

CS/ECE 439: Wireless Networking

MAC Layer – Power!

Energy Conservation Techniques

- ▶ Wi-Fi devices consume significant amounts of energy when idle
 - ▶ Idle $> 1W$
- ▶ Conservation Approach: Device suspension (sleep)
 - ▶ Reduced energy consumption
 - ▶ Sleep $\sim 0.05W$
 - ▶ Suspended communication capabilities
 - ▶ Buffer overflow
 - ▶ Wasted bandwidth
 - ▶ Lost messages
 - ▶ If all nodes are asleep, no one can communicate!



Communication Device Suspension

▶ Goal

- ▶ Remain awake when there is active communication
- ▶ Otherwise, suspend
- ▶ Adapt the sleep duration to reflect the communication patterns of the application

▶ Ideal

- ▶ Sleep whenever there is no data to receive from the base station
- ▶ Wake up for any incoming receptions



Communication Device Suspension

► Problems

- How can a sender differentiate between a suspended node and a node that has gone away?
 - Suspended receiver buffer packet
 - Confused sender dropped packet, extra energy consumption
- How can a suspended node know there is communication for it?
 - Wake up too soon waste energy
 - Wake up too late delay/miss packets



Communication Device Suspension

- ▶ Approach

- ▶ Ensure overlap between sender's and receiver's awake times

- ▶ Protocols

- ▶ Triggered Resume
 - ▶ Periodic Resume
 - ▶ Synchronous
 - ▶ Asynchronous



Triggered Resume

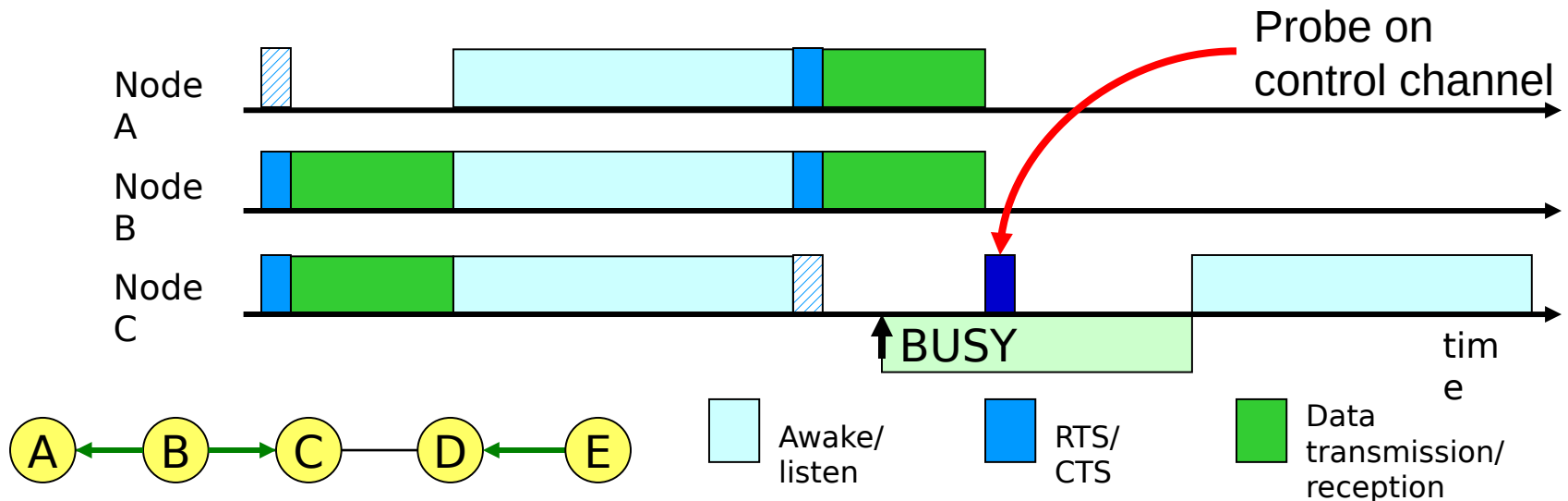
► Approach

- Use a second control channel (second radio)
 - Sender transmits RTS or beacon messages in control channel
 - Receiver replies in control channel and turns on main channel
- Main channel is only used for data
- Second channel
 - Must consume less energy than the main channel
 - Must not interfere with the main channel
 - Ex: BLE, ZigBee, RFID, 915Mhz



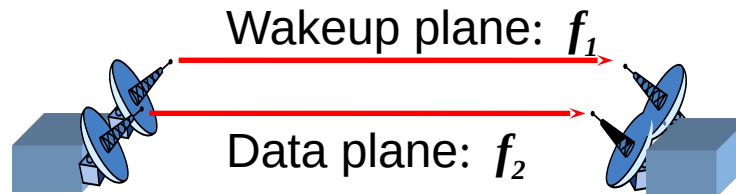
Triggered Resume

- ▶ Approach – Data only – PAMAS
 - ▶ Data channel
 - ▶ Power off radio when data is destined to a different node
 - ▶ Control channel
 - ▶ Probe neighbors to find longest remaining transfer



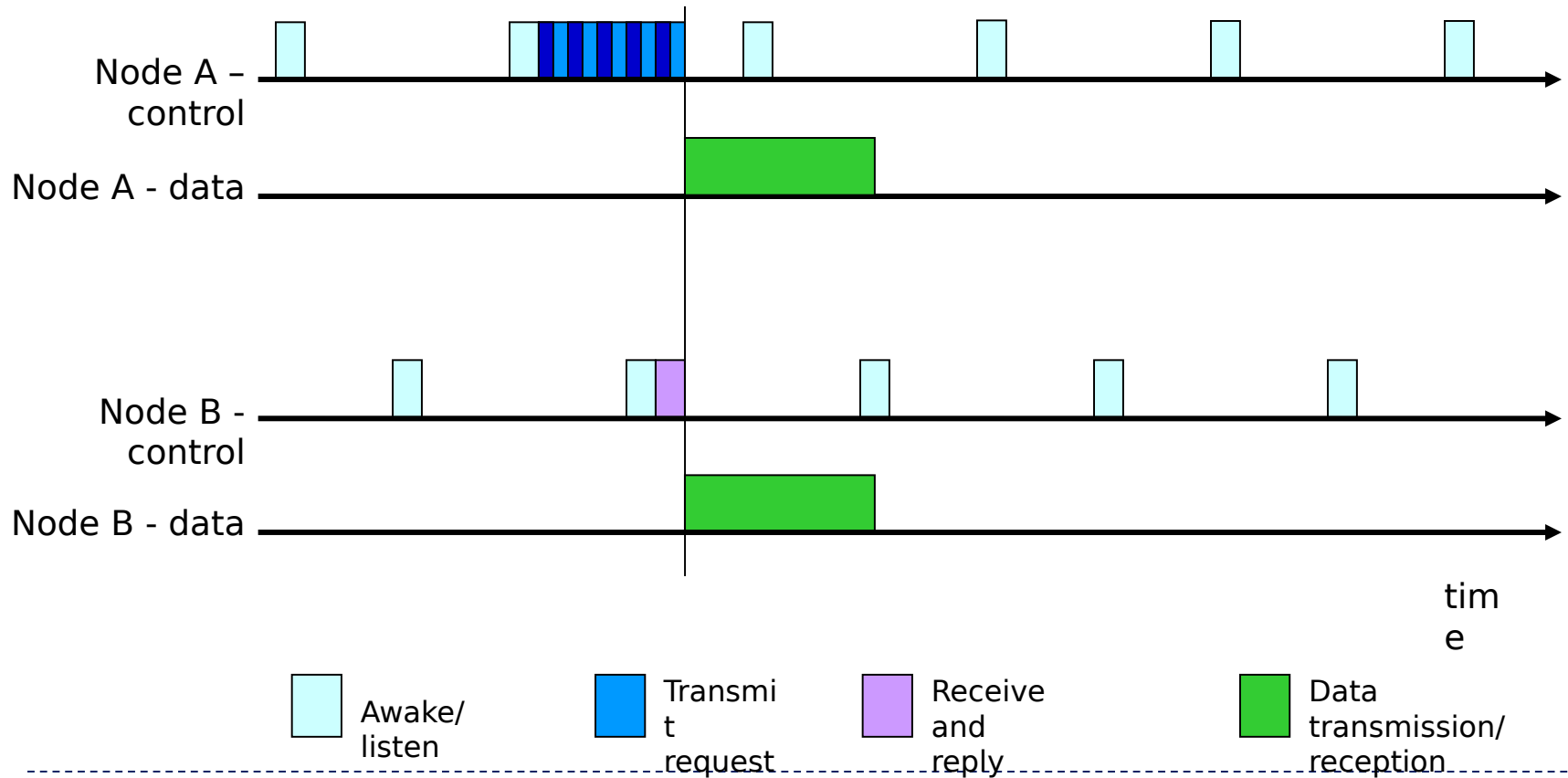
Triggered Resume

- ▶ Dual radio
 - ▶ Low duty cycle paging channel to wake up a neighboring node
 - ▶ Use separate radio for the paging channel to avoid interference with regular data forwarding
 - ▶ Trades off energy savings for setup latency



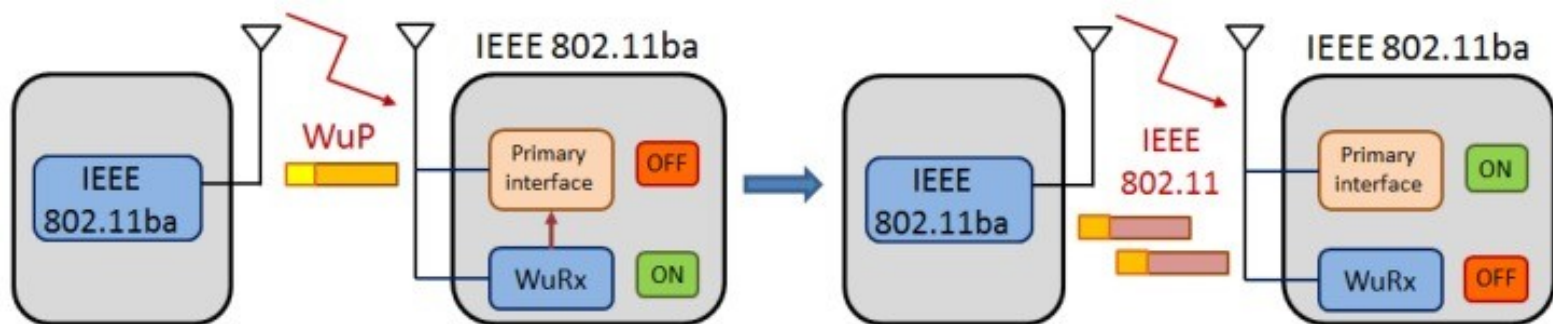
Triggered Resume

► Dual radio



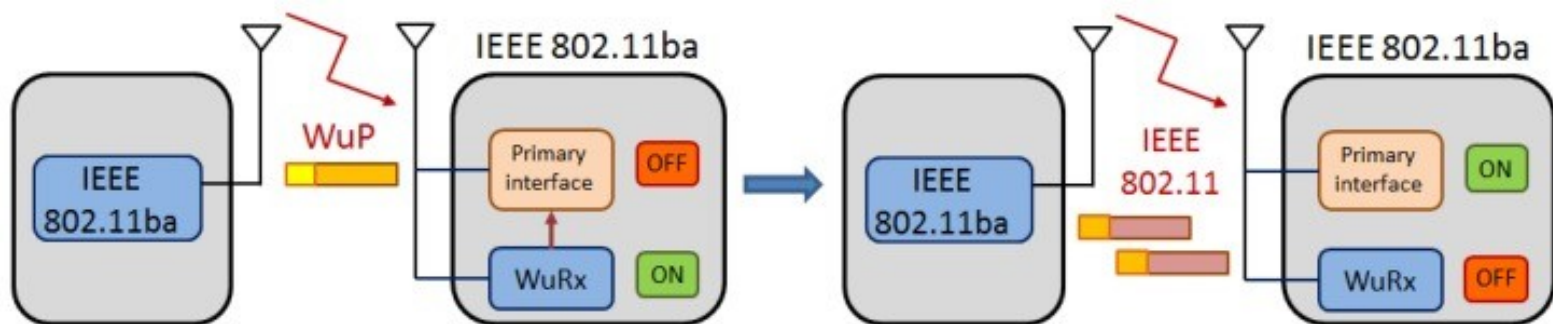
Triggered Resume – 802.11ba

- ▶ WuRx
 - ▶ a separate, ultra-low-power receiver
 - ▶ stays on when the Primary Interface is off
- ▶ On-Off Keying (OOK)
 - ▶ Significantly reduced listening cost



Triggered Resume – 802.11ba

- ▶ AP
 - ▶ Sends a short Wake-up Packet (WuP)
- ▶ WuRx
 - ▶ Detects the WuP
 - ▶ Powers on the Primary Interface
 - ▶ WuRx powers off
- ▶ Primary Interface
 - ▶ Handles 802.11 data exchange
- ▶ Net effect
 - ▶ Always-listening cost drops to near-zero
 - ▶ Full-rate Wi-Fi is only powered up when there's something to receive



Triggered Resume – Why isn't used today?

► Challenges

- Two radios are more complex than one
- Channel characteristics may not be the same for both radios
 - A successful RTS on the control channel does not guarantee the reverse channel works
 - A failed RTS on the control channel does not indicate that the reverse channel does not work



Periodic Resume

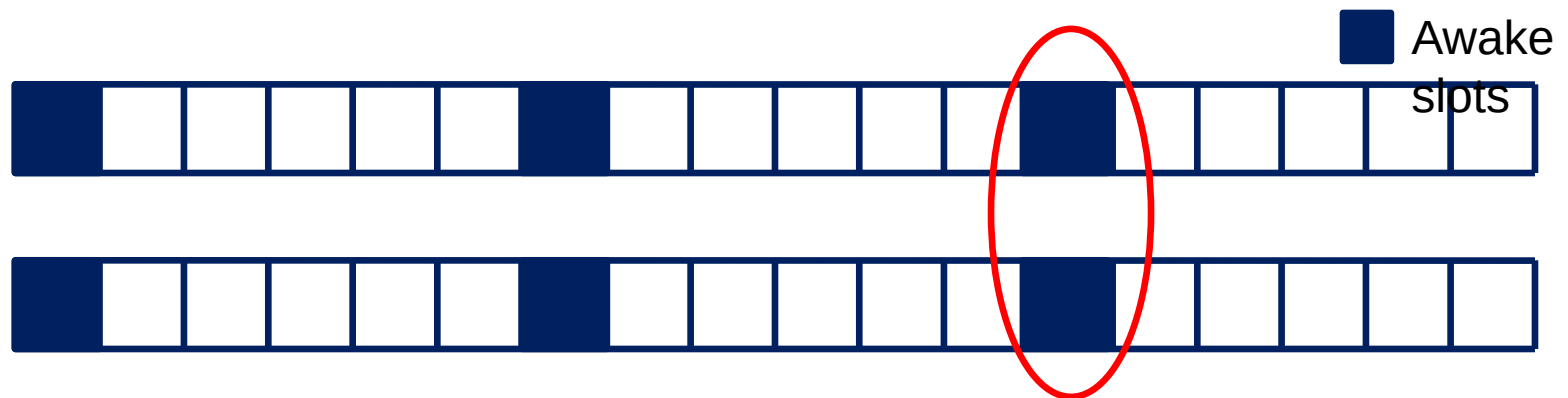
- ▶ **Approach**
 - ▶ Suspend most of the time
 - ▶ Periodically resume to check for pending communication
- ▶ **Communication indications**
 - ▶ Out-of-band channel
 - ▶ In-band signaling
- ▶ **Protocols**
 - ▶ Synchronous
 - ▶ Asynchronous



Synchronous Periodic Resume

► Basic Idea

- Time is slotted
- Nodes selectively remain awake for full slot duration
- Discovery occurs when two active slots overlap
- If all nodes are synchronized, all nodes are guaranteed to have overlapping awake periods



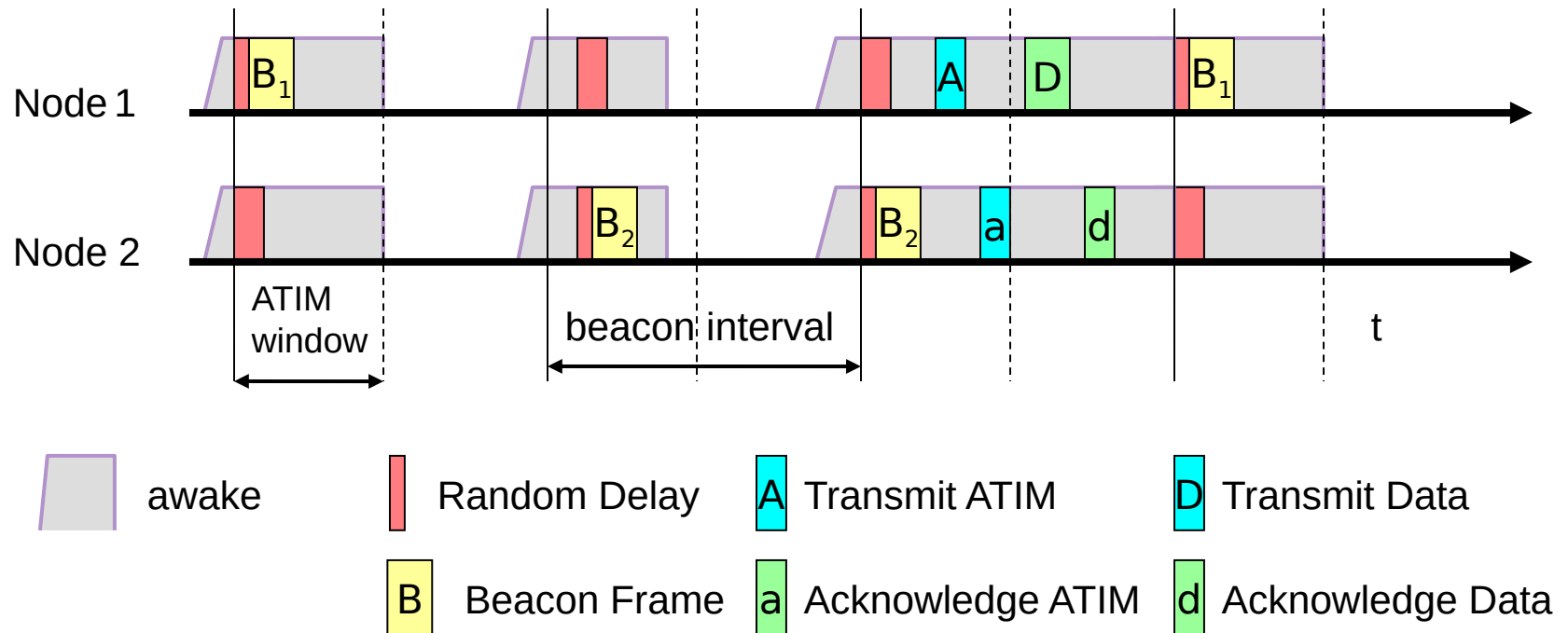
Synchronous Periodic Resume

- ▶ Protocol: IEEE 802.11 Power Save Mode (PSM)
 - ▶ Nodes are synchronized and wakeup periodically (Beacon Period)
 - ▶ Each beacon period is broken up into two segments
 - ▶ Ad-hoc Traffic Indication Map (ATIM) Window
 - Announcement in the ATIM indicates data
 - Target node responds with an ATIM ACK
 - If a node receives no announcements, it goes back to sleep
 - ▶ Transmission period
 - Sender can transmit packet until the end of the beacon period



Synchronous Periodic Resume

► IEEE 802.11 PSM



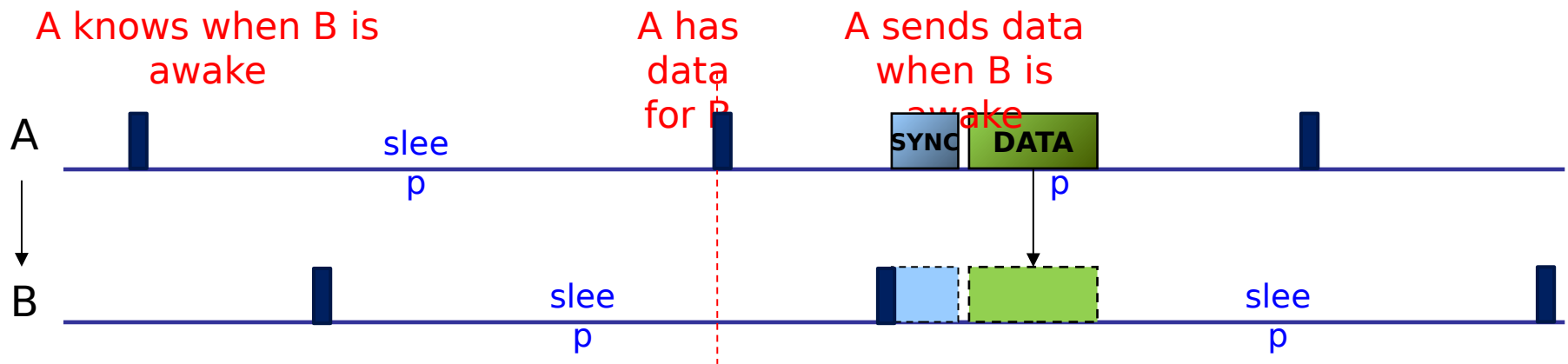
Synchronous Periodic Resume

- ▶ **Centralized solution**
 - ▶ Synchronization driven by base station
 - ▶ In beacon message
- ▶ **Distributed solution**
 - ▶ No base station
 - ▶ Synchronization protocols can be used to loosely synchronize nodes
 - ▶ Nodes wake up for a short period and check for channel activity
 - ▶ Return to sleep if no activity detected



Distributed Synchronous Periodic Resume

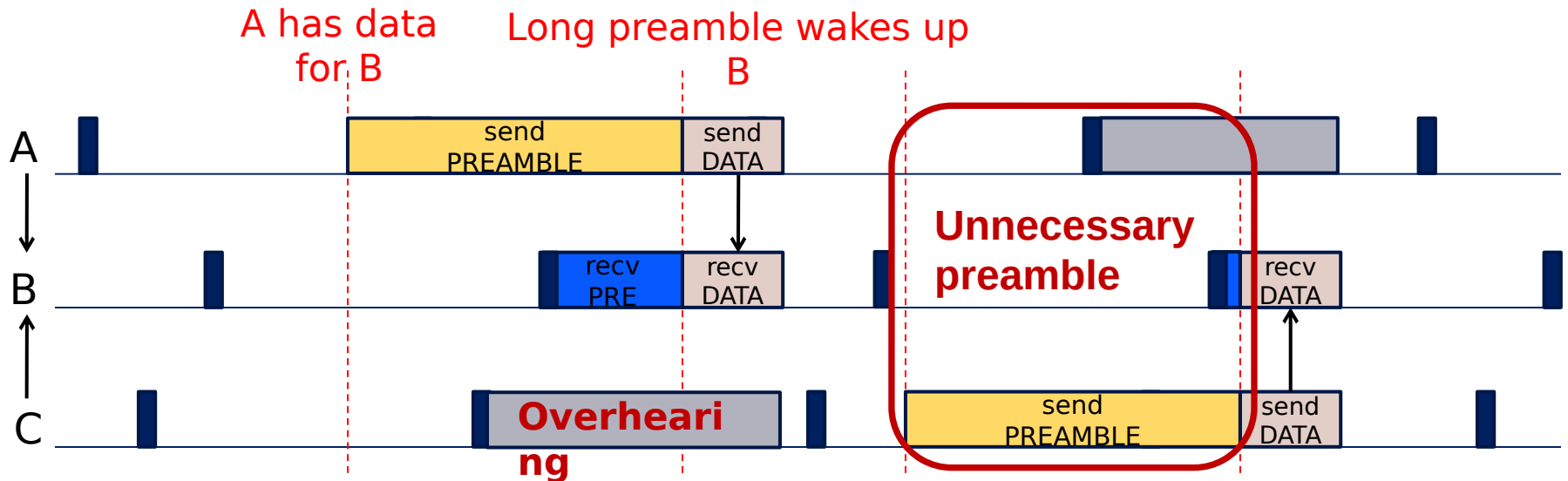
- ▶ Persistent loose synchronization
 - ▶ Constant, high synchronization overhead



Distributed Synchronous Periodic Resume

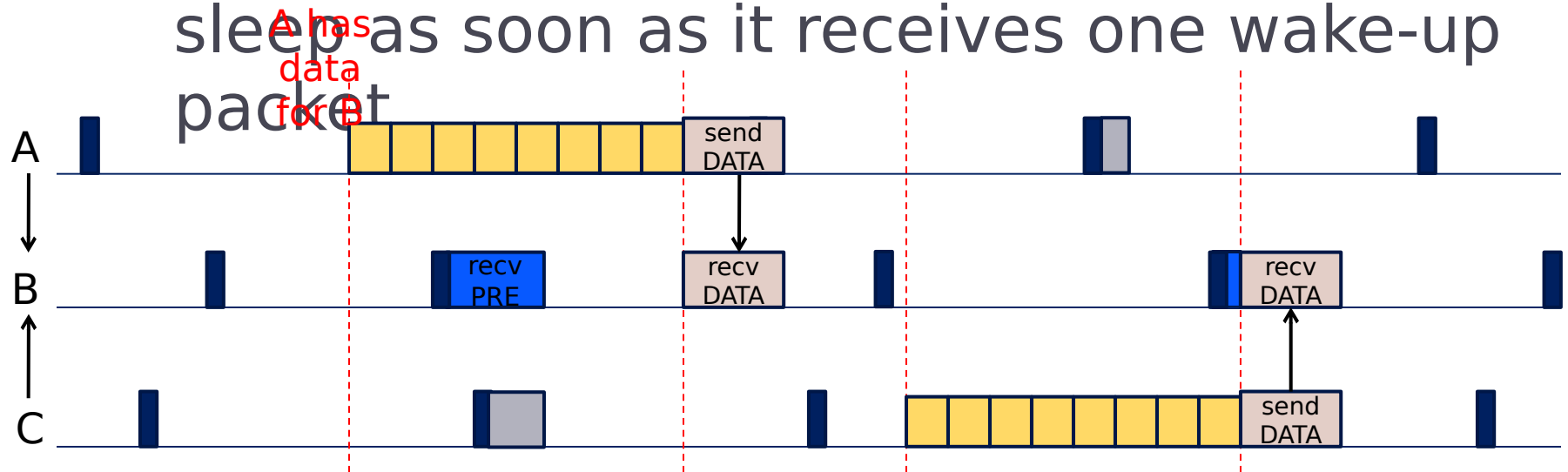
► Signaling

- No synchronization overhead
- High signaling overhead
 - Long preambles, all nodes wake up



Distributed Synchronous Periodic Resume

- ▶ Signaling: Wake-up packets
 - ▶ Send wake-up packets instead of preamble
 - ▶ Wake-up packets tell when data is starting so that receiver can go back to sleep as soon as it receives one wake-up packet



Distributed Synchronous Periodic Resume

- ▶ **Signaling: Multiple send**
 - ▶ Send data several times
 - ▶ Receiver can listen at any time and get all data
- ▶ **Problem with all approaches**
 - ▶ Communication costs are mostly paid by the sender
 - ▶ The amount of time the sender spends transmitting may be much longer than the actual data length



Synchronous Periodic Resume

► Problems

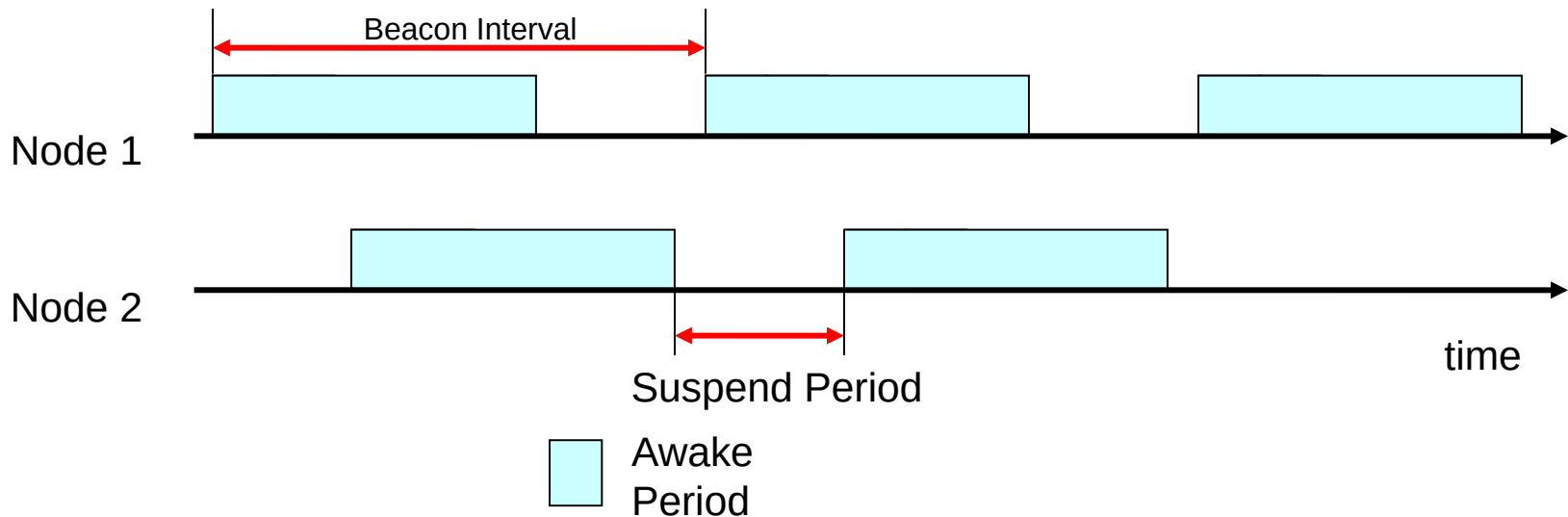
- Maintaining synchronization may be difficult
- Throughput is limited by the size of the notification window
 - If the notification window is too small, packets get buffered
 - Buffers may eventually overflow



Asynchronous Periodic Resume

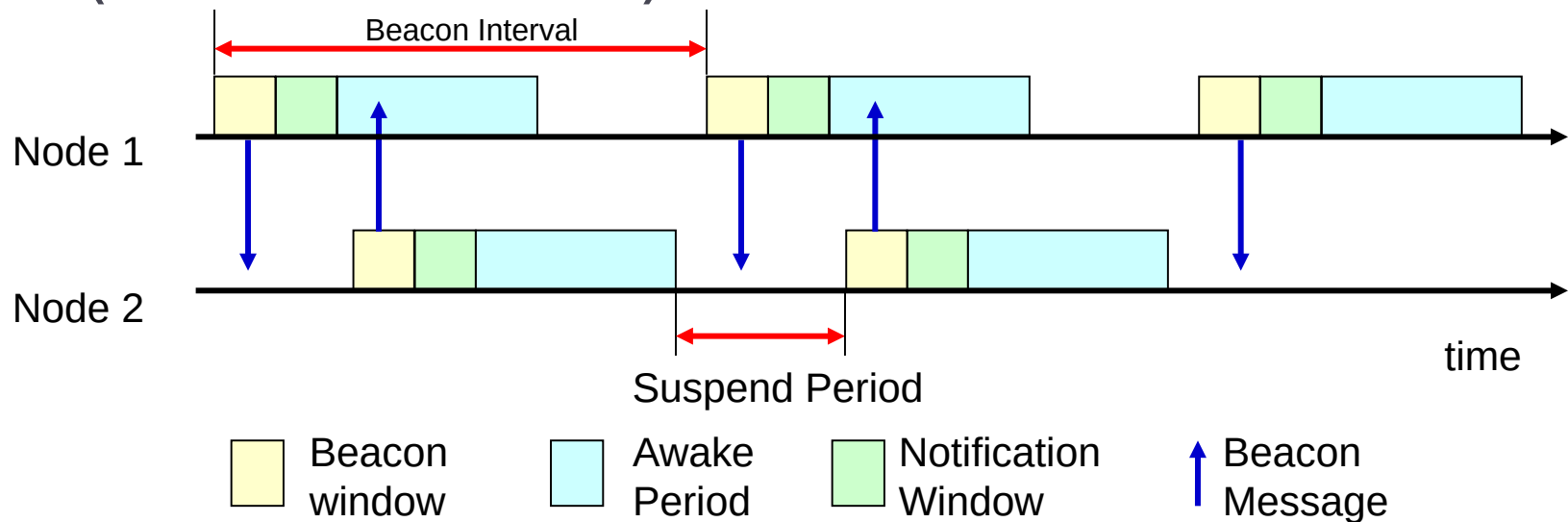
► Approach

- Stay awake longer to guarantee overlap of awake periods
- Overlap is guaranteed if the awake periods are more than half the beacon period



Asynchronous Periodic Resume

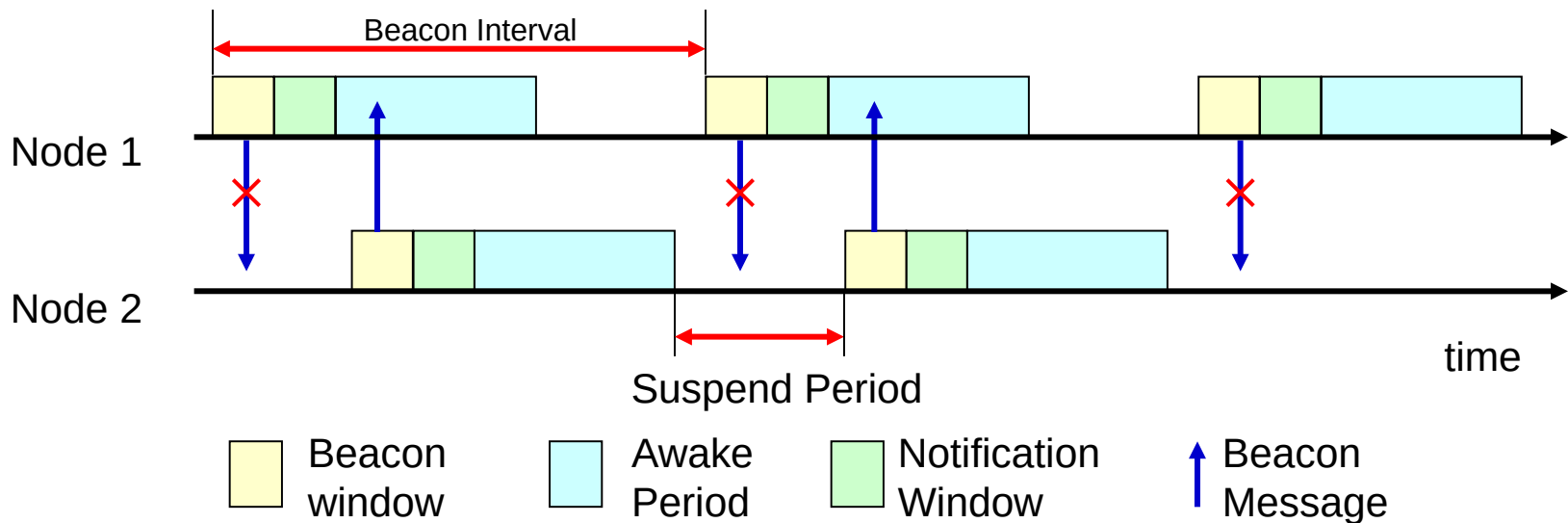
- ▶ Basic protocol
 - ▶ Use beacon messages at the start of awake periods
 - ▶ Some protocols use notification messages (similar to ATIM)



Asynchronous Periodic Resume

► Problem

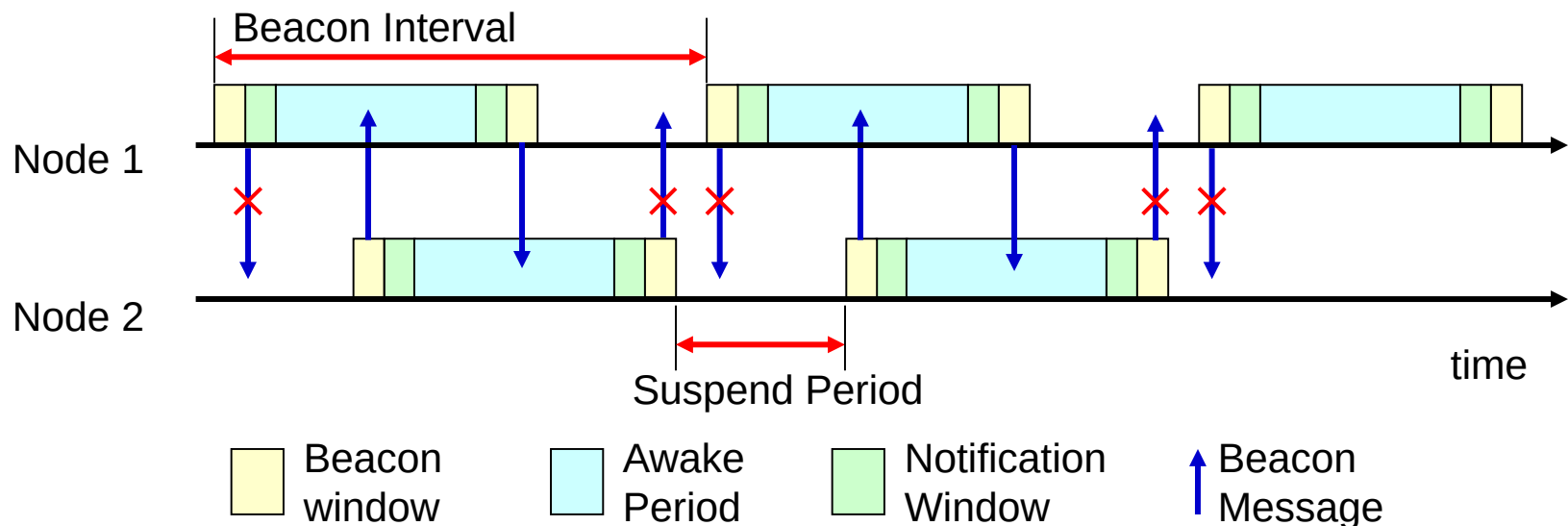
- No guarantee that all nodes will hear each other's beacon or notification messages



Asynchronous Periodic Resume

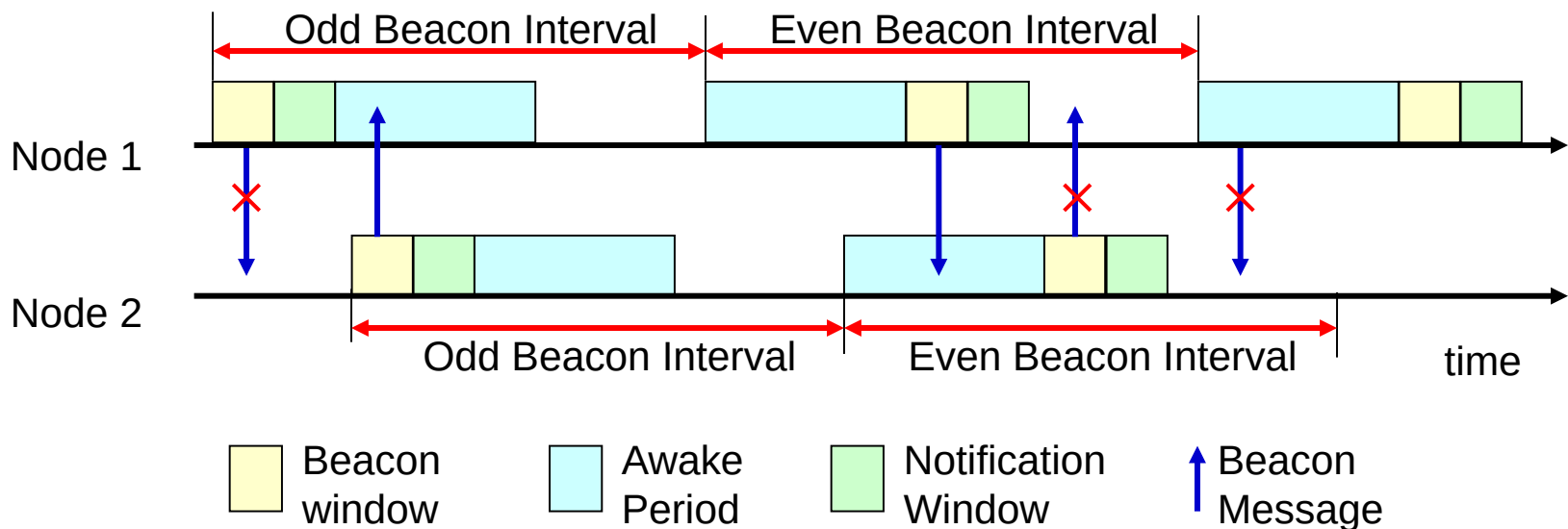
► Solution

- Have a beacon at the beginning and end of the beacon interval



Asynchronous Periodic Resume

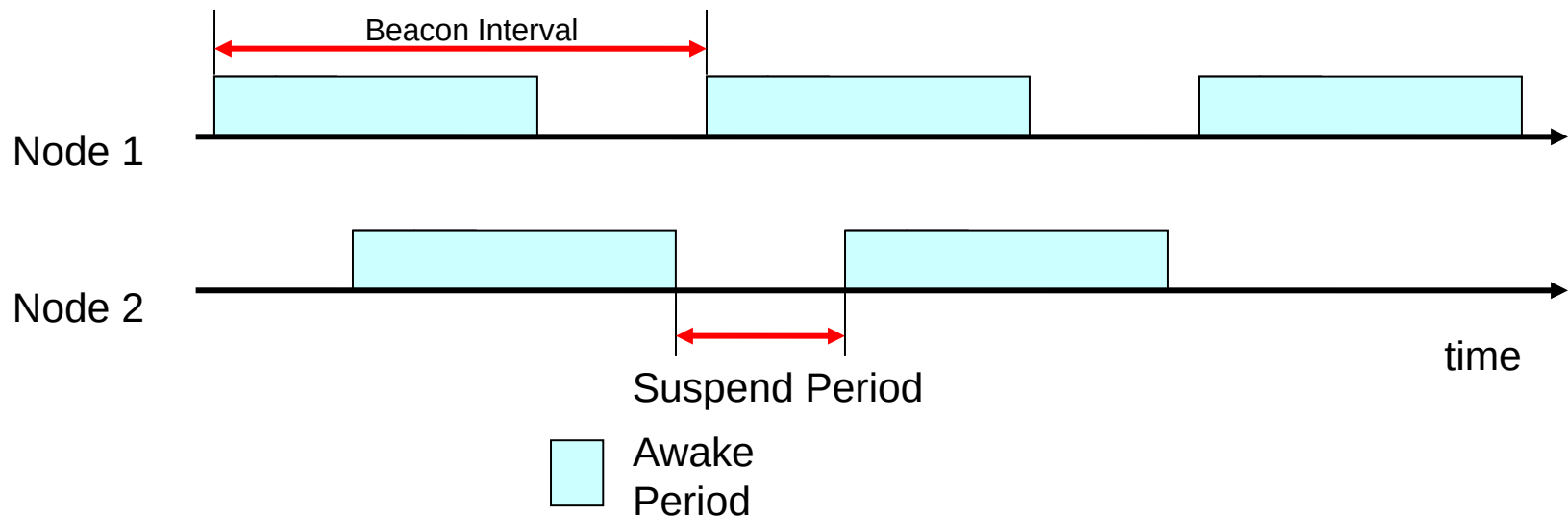
- ▶ Alternate solution
 - ▶ Beacon at the beginning of odd periods
 - ▶ Beacon at the end of even periods



Asynchronous Periodic Resume

► Problem

- Nodes stay awake more than half the time
- Wastes too much energy!

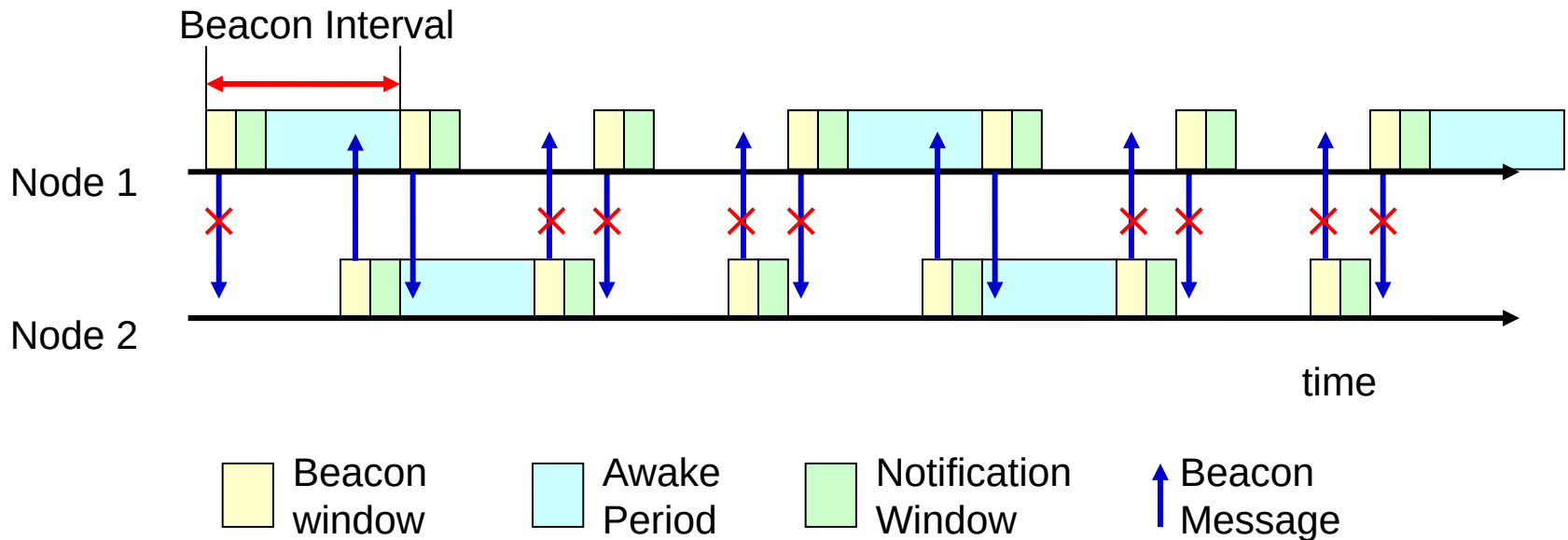


- ▶ Reduce awake time
 - ▶ Do not wake up every beacon interval
 - ▶ Delay depends on number of overlapping intervals



Asynchronous Periodic Resume

- ▶ Extended sleep
 - ▶ Wake up once every T intervals
 - ▶ Adds delay up to T length of beacon interval

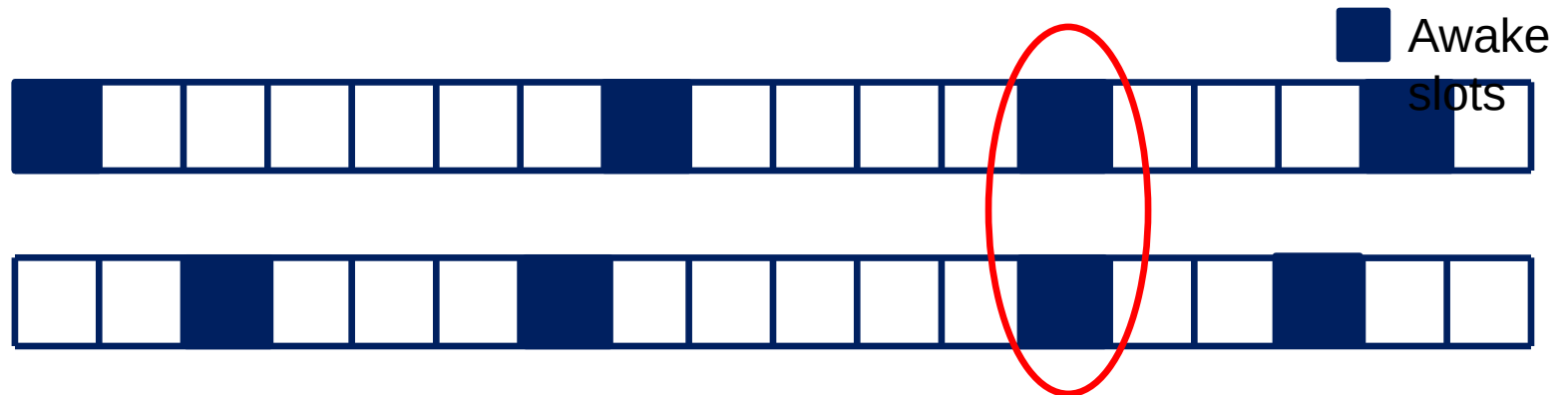


Asynchronous Periodic Resume

- ▶ Randomized Approach

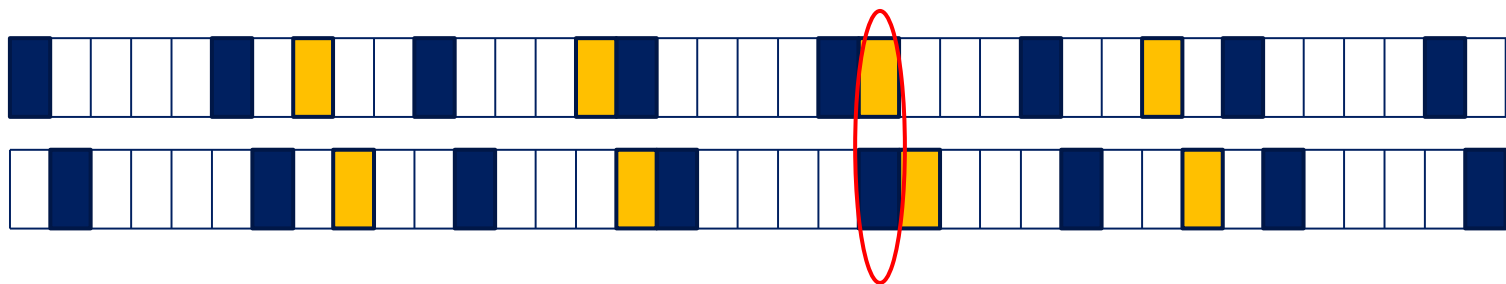
- ▶ Birthday protocol

- ▶ Randomly select a slot to wake up in with a given probability
 - ▶ Advantage
 - Good average case performance
 - ▶ Disadvantage
 - No bounds on worst-case discovery latency



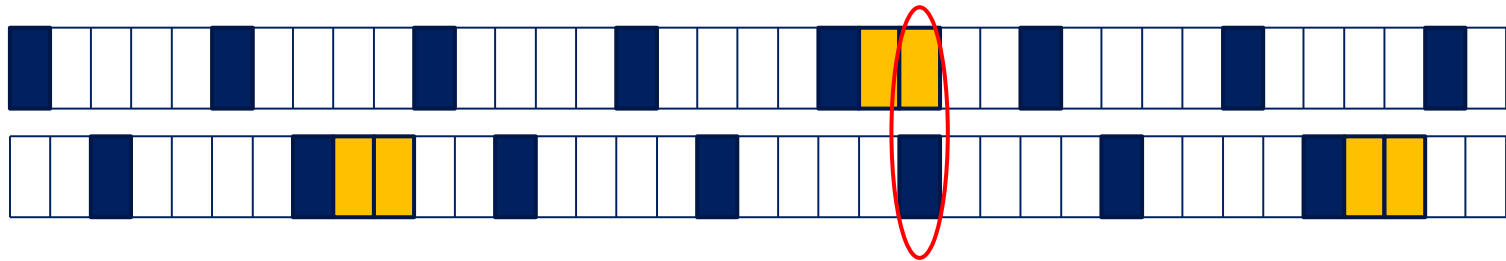
Asynchronous Periodic Resume

- ▶ **Deterministic: Prime-based**
 - ▶ Disco
 - ▶ Pick two primes p_1 and p_2
 - ▶ Wake up every p_1 and p_2 slot
 - ▶ Guarantees discovery in $p_1 \times p_2$ slots



Asynchronous Periodic Resume

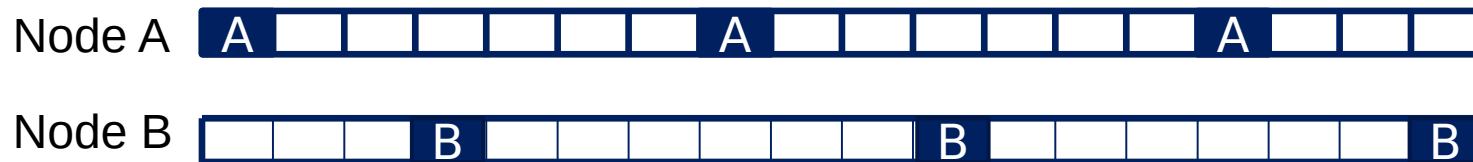
- ▶ **Deterministic: Prime-based**
 - ▶ U-Connect
 - ▶ Select 1 prime p
 - ▶ Wake up every p th slot and $(p-1)/2$ slots every $p \times p$ slots
 - ▶ Overlap is guaranteed within p^2 slots



Asynchronous Periodic Resume

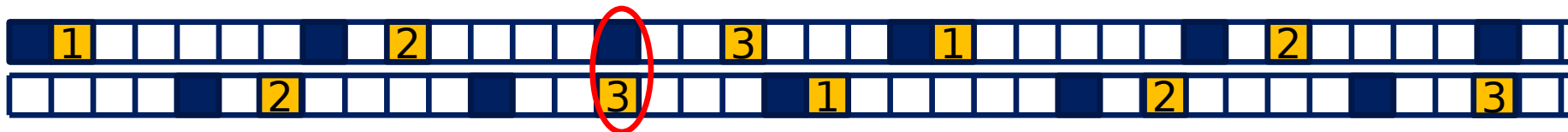
- ▶ Searchlight

- ▶ Have a **deterministic** discovery schedule that has a **pseudo-random** component



Asynchronous Periodic Resume

- ▶ Searchlight
 - ▶ Two slots per t slots (period)
 - ▶ Anchor slot: Keep one slot fixed at slot 0
 - ▶ Probe slot: Move around the other slot sequentially
 - ▶ Guaranteed overlap in $t \cdot t/2$ slots
 - ▶ Based on the time needed to ensure a probe-anchor overlap
 - ▶ Probe-probe overlap can also lead to discovery
 - ▶ Sequential scanning means less chance of a probe-probe overlap

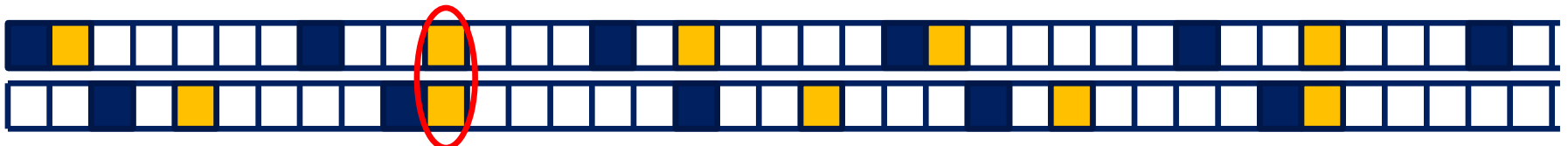


Discovery through anchor-probe overlap

Asynchronous Periodic Resume

► Searchlight

- Extension: randomized probing
 - Move the probe slot randomly
- Each node randomly chooses a schedule for its probe slot that repeats every $(t \cdot t/2)$ slots
 - Schedules of two nodes appear random to each other
- Advantage
 - Retains the same worst-case bound
 - Improves average case performance



Discovery through probe-probe overlap

Asynchronous Periodic Resume

► Challenges

- Reducing time spent awake
- Reducing delay
- No support for broadcast
 - None of the current approaches provide an interval where all nodes are awake



Wi-Fi 6 and beyond: Target Wake Time (TWT)

- ▶ Goal: reduce power consumption and contention
- ▶ Operating Modes
 - ▶ Individual TWT
 - ▶ Broadcast TWT
 - ▶ Opportunistic PS

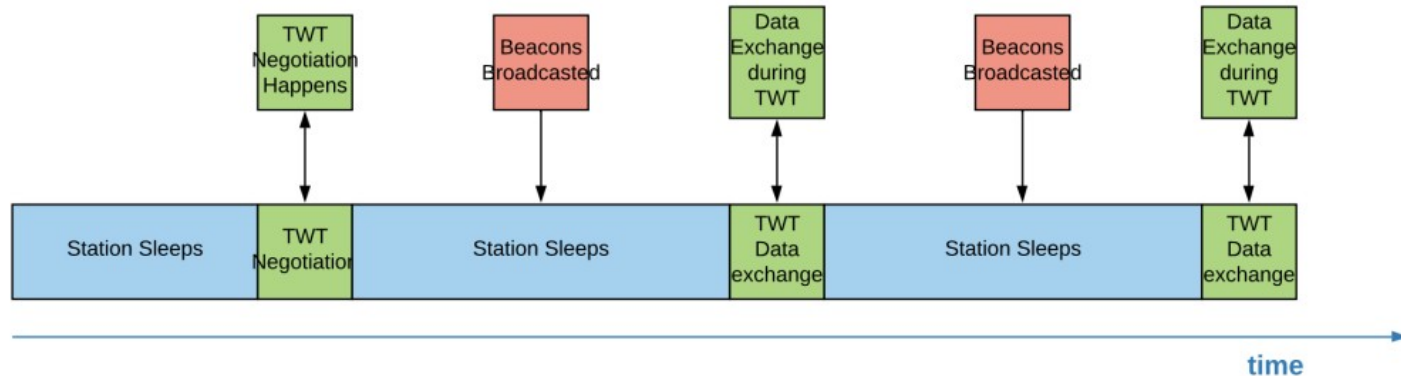


Target Wake Time (TWT)

- ▶ **TWT Service Period (TWT SP)**
 - ▶ Awake period to receive or send data
- ▶ **TWT Wake Interval**
 - ▶ Average time between successive TWT SP start times
- ▶ **TWT Channel**
 - ▶ Temporarily primary channel
- ▶ **TWT Agreement**
 - ▶ Final agreement between AP and client

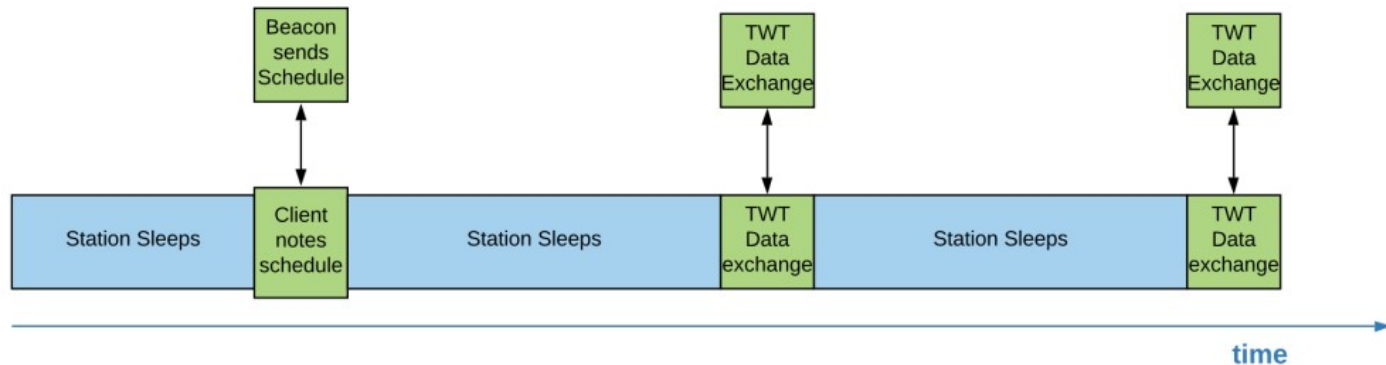


Individual Target Wake Time (TWT)



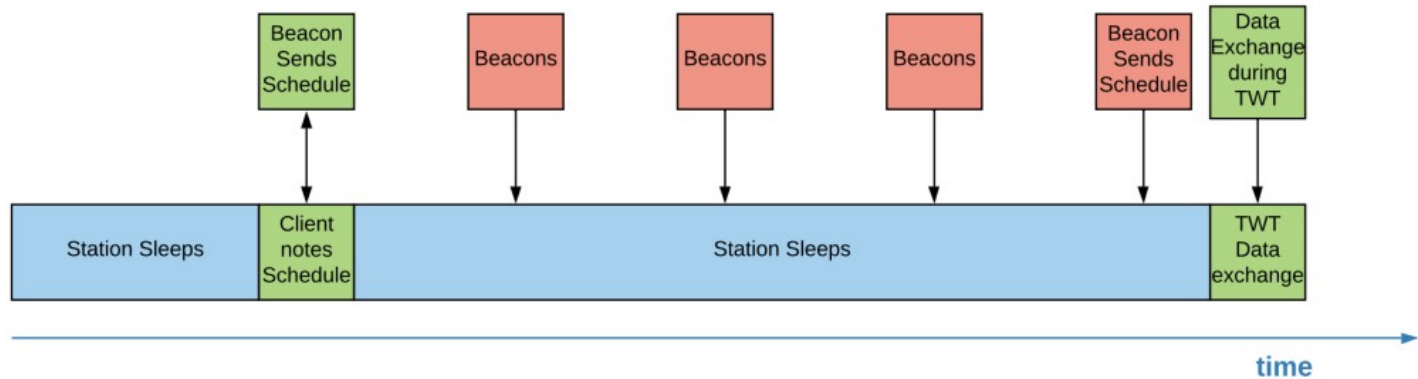
- ▶ Client sends waking schedule information to AP
- ▶ AP devises a schedule and sends TWT values to client
- ▶ The client wakes up and transmits according to the schedule
- ▶ **Explicit Mode**
 - ▶ AP sends client the next TWT information on when to wake up again
 - ▶ Client wakes up at the next scheduled time to send a frame and receive a new TWT information
- ▶ **Implicit Mode**
 - ▶ Client calculates next TWT by adding a fixed value to the current TWT

Broadcast Target Wake Time (TWT)



- ▶ **Hybrid PSM and TWT**
 - ▶ Beacon sub-intervals:
 - ▶ Traffic Indication Map (TIM):
 - ▶ Opportunistic sleep
 - ▶ Stay awake if Association Identifier (AID) is in TIM
 - ▶ No AID, client returns to a doze (sleep) state for the remainder of that service period
 - ▶ If AID is listed, client stays awake for that service period to receive data
 - ▶ No negotiation required

Opportunistic Power Save Mode



- ▶ Scheduling
 - ▶ AP sends TWT parameters in the Beacon frame using the TWT Element
 - ▶ Clients use the TWT parameters from the most recently received TWT element
- ▶ Trigger-based TWT
 - ▶ AP sends a trigger frame to discover which clients are awake
 - ▶ AP sends frames to awake clients
 - ▶ Clients can doze after transmission

Target Wake Time (TWT)

- ▶ TWT reduces congestion
 - ▶ Packet collisions cause retransmissions
 - ▶ Retransmission increase client energy consumption
- ▶ Legacy Wi-Fi networks
 - ▶ Clients wake up at random times to transmit
 - ▶ Clients have no knowledge of the wake-up times of other clients
 - ▶ No collision avoidance mechanisms
- ▶ TWT
 - ▶ APs negotiate wake up times for clients
 - ▶ APs can avoid having all clients wake up at the same time



TWT vs PSM

- ▶ **PSM**

- ▶ Clients must wake up during each beacon interval to check for buffered traffic
- ▶ Amount of traffic is limited by the TIM

- ▶ **TWT**

- ▶ Diverse sleep cycles
- ▶ Longer, negotiated sleep cycles
- ▶ Sleep cycle length independent of the standard beacon interval

802.11be: Restricted TWT (R-TWT)

- ▶ Problem with regular TWT: a scheduled TWT service period can still be delayed by other traffic (e.g., legacy contention) already using the medium
- ▶ R-TWT (new in 802.11be) protects specific service periods for latency-sensitive traffic
 - ▶ AP announces an R-TWT schedule; non-participating stations must defer/quiet during that window
 - ▶ Targets applications like gaming, video calls, AR/VR — needs bounded, predictable latency, not just average throughput
- ▶ Builds directly on the TWT SP concept from Wi-Fi 6, but adds an admission/protection mechanism around it



Power Save Under Multi-Link Operation (MLO)

- ▶ MLO (802.11be) lets a device stay associated on multiple links (2.4/5/6 GHz) at once — power save now needs to span all of them
- ▶ New complication: a device can't just "sleep" the way single-link PSM/TWT does
 - ▶ Different links may have different traffic, different TWT schedules, different radio states
 - ▶ Device must decide which link(s) to keep awake on, and when it's safe to power down a given radio without missing traffic
- ▶ Two device power modes under MLO
 - ▶ STR (Simultaneous Tx/Rx): independent radios per link, can sleep/wake each link separately
 - ▶ NSTR (Non-Simultaneous Tx/Rx): shared constraints across links, sleep decisions are coupled
- ▶ Extending TWT/R-TWT scheduling across multiple, possibly asynchronous links is an active area of standards work



Cellular Power Saving: Discontinuous Reception (DRX)

- ▶ Same core idea as TWT: negotiate sleep/wake cycles instead of always listening
- ▶ Idle-mode DRX
 - ▶ No active connection — device wakes on a long cycle just to check for paging
- ▶ Connected-mode DRX (C-DRX)
 - ▶ Device has an active connection; alternates between an awake window and sleep
 - ▶ Short DRX Cycle used right after activity, in case more data is coming
 - ▶ Falls back to a Long DRX Cycle if nothing arrives — less frequent wake-ups, more power saved
- ▶ Tradeoff is the same one seen throughout this deck: longer sleep cycle = more power saved, more latency



DRX: Timers and Wake-Up Signal

- ▶ On Duration Timer
 - ▶ The window each cycle where the device listens for a scheduling grant on the control channel (PDCCH)
- ▶ DRX Inactivity Timer
 - ▶ After any new grant, the device stays awake a bit longer in case more data follows
 - ▶ Resets the countdown back to a Short DRX Cycle instead of immediately going to Long
- ▶ Wake-Up Signal (WUS) — 5G addition
 - ▶ A short signal the network can send before the On Duration begins
 - ▶ Tells the device whether to skip that cycle's wake-up entirely
 - ▶ Same idea as 802.11ba's WuP — a cheap heads-up so the device doesn't fully wake just to find nothing's there



LoRaWAN: Receive Windows Instead of Periodic Wake-Up

- ▶ A different model than Wi-Fi or cellular
 - ▶ No periodic beacon or poll cycle — the device itself controls when it's reachable
 - ▶ The network can't just page it whenever; it has to wait for its chance
- ▶ Class A (default for every LoRaWAN device)
 - ▶ After every uplink, the device opens two short receive windows: RX1, then RX2
 - ▶ Otherwise, the radio is fully off
 - ▶ Most power-efficient class — but downlink is only possible right after an uplink



LoRaWAN Class B and Class C

- ▶ Class B

- ▶ Adds scheduled "ping slots," time-synced to a beacon broadcast by the gateway
- ▶ Gives the network periodic chances to reach the device without waiting on an uplink
- ▶ More power than Class A, but device is still asleep most of the time

- ▶ Class C

- ▶ Receiver stays on continuously, except while transmitting
 - ▶ Lowest latency, highest power draw
 - ▶ Typically used for mains-powered devices, not battery-powered ones
- ▶ Same tradeoff as TWT and DRX — just a different mechanism: uplink-triggered windows instead of periodic wake cycles

