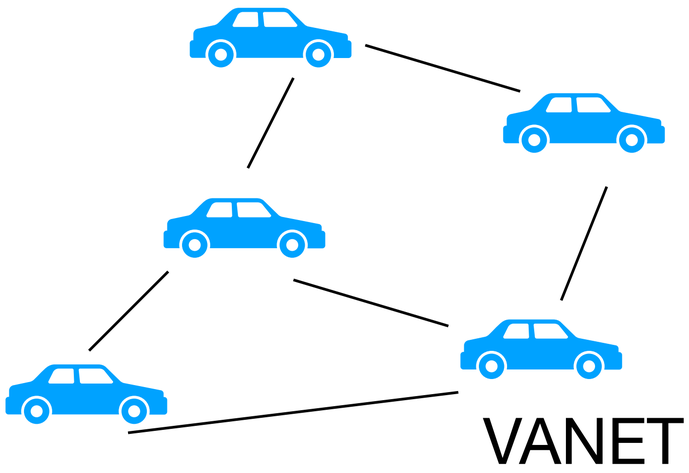
- A MANET is a **peer-to-peer** network that usually consists of tens to hundreds of mobile nodes and covers a range of up to hundreds of meters.

- All nodes are mobile and there is no fixed infrastructure (“**ad-hoc**”).

  - The network must organize the nodes to form a communication infrastructure, perform routing, and maintain the organization and routing under mobile conditions.

- The primary goal of a MANET is to provide high quality of service in the face of high node mobility.





Wireless Sensor Networks

- A sensor network consists of a large number of sensor nodes densely deployed in a geographical field.
- Sensor nodes are usually powered by batteries (limited power capacity). It is often difficult or impossible to change or recharge batteries for these nodes (the lifetime of a sensor network largely depends on the lifetime of its sensor nodes).
  Energy consumption is a primary concern.
- Nodes are often deployed in a ad-hoc fashion. They must be able to organize themselves into a communication network.
- The topology of a sensor network changes more frequently due to both node failure and mobility. Sensor nodes are prone to failures.
- Sensor nodes have very limited computational capacity and memory.
- **Require specific MAC protocols to ensure long life to nodes.**

Precision agricolture sensors

Energy Draw in WSNs

- All sensors' activities consume energy:
  - sensing
  - data processing
  - communication
- Communication is the major source of energy consumption, for this reason it must be reduced as much as possible in a sensor network.
- During communication, the major sources of energy waste are:
  - **Collisions** (retransmissions of packets increase both energy consumption and delivery latency)
  - **Overhearing** (sensor node receives packets that are destined for other nodes)
  - **Idle Listening** (sensor node is listening to the radio channel to receive possible data packets while there are actually no data packets sent in the network)
  - **Control Overhead**. A MAC protocol requires sending, receiving, and listening to a certain necessary control packets, which also consumes energy not for data communication.

5.2.3 Power Saving Algorithms

Component	Power Usage (approx.)
LTE Radio (transmitting at 1Mbps)	1700 mW
3G Radio (transmitting at 1Mbps)	1700 mW
WiFi Radio (transmitting at 1Mbps)	400 mW
ARM CPU+RAM (100% CPU utilization)	2000 mW
ARM CPU+RAM (idle)	70 mW
Smartphone Screen (100% brightness)	850 mW
GPS (after lock is acquired)	100-150mW
Accelerometer (10Hz)	75µW
Image sensor (1080p@30Hz)	270mW

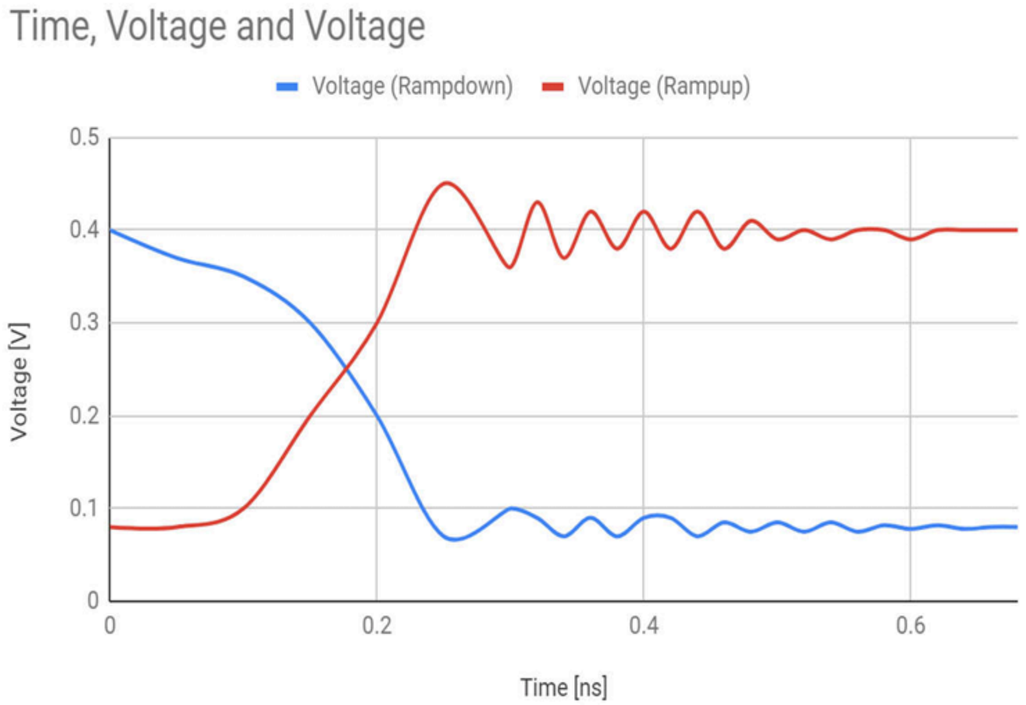
- IoT devices are made of several components that have different power consumption.
- **Idea**: shut down or idle parts of the system to save power.
- **Challenge**: Low-power mode may reduce functionality or stop components from working.

- Some components are very power consuming even in idle mode.
- Some components have a sleep mode that can greatly reduce the power consumption, but reduces the functionality and the promptness of the device to carry out tasks.
  - Protocols that use the sleep mode for energy saving must do it carefully.

Wireless NIC power consumption

Function	Power Usage (approx.)
Transmit	1.33 Watts
Receive	0.97 Watts
Idle	0.84 Watts
Sleep	0.07 Watts

- Entering/leaving sleep mode takes time to wake up and to stabilise.
- YouTube video
- Minimising power consumption in SDNs



Example use case: Cold Supply Chain Auditing (1)

- Many items must be stored at low temperature, also during transportation.
  - medical products: vaccines, blood, insulin, drugs
  - Food: meat, veggie, dairy
  - Chemicals and sensitive electronics

- Cold Chain Transport, i.e., a supply chain where products remain cold
  - ensures cold temperatures at every step of transport process
  - critical for product effectiveness and consumer safety
  - 200B$ market

- IoT plays a crucial role in this field: with IoT, we can monitor products continuously as they move.

Example use case: Cold Supply Chain Auditing (2)

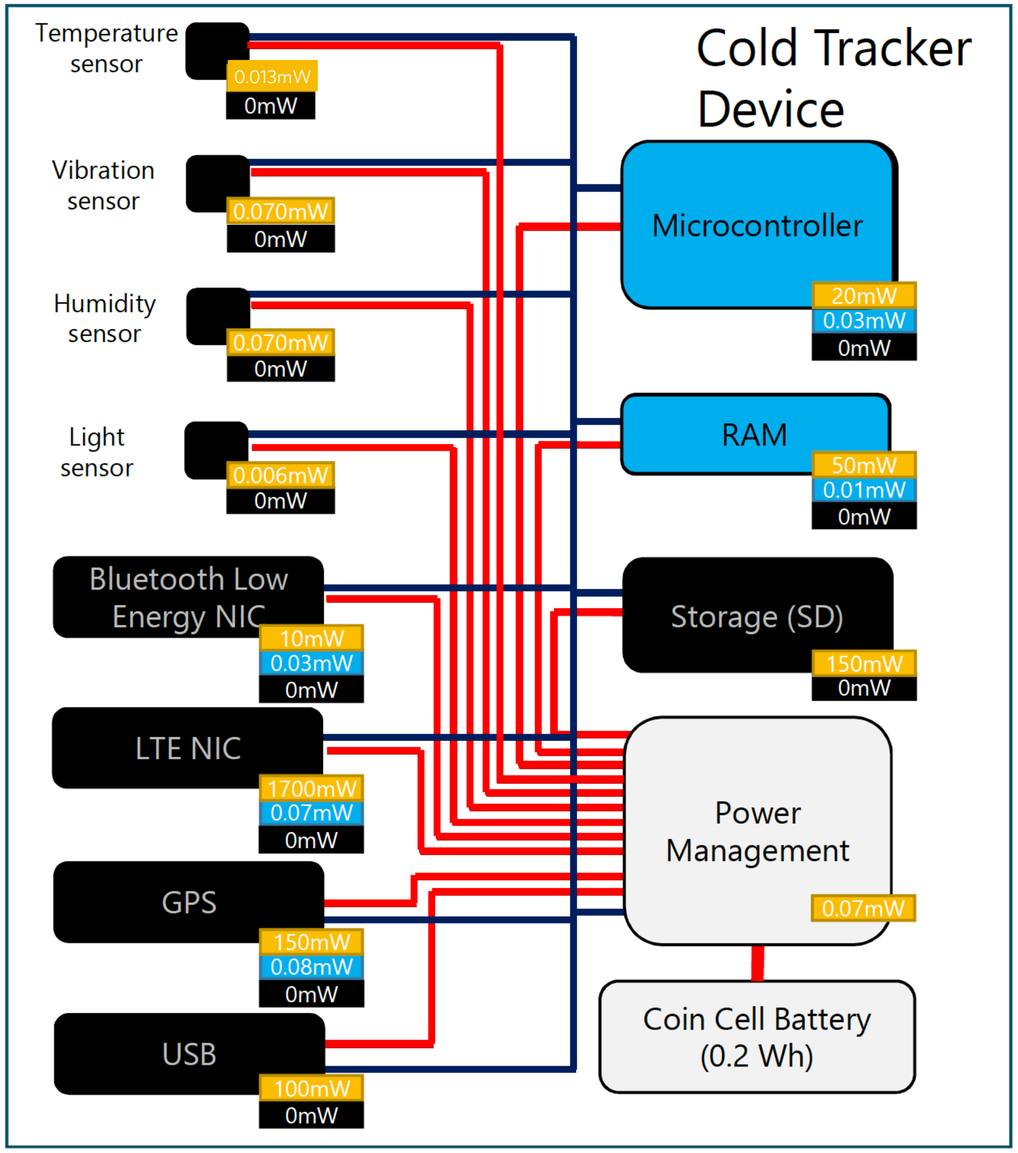
- Suppose you want to build an IoT Cold Chain Auditor
- Use a tag-like sensor attached to products that continually monitors temperature and alarms when it goes above a given threshold and periodically records samples
- Records and communicates findings

Example use case: Cold Supply Chain Auditing (3)

deep sleep

Idle mode

Active mode

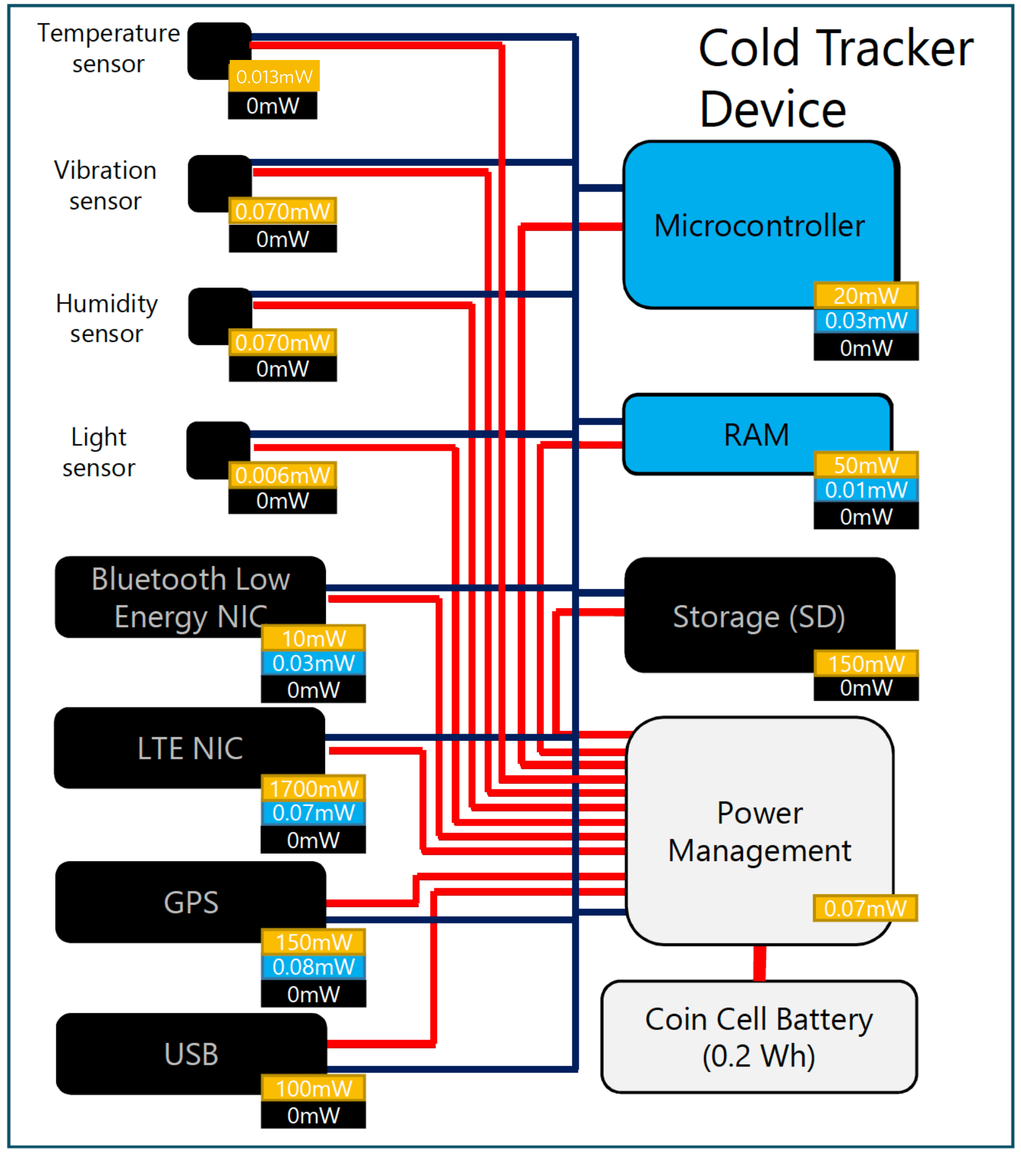


Example use case: Cold Supply Chain Auditing (3)

deep sleep

Idle mode

Active mode



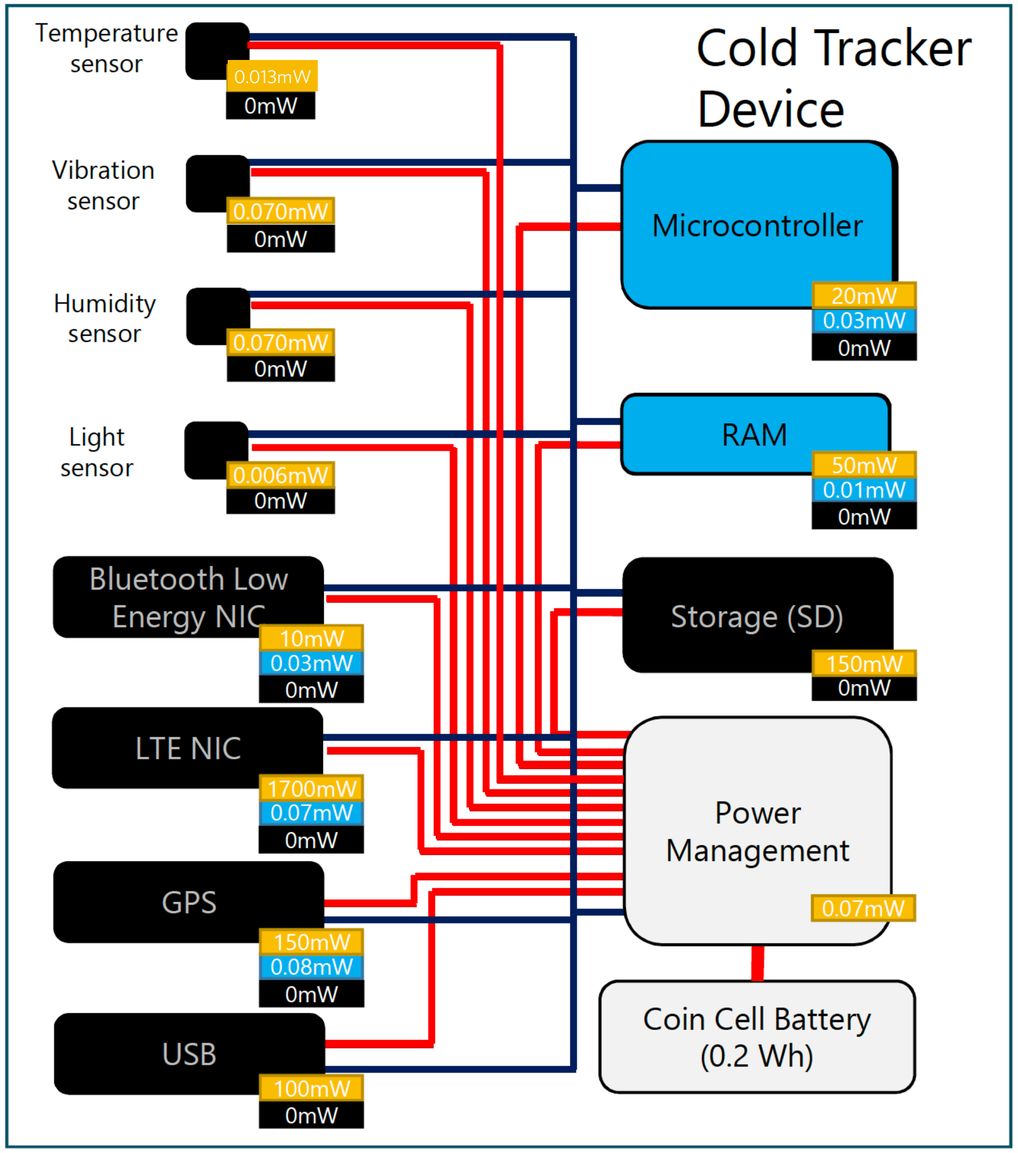
- System on: (0.013+0.07+0.07+0.006+10+1700+150+100+20+50+150+0.07)mW =2180.229mW ->(200mWh/2180.229mW) = 0.0917h~5.5 minutes

Example use case: Cold Supply Chain Auditing (3)

deep sleep

Idle mode

Active mode



- System on: (0.013+0.07+0.07+0.006+10+1700+150+100+20+50+150+0.07)mW =2180.229mW ->(200mWh/2180.229mW) = 0.0917h~5.5 minutes
- Deep Sleep (emergency mode - power management on): (0+…+0+0.07)mW=0.07mW-> (200mWh/0.07mW~2857 hours)~199 days

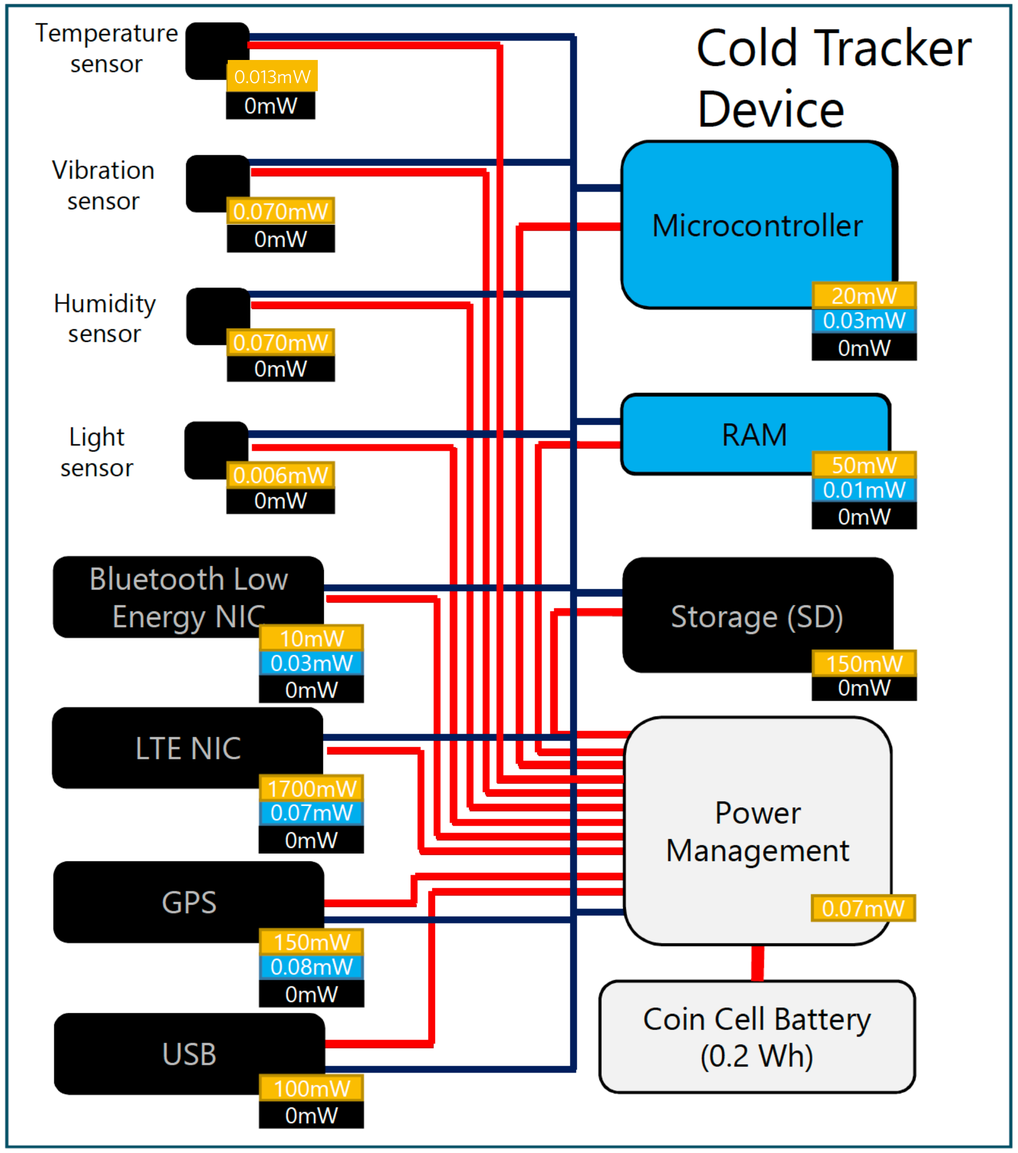
Power management

Example use case: Cold Supply Chain Auditing (3)

deep sleep

Idle mode

Active mode



- System on: (0.013+0.07+0.07+0.006+10+1700+150+100+20+50+150+0.07)mW =2180.229mW ->(200mWh/2180.229mW) = 0.0917h~5.5 minutes
- Deep Sleep (emergency mode - power management on): (0+…+0+0.07)mW=0.07mW-> (200mWh/0.07mW~2857 hours)~199 days
- Sensor-only mode: (0.013+0.07+0.07+0.006+0.03+0.01)mW = 0.199mW -> ~41.8 days

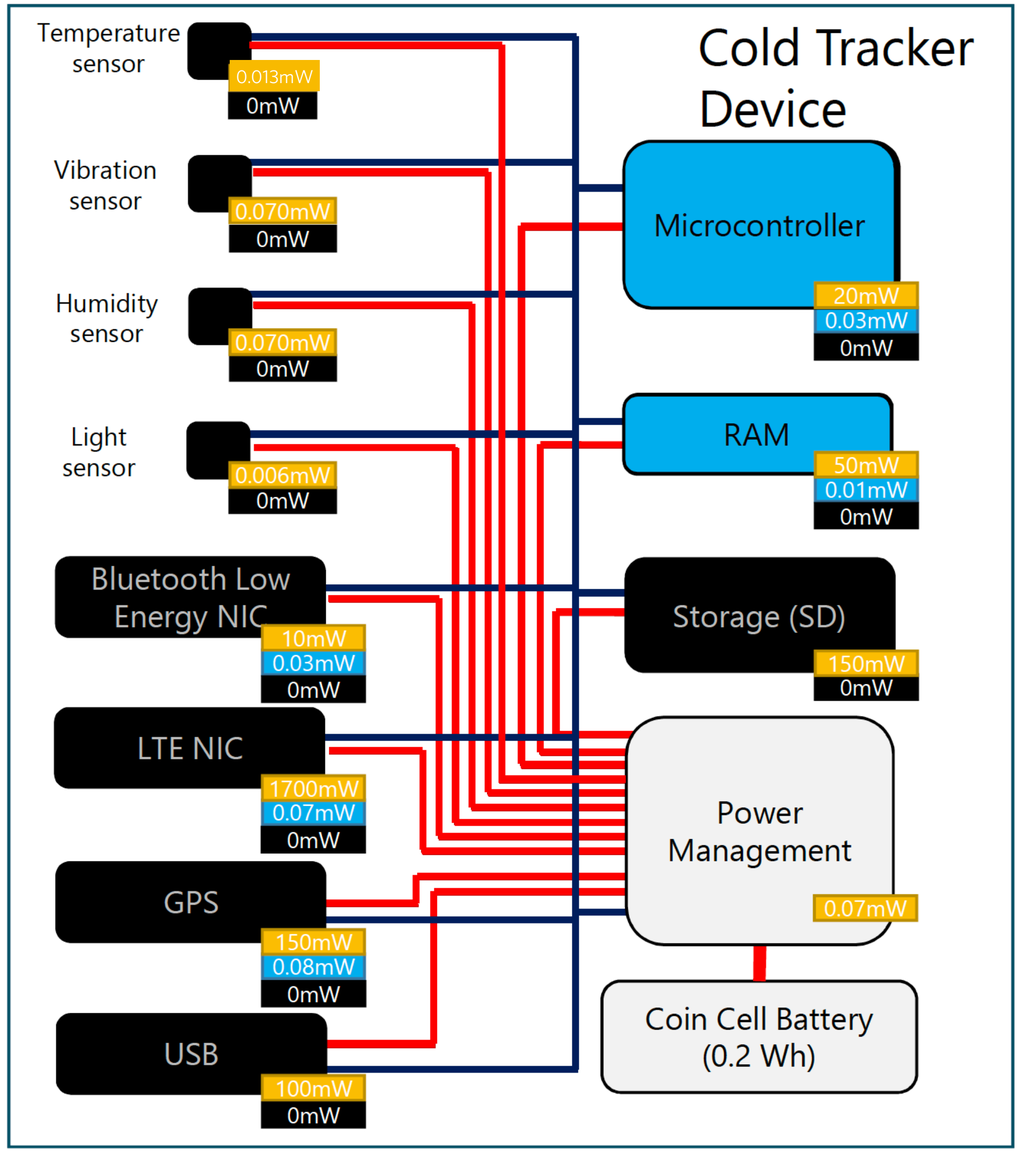
sensors MCU RAM

Example use case: Cold Supply Chain Auditing (3)

deep sleep

Idle mode

Active mode



- System on: (0.013+0.07+0.07+0.006+10+1700+150+100+20+50+150+0.07)mW =2180.229mW ->(200mWh/2180.229mW) = 0.0917h~5.5 minutes
- Deep Sleep (emergency mode - power management on): (0+…+0+0.07)mW=0.07mW-> (200mWh/0.07mW~2857 hours)~199 days
- Sensor-only mode: (0.013+0.07+0.07+0.006+0.03+0.01)mW = 0.199mW -> ~41.8 days
- Sensors + BLE NIC turned on 10 ms per hour + idle MCU + idle RAM: sensors BLE BLE
  (0.013+0.07+0.07+0.006)mW+($10*2.78*10^{-6}$ + 0.03 * (1-$2.78*10^{-6}$) + 0.03+0.01)mW= 0.229mW -> 36.4 days
  MCU RAM

10mW * 10/1000/60/60h

(10/1000/60/60)h = 0.00000278h (10 ms is this fraction of an hour)

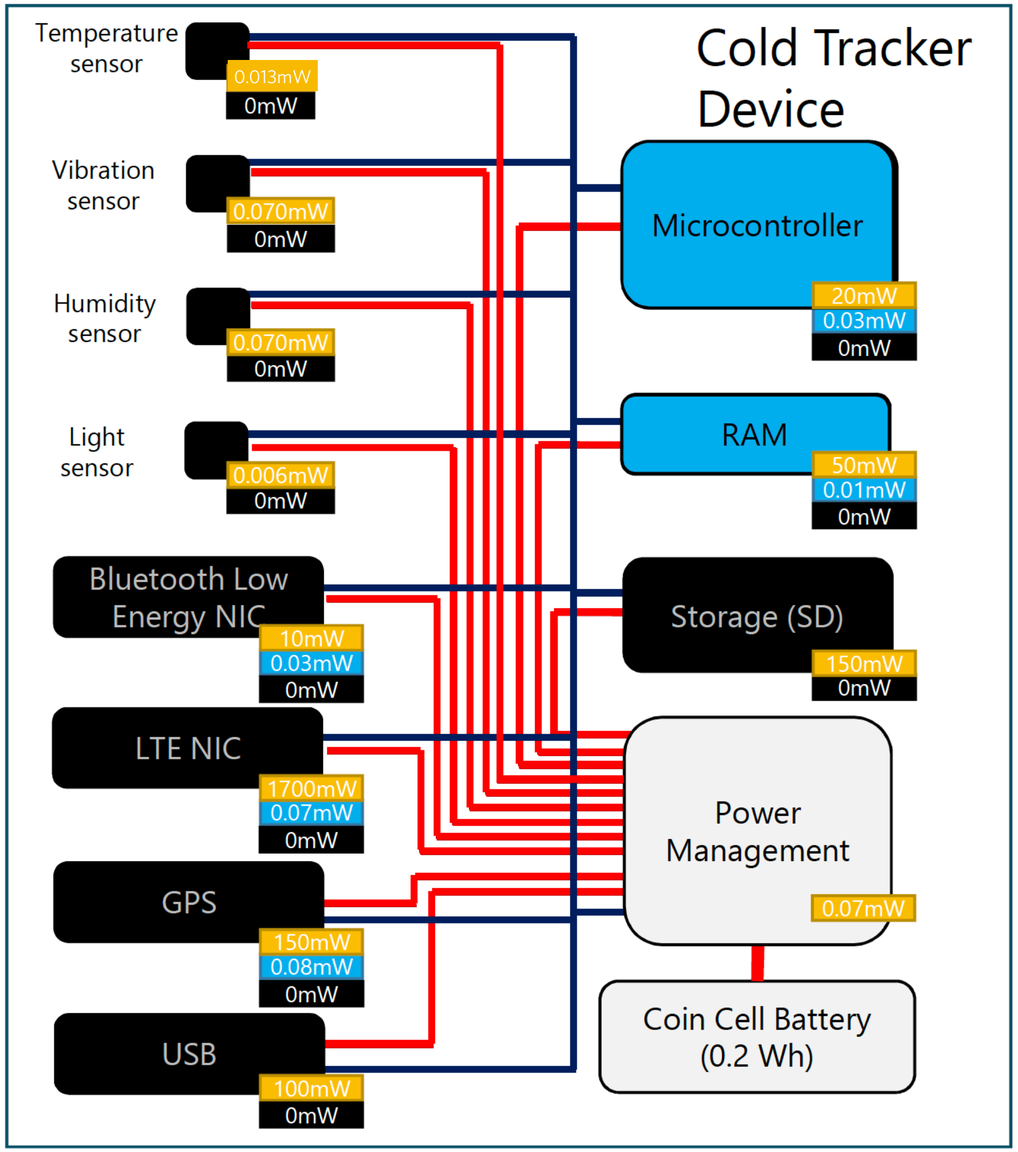
in seconds ↓ in minutes ↓ in hours

Example use case: Cold Supply Chain Auditing (3)

deep sleep

Idle mode

Active mode



- System on:
(0.013+0.07+0.07+0.006+10+1700+150+100+20+50+150+0.07)mW =2180.229mW ->(200mWh/2180.229mW) = 0.0917h~5.5 minutes

- Deep Sleep (emergency mode - power management on):
(0+…+0+0.07)mW=0.07mW-> (200mWh/0.07mW~2857 hours)~199 days

- Sensor-only mode:
(0.013+0.07+0.07+0.006+0.03+0.01)mW = 0.199mW -> ~41.8 days

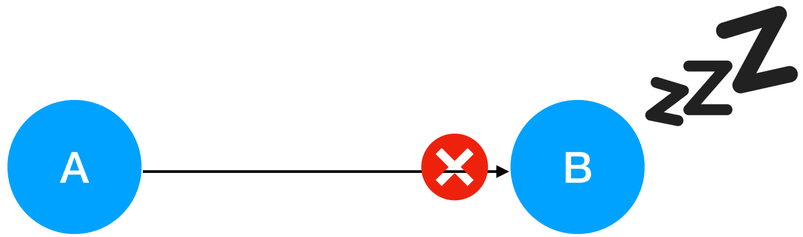
- Sensors + BLE NIC turned on 10 ms per hour + idle MCU + idle RAM:
(0.013+0.07+0.07+0.006)mW+(10*2.78*10^{-6} + 0.03 * (1-2.78*10^{-6}) + 0.03+0.01)mW= 0.229mW -> 36.4 days

- Sensors + BLE on 10 ms/h + GPS on 10s/3h + LTE on 50ms/24 h:
(0.013+0.07+0.07+0.006)mW + (Sensors)
(10*2.78*10^{-6} + 0.03 * (1-2.78*10^{-6}))mW + (BLE)
(150*10/60/(3*60) + 0.08*(1-10/60/(3*60))mW + (GPS)
(1700*50/1000/60/(24*60) + 0.07*(1-50/1000/60/(24*60))mW+ (LTE)
+ 0.03mW+0.01mW = 0.5188mW -> 16.4 days (MCU+RAM)

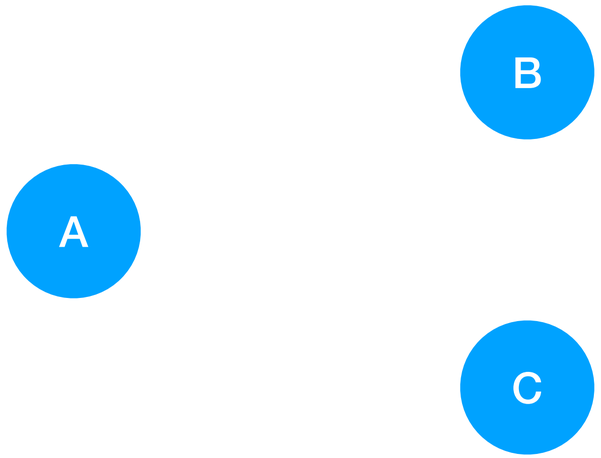
- We have designed a nice power saving “algorithm” for our cold tracker device that sensors the environment, transmits data every once in a while and tracks its GPS position periodically.

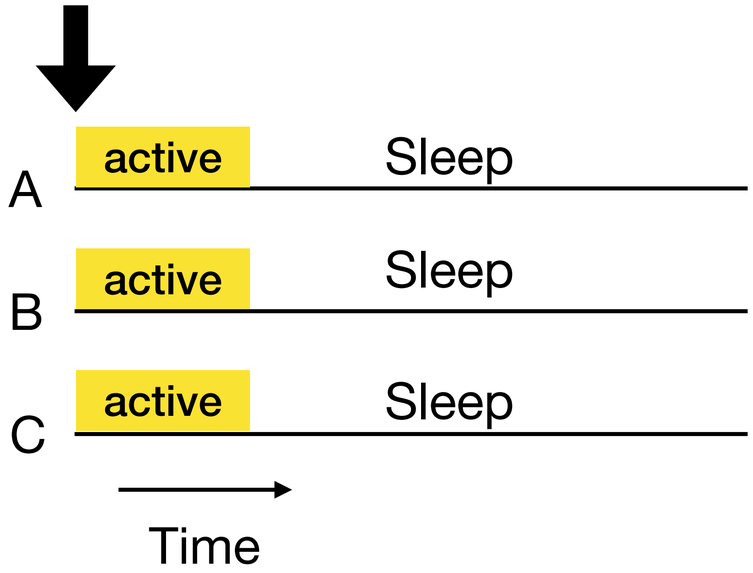
- Challenge: this device should be able to communicate with other similar devices.

- If a device sends data when all the other devices are in sleep mode, the data goes lost.



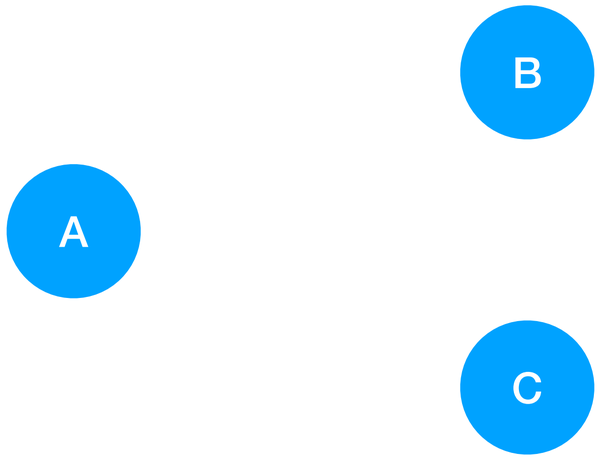
Idea: synchronise sleep/wake times

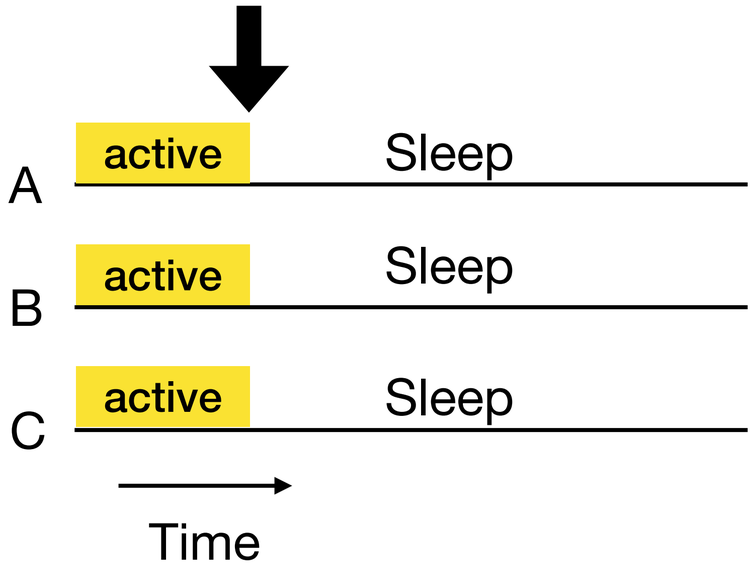




- Nodes agree on “active” time (when data can be exchanged) and “sleep” time (highly configurable parameters)
- Problems: Clock drifts, delays and interferences that interfere with synchronisation messages, topology changes etc

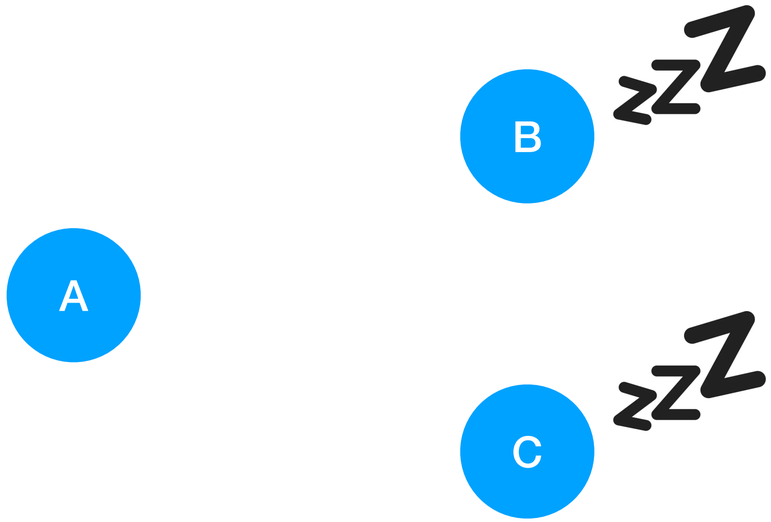
Idea: synchronise sleep/wake times

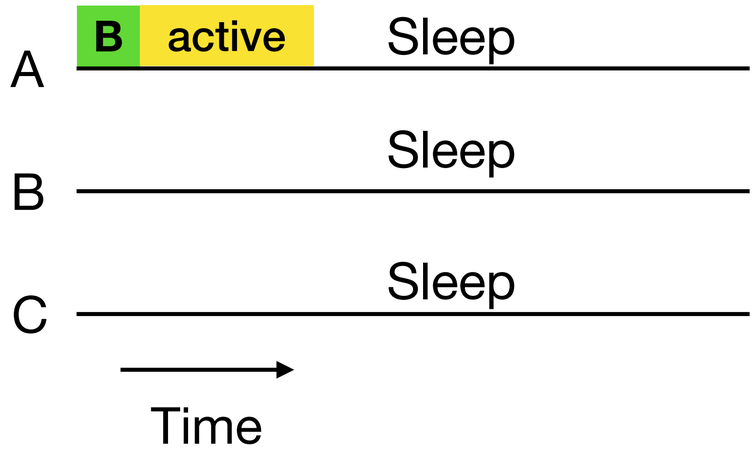




- Nodes agree on “active” time (when data can be exchanged) and “sleep” time (highly configurable parameters)
- Problems: Clock drifts, delays and interferences that interfere with synchronisation messages, topology changes etc

Idea: tell other nodes when you're awake





- Tell other nodes that you are awake by "beacons"
  - protocol: Send a beacon and stay awake for some time after that

- Problems: what if other nodes are asleep when beacon is sent?

MAC-power saving algorithms

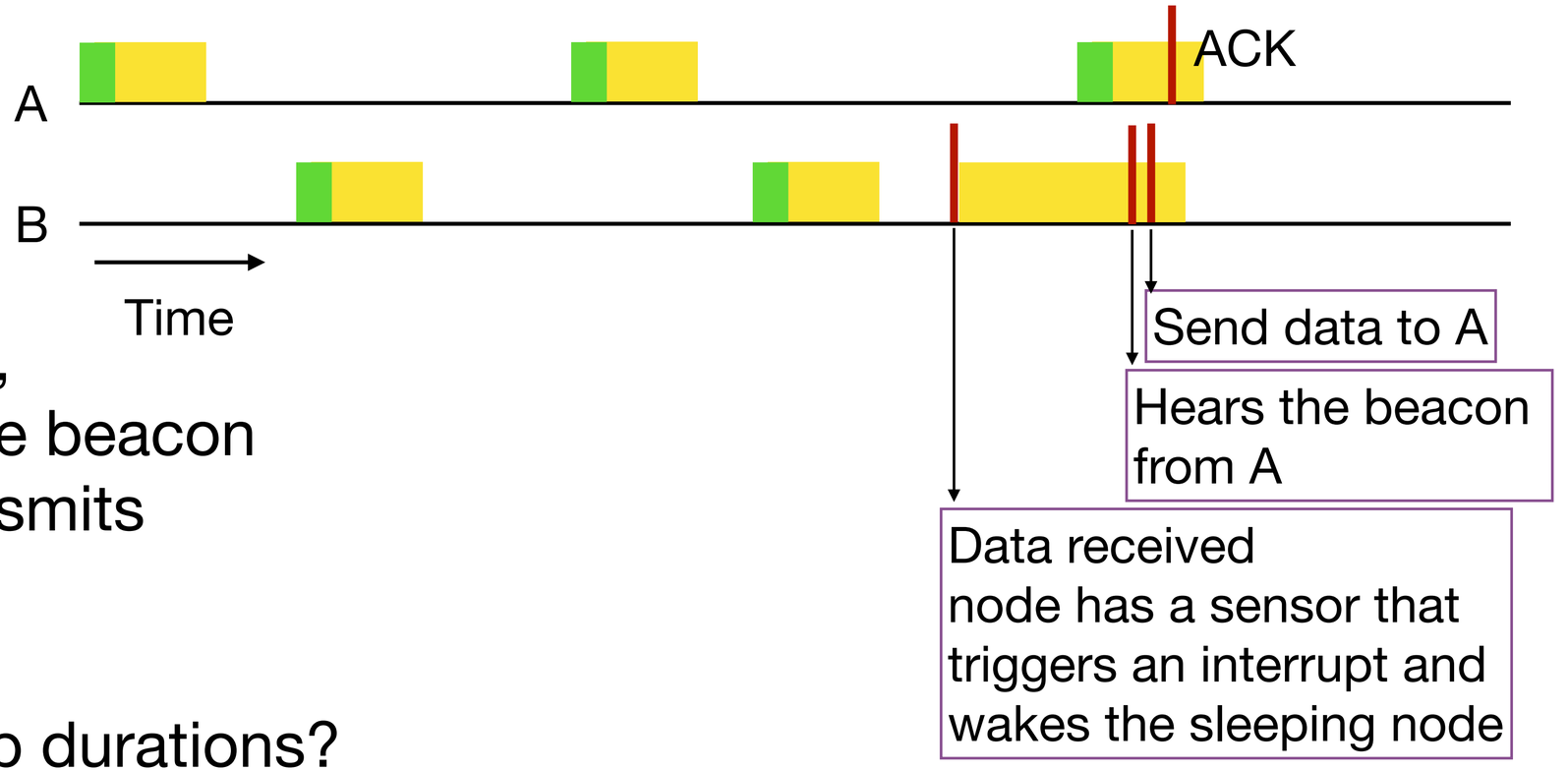
- Power savings IoT algorithms are fundamental for IoT applications relying on WSNs and used by many IoT protocols, as ZigBee and Bluetooth.

- Algorithms can operate in:
  - Beacon mode: Non-beacon tracking (NBT), Beacon Tracking (BT)
  - Non Beacon mode: Long Preamble Emulation (LPE)

Non-Beacon Tracking (NBT)

- **Asynchronous** wakeup algorithm:
  - uses beacons, but does not force nodes to continuously listen for them

- Each node periodically wakes and beacons
  - stays awake for a little after the beacon
  - if a node has data to send, it stays awake till it hears the beacon of the destination, then transmits



- Questions:
  how long to set active/sleep durations?
  How long does a node have to wait in active state?

Non-Beacon Tracking (NBT) cons

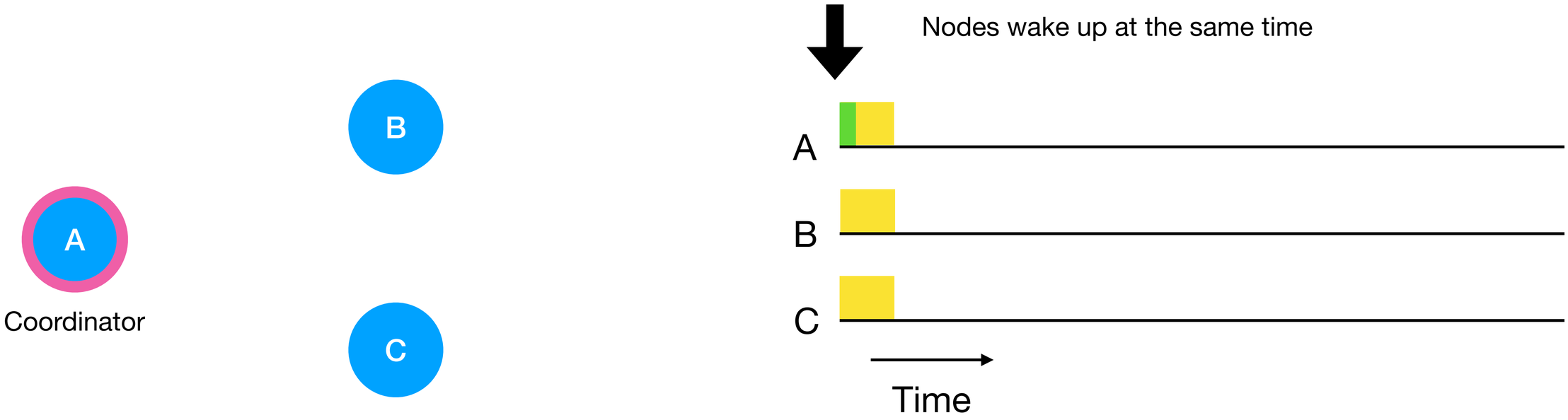
- Beacons might collide
  - if they have an identical schedule, they would collide continuously
  - idea: restart beaconing after random delay if collisions are detected, but collisions might still happen

- Might have to wait long time before transmitting data. For beacon inter arrival time t_{bi}, has to wait $t_{bi}/2$ on average (increased overhead)

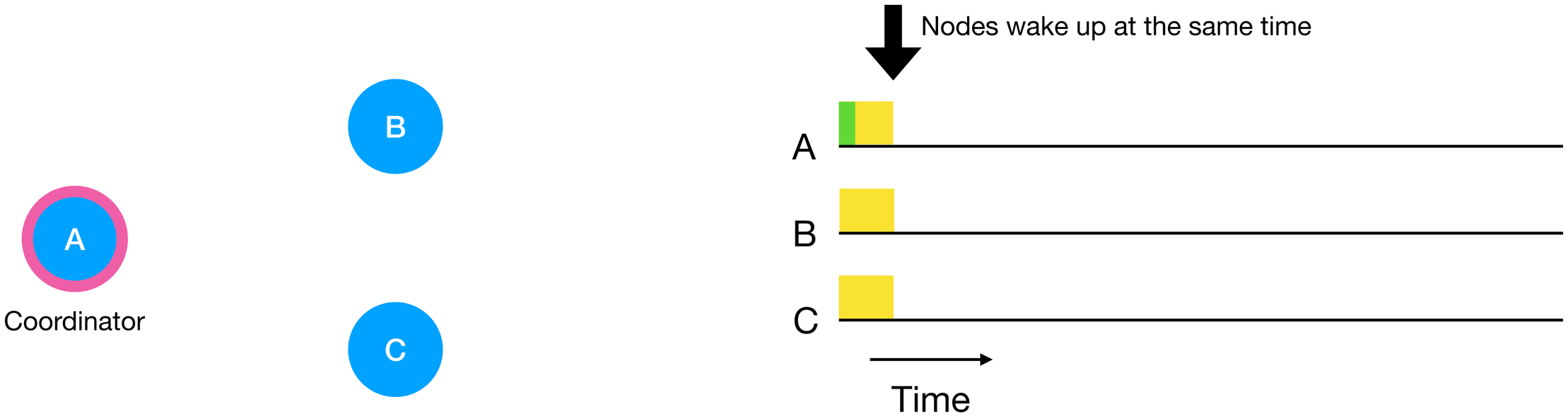
Beacon Tracking (BT) (1)

- **Synchronous** wakeup algorithm using frame structures (i.e., TDMA).
- One node is elected to be the **coordinator** (this choice does not depend on the protocol, could be the most reliable node, or run election algorithms to choose the coordinator).
- The coordinator sends a time frame. The first entry is the beacon (reference burst in TDMA), the rest is used by the coordination to sponsor a sleep and wake schedule that other nodes store and follow.
- During active time, nodes can transmit and receive data. Duration of a frame is between 15ms and 256s.
- Nodes need to synchronise. To avoid missing simultaneous waking due to clock drifts, nodes can wake a little earlier.

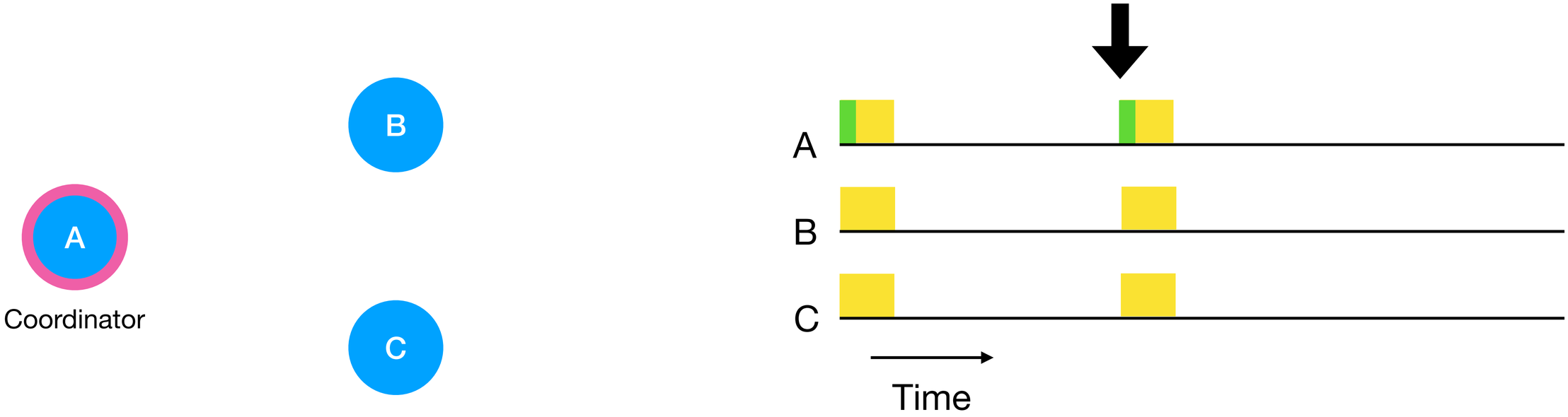
Beacon Tracking (BT) (2)



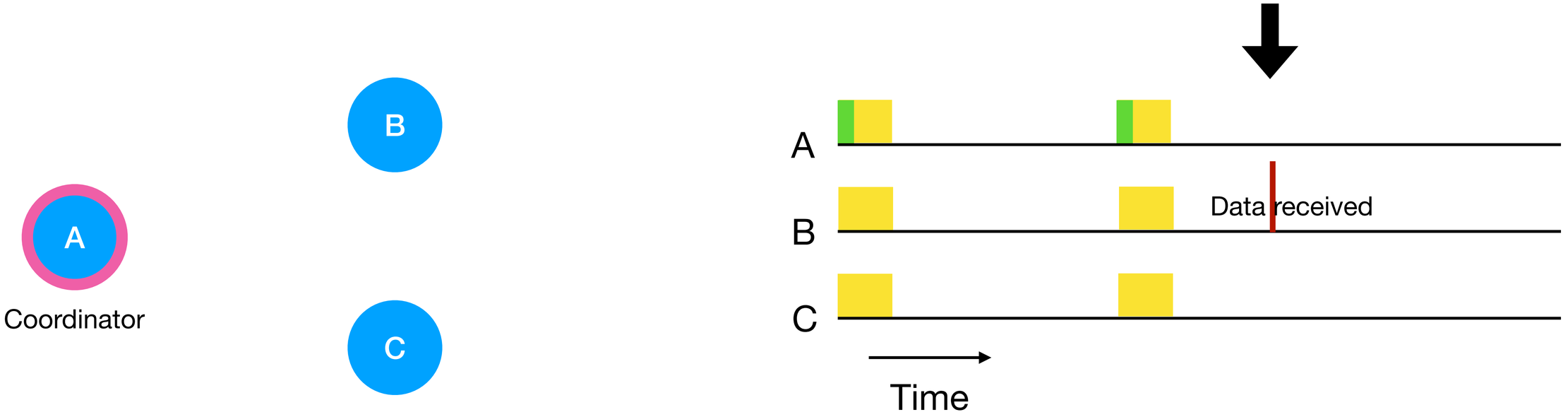
Beacon Tracking (BT) (2)



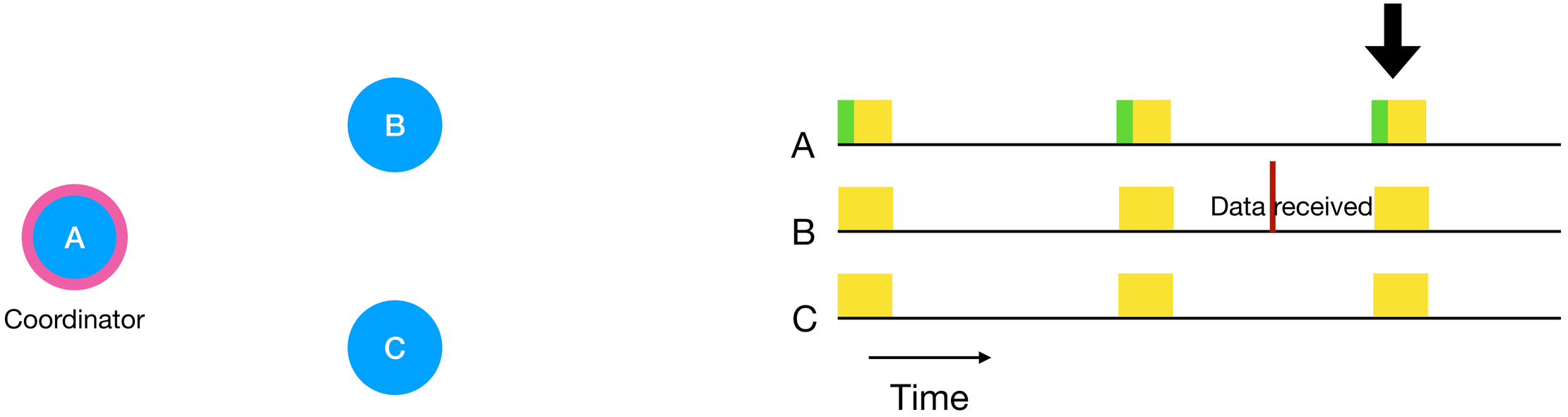
Beacon Tracking (BT) (2)



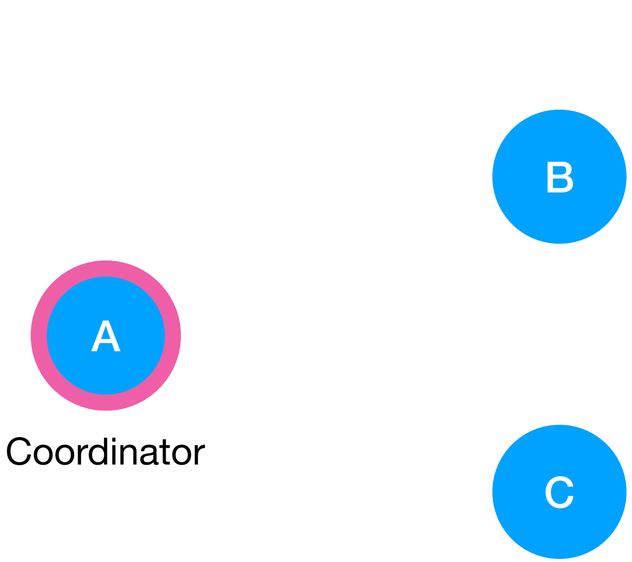
Beacon Tracking (BT) (2)

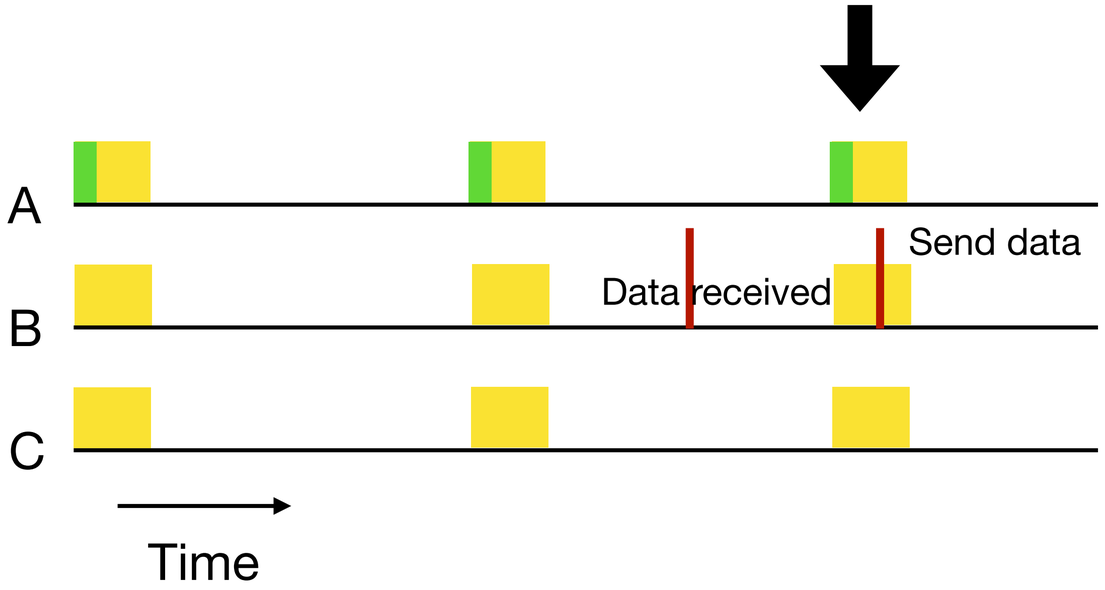


Beacon Tracking (BT) (2)

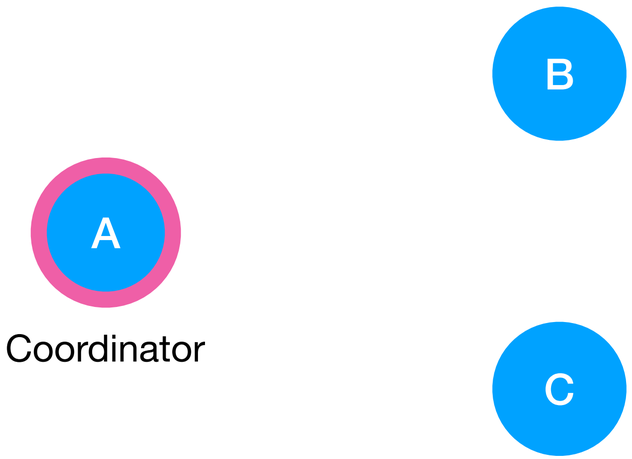


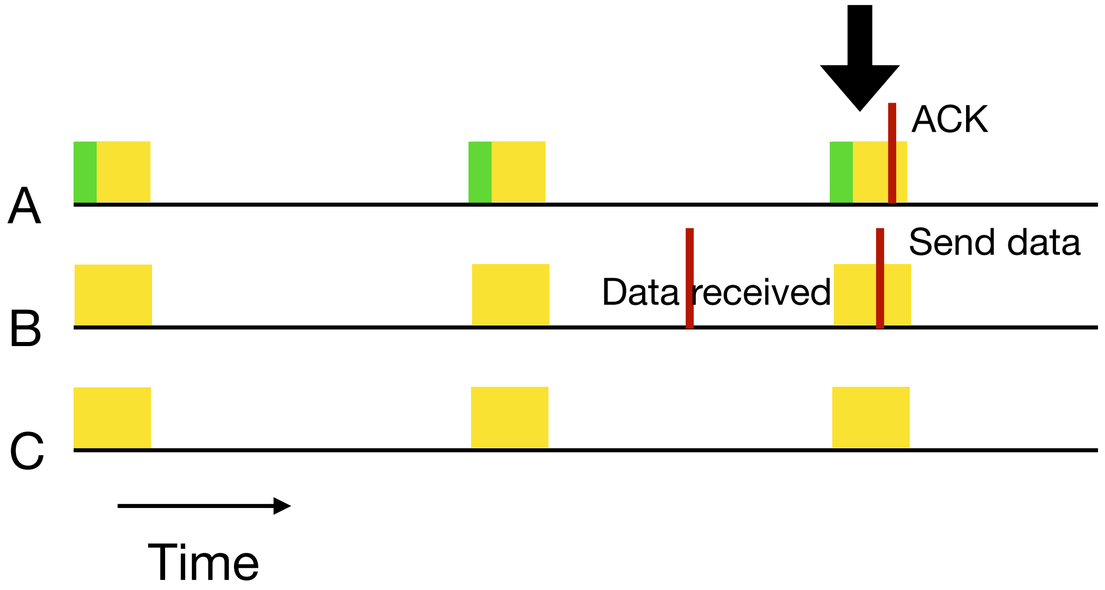
Beacon Tracking (BT) (2)





Beacon Tracking (BT) (2)

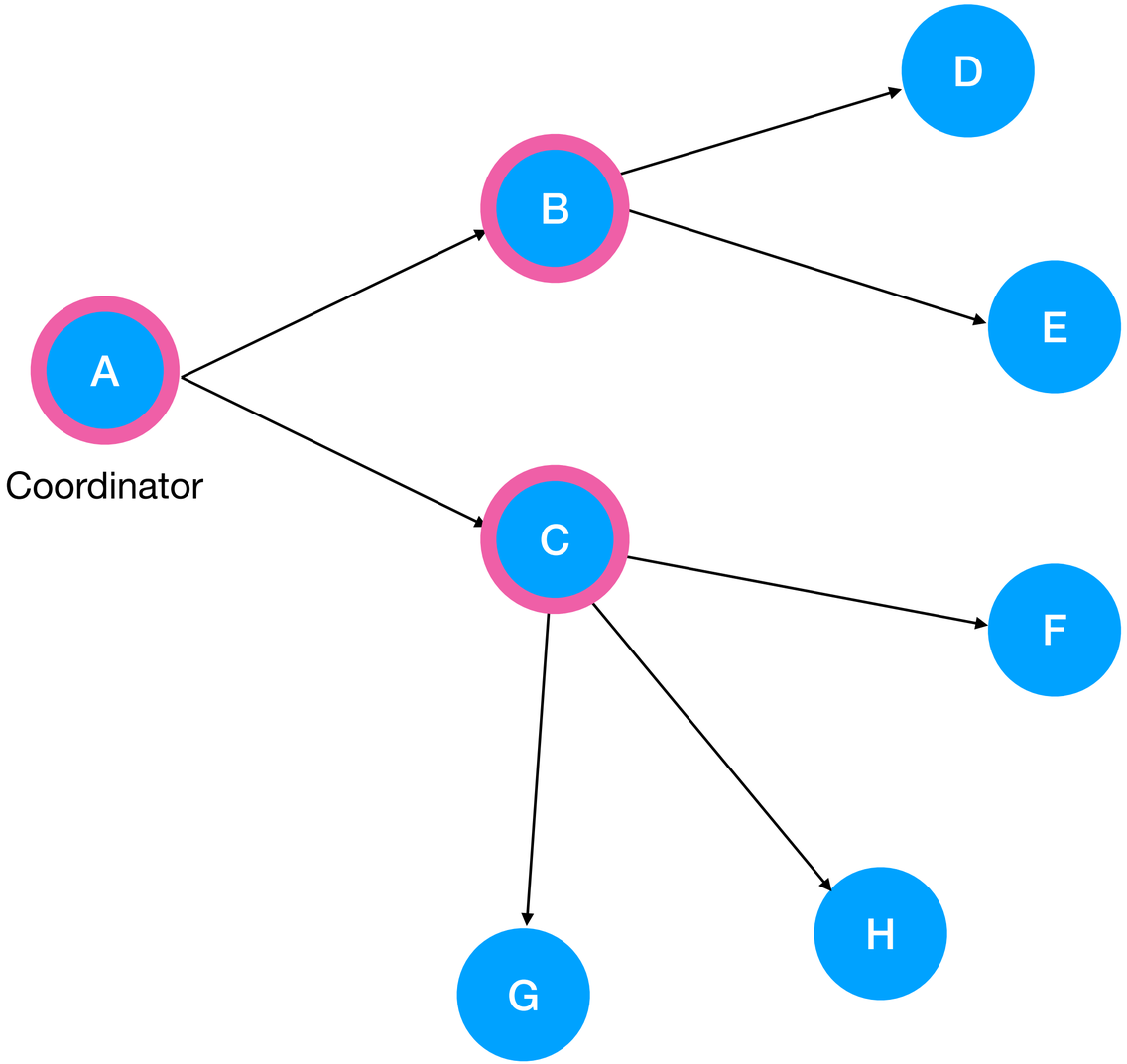




Beacon Tracking (BT) cons

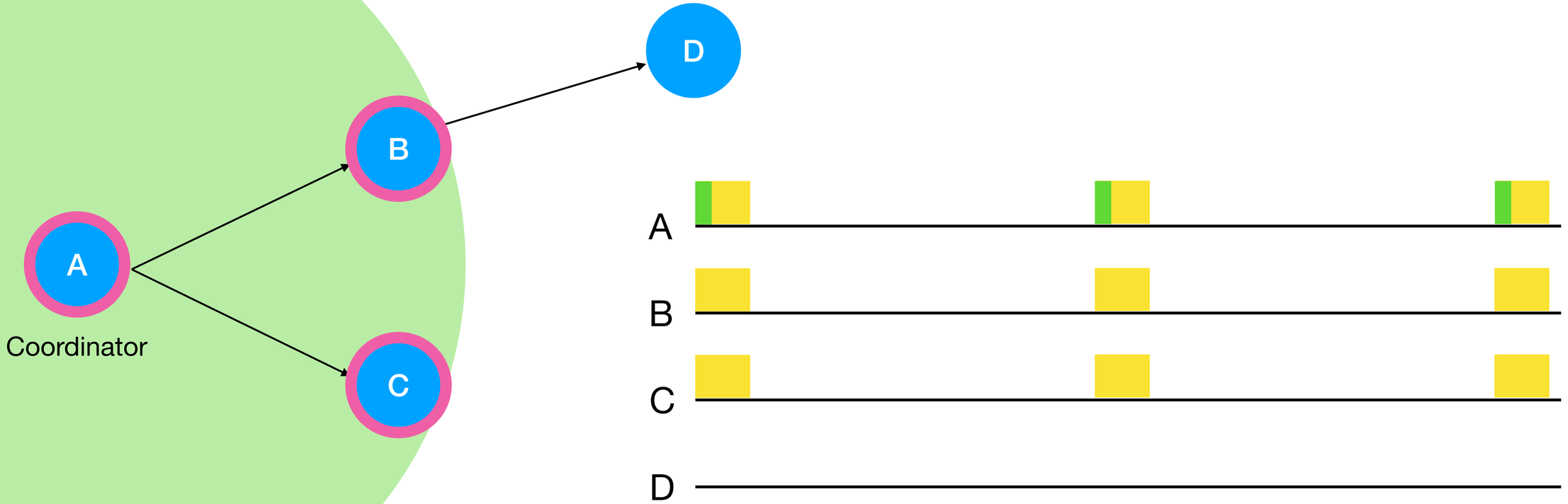
- Synchronisation is hard to achieve due to clock drift. Could buy better clocks but they cost more money.
- Need to have one coordinator, i.e., a single point of failure.
  - What if the coordinator is not within radio range of other nodes and cannot be heard?
    - **Multi hop beacon tracking**

Multi hop Beacon Tracking (1)



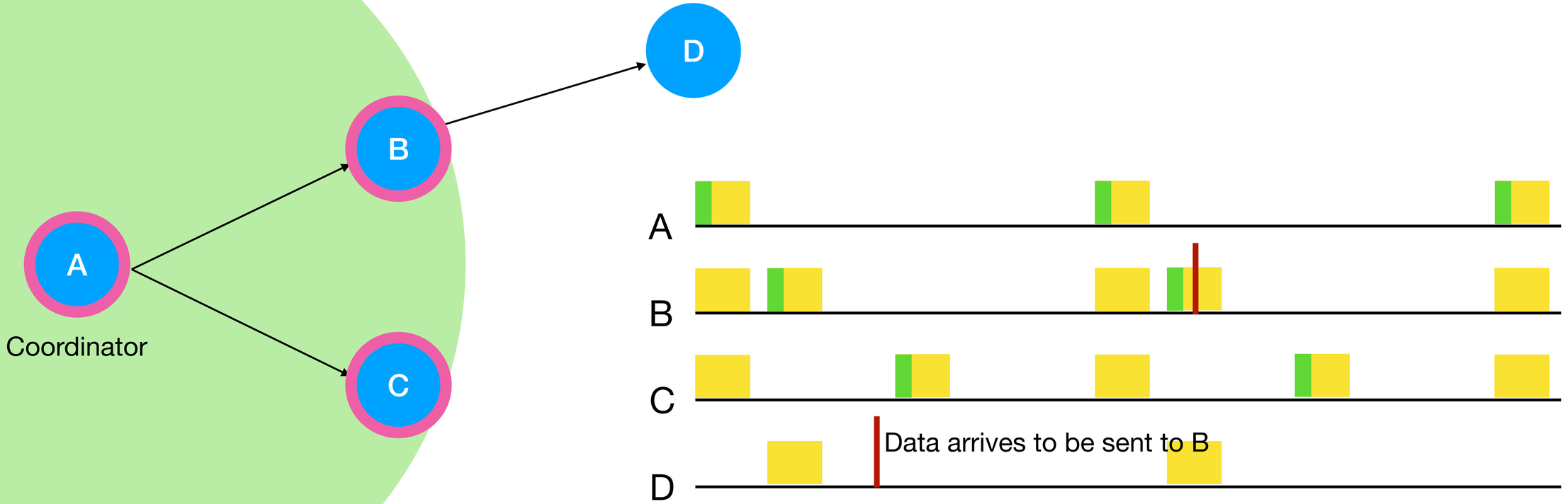
- Other Nodes send beacons too (act as coordinator)
- Nodes wake up for any beacons in their broadcast region
- Nodes form a “associate tree”.
- Nodes wake up for parents in the tree. Root coordinator wakes up for its children

Multi hop Beacon Tracking (2)



- Every time a coordinator wakes up, so do their children

Multi hop Beacon Tracking (2)



- Another coordinator (in a different window time) wakes up and sends out beacons that are received by its children

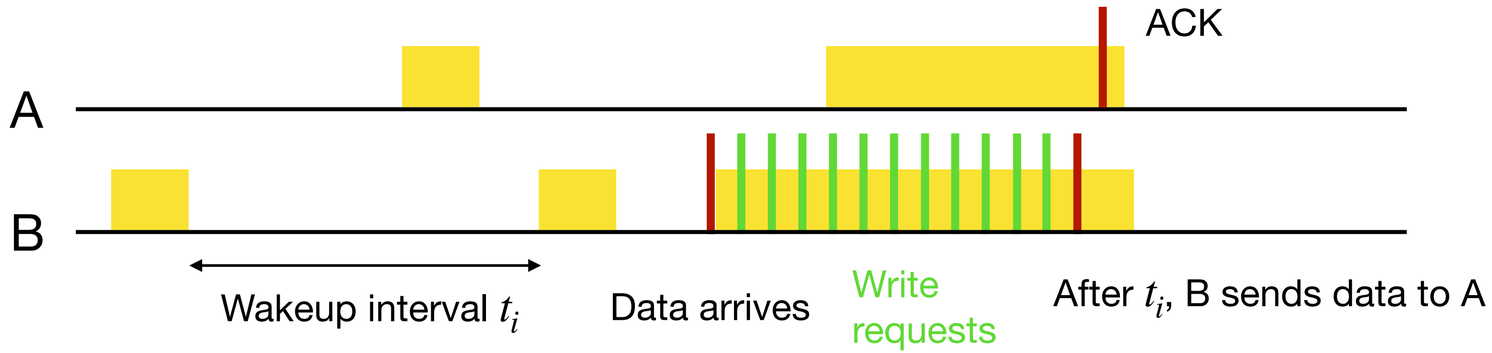
Multi hop Beacon Tracking (BT) cons

- Nodes are still waking up a lot
- Coordinators have to watch out not to collide with frames from other coordinators. This introduces complexity.

Long Preamble Emulation (LPE) (1)

- Also known as B-MAC
- Non-beacon mode and asynchronous (nodes do not send beacons at all)
- Nodes wake up on regular intervals of time t_i
  - Intervals are unsynchronised but regular and known
- Uses “write request” messages until receiver would wake up, then sends data

Long Preamble Emulation (LPE) (2)



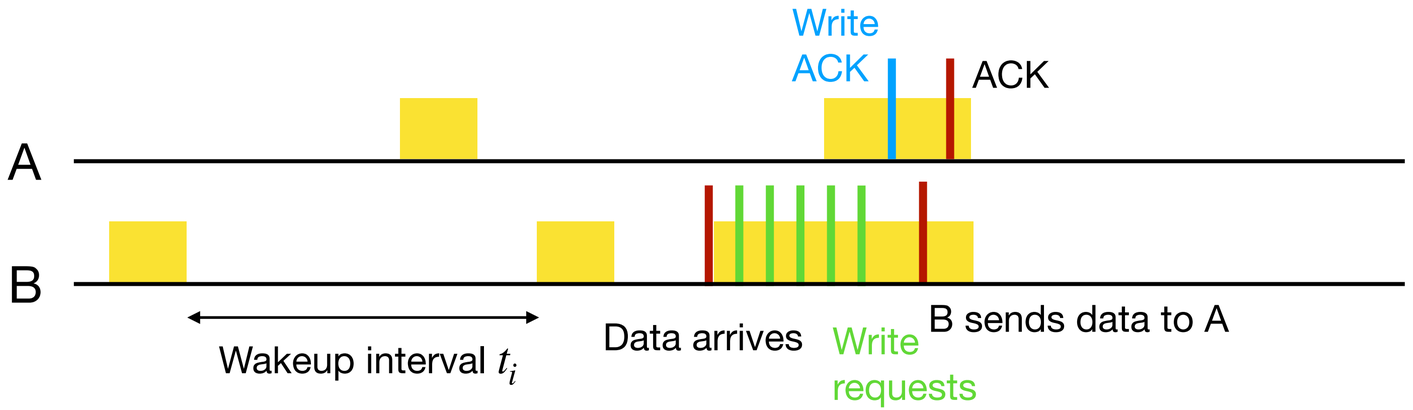
- Nodes wake up at regular time intervals of duration t_i.
- When B receives data, it stays awake, sends Write Requests continuously to A (does not know when A wakes up). In the meantime, A wakes up and stays up as it receives the requests from B
- After time t_i from when it received the data, B knows that A must be awake, so it transmits data and A replies with an ACK

Long Preamble Emulation with Acknowledgement (1)

- Long Preamble Emulation is good because it does not need node synchronisation, but when a node receives data, its radio stays awake for a long time before it transmit data to other nodes (waste of power).

- Simple solution: Long Preamble Emulation with Acknowledgement, acknowledge write requests (introduces some complexity).

Long Preamble Emulation with Acknowledgement (2)

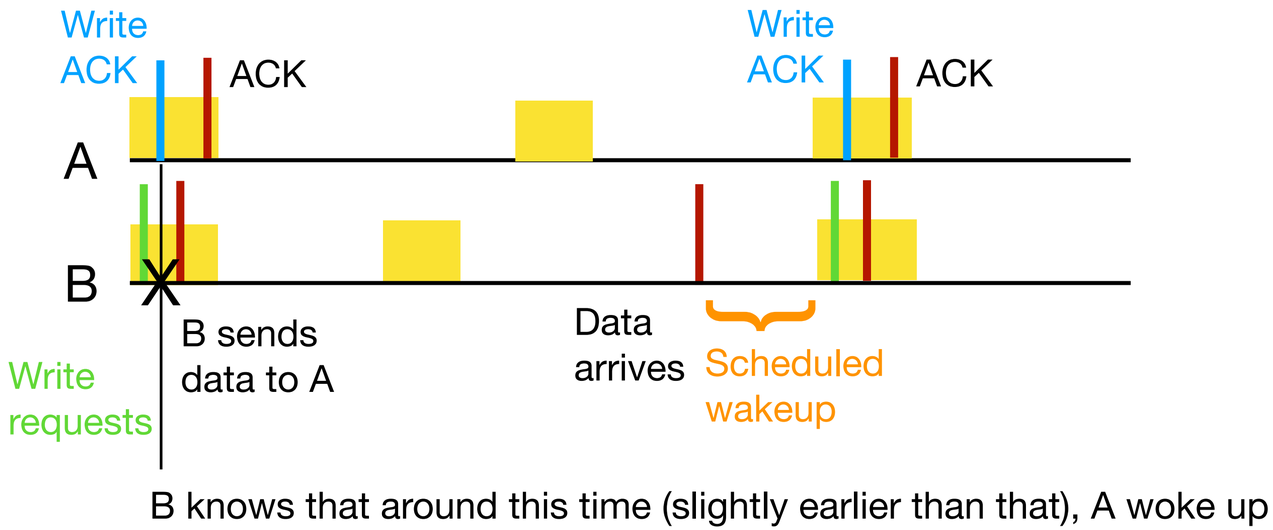


- Nodes wake up at regular time intervals of duration t_i.
- When B receives data, it stays awake, sends Write Requests continuously to A (does not know when A wakes up).
- When A wakes up and receives WRs, it replies with a Write ACK, then B transmits data to A, which responds with an ACK and they go back to sleep

Long Preamble Emulation with Acknowledgement after Local Synchronization (1)

- Long Preamble Emulation with Acknowledgement is more efficient, but the sender still has to wait for A to wake up before being able to send data
- This is because B does not know when A wakes up… or does it?
- A and B must have sent data to each other before, and so B must have a little bit of information of when A wakes up.
- Long Preamble Emulation with Acknowledgement after Synchronization leverages this information (more efficient and complex)

Long Preamble Emulation with Acknowledgement after Local Synchronization (2)



- If A was awake at time t, it must be awake at time $2t, 3t$ etc.
- When B receives data and it knows that A is not awake, it schedules a wakeup for itself until the expected wake up time of A.

5.2.4 MAC protocols for WSNs

Contention-based and contention-free

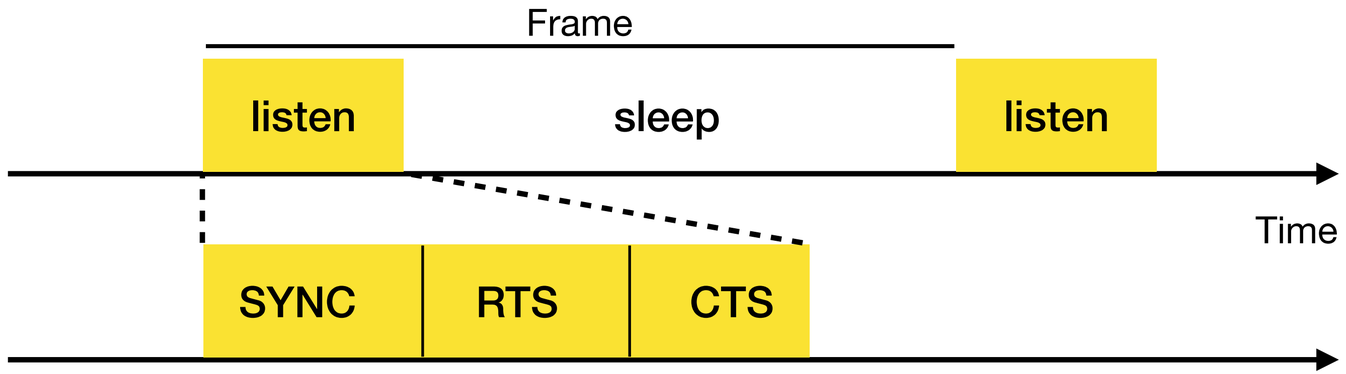
- MAC protocols for WSNs can be:
  - Contention-based
    If they use Multiple Access approaches where devices compete for access to the channel (e.g., ALOHA, CSMA, MACA)
    - Sensor MAC (S-MAC)
  - Contention-free
    If nodes/coordinator manage access to avoid collisions (TDMA, FDMA, CDMA)
    - Traffic Adaptive Medium Access (TRAMA)

Sensor MAC (S-MAC) (1)

- Sensor-MAC (S-MAC) protocol is an energy efficient protocol specifically designed for WSNs.
- **Sensor network scenario:**
  - most communication occurs between nodes as **peers**, rather than to a single base station.
  - Suitable for applications that are latency-tolerant.
- Main goal: improve energy efficiency while maintaining good scalability and collision avoidance.

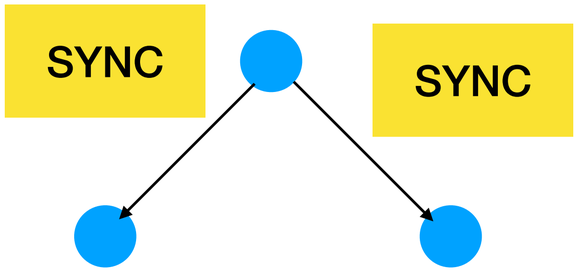
Sensor MAC (S-MAC) - frame

- Periodic listen and sleep mechanism to establish a low-duty-cycle operation on each node.
- Frame: complete cycle of listen and sleep periods.
- Frame begins with a Listen period, further divided into smaller intervals for sending or receiving SYNC, RTS, CTS packets.



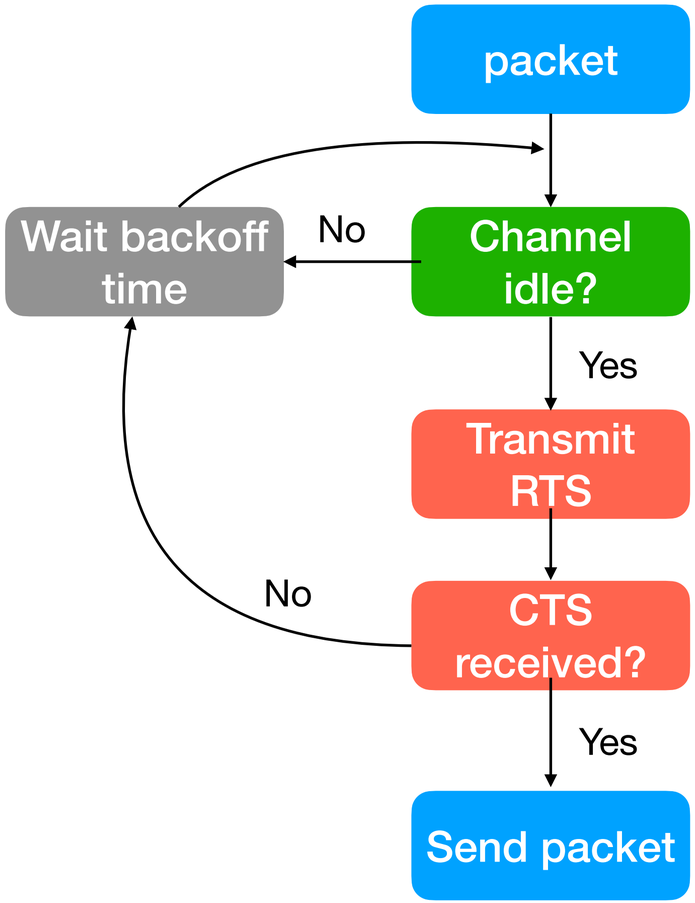
Sensor MAC (S-MAC) - neighbour coordination

- All nodes are free to choose their own listen and sleep schedules.
- To reduce control overhead, however, neighboring nodes coordinate their sleep schedules and try to adopt the same schedules, rather than randomly sleep on their own.
- To establish coordinated or synchronized sleep schedules, each node exchanges its schedule with all its immediate neighbours by periodically broadcasting a SYNC packet.



Sensor MAC (S-MAC) - collision avoidance

- Uses Carrier Sense with collision avoidance, both physical and virtual.
- Physical CS: sense the channel, if energy above a certain threshold, the channel is deemed busy, idle otherwise.
- Virtual CS: uses RTS/CTS mechanism as Virtual Carrier sense mechanism.
- In virtual CS, each transmitted packet carries a **duration field** that indicates the duration of transmission.
- If a node hears a packet destined to another node, it knows how long it needs to keep silent. It stores this information in a variables (**Network allocation vector, NAV**), sets a time for it and goes to sleep.
- If the node wakes up and has data to send, it checks the NAV value.



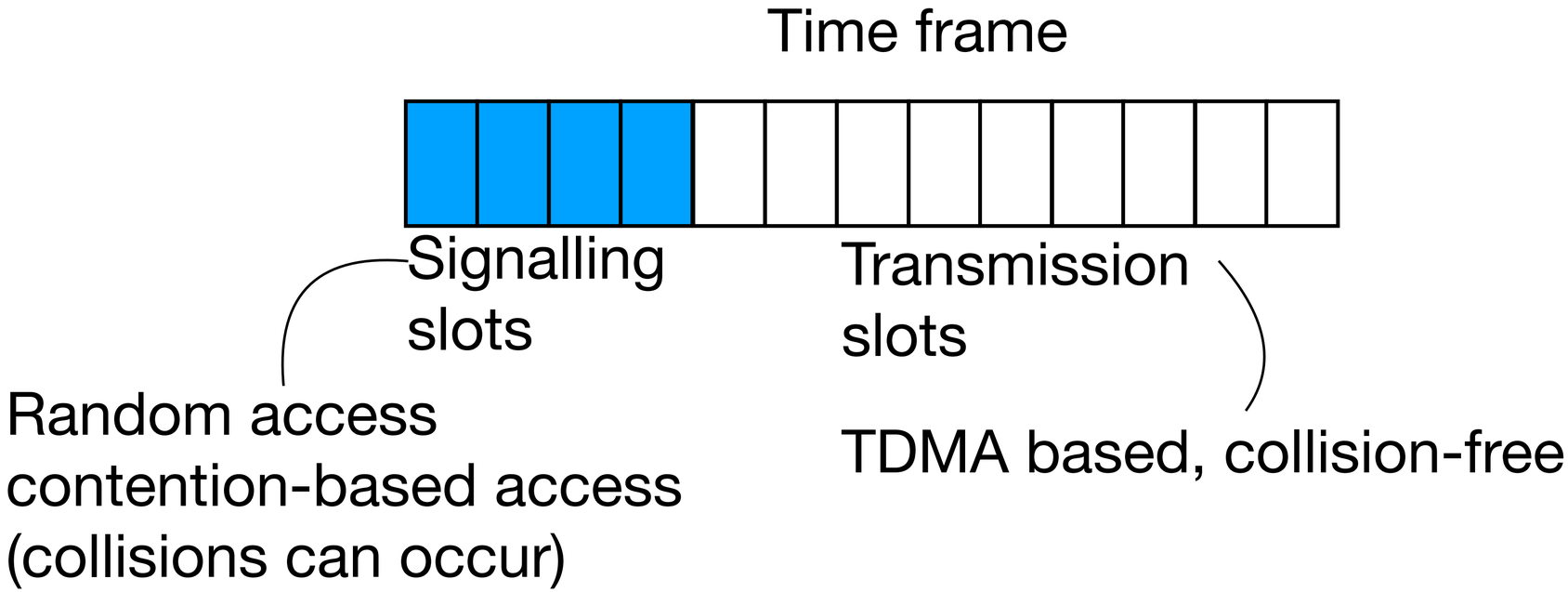
Traffic Adaptive Medium Access (TRAMA) (1)

- TRAMA employs a **traffic adaptive distributed election scheme** to decide transmission schedules**.**
- TRAMA consists of three components:
  - Neighbour Protocol (NP).
  - Schedule Exchange Protocol (SEP).

  allow nodes to exchange **two-hop neighbour** information and their schedules
  - Adaptive Election Algorithm (AEA), which uses neighbourhood and schedule information to select the transmitters and receivers for the current time time.

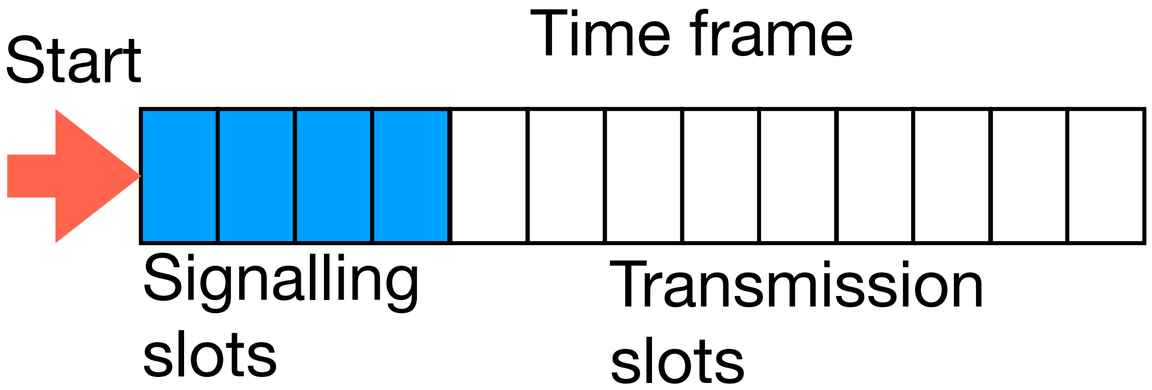
Traffic Adaptive Medium Access (TRAMA) (2)

- TRAMA assumes a single, time-slotted channel for both data and signaling transmissions.



Traffic Adaptive Medium Access (TRAMA) (2)

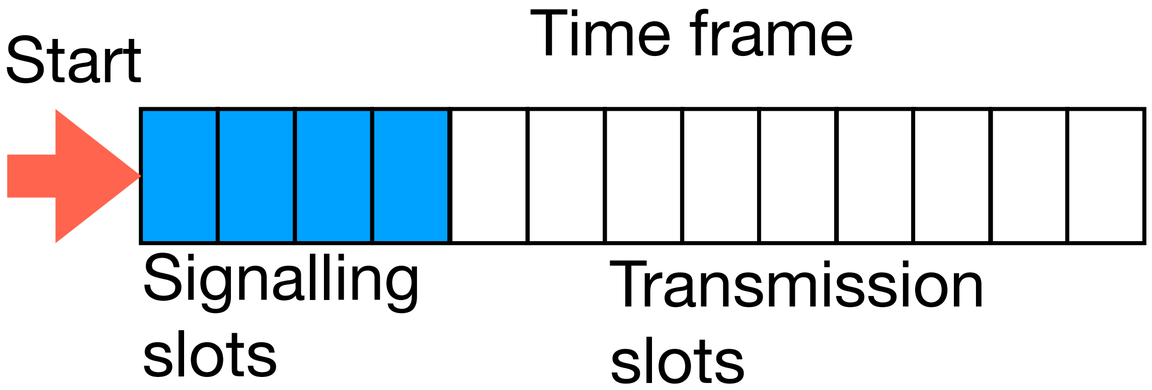
- TRAMA assumes a single, time-slotted channel for both data and signaling transmissions.



- Each node transmits by selecting a slot randomly
- Nodes can access the network during this period
- All nodes must be in transmitting or listening mode
- Nodes share neighbourhood information
- Nodes share schedules for transmission slots
- Time syncronization

Traffic Adaptive Medium Access (TRAMA) (2)

- TRAMA assumes a single, time-slotted channel for b
signaling transmissions.



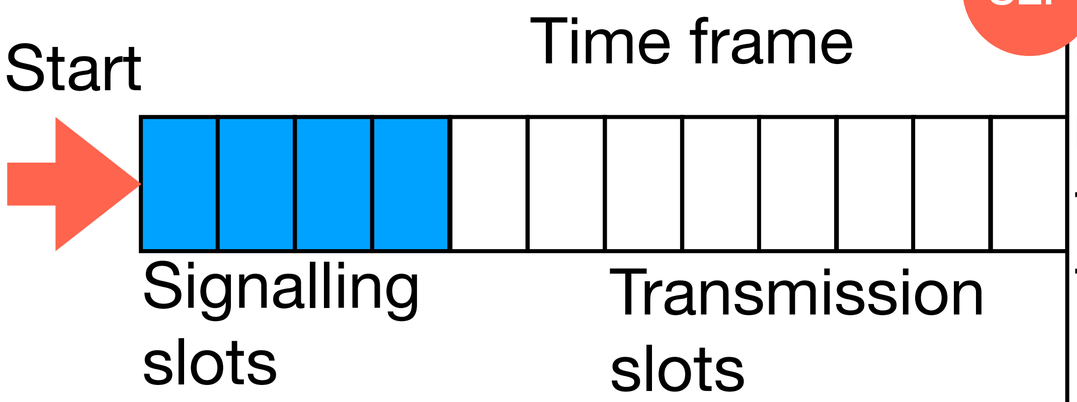
- Each node transmits by selecting a slot randomly
- Nodes can access the network during this period
- All nodes must be in transmitting or listening mode
- Nodes share neighbourhood information
- Nodes share schedules for transmission slots
- Time syncronization

NP

-Signaling packets carry incremental 1-hop neighbour updates.
-If no updates, signaling packets are sent as “keep-alive” beacons.
- A node times out a neighbour if it does not hear from it for a certain amount of time.
- knowing 1-hop neighbours of a node’s neighbours allows 2-hop neighbourhood knowledge.

Traffic Adaptive Medium Access (TRAMA) (2)

- TRAMA assumes a single, time-slotted channel for both data and signaling transmissions.



- Each node transmits by selecting a slot randomly
- Nodes can access the network during this period
- All nodes must be in transmitting or listening mode
- Nodes share neighbourhood information
- Nodes share schedules for transmission slots
- Time syncronization

SEP

-Nodes share schedule messages containing their traffic needs (who they wants to talk to, and when).
-Nodes listen to their neighbours' schedules and build a conflict-free schedule using a priority function
- Conflicts (multiple nodes wanting the same slot) are resolved using the Adaptive Election Algorithm

Adaptive Election Algorithm - TRAMA

- Each transmission slot can be used by only one node in a two-hop neighbourhood to avoid interferences.
- The priority of a node for a given time slots is computed as an hash function that is public and deterministic.
- All nodes compute the same priority values for their 2-hop neighbours, so they all agree on who has the highest priority for a given slot.
- Only nodes that have data packets to send are eligible for having a time slot allocated.
- The node with the highest priority wins the slot and gets to transmit. Other nodes go to sleep or prepare to receive if they are the intended receiver.

5.2.4 Association and neighbour discovery in IEEE 802.11 (WiFi)

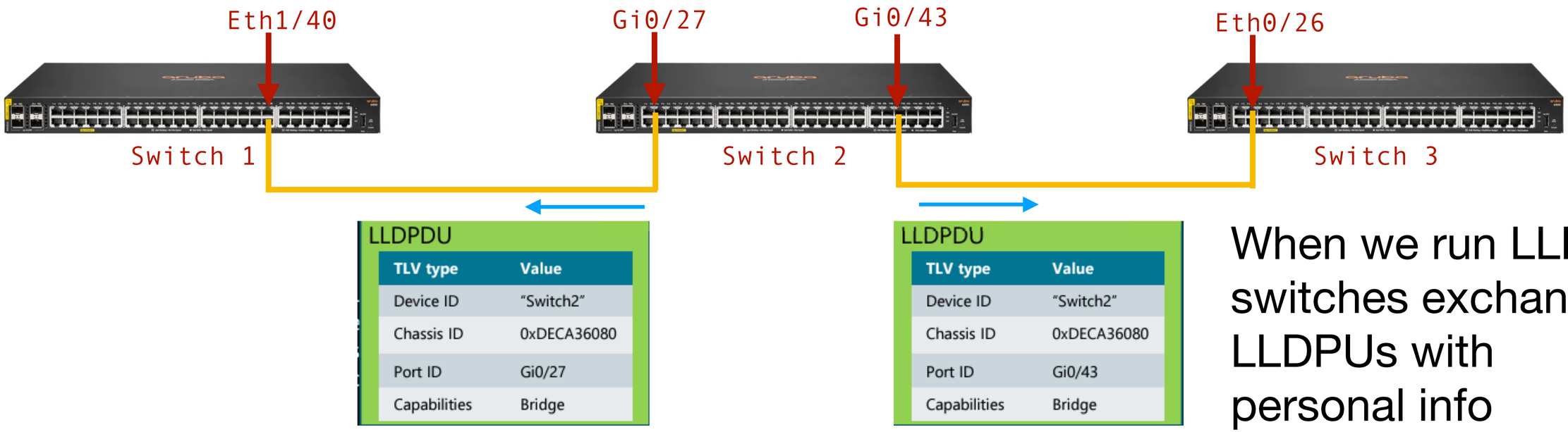
Neighbour discovery in Wired Networks: Link-Layer Discovery Protocol (LLDP) (1)

- LLDP is protocol used by network devices for advertising their identity, capabilities, and neighbours on a local area network connected through Ethernet.

- Periodically (e.g., every 30 seconds) switches/routers exchange LLDP messages with their physical neighbours.
  - similar to hello messages, used for “introducing” themselves, contain records as chassis ID, system name, system description etc.

- Common Cisco-Proprietary variant: Cisco Discovery protocol (CDP)

Neighbour discovery in Wired Networks: Link-Layer Discovery Protocol (LLDP) (2)



When we run LLDP, switches exchange LLDPUs with personal info

- Example: switches provided with ethernet/fiber ports (Eth/fa/Gi), forward packets at high speed. Can log into them and configure them.

Neighbour discovery in Wired Networks: Link-Layer Discovery Protocol (LLDP) (2)


```
switch# show lldp neighbor
Capability codes:
  (R) Router, (B) Bridge, (T) Telephone, (C) DOCSIS Cable Device
  (W) WLAN Access Point, (P) Repeater, (S) Station, (O) Other
Device ID       Local Intf     Chassis ID         Port ID         Hold-time    Capability
Switch2         Eth1/40        000d.eca3.6080     Gi0/27          120          B
switch#
```

If we log into a switch and type `show lldp neighbor` the switch provides us the information gained through LLDP by other neighbours

- Example: switches provided with ethernet/fiber ports (Eth/fa/Gi), forward packets at high speed. Can log into them and configure them.

Can LLDP be used to discover neighbours in wireless networks?

- Wireless has additional challenges
  - may have equivalent access points
  - may need to discover “closest” neighbours
  - neighbours may lie about who they are

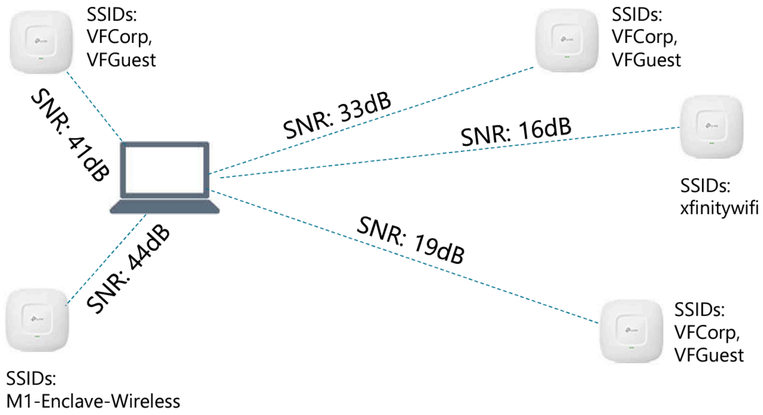
- Wireless protocols have been developed to solve these challenges:
  - association, discovery, authentication

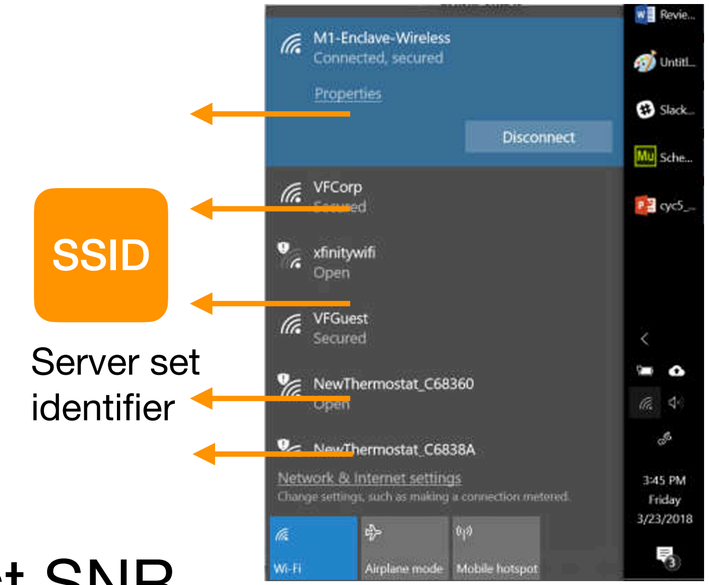
Scanning

Active scanning:
NIC sends probe request frames and waits for probe responses from APs

Passive scanning:
listens for beacon frames from nearby APs.

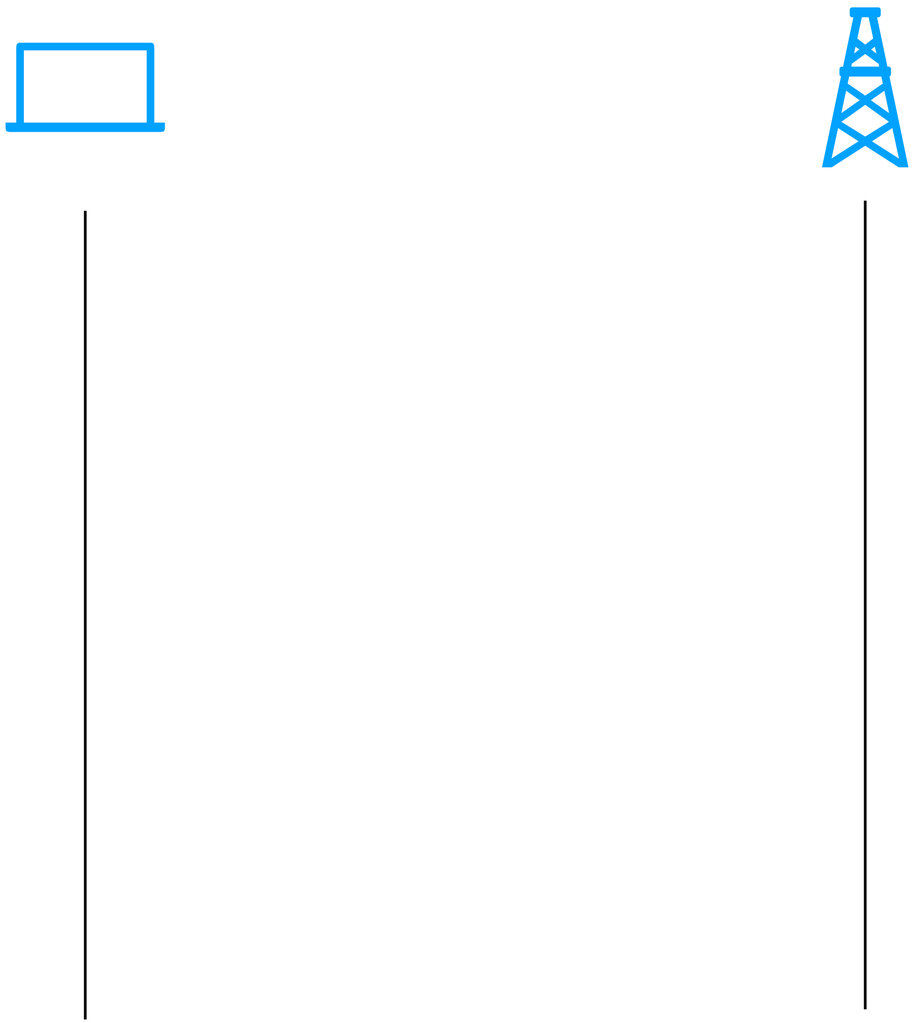
Access points periodically (default 100ms) send beacons with personal information (supported rates, transmission parameters) and what SSID they support (e.g., eduroam)



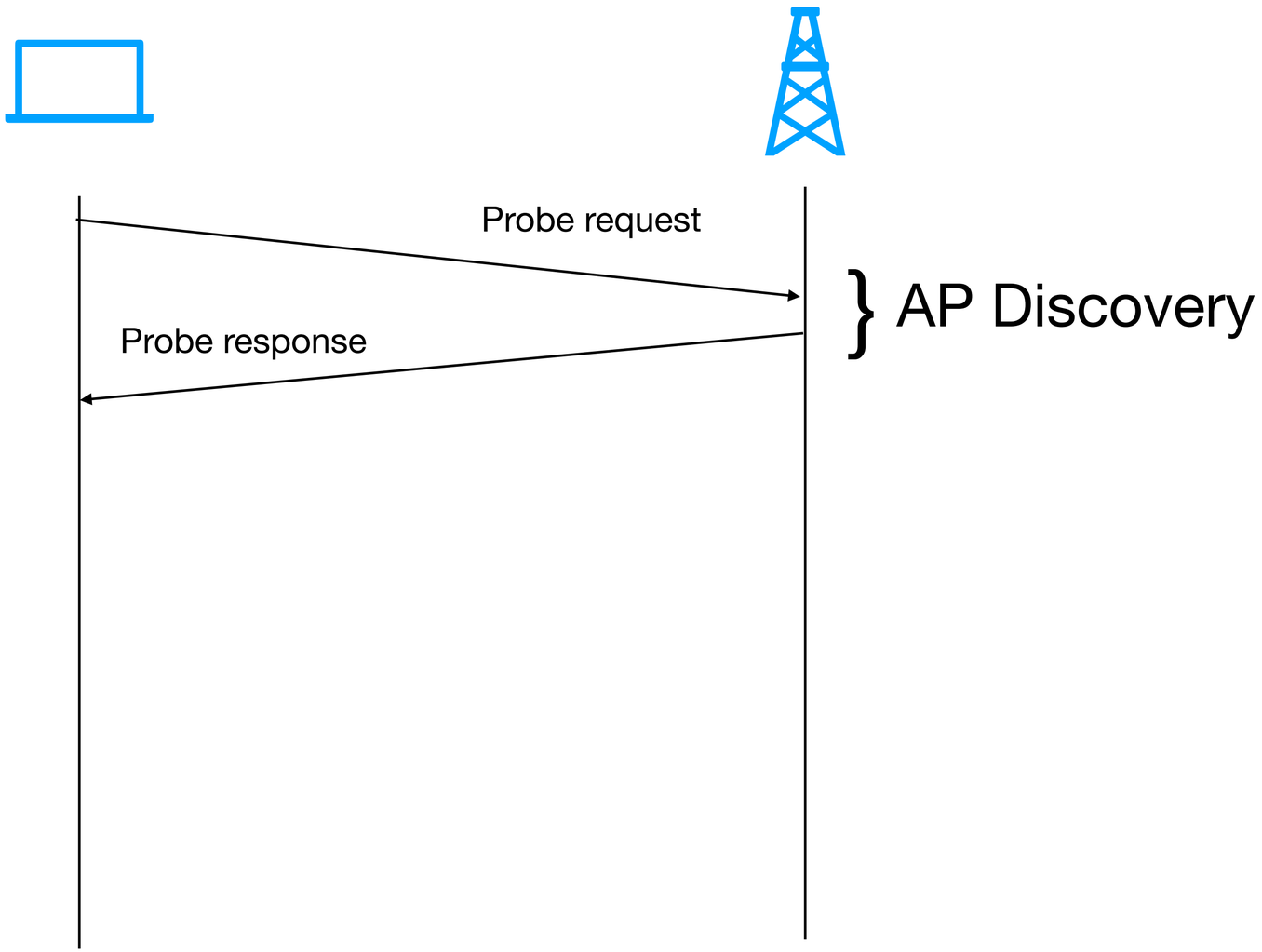


What AP to choose? Select it manually, or the ones with highest SNR

AP Association process

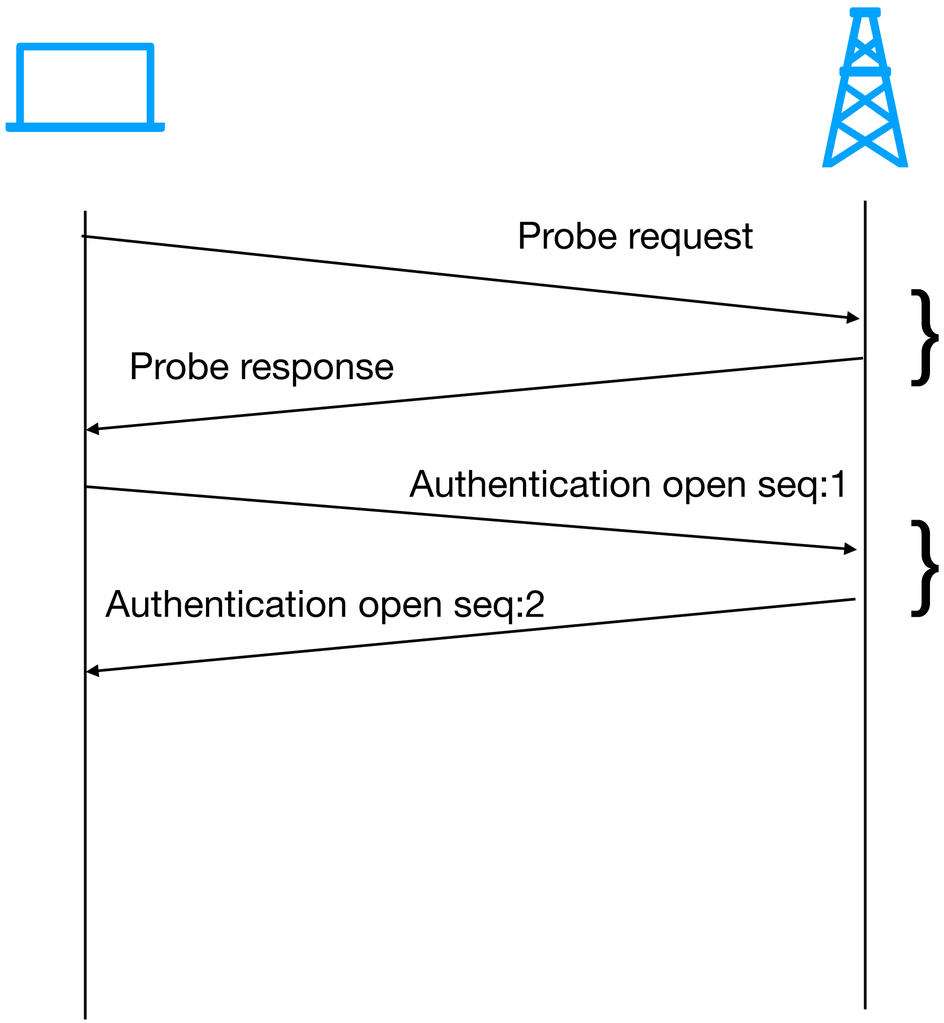


AP Association process



The devices scans the APs and gets information. It chooses one AP, e.g., the one with better signal to noise ratio

AP Association process



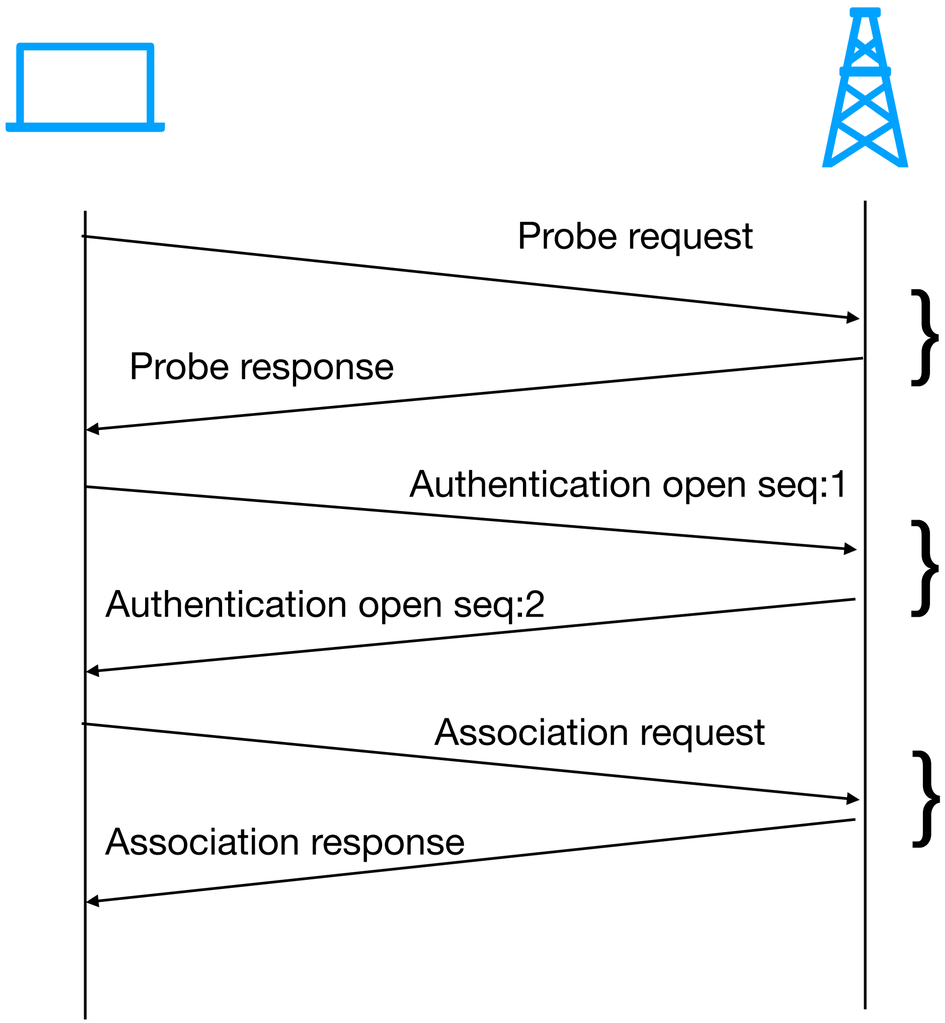
} AP Discovery

The devices scans the APs and gets information. It chooses one AP, e.g., the one with better signal to noise ratio

} Authentication

The devices “introduces” itself to the access point and convinces it that it is allowed to connect to it, by presenting valid credentials and keys. The access point validates that and tells the device whether it can access

AP Association process



} AP Discovery

The devices scans the APs and gets information. It chooses one AP, e.g., the one with better signal to noise ratio

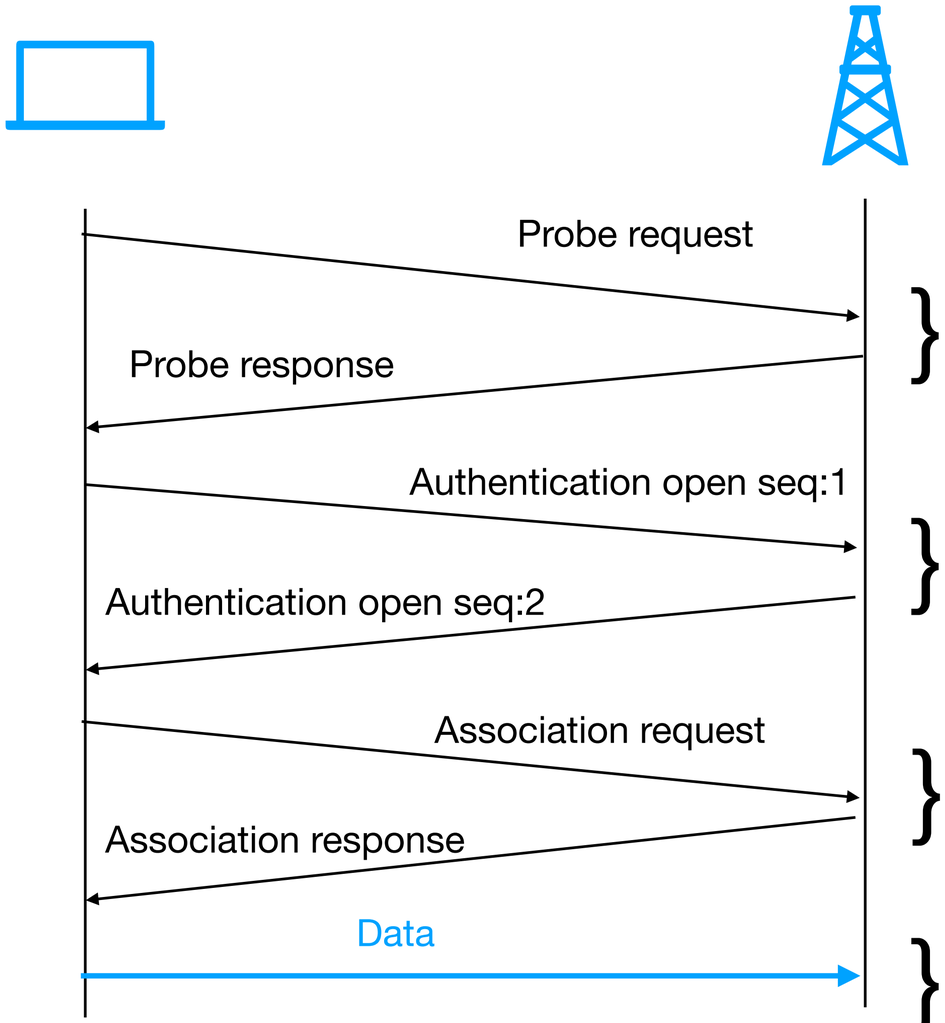
} Authentication

The devices “introduces” itself to the access point and convinces it that it is allowed to connect to it, by presenting valid credentials and keys. The access point validates that and tells the device whether it can access

} Association

The device requests to connect to the AP. The access point can accept the association or reject it, if for instance it is already serving too many users

AP Association process



} AP Discovery

The devices scans the APs and gets information. It chooses one AP, e.g., the one with better signal to noise ratio

} Authentication

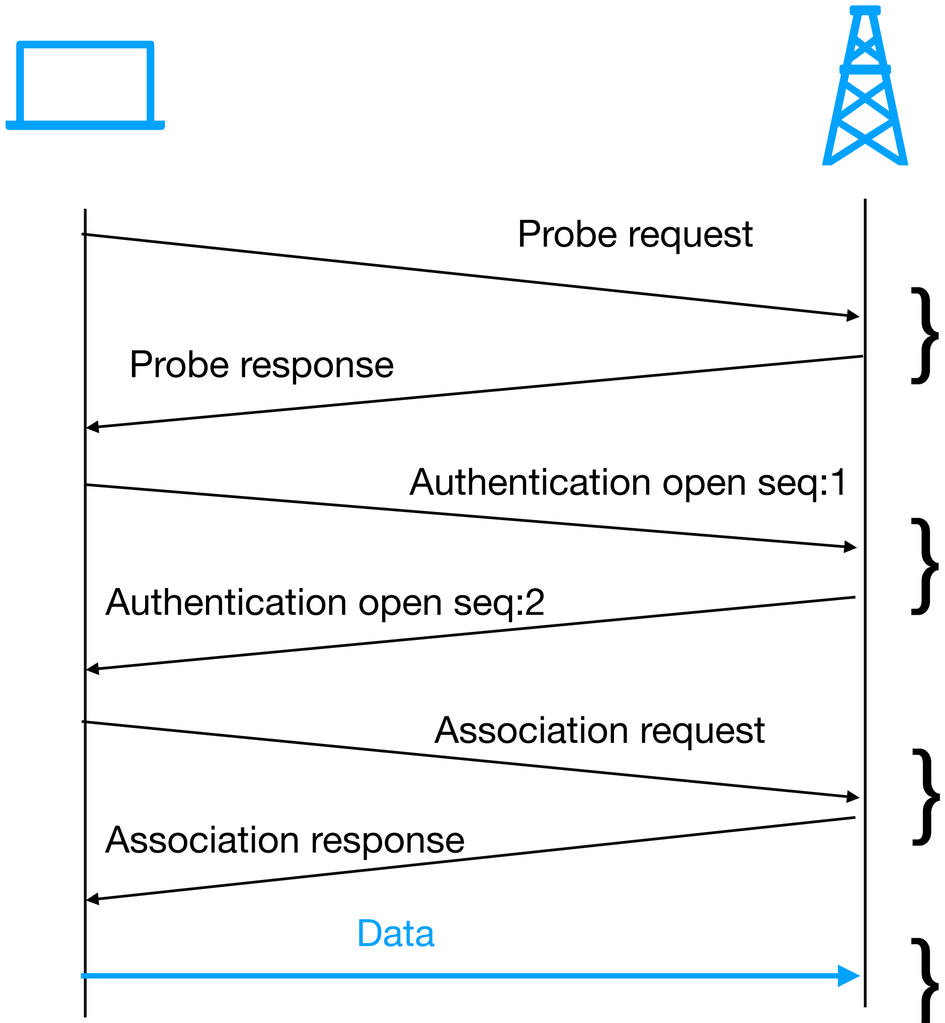
The devices “introduces” itself to the access point and convinces it that it is allowed to connect to it, by presenting valid credentials and keys. The access point validates that and tells the device whether it can access

} Association

The device requests to connect to the AP. The access point can accept the association or reject it, if for instance it is already serving too many users

} Data transmission

AP Association process



} AP Discovery

} Authentication

} Association

} Data transmission

This process is repeated during the first association process, but it is also repeated periodically, to monitor signal strength. This way allows mobility

Bibliography

- Abbas Jamalipour, WIRELESS SENSOR NETWORKS, A Networking Perspective. IEEE, WILEY 2009.

- Rajendran, V., Obraczka, K., & Garcia-Luna-Aceves, J. J. (2003, November). Energy-efficient collision-free medium access control for wireless sensor networks. In Proceedings of the 1st international conference on Embedded networked sensor systems (pp. 181-192).

- IoT Communication, Illinois Urbana-Champaign

- Pejman Roshan et al. “802.11 Wireless LAN Fundamentals” Cisco Press 2023

- Kurose, Ross. COMPUTER NETWORKING A Top-Down Approach. 8th edition, Pearson.