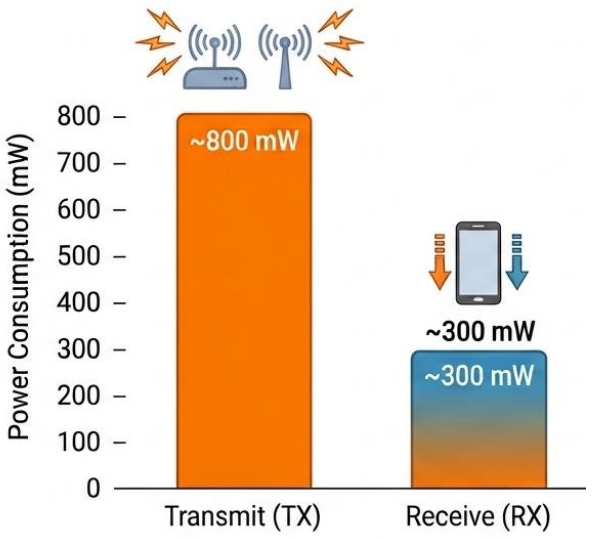
CS/ECE 439: Wireless Networking

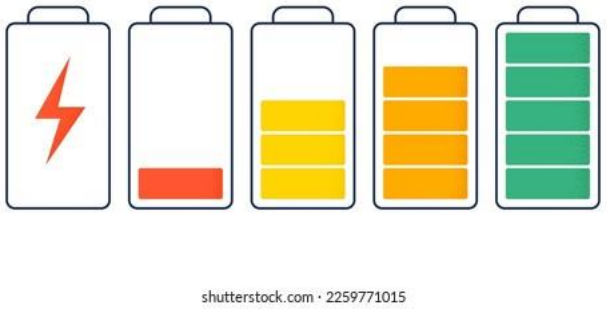
MAC Layer – Power!

© CS 439 Staff, University of Illinois Fall 2026

Energy Conservation Techniques

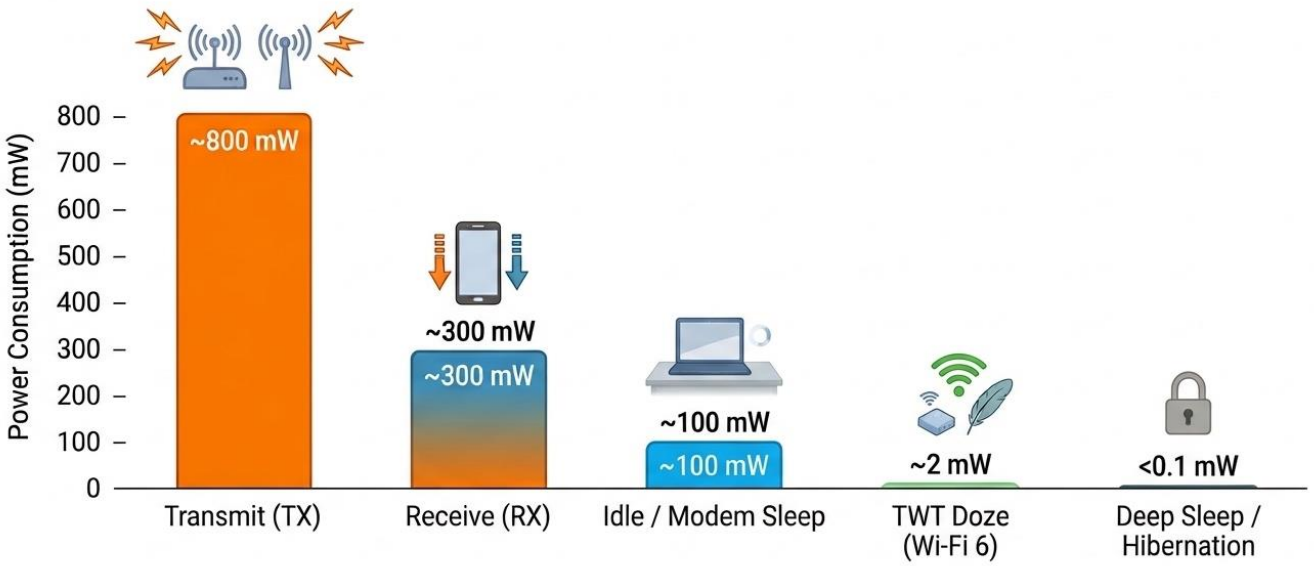
- Wi-Fi devices consume significant amounts of energy when idle
  - Idle listening costs about as much as receiving: hundreds of mW up to ~1 W





Energy Conservation Techniques

- Device suspension (sleep)
  - Reduced energy consumption
    - Sleep: orders of magnitude lower (mW or less)



Where Does the Energy Go?

- Radio power by state (highest to lowest)
  - Transmit — highest (power amplifier)
  - Receive — decoding a frame
  - Idle listening — receiver on, no frame
close to receive power
  - Sleep — radio off, orders of magnitude lower
- Key insight
  - Waiting for traffic costs almost as much as receiving
  - Savings come from sleeping, not just from sending less

Energy Costs: Example ESP32

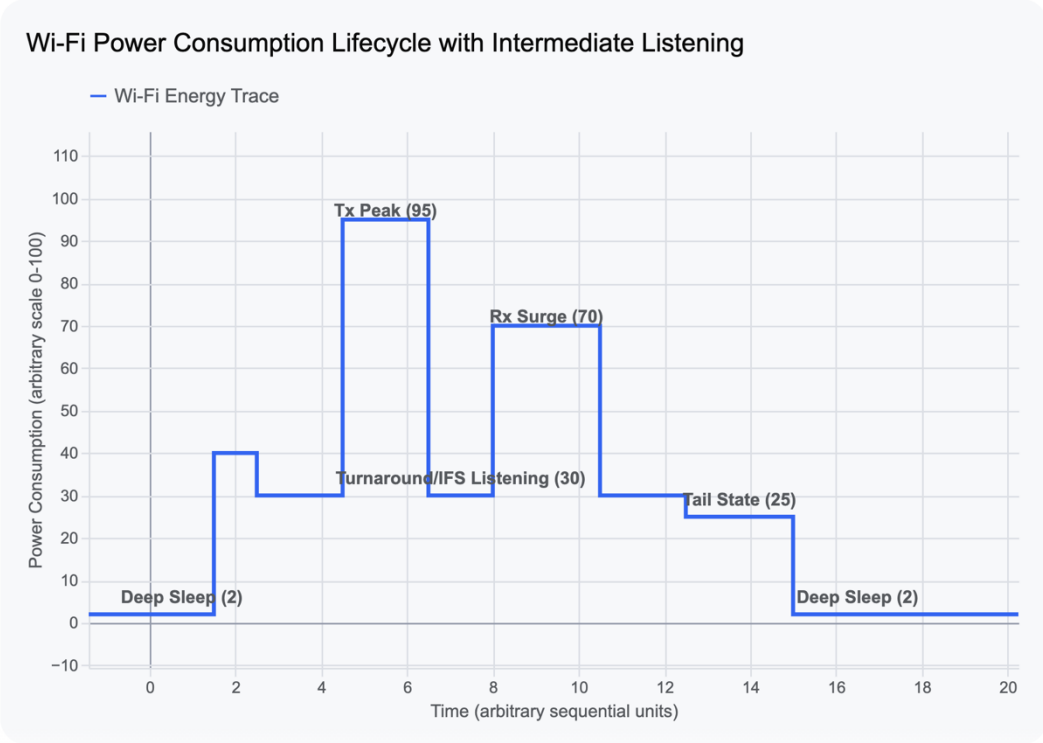
State	Current	Power at 3.3 V
Transmit, 802.11ax MCS9 (15.5 dBm)	251 mA	0.83 W
Transmit, 802.11b (20.5 dBm)	382 mA	1.26 W
Receive / idle listening	78 mA	0.26 W
Modem-sleep (associated, CPU on)	17–27 mA	56–89 mW
Light-sleep	0.18 mA	0.6 mW
Deep-sleep	7 µA	23 µW

Receiving costs about 30% of transmitting, and idle listening costs about the same as receiving. Modem-sleep (still associated) saves roughly 3–5x; light and deep sleep save 2–4 orders of magnitude.

Source: Espressif ESP32-C6-WROOM-1 datasheet (Wi-Fi 6 IoT module). Phone and laptop radios differ in absolute values.

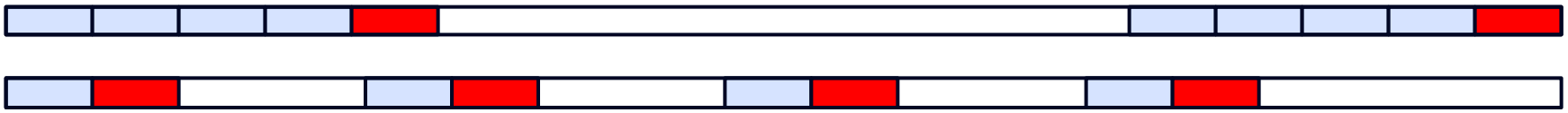
Hidden Energy Costs: Tail Energy

- Radio does not drop to sleep immediately
  - It stays in a high-power state until an inactivity timer expires
  - Fixed cost regardless of amount of data transmitted

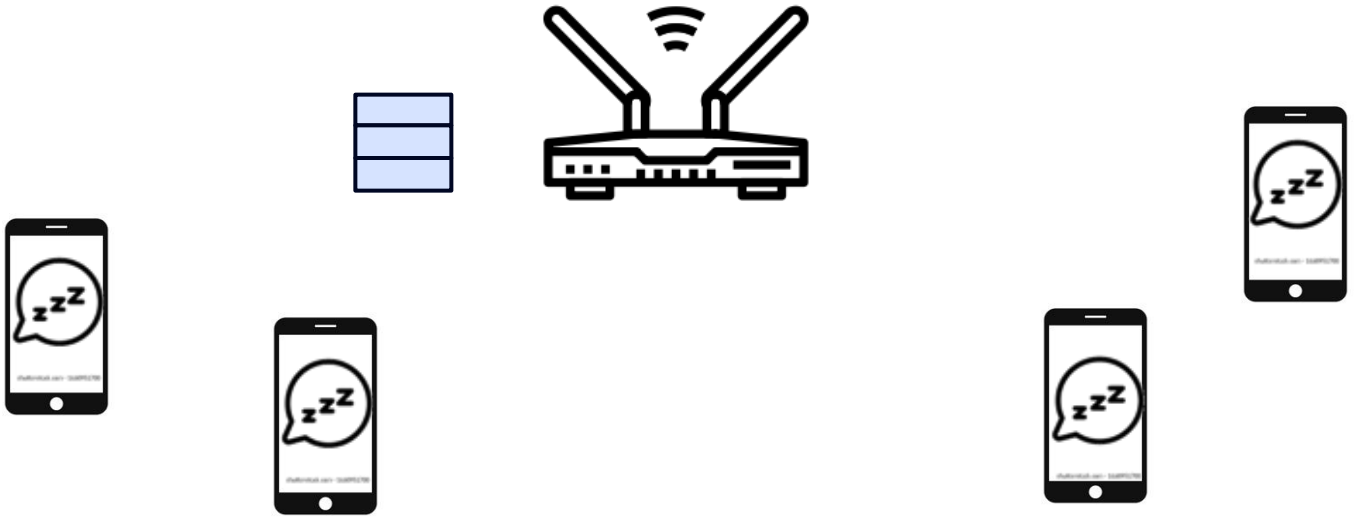


Hidden Energy Costs: Tail Energy

- Tradeoffs
  - Many small packets cost far more than one aggregate batch
  - Background syncs, keepalives, and chatty apps keep the radio awake
  - Batching traffic saves energy even if the total bytes stay the same
- Cellular
  - Tail is long
  - LTE connected mode
    - About 11s in measurement studies (set by the carrier)
- Wi-Fi
  - Tail is much shorter
  - Fraction of a second

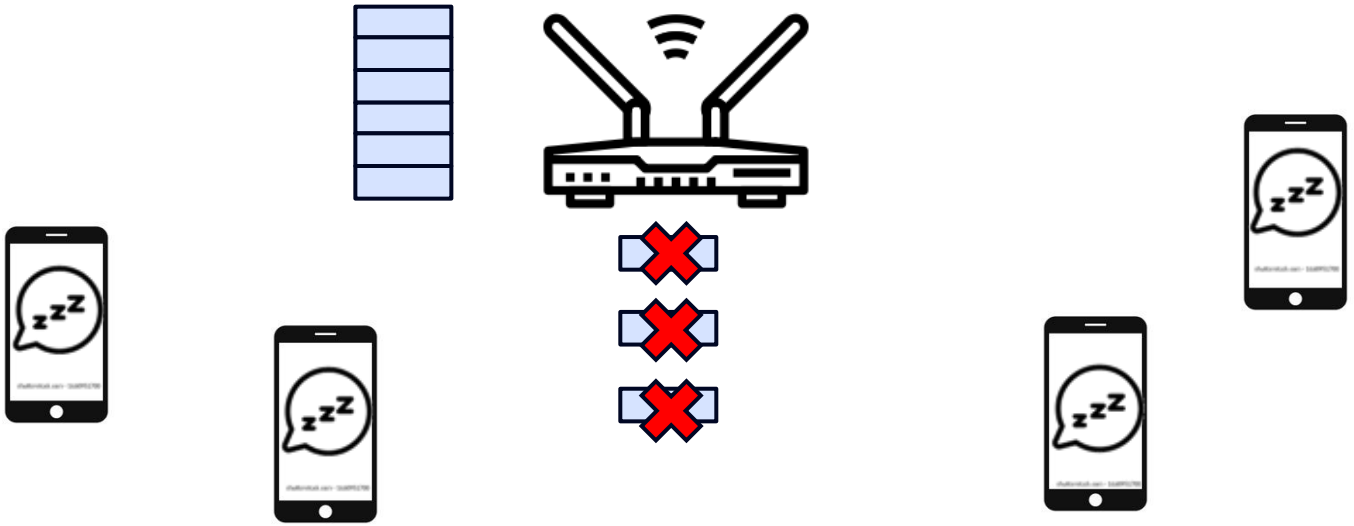


Energy Conservation Challenges



Energy Conservation Challenges

- 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 $\Rightarrow$ buffer packet
    - Confused sender $\Rightarrow$ dropped packet, extra energy consumption
  - How can a suspended node know there is communication for it?
    - Wake up too soon $\Rightarrow$ waste energy
    - Wake up too late $\Rightarrow$ delay/miss packets

Communication Device Suspension

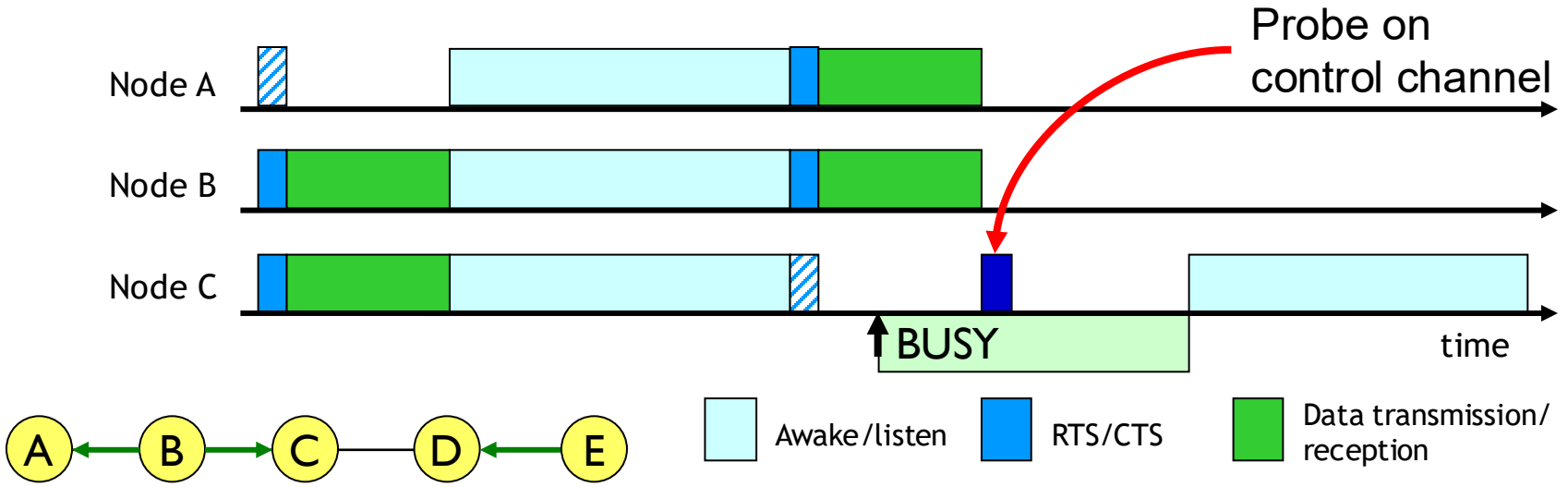
- Approach
  - Ensure overlap between sender's and receiver's awake times
- Protocols
  - Triggered Resume
  - Periodic Resume
    - Synchronous
    - Asynchronous

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

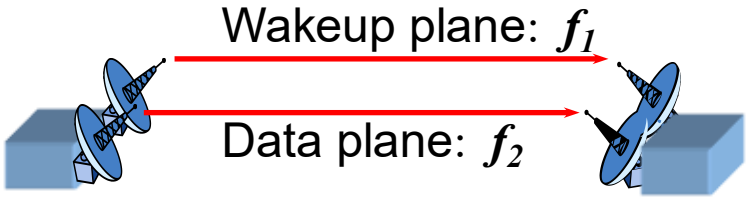
Triggered Resume

- 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



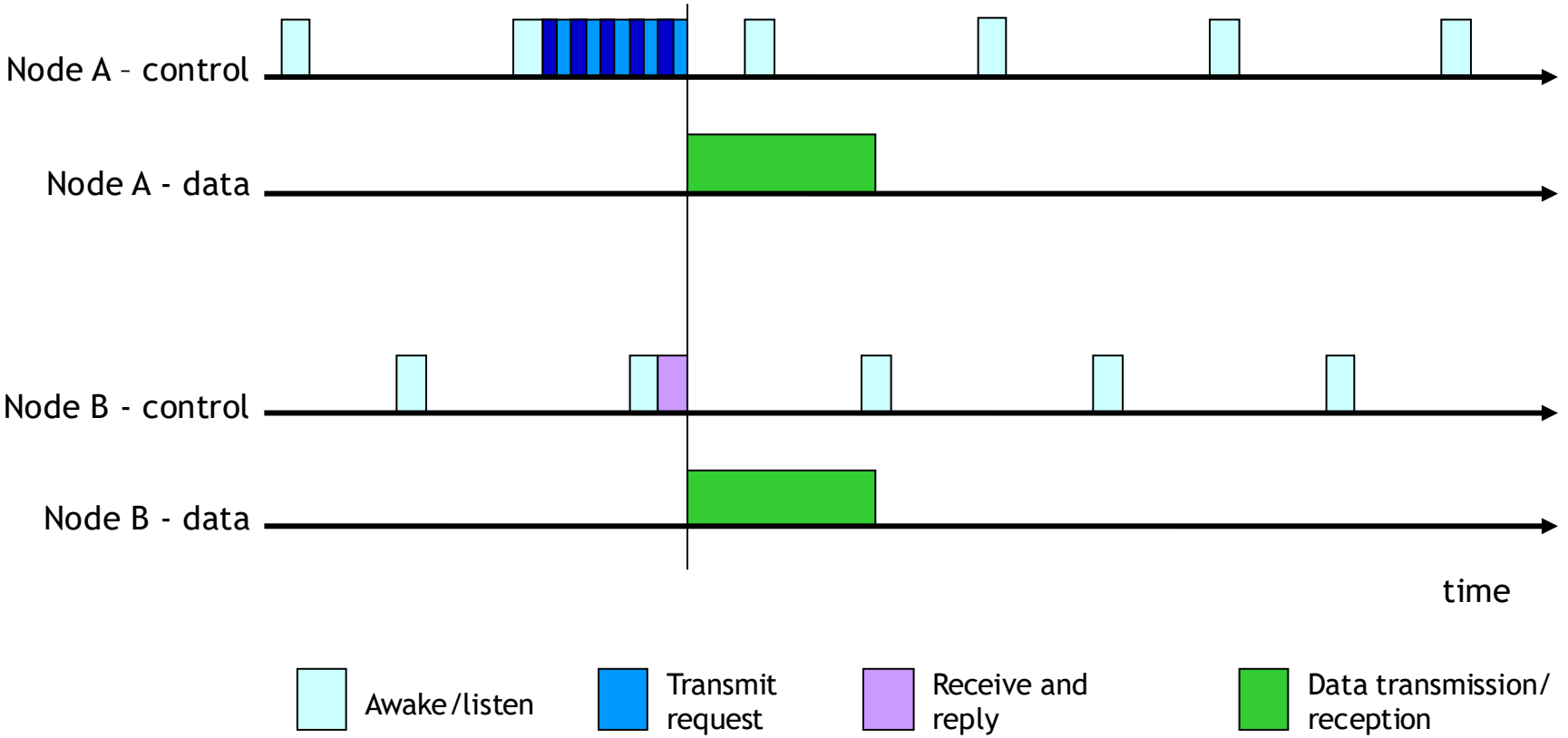
Triggered Resume

- 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

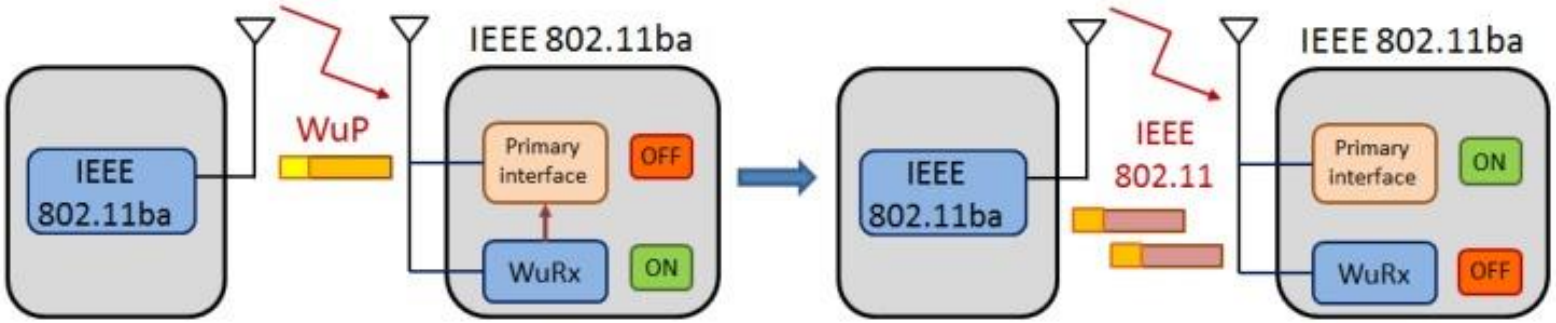


Triggered Resume – 802.11ba

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

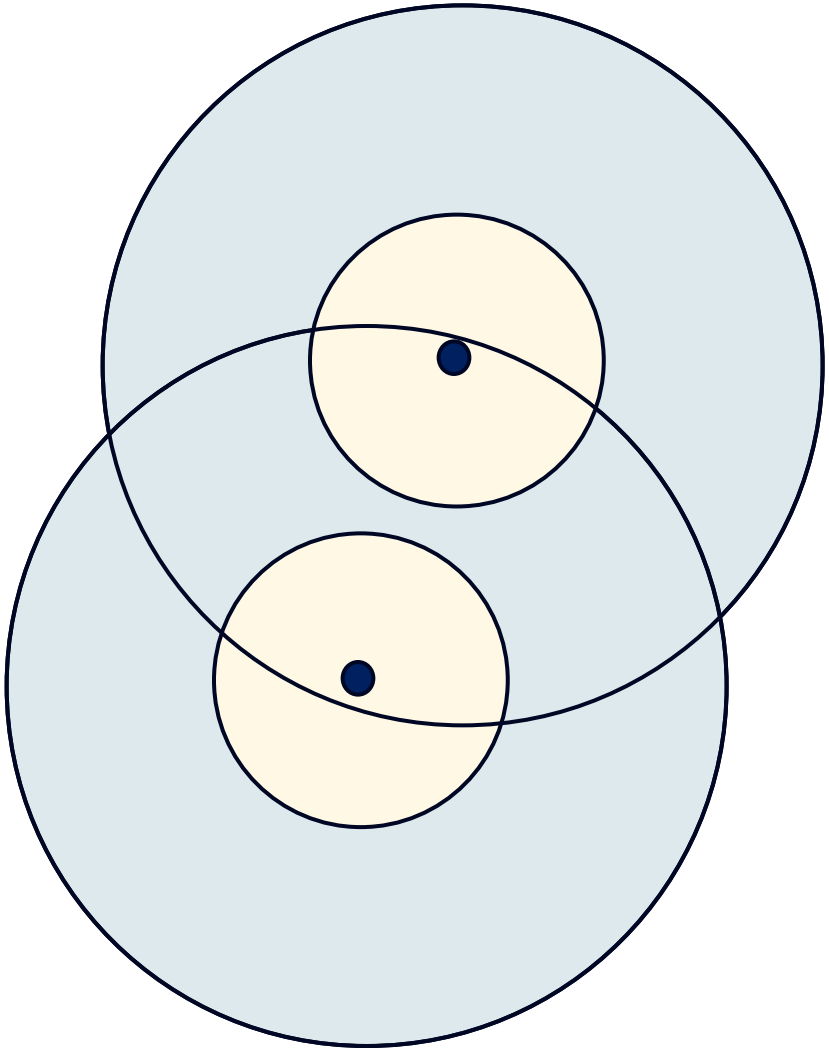
Triggered Resume – 802.11ba

- 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 – Challenges

- 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
  - 802.11ba (wake-up receiver) has seen limited adoption so far

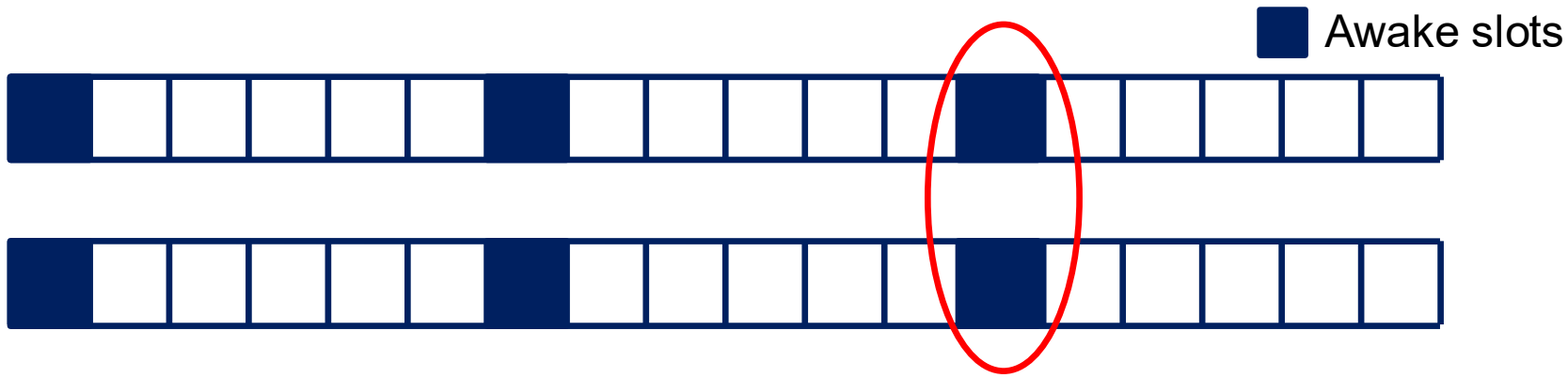


Periodic Resume

- 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

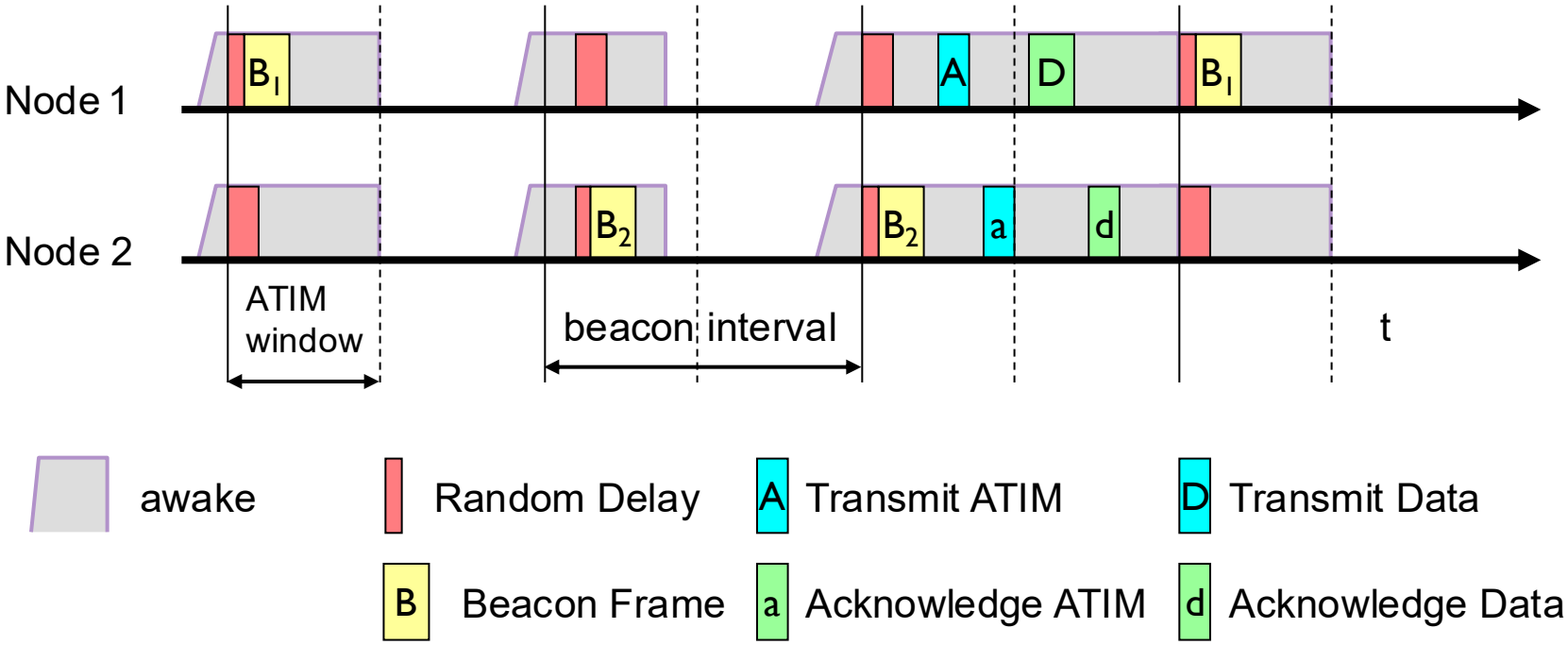


Synchronous Periodic Resume

- IEEE 802.11 IBSS (ad hoc) Power Save Mode
  - 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 IBSS PSM

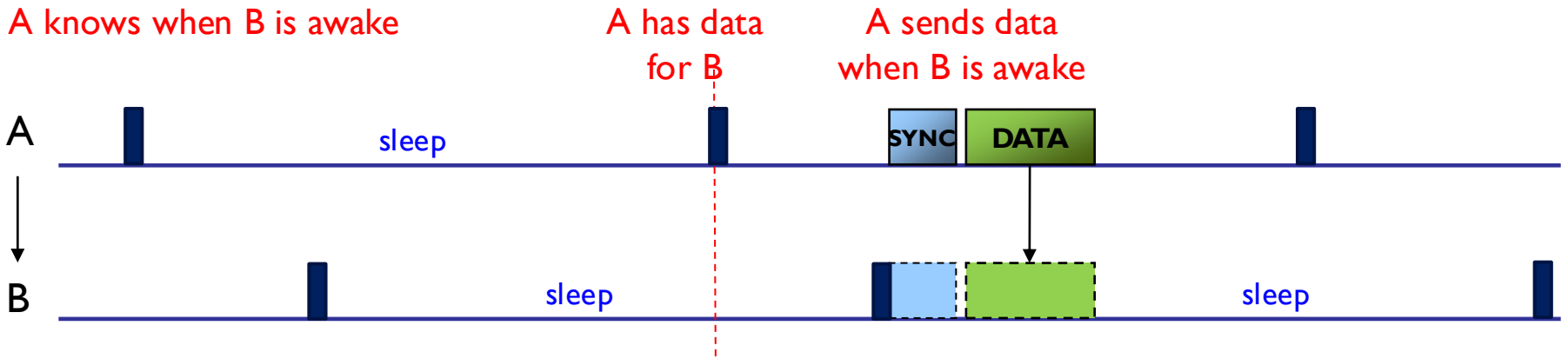


Synchronous Periodic Resume

- 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 (e.g., S-MAC)

- Persistent loose synchronization
  - Constant, high synchronization overhead

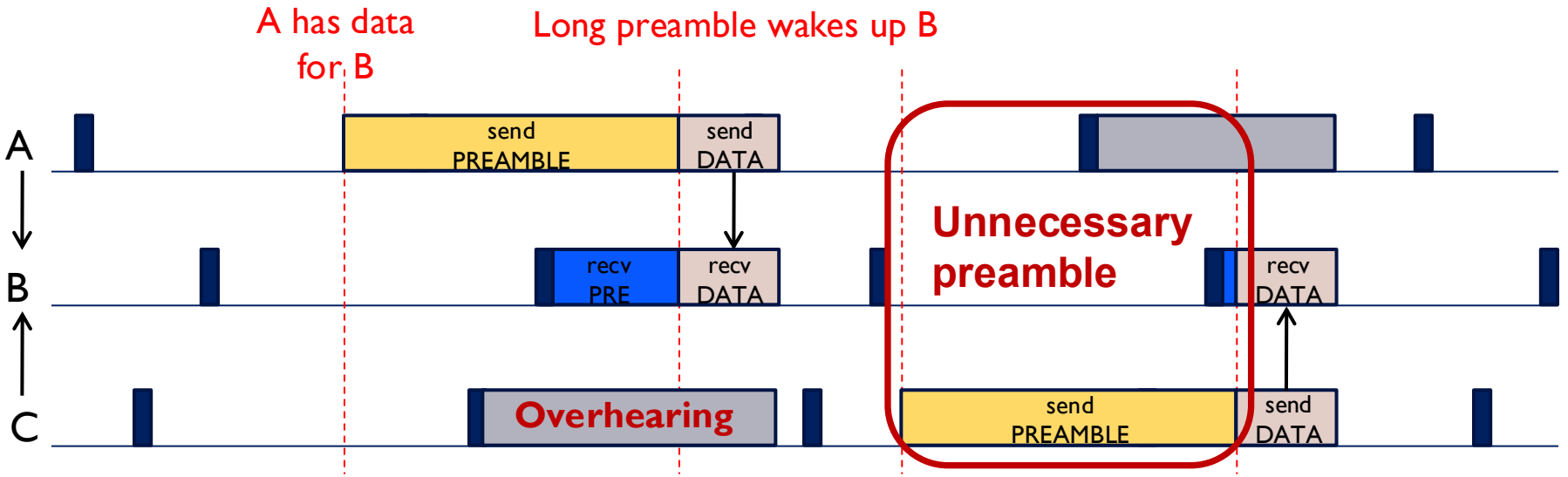


Synchronous Periodic Resume

- 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

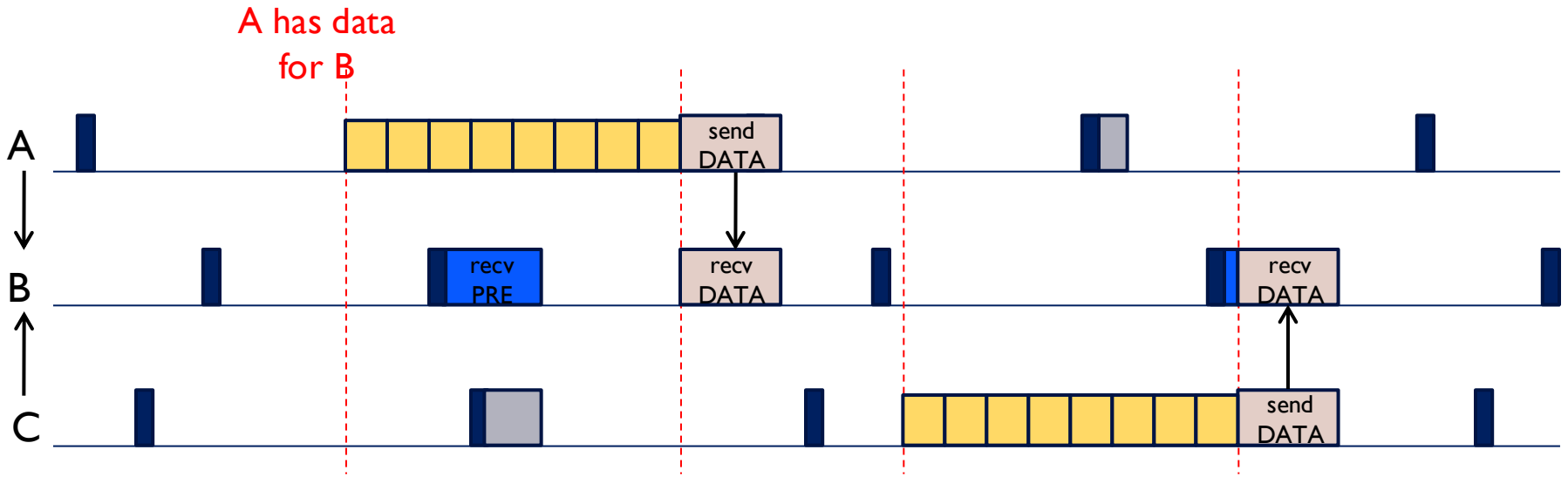
Preamble Sampling: Long Preamble (e.g., B-MAC)

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



Preamble Sampling: Wake-up Packets (e.g., X-MAC)

- 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



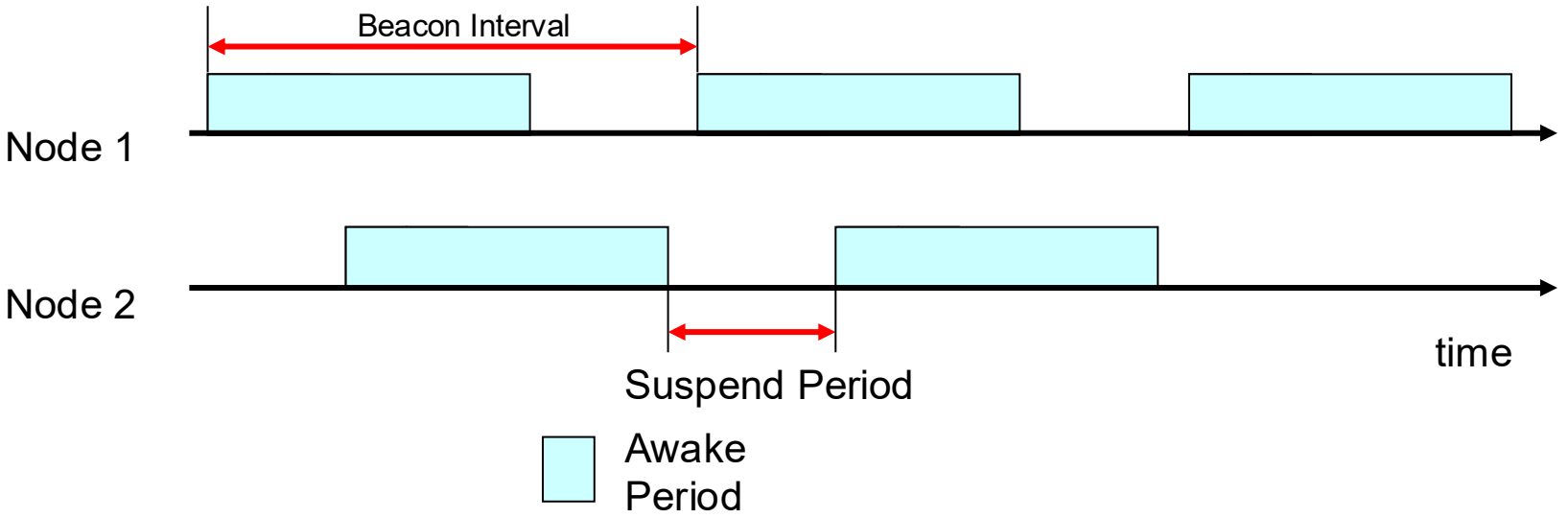
Preamble Sampling: Multiple Send

- 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

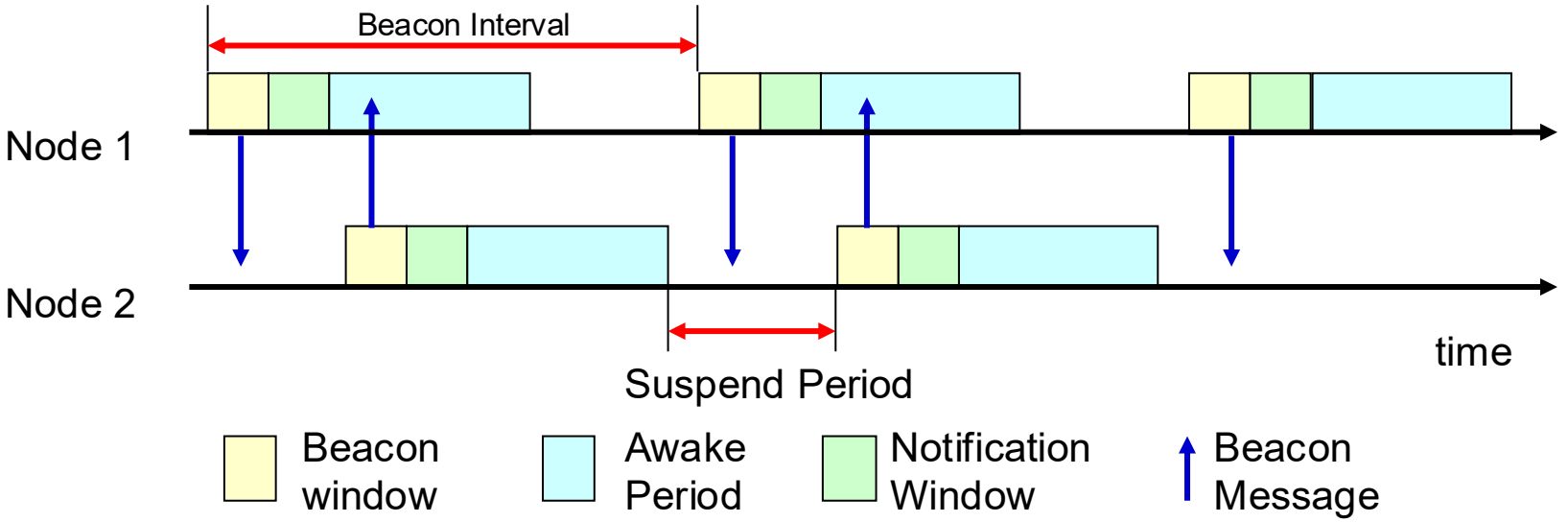
Asynchronous Periodic Resume

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



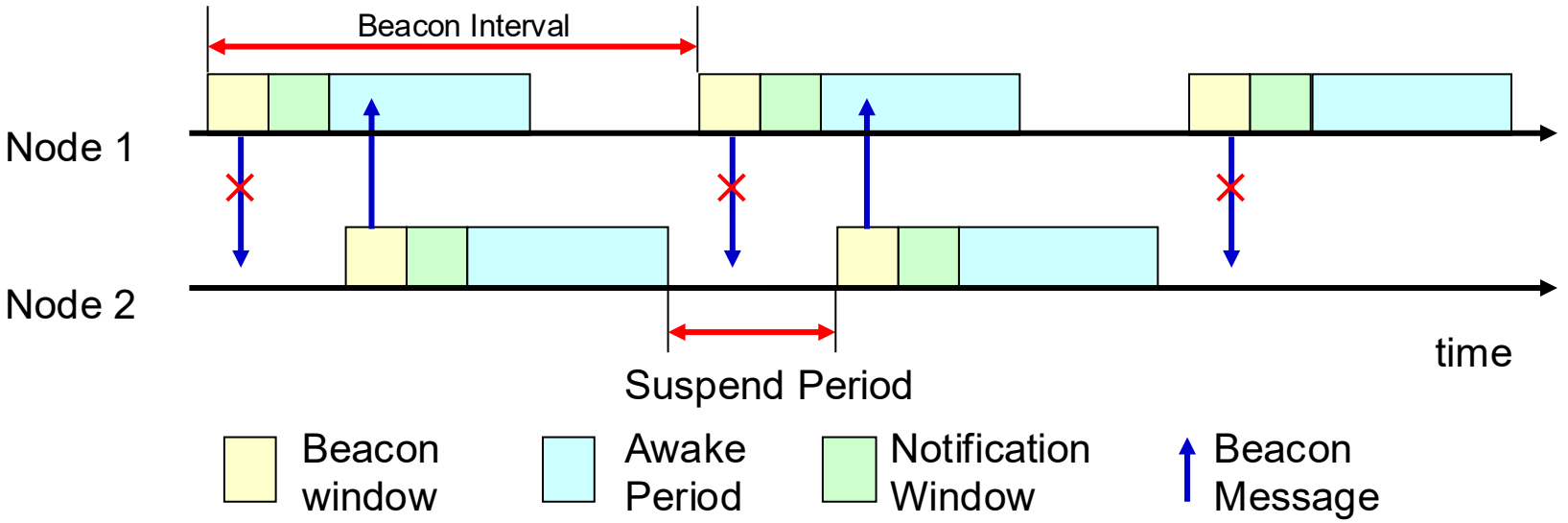
Asynchronous Periodic Resume

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



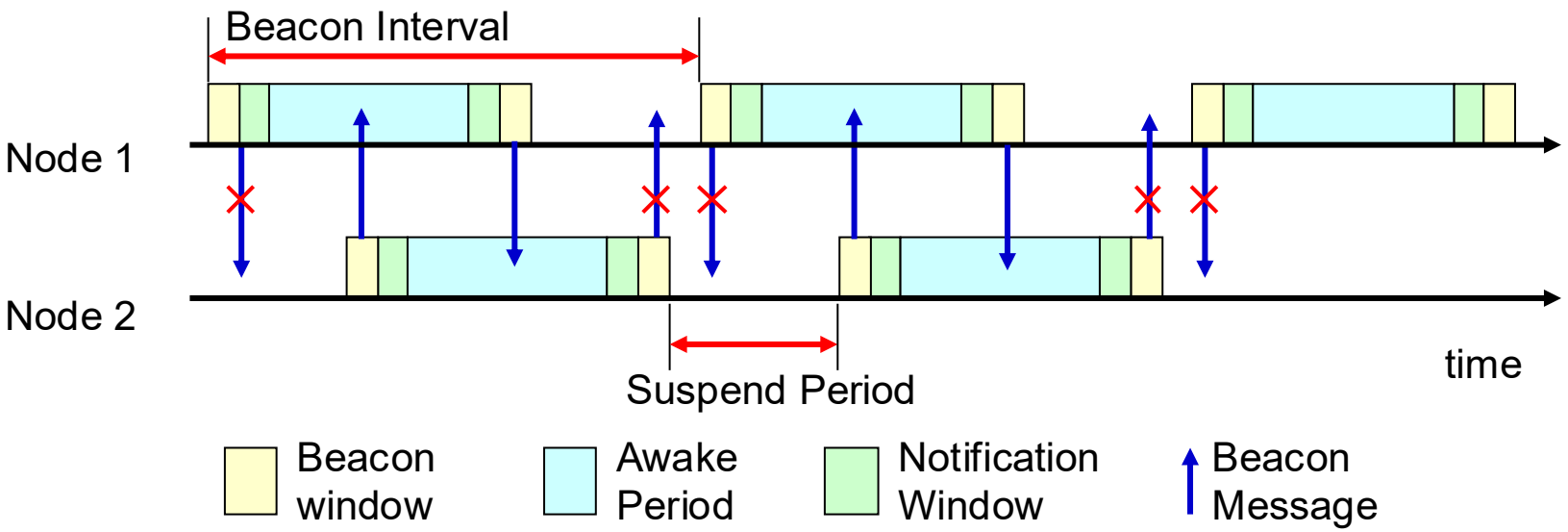
Asynchronous Periodic Resume

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



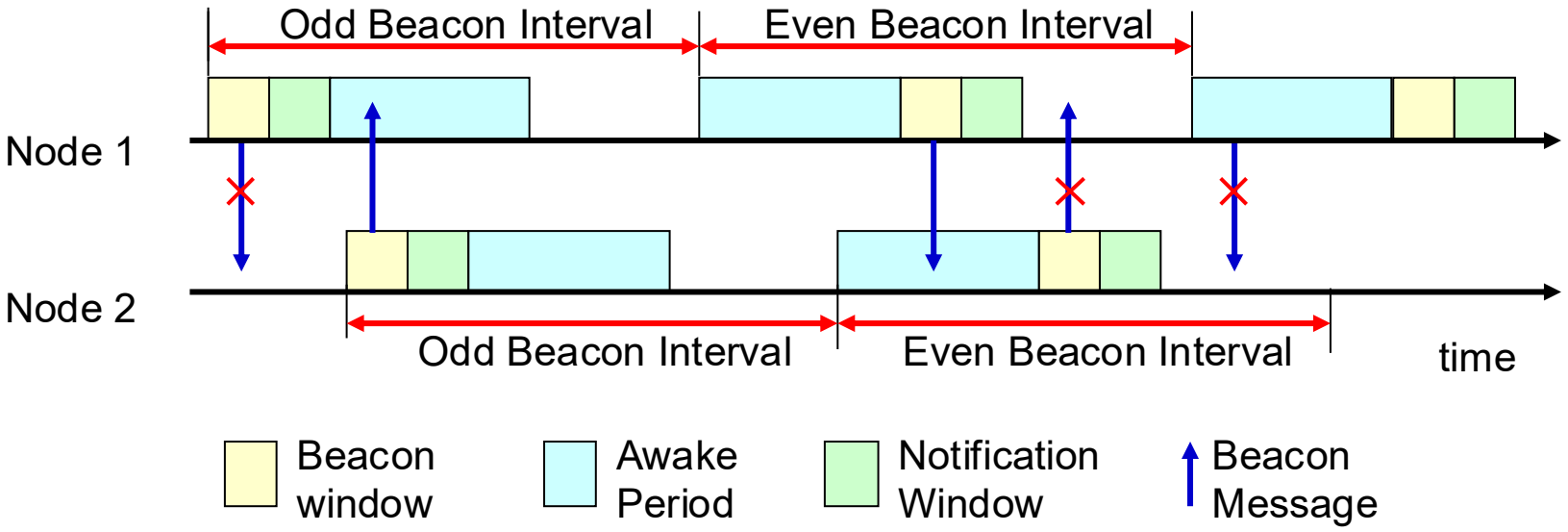
Asynchronous Periodic Resume

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



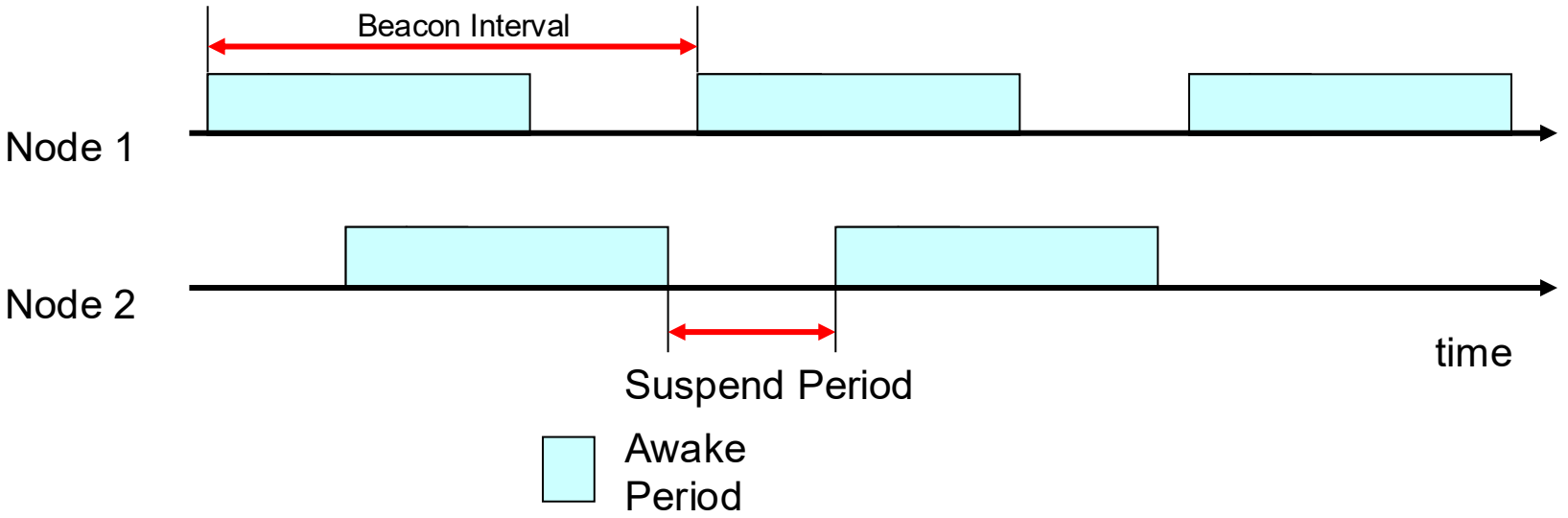
Asynchronous Periodic Resume

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



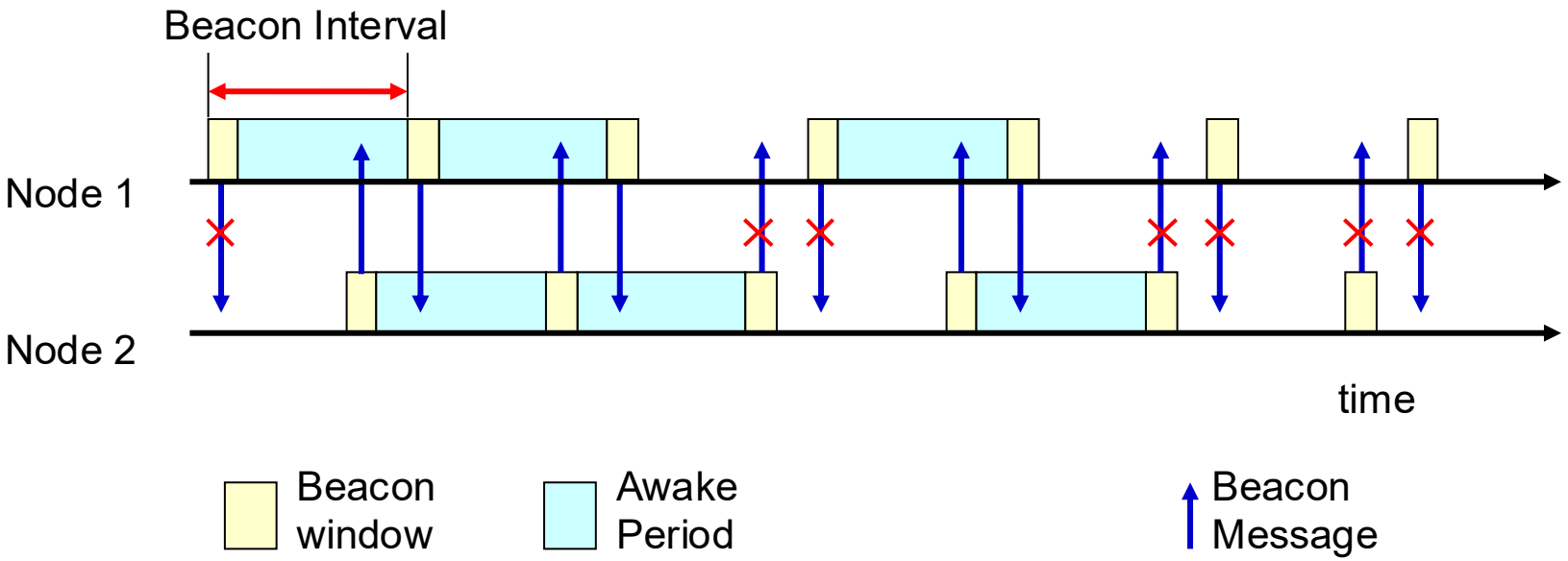
Asynchronous Periodic Resume

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



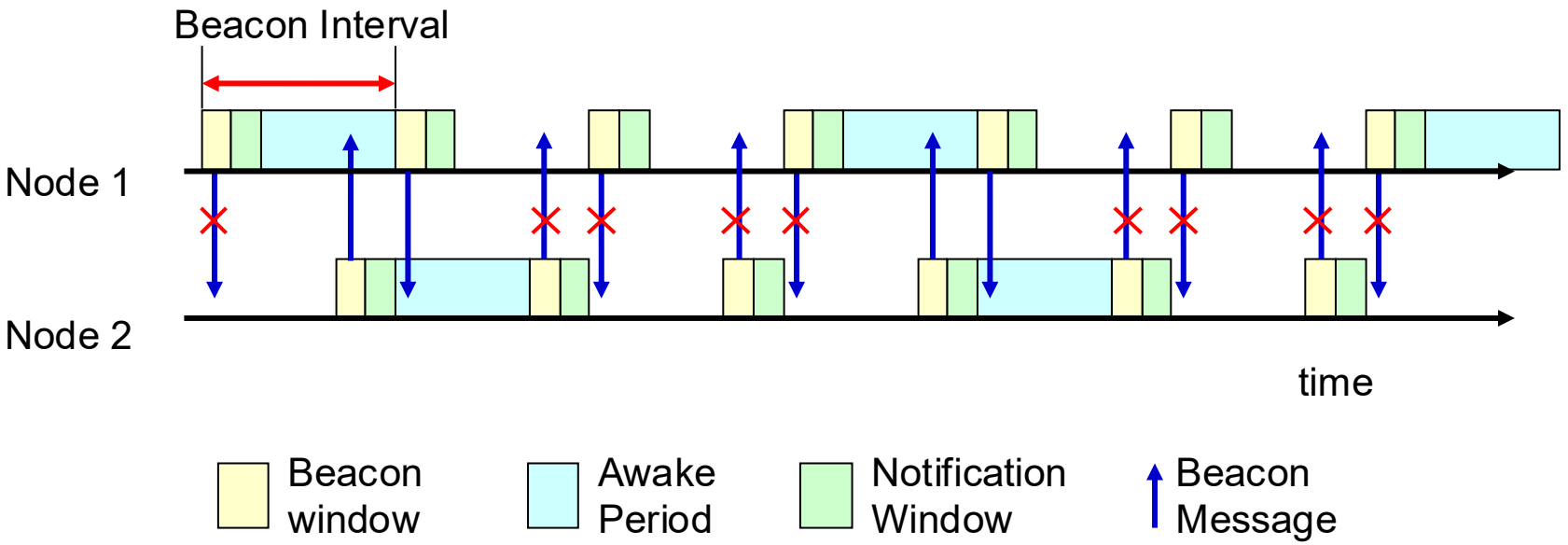
Asynchronous Periodic Resume

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



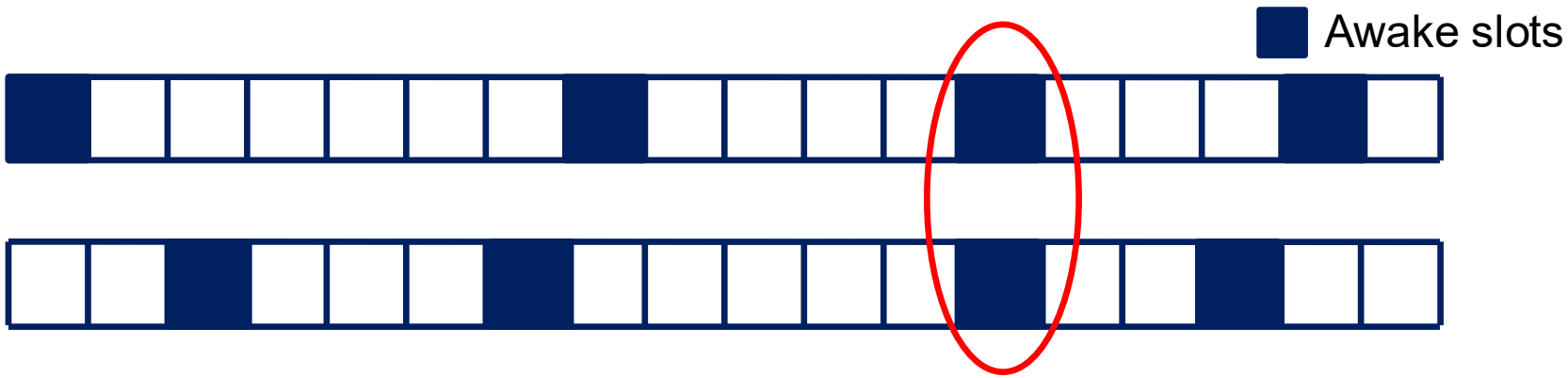
Asynchronous Periodic Resume

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



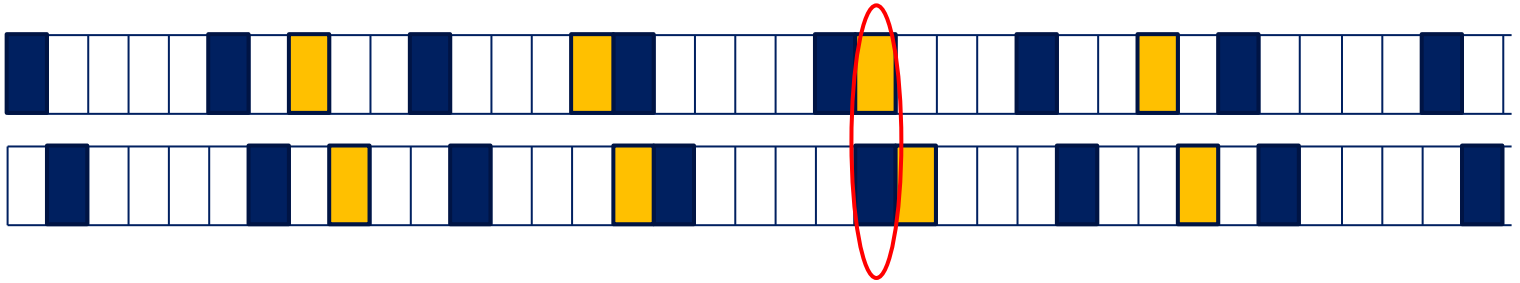
Asynchronous Periodic Resume

- 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



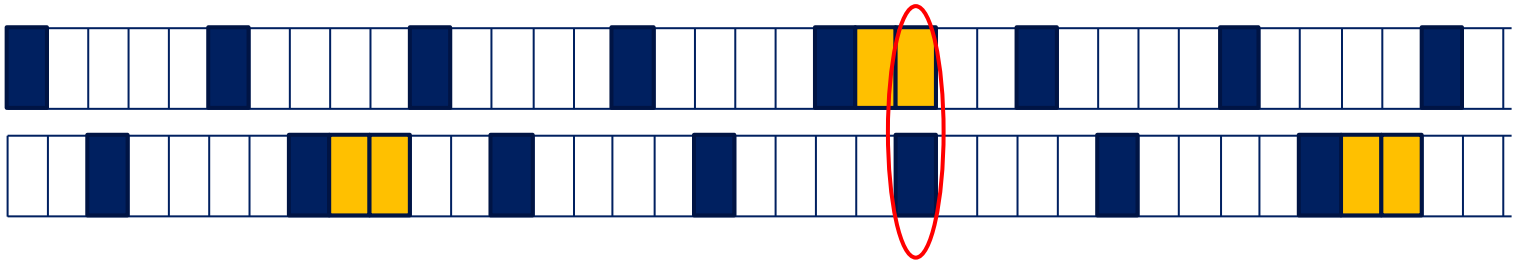
Asynchronous Periodic Resume

- Deterministic: Prime-based
  - Disco
    - Pick two primes p1 and p2
    - Wake up every p1 and p2 slot
    - Guarantees discovery in p1 x p2 slots



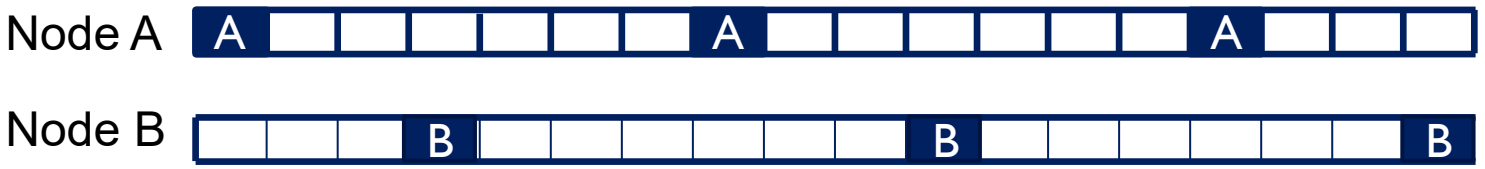
Asynchronous Periodic Resume

- Deterministic: Prime-based
  - U-Connect
    - Select 1 prime p
    - Wake up every pth slot and (p+1)/2 consecutive slots every p*p slots
    - Overlap is guaranteed within p2 slots



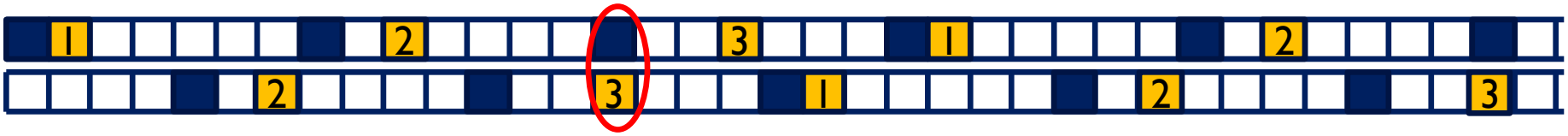
Asynchronous Periodic Resume

- Searchlight
  - Have a deterministic discovery schedule that has a pseudo-random component



Asynchronous Periodic Resume

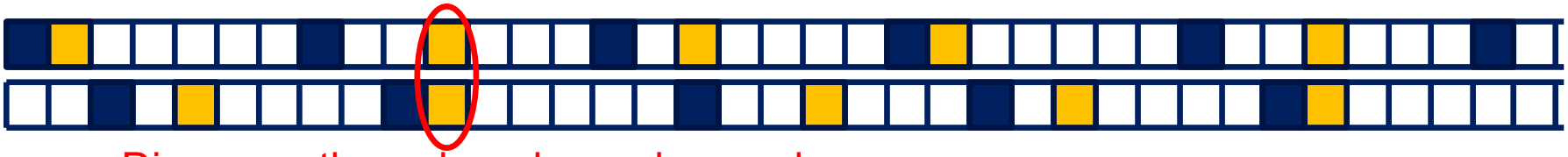
- Searchlight
  - Two slots per t slots (period)
    - Anchor slot: Keep one slot fixed at slot 0
    - Probe slot: Move around the other slot sequentially
  - Guaranteed overlap in t*t/2 slots
    - Based on the time needed to ensure a probe-anchor overlap
  - Probe-probe overlap can also lead to discovery
    - 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*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

© CS 439 Staff, University of Illinois

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

Infrastructure PSM: TIM and PS-Poll

- Station announces it is dozing
  - Sets the Power Management bit in a frame header (e.g., a null data frame)
  - AP now buffers frames for that station

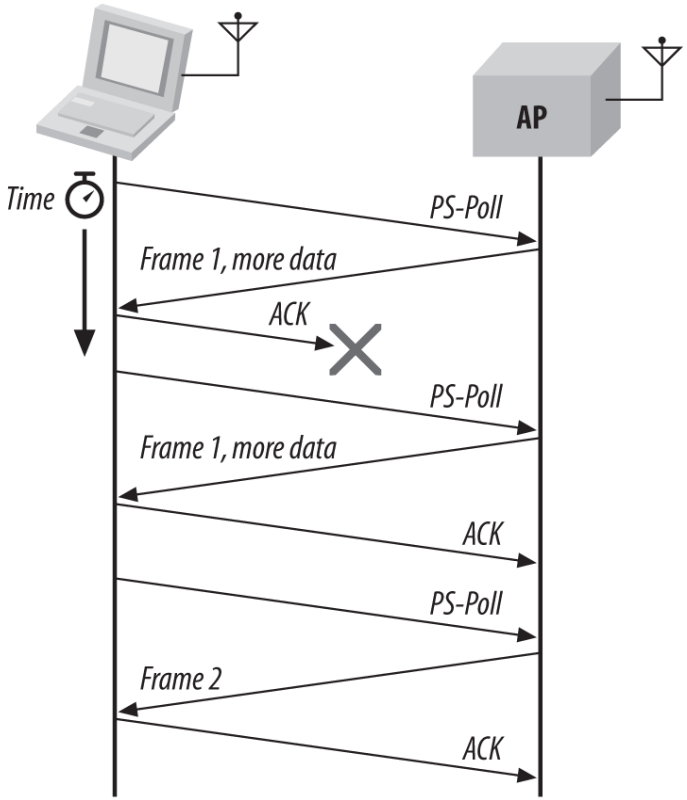
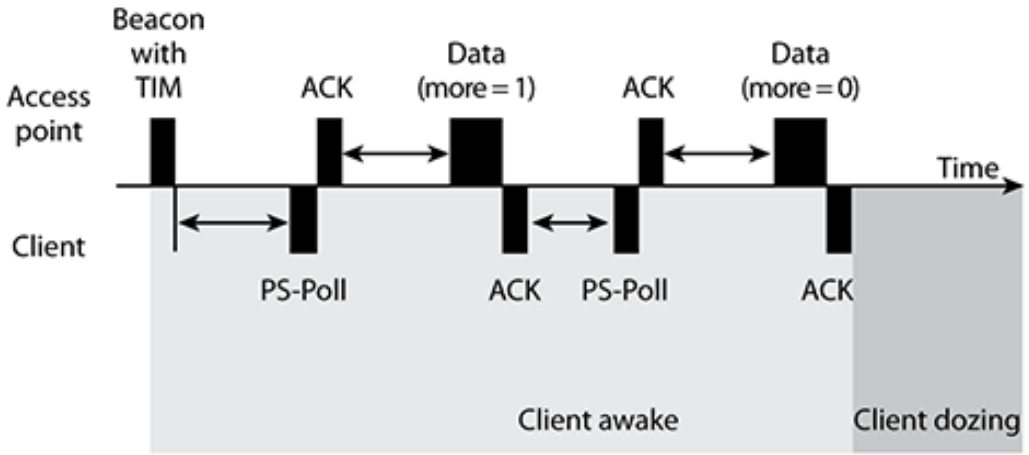


Figure 8-12. PS-Poll frame retrieval

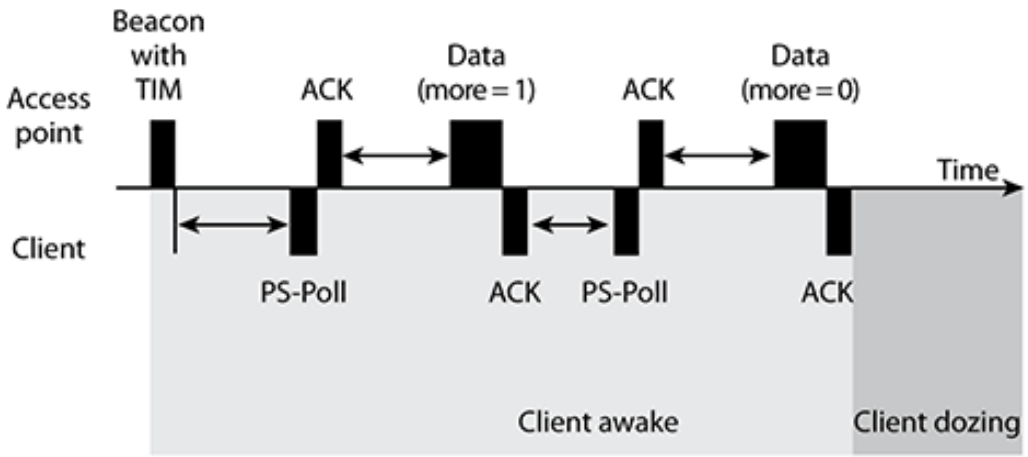
Infrastructure PSM: TIM and PS-Poll

- AP advertises buffered traffic
  - Every beacon carries a TIM
    - Bitmap of AIDs with buffered frames
  - Station wakes for beacons (every one, or every Nth via its listen interval)



Infrastructure PSM: TIM and PS-Poll

- Retrieval
  - Bit not set
    - Go back to sleep
  - Bit set
    - Send PS-Poll
    - AP returns a buffered frame
    - More Data bit says whether to keep polling



Multicast, DTIM, and Listen Interval

- Broadcast/multicast frames are not polled per station
  - AP buffers them and sends right after a DTIM beacon
  - DTIM period: every N beacons (set by the AP)
  - Stations that want multicast must wake at every DTIM beacon

Multicast, DTIM, and Listen Interval

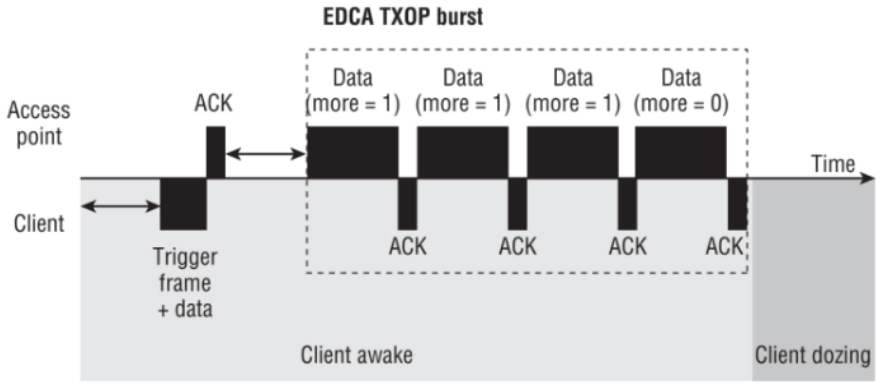
- Tradeoff
  - Larger DTIM period
    - More sleep, more multicast delay
  - Beacon interval is typically 100 TU (about 102 ms)
- Listen interval: same tradeoff for unicast
  - Skipping beacons saves energy but adds latency, and the AP must buffer longer

Staying Associated While Asleep

- The AP drops stations that stay silent too long
  - Frees the AID and its buffered frames
- BSS Max Idle Period
  - The AP tells the station how long it will wait before removing it
  - A sleeping station must send some frame (a keepalive) before it expires
- Tradeoff
  - Long sleep needs a long idle period
  - Short sleep forces extra wake-ups
  - Application and NAT keepalives wake the radio too

Unscheduled Automatic Power Save Delivery

- U-APSD: event-driven delivery
  - Station sends a trigger frame (data or QoS Null)
  - AP replies with a burst of buffered frames (a service period)
  - EOSP bit marks the last one

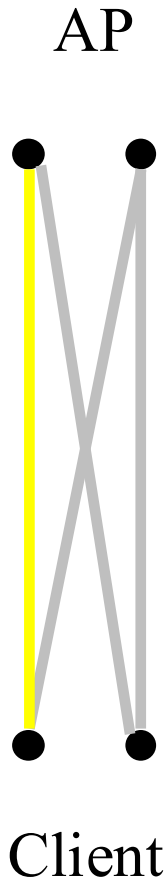


Unscheduled Automatic Power Save Delivery

- U-APSD: event-driven delivery
  - Station sends a trigger frame (data or QoS Null)
  - AP replies with a burst of buffered frames (a service period)
  - EOSP bit marks the last one
- Key Benefits
  - Massive Energy Savings
    - Reduced listening costs
    - Reduced wake up for polling
  - Reduced Latency
    - Immediate request for queued packets
- Optimized for Voice over Wi-Fi (VoIP)
  - Voice calls send data predictably in both directions (e.g., every 20ms)
  - Client can perfectly synchronize sending and receiving in one awake window.

Radio-Level Power Save (802.11n)

- SM-PS (802.11n): power down all but one receive chain
  - Device acts as 1x1, so peak rate drops
  - Static
    - Tells the AP with an SM Power Save action frame
  - Dynamic
    - AP sends an RTS first
    - Device wakes its chains and answers with a CTS
- PSMP
  - Scheduled multi-poll of uplink and downlink
  - Rarely implemented



Radio-Level Power Save (802.11ac)

- Very High Throughput Transmit Opportunity Power Save (VHT TXOP PS)
  - Doze during a TXOP that is not for you
    - PHY header tells client the TXOP is for it
    - All nearby clients wake up just long enough to capture and read the tiny preamble packet
    - Longer TXOPs mean longer sleep

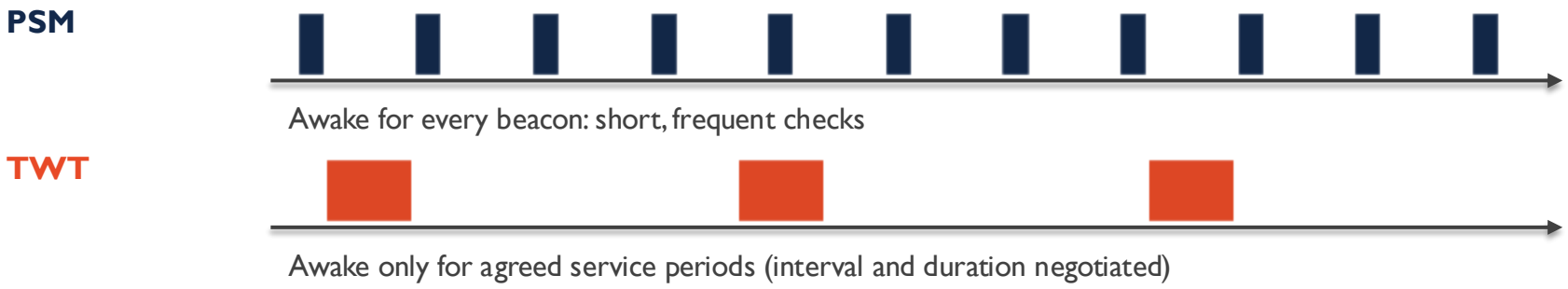
Limits of Legacy Power Save

- With dense networks and high traffic load, prior methods suffer
  - High packet loss
  - High MAC-layer overhead
  - High chance of collisions

- Intuition for improvements for power saving
  - Currently transmitting/receiving STAs need to be awake
  - A client can only transmit/receive frames when awake

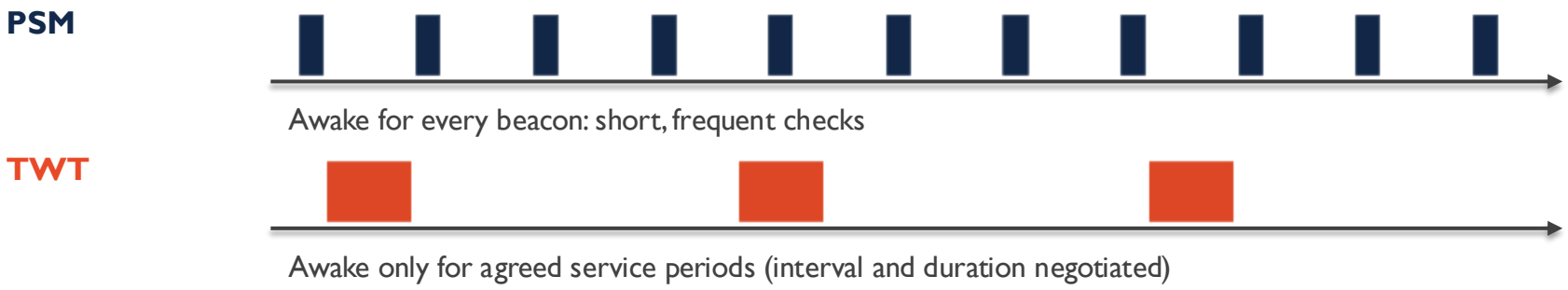
Target Wake Time (TWT)

- Agreed wake-up schedule between client and AP
  - Not every beacon
  - Client wakes only at times both sides agreed on
- The agreement
  - First wake time, wake interval (how often), wake duration (how long)



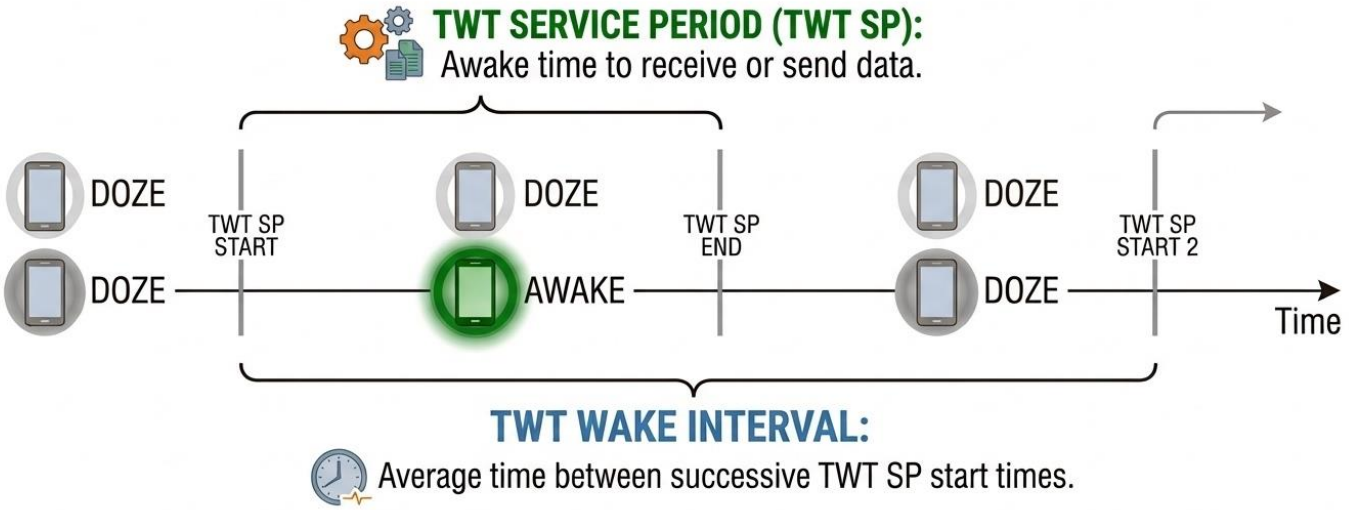
Target Wake Time (TWT)

- Between service periods, client dozes and the AP holds its traffic
  - Longer sleep and less contention (the AP can stagger clients)
  - Defined in 802.11ah, extended in 802.11ax



Target Wake Time (TWT)

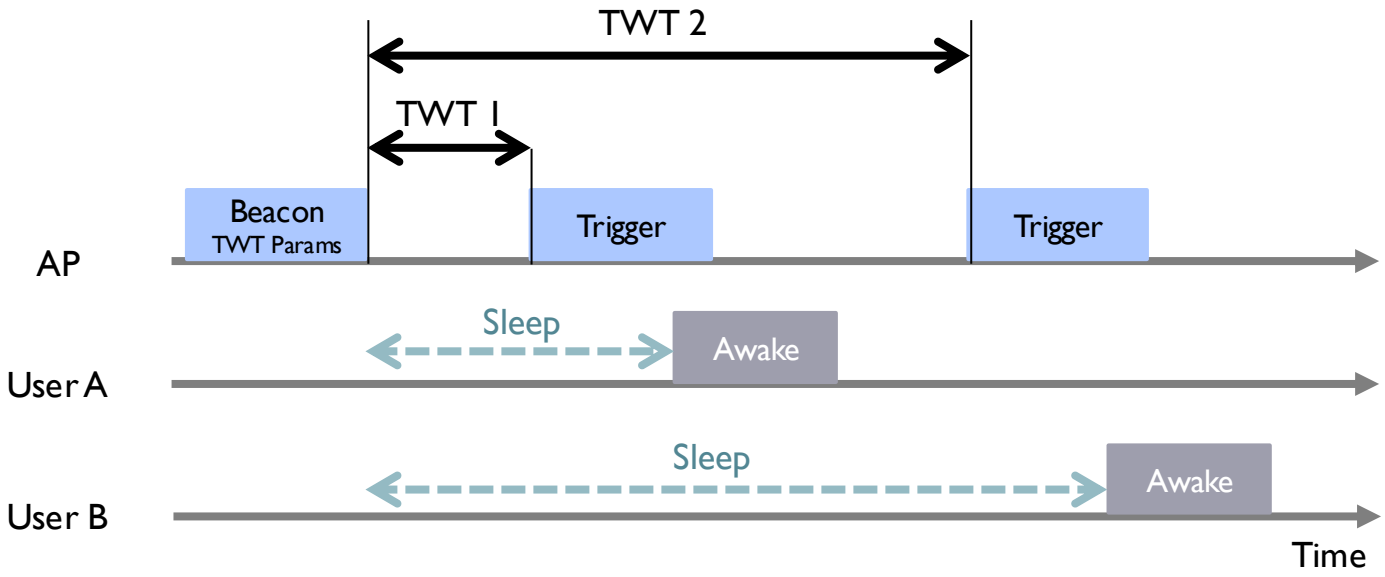
- TWT Service Period (TWT SP)
  - Awake period to receive or send data
- TWT Wake Interval
  - Average time between successive TWT SP start times



Target Wake Time (TWT)

- TWT Channel
  - Channel the client treats as its primary during the service period
- TWT Agreement
  - Final agreement between AP and client
    - TWT Wake Interval
    - TWT Service Period (SP)
    - TWT Channel Configuration
    - Trigger Mode

TWT Visualized



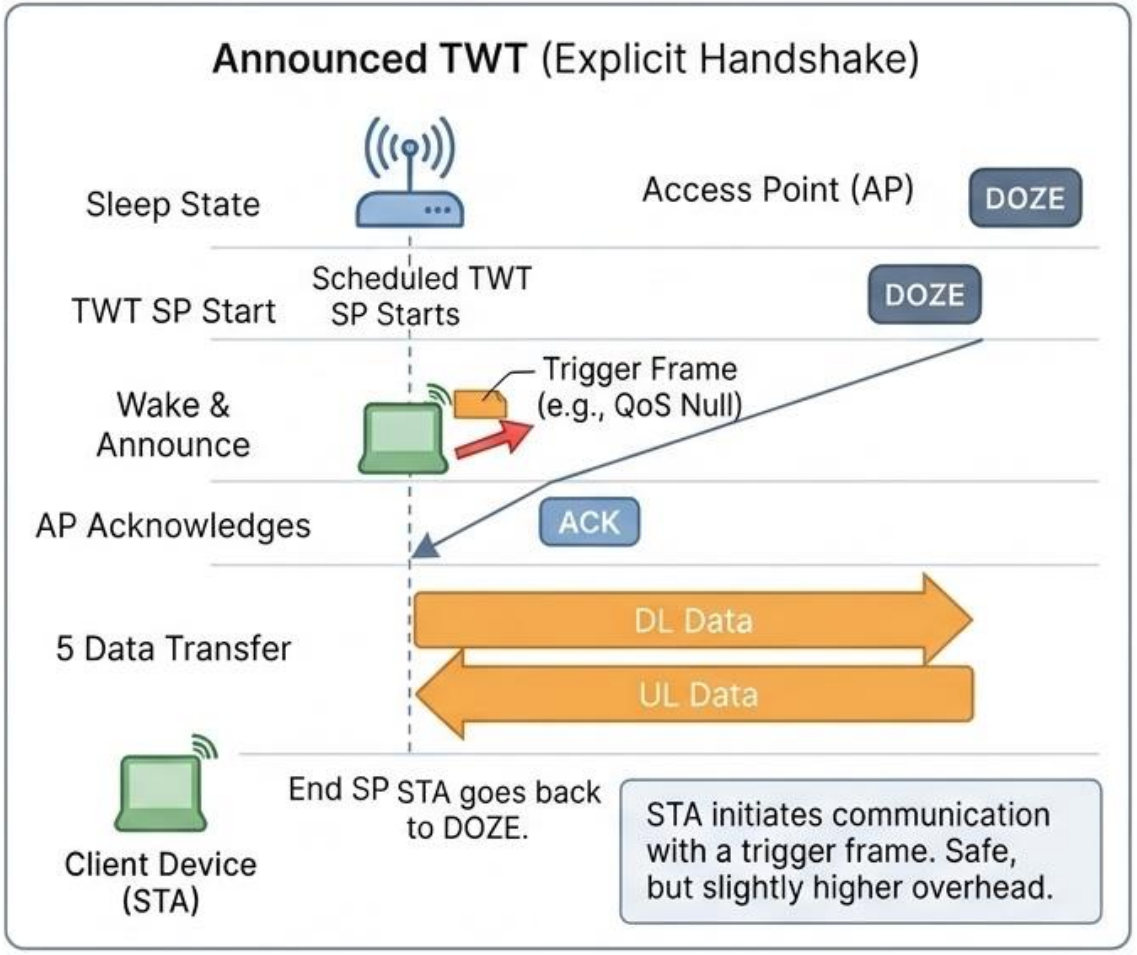
Devices transmit/receive data while awake after confirmed with AP

Target Wake Time (TWT)

- TWT Service Periods (SPs) can be individually agreed or broadcast
  - Individually agreed → Between 2 devices
  - Broadcast SPs are set by the AP and announced in beacons

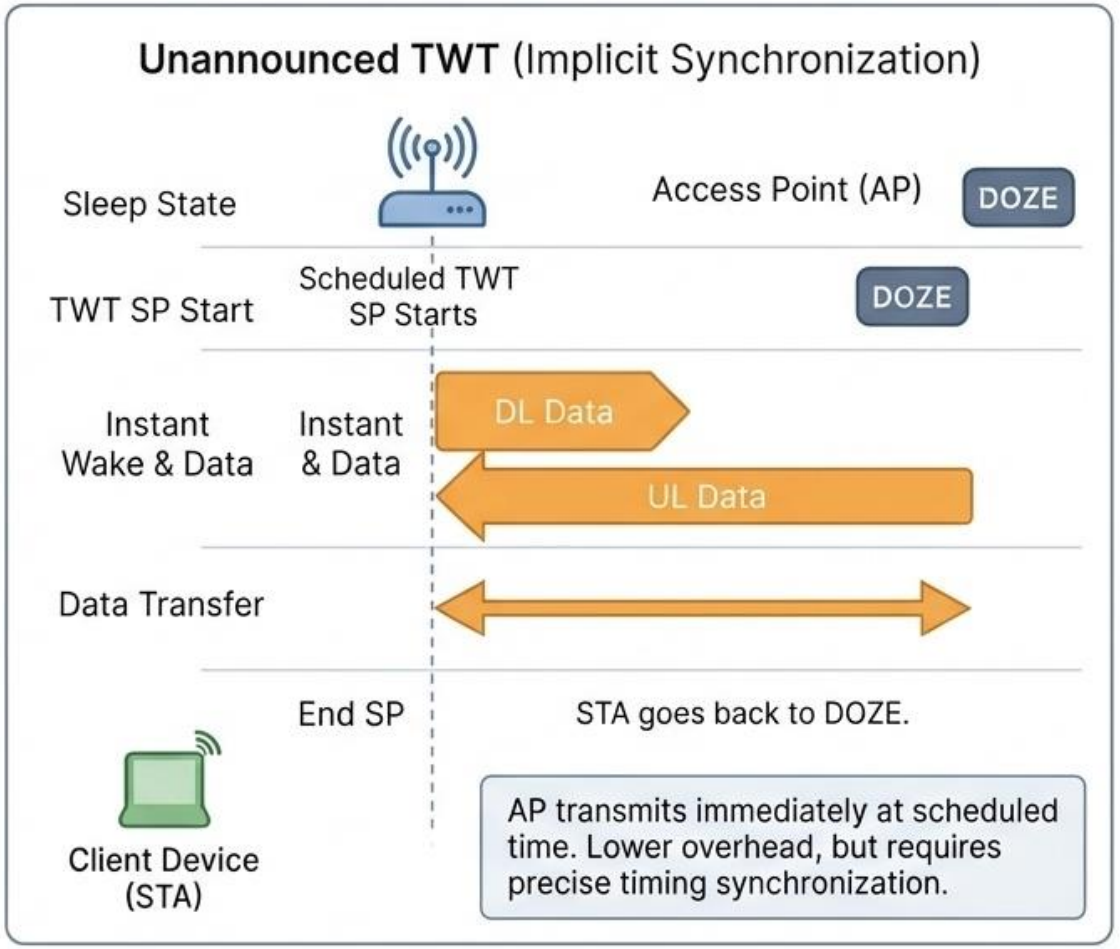
TWT Setup Options

- Announced
  - Station sends awake signal (e.g., PS-Poll or QoS Null
  - AP waits to send data
- Pros
  - Safer
  - Ensures the AP never transmits while the client is still dozing or off-sync
- Cons
  - Requires sending an extra frame per SP



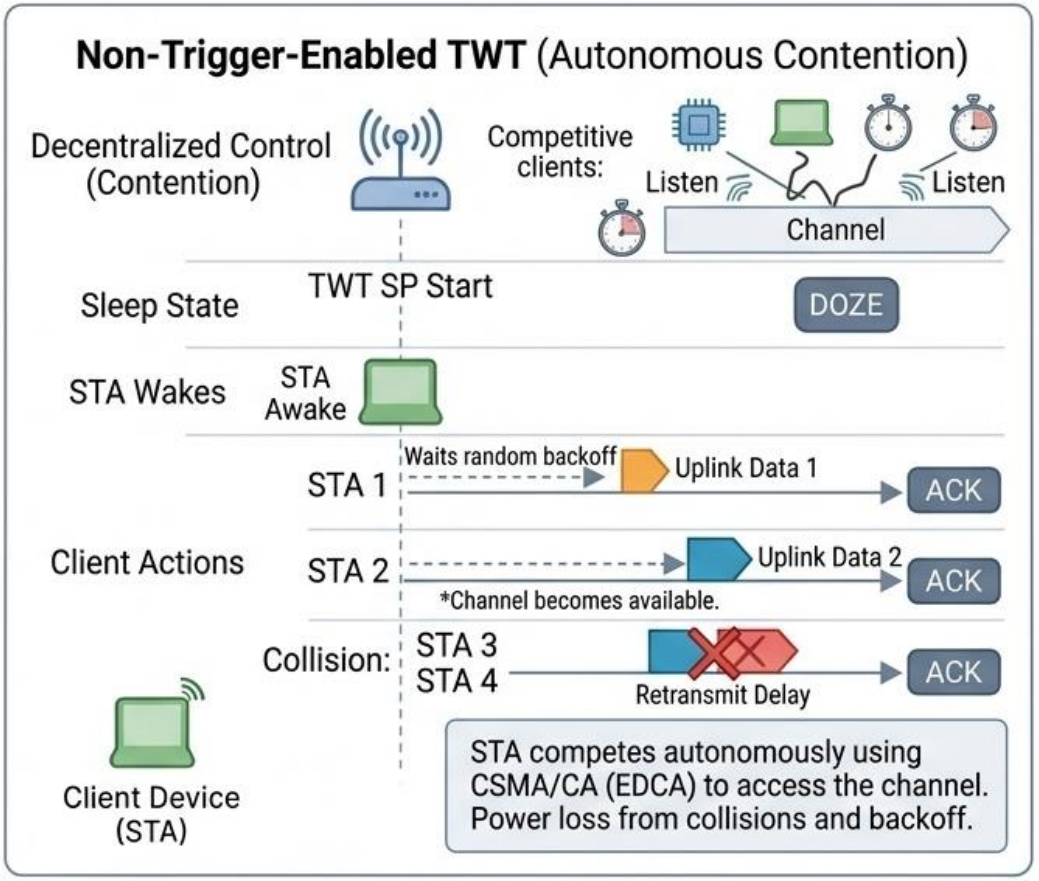
TWT Setup Options

- Unannounced
  - AP assumes client is awake
  - AP sends at start of the service period without a signal
- Pros
  - Saves wireless overhead
  - Eliminate initial handshake frame
- Cons
  - Relies entirely on strict timing synchronization between AP and client



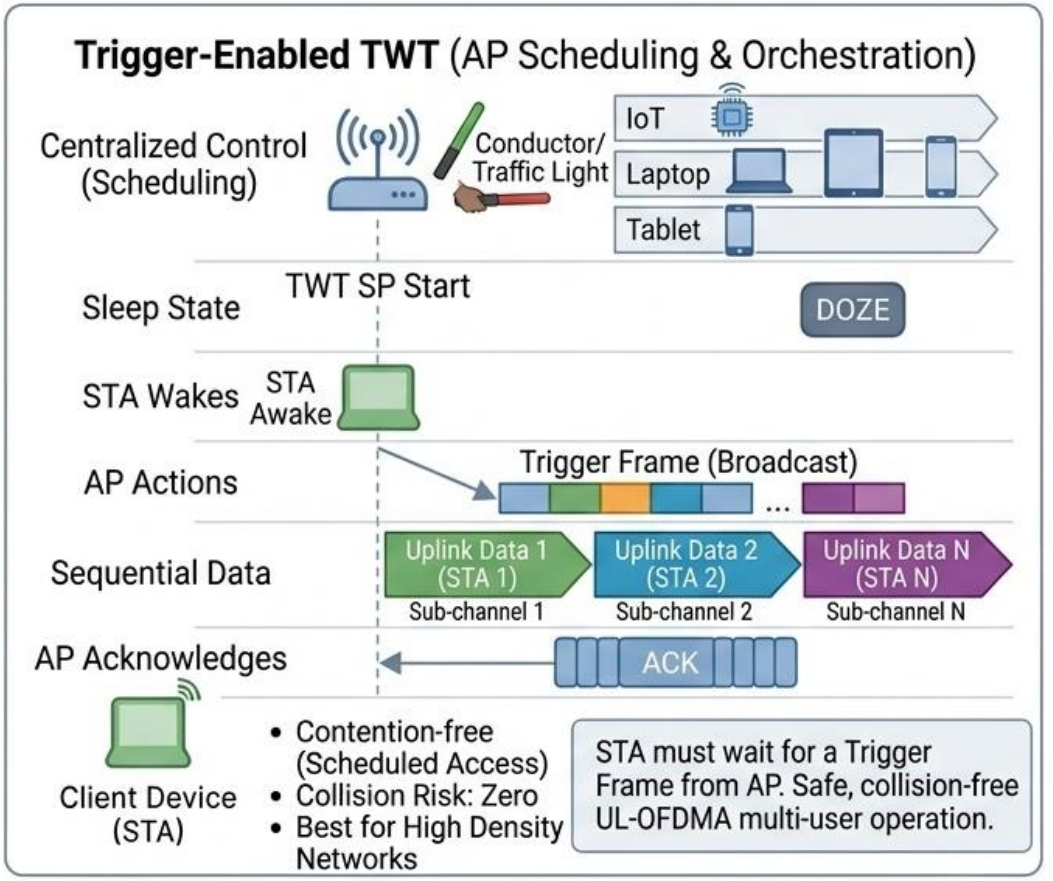
TWT Setup Options

- Non-trigger-enabled
  - Client wakes up and competes for the channel
- Pros
  - Lower latency for the client
- Cons
  - Higher risk of packet collisions and retransmissions



TWT Setup Options

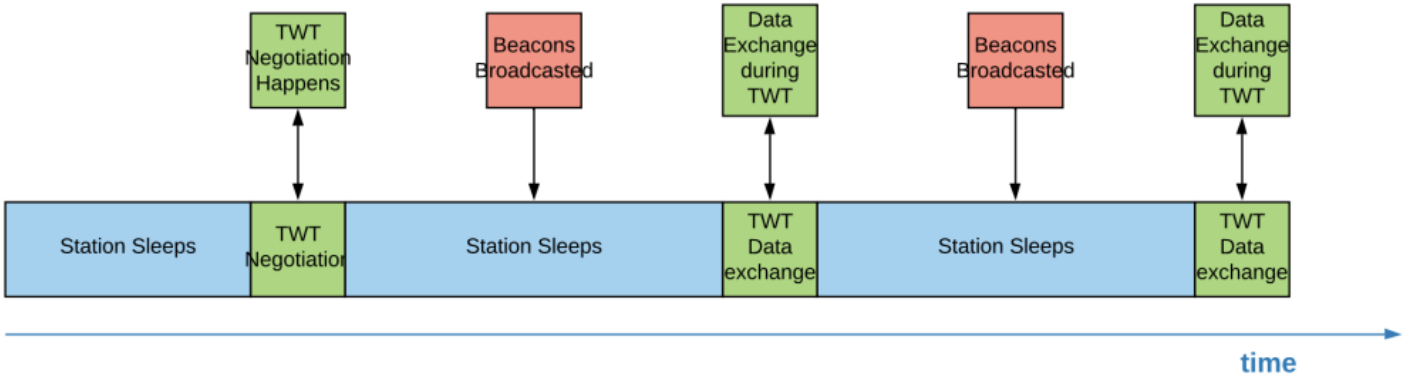
- Trigger-enabled
  - Device wakes up and waits
  - AP sends Trigger frames to schedule OFDMA transmissions
- Pros
  - Eliminates wireless collisions
- Cons
  - Client is dependent on the AP's timing



TWT Setup Options

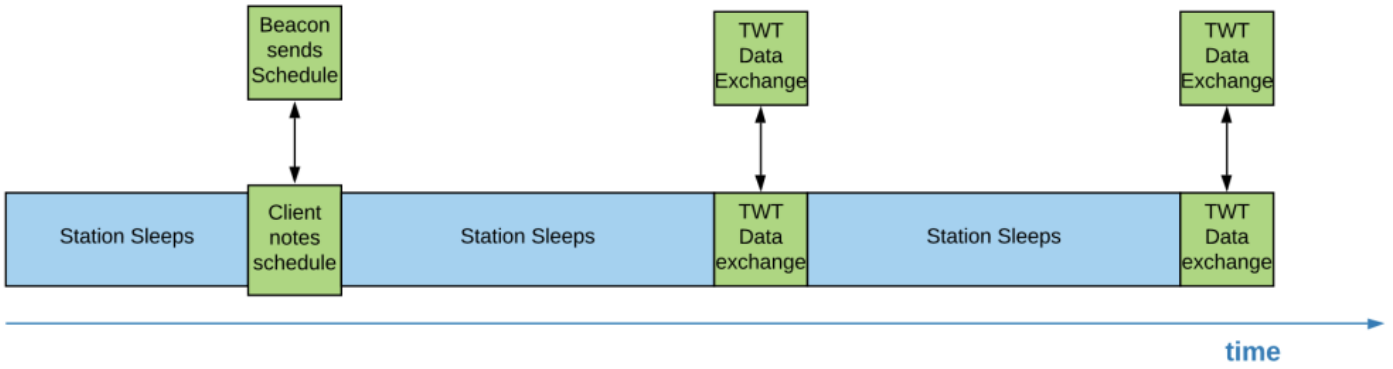
- Flow identifier
  - A station can hold several TWT agreements at once (flow IDs 0-7)
- Teardown
  - Either side can end an agreement and renegotiate

Individual Target Wake Time (TWT)



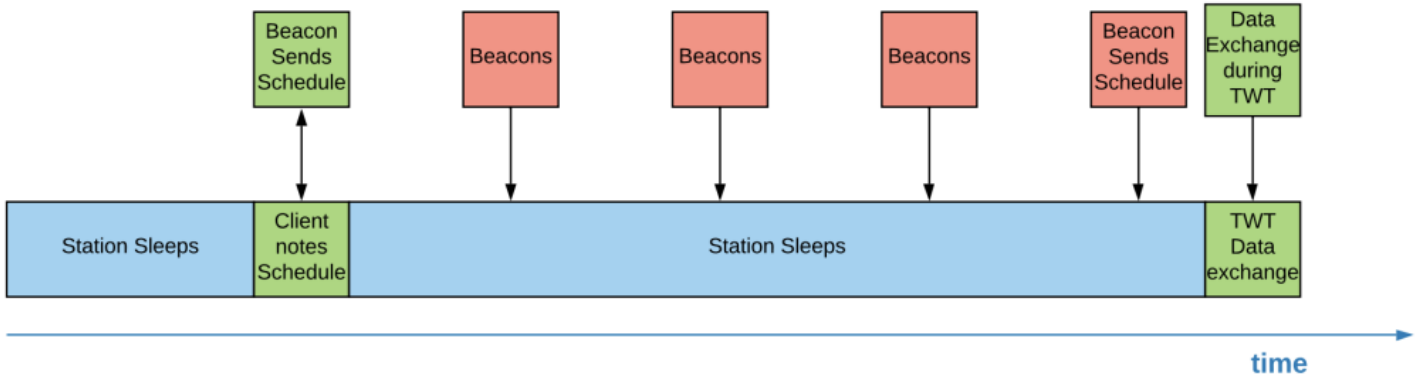
- Client sends waking schedule information to AP
- AP devises a schedule and sends TWT values to client
- The client wakes up and transmits according to the schedule
- Explicit Mode
  - AP sends client the next TWT information
  - Client wakes up at the next scheduled time
- Implicit Mode
  - Client calculates next TWT by adding a fixed value to the current TWT

Broadcast Target Wake Time (TWT)



- Hybrid PSM and TWT
  - 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
  - AP defines the schedule; stations simply join it

Broadcast TWT: Scheduling and Triggers



- 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 trigger frames during the service period to schedule uplink and downlink transmissions
  - AP sends frames to awake clients
  - Clients can doze after transmission

Target Wake Time Impact

- 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 coordination of wake-up times (only CSMA/CA)
  - 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 wake for beacons (every Nth beacon via listen interval) to check the TIM for buffered traffic
  - Wake-up granularity is tied to the ~100 ms beacon interval, and the station must still contend after waking
- TWT
  - Diverse sleep cycles
  - Longer, negotiated sleep cycles
  - Sleep cycle length independent of the standard beacon interval

Restricted TWT (R-TWT)

- Regular TWT
  - Schedules a wake time, but doesn't protect it
  - Other traffic can still cause delay
- R-TWT
  - Reserves specific time windows for latency-sensitive traffic (gaming, calls, AR/VR)
  - Clients outside the R-TWT agreement are expected to stay quiet during an announced window
- Comparison
  - TWT tells devices when to wake up
  - R-TWT guarantees the channel is theirs when they do

Power Save Under Multi-Link Operation (MLO)

- MLO (802.11be)
  - Devices stay associated on multiple links (2.4/5/6 GHz) at once
  - Power save needs to span all of them
- Device can't just "sleep" the way single-link PSM/TWT does
  - Different links may have different traffic, different TWT schedules, different radio states
  - Which link(s) to keep awake
  - When is it safe to power down a given radio without missing traffic

Power Save Under Multi-Link Operation (MLO)

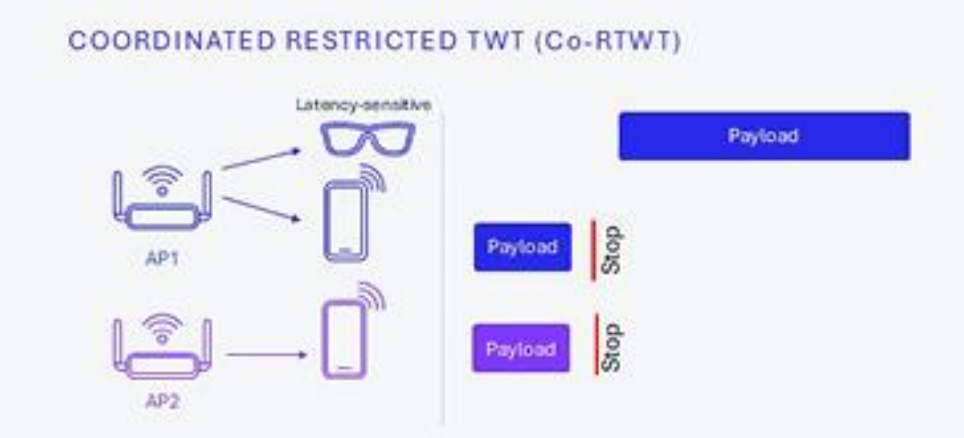
- Link capability constrains power save
  - STR (Simultaneous Tx/Rx)
    - Independent radios per link
    - Each link can sleep and wake separately
  - NSTR (Non-Simultaneous Tx/Rx)
    - Links interfere with each other
    - One link's activity limits the others and sleep decisions are coupled
- Per-link TWT and TID-to-link mapping let a device doze links it does not need

MLO Power Save: EMLSR

- STR on every link needs a radio chain per link, which costs power and money
- EMLSR (Enhanced Multi-Link Single Radio)
  - Client listens on several links in a low-power, limited listening mode
  - AP sends an initial control frame (e.g., MU-RTS) on one link
  - Client switches its full radio (all antennas) to that link for the exchange, then returns to listening
- Result
  - Most of the multi-link latency benefit at close to single-radio power
  - EMLMR is the multi-radio variant, with fewer constraints but more hardware

Wi-Fi 8: Multi-AP Coordination

- Co-RTWT: Coordinated Restricted TWT
  - Neighboring APs align their R-TWT service periods across overlapping networks
  - Each AP's reserved low-latency window stays protected from the others' traffic



802.11ah: Wi-Fi HaLow (Sub-1 GHz)

- Sub-1 GHz unlicensed bands (900 MHz region in the US)
- Lower frequency
  - Much better range and wall penetration
  - Lower data rate
  - Range: up to ~1 km outdoors
- Designed for IoT
  - Sensors, smart metering, industrial monitoring
- Similar protocols
  - Zigbee/802.15.4
    - HaLow trades some power efficiency for a native IP/Wi-Fi stack
    - Easier to integrate with existing networks
  - LoRa
    - HaLow has higher throughput and lower latency, but shorter range

802.11ah: HaLow MAC

- CSMA/CA based
- Restricted Access Window (RAW)
  - The AP divides time into windows and assigns groups of stations to specific RAW slots
  - Only a subset of stations contend for the channel at once
  - Supports up to 8,191 stations per AP
- TIM segmentation / hierarchical grouping
  - Stations are organized into pages and blocks
  - TIM doesn't need an entry for every station on every beacon
- TWT - Target Wake Time (originated in 802.11ah)
  - Improve energy conservation
  - Sensors can sleep for long, negotiated intervals between wake-ups