Energy Conservation Techniques

- Wi-Fi devices consume significant amounts of energy when idle
  - Idle > 1W

- Conservation Approach: Device suspension (sleep)
  - Reduced energy consumption
    - Sleep ~ 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 $\Rightarrow$ 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
Triggered Resume

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

![Diagram showing node A, B, C, D, and E with awake/listen, RTS/CTS, and data transmission/reception states.][1]
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

[Diagram showing two nodes: one labeled "Wakeup plane: $f_1$" and the other "Data plane: $f_2$".]
Triggered Resume

- Dual radio

- Node A - control
- Node A - data
- Node B - control
- Node B - data

Legend:
- Awake/listen
- Transmit request
- Receive and reply
- Data transmission/reception

Time
Triggered Resume

- 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

![Diagram of awake slots](image)
Synchronous Periodic Resume

- **Protocol:** IEEE 802.11 Power Save Mode (PSM)
- Nodes are synchronized and wake up 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**

---

Node 1
- \(B_1\)  

Node 2
- \(B_2\)

**ATIM window**

**beacon interval**

**\(t\)**

- **awake**
- **Random Delay**
- **Transmit ATIM**
- **Transmit Data**
- **Beacon Frame**
- **Acknowledge ATIM**
- **Acknowledge Data**
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

A knows when B is awake

A has data for B

A sends data when B is awake

© CS/ECE 439 Staff, University of Illinois  Fall 2018
Distributed Synchronous Periodic Resume

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

A has data for B
Long preamble wakes up B

Overhearing

Unnecessary preamble
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

A has data for B
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

![Diagram showing the concept of asynchronous periodic resume with nodes 1 and 2, highlighting awake and suspend periods with respect to beacon interval.]
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!

![Diagram showing Beacon Interval and Suspend Period for Node 1 and Node 2. The diagram illustrates the time periods when nodes are awake and suspended.]
Asynchronous Periodic Resume

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

![Diagram showing beacon interval and awake periods for Node 1 and Node 2.](image-url)
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

- **Extended sleep**
  - Wake up once every $T$ intervals
  - Adds delay up to $T \times$ length of beacon interval

![Diagram of beacon interval, awake period, and notification window between nodes 1 and 2.]

© CS/ECE 439 Staff, University of Illinois Fall 2018
Asynchronous Periodic Resume

- **Quorum**
  - Increase number of beacon intervals in cycle \( n \)
  - Increase number of awake periods \( 2n - 1 \) of \( n^2 \)

Delay is determined by where the overlap is (worst case \( n^2 \))
Asynchronous Periodic Resume

- **Quorum**
  - Example: \( n = 4, n^2 = 16, 2n-1 = 7 \)
  - Two overlapping intervals: delay = \( n^2 - 2 \)

![Node i and Node j diagrams with colors indicating awake states]
Asynchronous Periodic Resume

- **Deterministic**
  - Find a feasible overlapping pattern
    - Guarantee at least one overlapping interval
    - Requires knowledge of number of nodes

![Awake pattern diagram](image)
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 $p^2$ 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 \( \frac{t^2}{2} \) 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
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