Design and Development of IoT Applications

Dr. –Ing. Vo Que Son

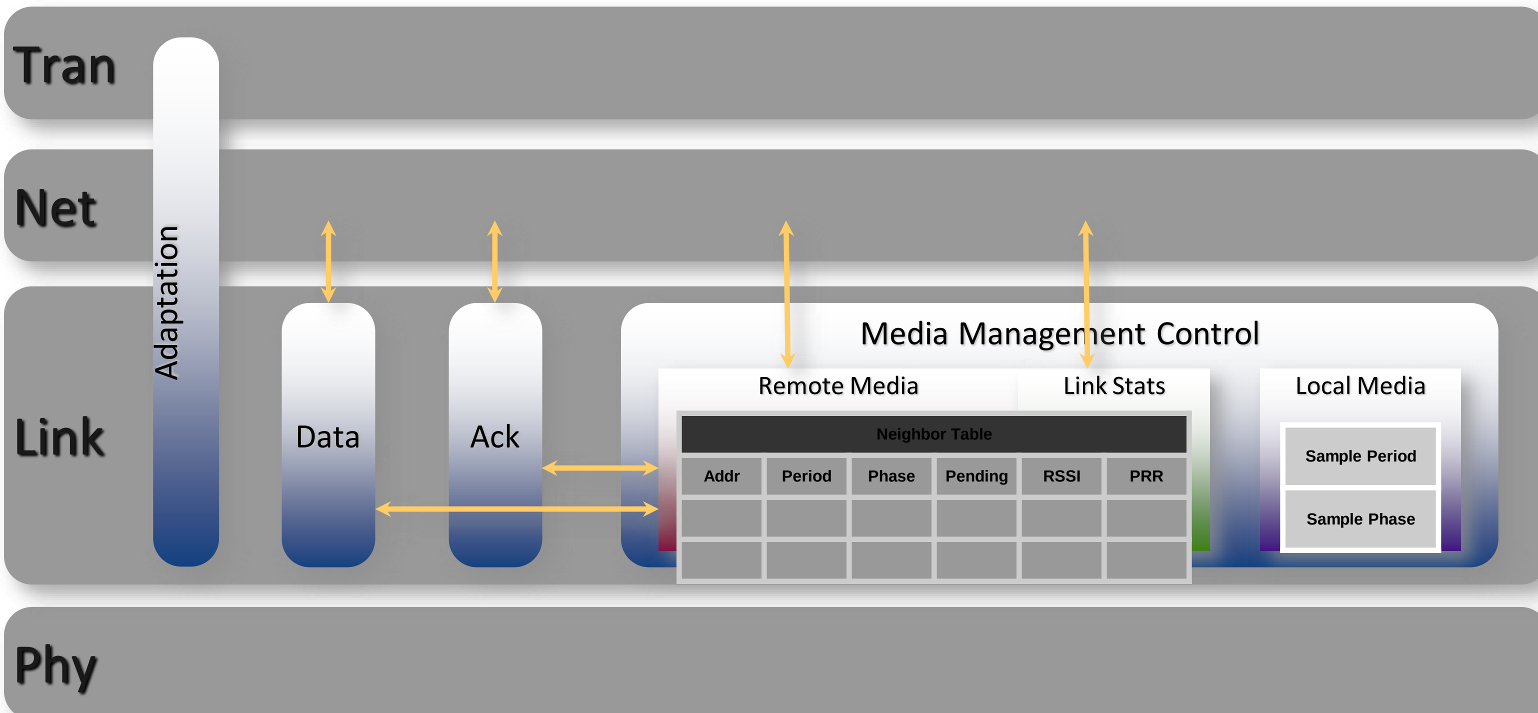
Email: sonvq@hcmut.edu.vn

Content

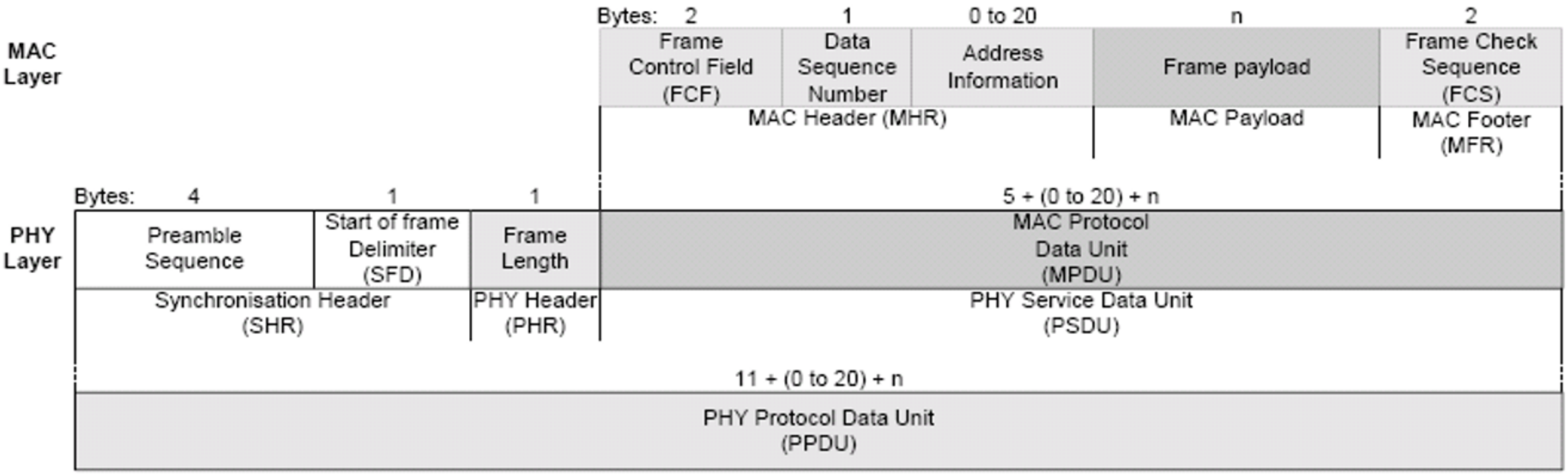
- Chapter 3: Embedded OS for end-devices
  - Intro to Contiki-OS
  - Programming using Contiki
  - I/O interfaces
  - Networking stack
  - Cooja Emulator
- Chapter 4: MAC protocols for WSNs
  - Low-power link
  - Robust communication
  - Radio Duty Cycling
  - Synchronized and Asynchronized Protocols

Building A Low-Power Link

- ❑Media Management Control – not MAC



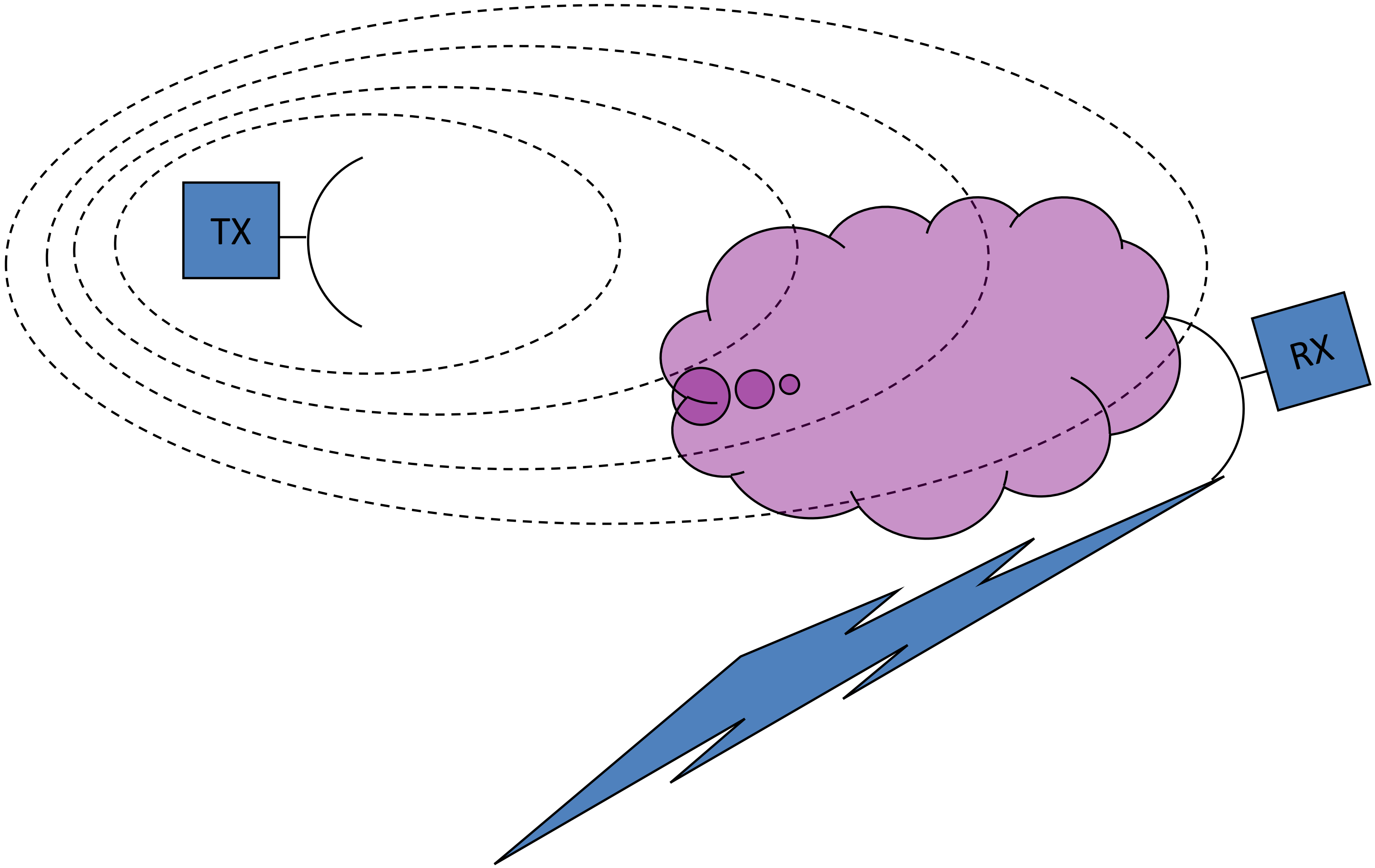
IEEE 802.15.4 Frame Format



Elements of Robust Communication

- **Application**: feasible workload
  - Packet rates, pattern, timing
  - Network: finding and using good paths
  - Topology discovery and route selection
  - Route cost determination, selection
  - Forwarding
- **Link**: Framing, Media Management Protocol
  - On to receive during transmission
  - Frame structure, error detection, acknowledgement
  - Avoiding contention (MAC, CCA, Hidden Terminal)
  - Link quality estimation
- **Physical**: Signal to Noise Ratio (SNR)
  - Device Transmission Power / Receive Sensitivity
  - Antenna design and orientation, obstructions, attenuation
  - Receive signal vs interference, noise, multipath
  - Modulation, channel coding

In a nutshell



Why multi-hop routing?

- Power!
  - Power to transmit distance D grows as D^3 or worse
  - Power to route distance D grows linearly
- Bandwidth (spatial multiplexing)
  - With n nodes in a single cell, each gets at most 1/n bandwidth
  - Many small cells => many simultaneous transmissions.
- Reliability
  - Individual links experience interference, obstacles, and multipath effects
  - Even short-range “wireless wires” require human nurturing
  - IrDA, Bluetooth, WiFi, Cell Phone
  - Provides spatial diversity and receiver diversity rather than antenna diversity
  - Protocol level reliability

Properties of a good Link layer

- It works!
- It supports the network layer above it.
- It allows low-power operation
- It allows highly reliable communication
- It allows for low latency
- It allows bursts
- It avoids contention
- It is simple, robust and flexible

MAC Protocol

- ❑ MAC protocol is to ensure that the channel can be accessed by multiple users, dealing with the situation of interference.
- ❑ Has a direct bearing on how reliably and efficiently data can be transmitted
- ❑ Long battery life
- ❑ MAC protocol design for WSNs:
  - ❖ Energy-efficient in sense of achieved throughput
  - ❖ Robust
  - ❖ As simple as possible

Major problems in WSN MAC design

- ❑ Idle listening
  - ❖ Listening when no traffic is sent
- ❑ Overhearing
  - ❖ Receiving packets destined to other nodes
- ❑ Collision
  - ❖ Retransmission
- ❑ Overhead
  - ❖ Headers for signalling

The "Idle Listening" Problem

- The power consumption of "short range" (i.e., low-power) wireless communications devices is roughly the same whether the radio is transmitting, receiving, or simply ON, "listening" for potential reception
  - includes IEEE 802.15.4, Z-Wave, Bluetooth, and the many variants
  - WiFi too!
  - Circuit power dominated by core, rather than large amplifiers
- Radio must be ON (listening) in order receive anything.
  - Transmission is infrequent. Reception α Transmit x Density
  - Listening (potentially) happens all the time
- Total energy consumption dominated by idle listening

MAC Caution

- The idle listen problem is often associated with Media Access Control (MAC) protocols,
  - TDMA, CSMA, …
- but MACs provide arbitration among multiple transmitters attempting to utilize a shared medium simultaneously.
  - Reduce Contention and associated loss.
  - May involve scheduling (TDMA) or transmission detection (CSMA)
- The problem here is the opposite.
  - Most of the time, nothing is transmitting.
  - Avoid listening when there is nothing to hear.
  - Scheduling and detection are involved, but to determine when to turn on receiver, rather than when to turn off transmission.
- You need Media Access Control, but that is not enough

Low Power

- 2 AA => 1.5 amp hours (~4 watt hours)
- Cell => 1 amp hour (3.5 watt hours)

Cell: 500 -1000 mW => few hours active

WiFi: 300 - 500 mW => several hours

GPS: 50 – 100 mW => couple days

WSN: 50 mW active, 20 uW passive

450 uW => one year

45 uW => ~10 years

$$\text{Ave Power} = \boxed{f_{act}} * \boxed{P_{act}} + f_{sleep} * \boxed{P_{sleep}} + \boxed{f_{waking}} * P_{waking}$$

* System design

* Leakage (~RAM)

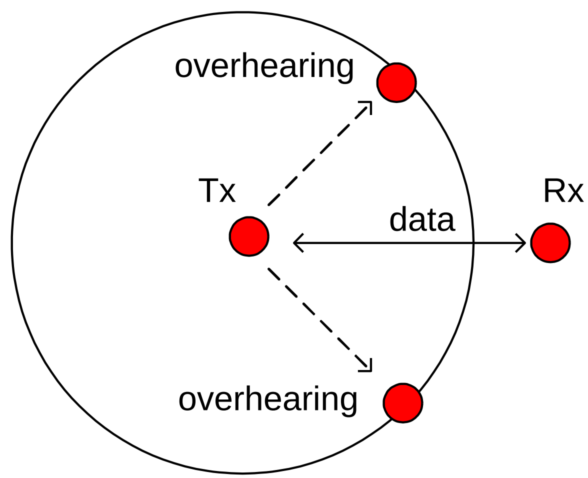
* Nobody fools mother nature

Where the energy goes?

- Sleep
  - 7 uA for TI MSP
- Sensing
- Transmitting results
- Management Traffic
- Routing Structure Maintenance
  - only parent tracking for leaf
- Listening
- Forwarding
  - non-leaf
- Overhearing packets destined for others

Overhearing

- Receiving packets that are not destined to the node
- Interception, waste of energy in receiving, error responding will cause potential collision



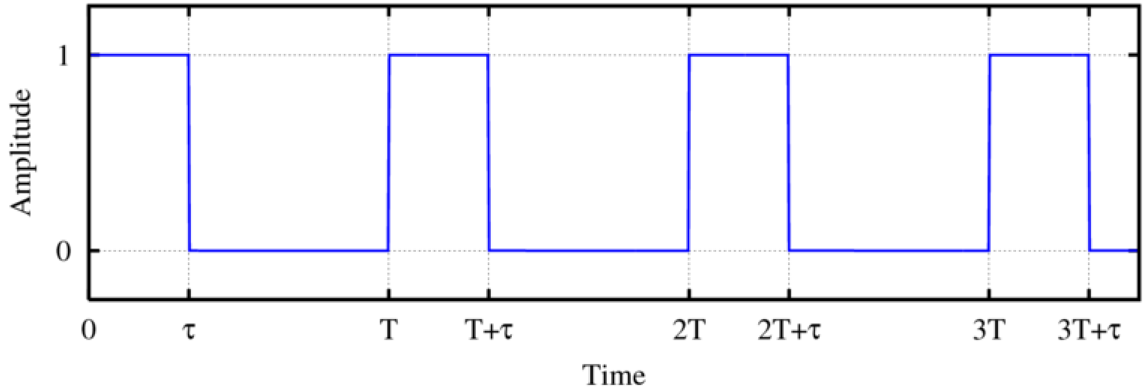
Traffic Pattern

- Local broadcast
  - Schedule exchange/update between neighbours
  - Omni-directional transmission is desired
- Nodes to sink report
  - Payload and signalling
  - In favour of directional transmission

Duty cycle

- Duty cycling is a widely used mechanism in wireless sensor networks.
- Reducing energy consumption due to idle listening.
- This mechanism also introduces additional latency in packet delivery.

$$D = \frac{\tau}{T}$$



Radio

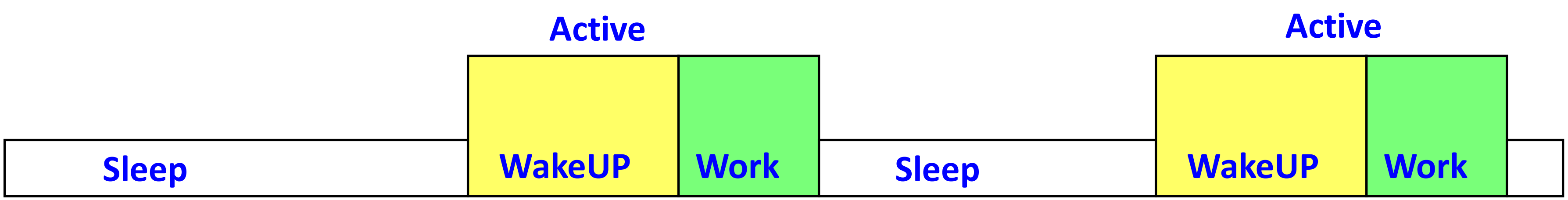
Type	Narrowband				Wideband		
Vendor Part no.	RFM TR1000	Chipcon CC1000	Chipcon CC2400	Nordic nRF2401	Chipcon CC2420	Motorola MC13191/92	Zeevo ZV4002
Max Data rate (kbps)	115.2	76.8	1000	1000	250	250	723.2
RX power (mA)	3.8	9.6	24	18 (25)	19.7	37(42)	65
TX power (mA/dBm)	12 / 1.5	16.5 / 10	19 / 0	13 / 0	17.4 / 0	34(30)/ 0	65 / 0
Powerdown power (μA)	1	1	1.5	0.4	1	1	140
Turn on time (ms)	0.02	2	1.13	3	0.58	20	*
Modulation	OOK/ASK	FSK	FSK,GFSK	GFSK	DSSS-O-QPSK	DSSS-O-QPSK	FHSS-GFSK
Packet detection	no	no	programmable	yes	yes	yes	yes
Address decoding	no	no	no	yes	yes	yes	yes
Encryption support	no	no	no	no	128-bit AES	no	128-bit SC
Error detection	no	no	yes	yes	yes	yes	yes
Error correction	no	no	no	no	yes	yes	yes
Acknowledgments	no	no	no	no	yes	yes	yes
Interface	bit	byte	packet/byte	packet/byte	packet/byte	packet/byte	packet
Buffering (bytes)	no	1	32	16	128	133	yes *
Time-sync	bit	SFD/byte	SFD/packet	packet	SFD	SFD	Bluetooth
Localization	RSSI	RSSI	RSSI	no	RSSI/LQI	RSSI/LQI	RSSI

* Manufacturer's documentation does not include additional information.

- Trade-offs:
  - resilience / performance => slow wake up
  - Wakeup vs interface level
  - Ability to optimize vs dedicated support

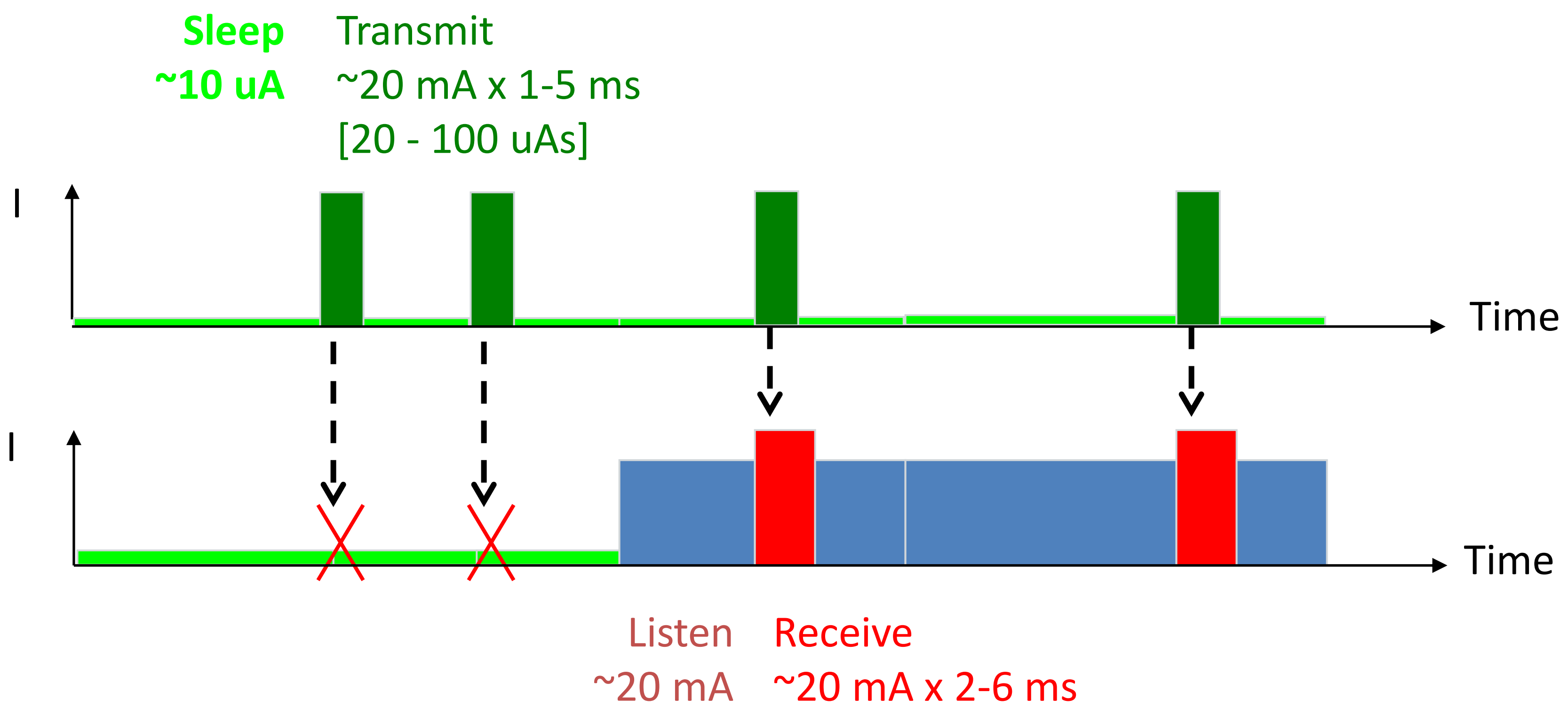
Power States at Node Level

Operation	Telos	Mica2	MicaZ
Minimum Voltage	1.8V	2.7V	2.7V
Module Standby	5.1 μA	19.0 μA	27.0 μA
MCU Idle	54.5 μA	3.2 mA	3.2 mA
MCU Active	1.8 mA	8.0 mA	8.0 mA
MCU + Radio RX	21.8 mA	15.1 mA	23.3 mA
MCU + Radio TX (0dBm)	19.5 mA	25.4 mA	21.0 mA
MCU + Flash Read	4.1 mA	9.4 mA	9.4 mA
MCU + Flash Write	15.1 mA	21.6 mA	21.6 mA
MCU Wakeup	6 μs	180 μs	180 μs
Radio Wakeup	580 μs	1800 μs	860 μs

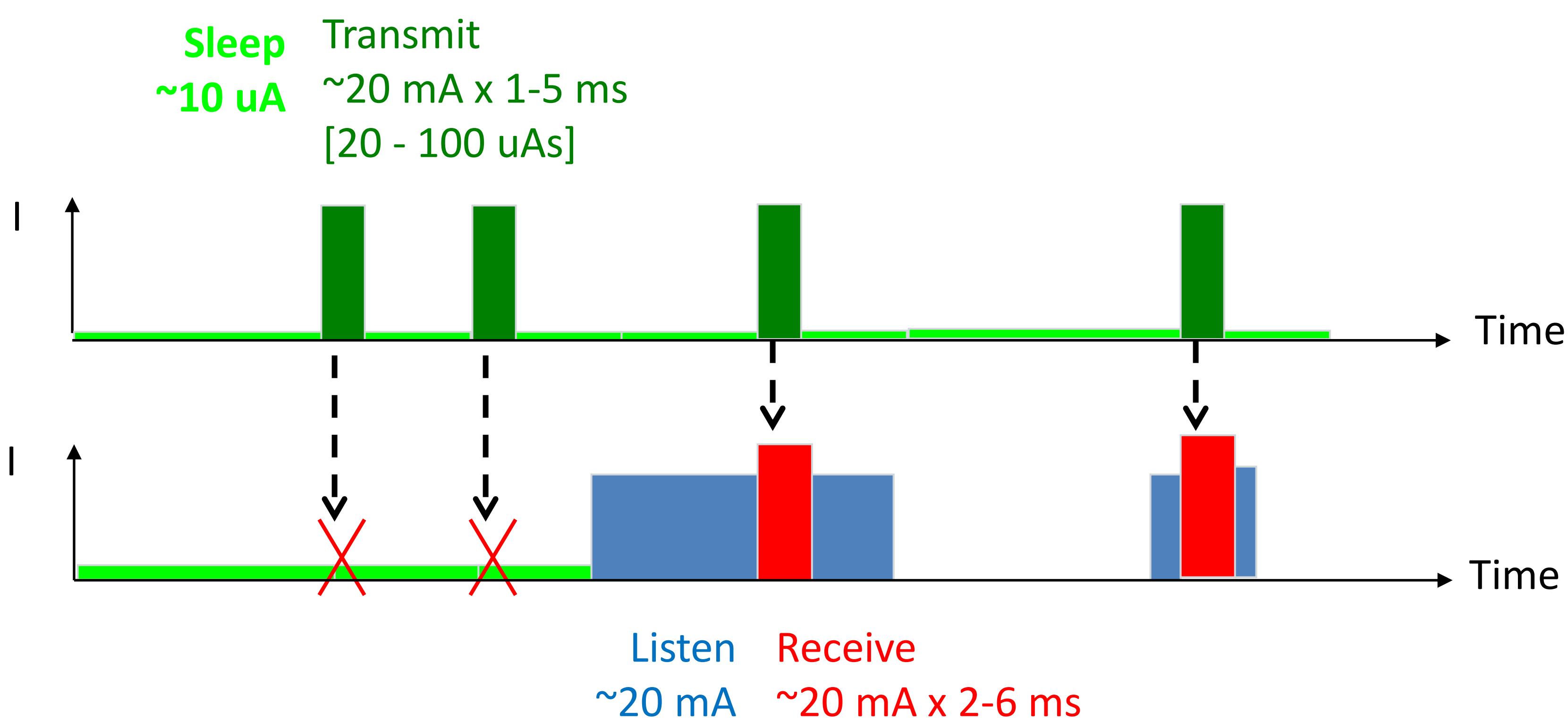


Telos: Enabling Ultra-Low Power Wireless Research, *Polastre, Szewczyk, Culler, IPSN/SPOTS 2005*

Communication Power Consumption



Communication Power – Passive Vigilance



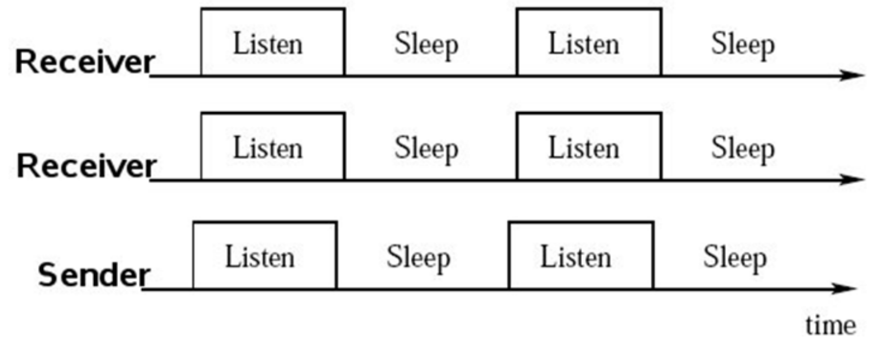
❑ Listen just when there is something to hear …

What causes Energy Waste?

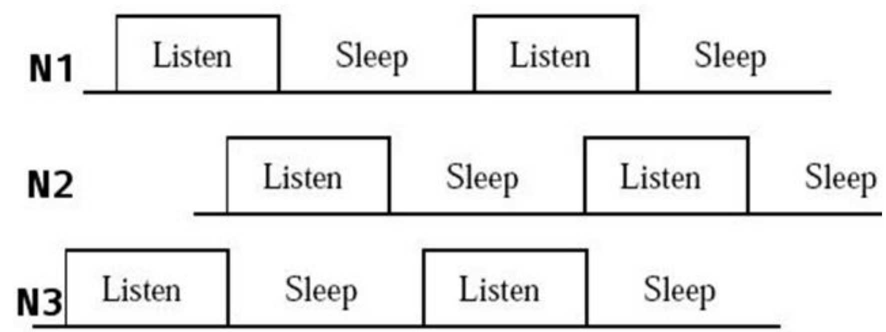
- ❑ Collisions/Interference
  - ❖Basic Function of MAC layer
- ❑ Control packet overhead
  - ❖e.g., RTS/CTS, Beaconing
- ❑ Overhearing unnecessary traffic
  - ❖Non-target nodes
- ❑ Long idle time
  - ❖Prepare for receiving packets
- ❑ Synchronization Overhead
  - ❖Preamble and Start of Frame Detection

2 Basic Solution Techniques

- Synchronized protocols
  - Nodes awake/sleep on a schedule
  - periodic sleep, no idle listening
  - E.g. S-MAC, T-MAC



- Asynchronized protocols:
  - Nodes having independent schedule
  - Need some mechanism for carrier sensing
  - E.g. B-MAC, X-MAC, TSCH



Asynchronous vs. Synchronous

- **Advantages**
  - Use extended preamble
  - Sender and receiver can have decoupled duty cycles.
  - No synchronization overhead.
  - Awake periods are much shorter

- **Disadvantages**
  - Frame exchange delay even if receiver awakes before preamble ends
  - Overhearing problem
  - Preamble latency is expensive for multi-hop routes

S-MAC: Sensor-MAC

- Synchronizes sensor clusters
- Nodes periodically wake-up to receive synchronization info from its neighbors.
  - Mitigates need for system wide synchronization.
  - Nodes can belong to more that one virtual cluster.
- Communicate using RTS-CTS: avoid hidden terminal
- Can use adaptive listening
  - Neighbor briefly wakes up at the end of overheard RTS/CTS
  - Reduces one-hop latency
- Implemented in TinyOS 1.0

S-MAC

- Periodic listen and sleep
- Collision avoidance
- Coordinated sleeping
- Overhearing avoidance

S-MAC: Periodic Listen & Sleep

- Frame
- Duty cycle
  - (Listen Interval / Frame Length)

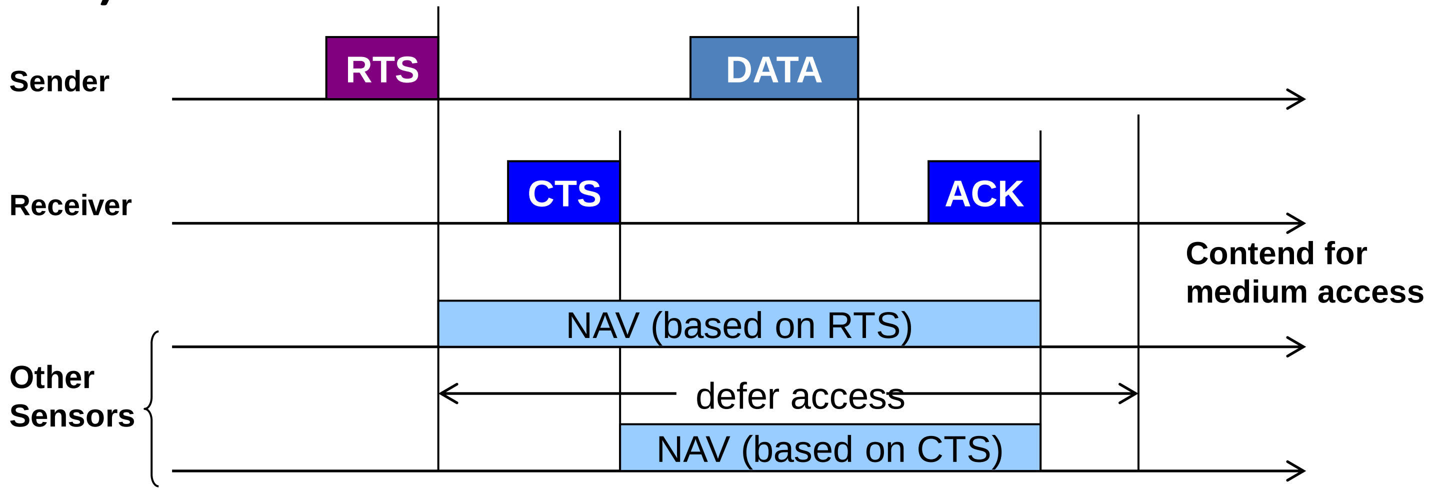
- Frame schedule
  - Nodes are free to choose their listen/sleep schedule
  - Requirement: neighboring nodes synchronize together
  - Exchange schedules periodically (SYNC packet)
    - Synchronization period (SP)

- Nodes communicate in receivers scheduled listen times

S-MAC: Collision-Avoidance

Collision-Avoidance Strategy ~= IEEE 802.11

- ❑ RTS/CTS
- ❑ Physical carrier sense
- ❑ Virtual carrier sense: network allocation vector (NAV)



S-MAC: Coordinated Sleeping (1)

Frame Schedule Maintenance

1. Choosing a schedule
   - Listen to the medium for at least SP
   - Nothing heard, choose a schedule
   - Broadcast a SYNC packet (should contend for medium)
2. Following a schedule
   - Receives a schedule before choosing/announcing
   - Follows the schedule
   - Broadcast a SYNC packet
3. Adopting multiple schedules
   - Receives a schedule after choosing/announcing
   - Can discard the new schedule; or
   - Follow both the schedules – suffer more energy loss

S-MAC: Coordinated Sleeping (2)

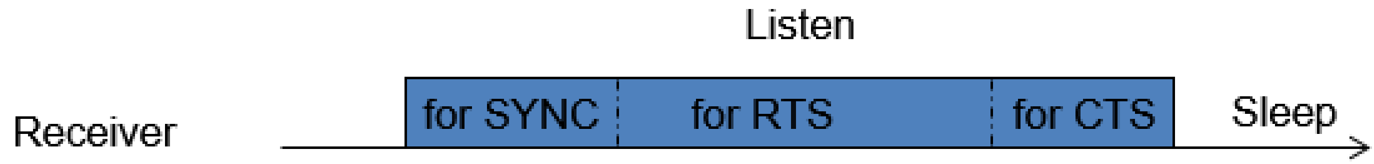
Neighbor Discovery

- chance of failing to discover an existing neighbor
  - corrupted SYNC packet, collisions, interference
  - sensor – border of two schedules; discovers only the first schedule, if schedules do not overlap
- Periodically, listen for the complete SP
  - frequency?
    - Increased - if a sensor has no neighbors
    - S-MAC experimental values:
      - SP = 10 seconds
      - Neighbor discovery period = 2 minutes, if at least 1 neighbor

S-MAC: Coordinated Sleeping (3)

Maintaining Synchronization

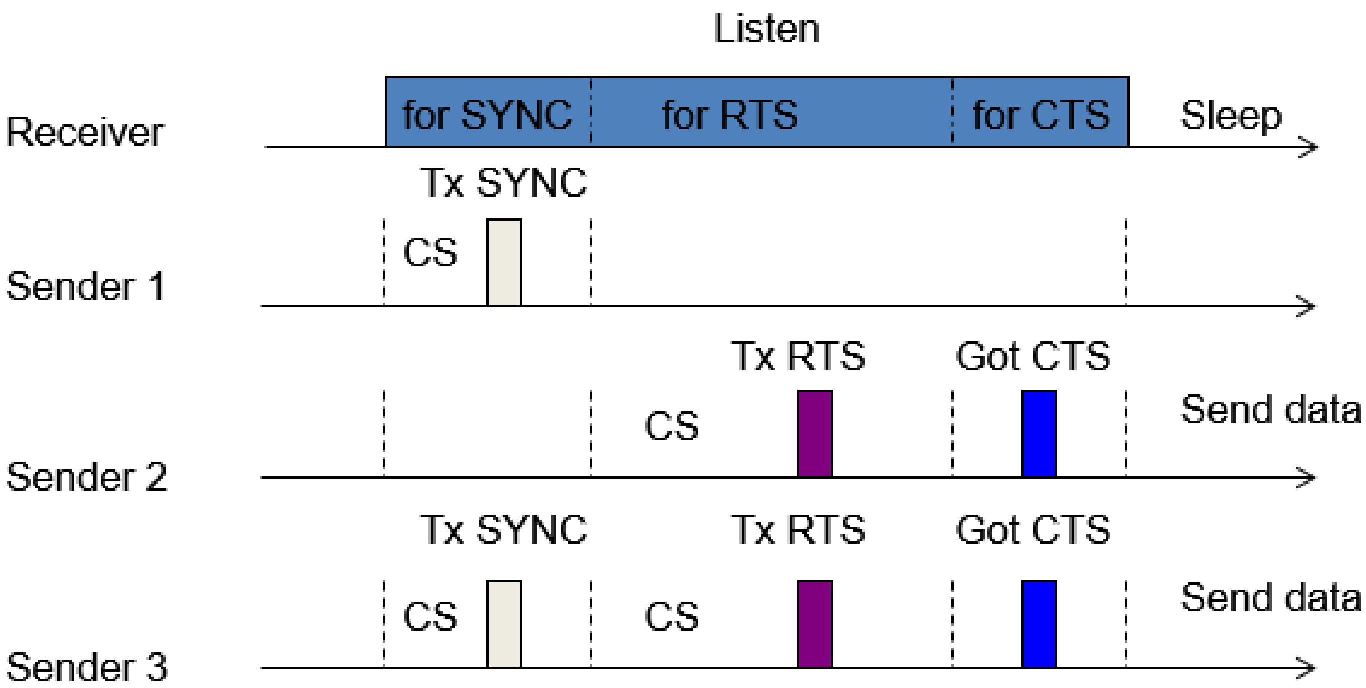
- Clock drifts – not a major concern (listen time = 0.5s – 10^5 times longer than typical drift rates)
- Need to mitigate long term drifts – schedule updating using SYNC packet (sender ID, its next scheduled sleep time – relative);
- Listen is split into 2 parts – for SYNC and RTS/CTS



- Once RTS/CTS is established, data (if available) sent in sleep interval

S-MAC: Coordinated Sleeping (4)

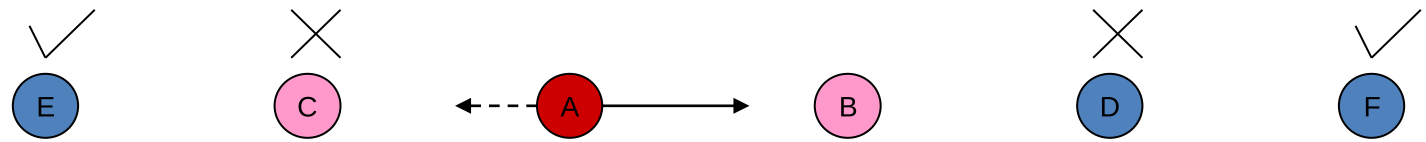
Example Scenarios



CS: Carrier Sense

S-MAC: Overhearing Avoidance

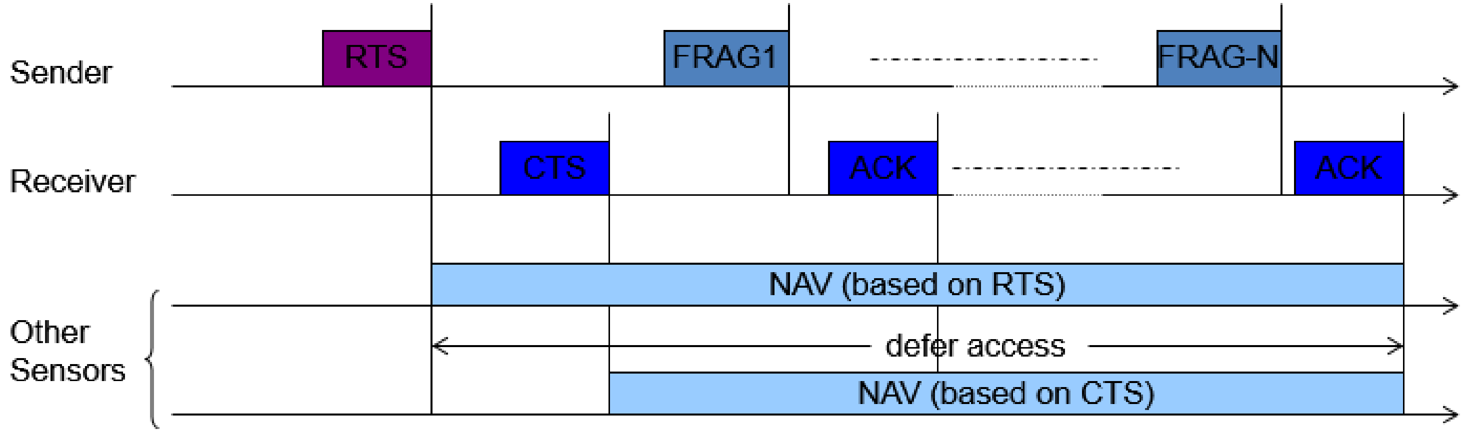
- Who should sleep when a node is transmitting?



- All immediate neighbors of both sender and receiver should sleep after hearing a RTS or CTS signal (e.g. Node C and D)
- Use NAV to schedule the sleep timer
- The medium is busy when the NAV value is not zero

S-MAC: Efficient Message Passing

- Sending a long message?
  - As a single packet: increase cost of retransmission for message corruption



S-MAC: Efficient Message Passing

- RTS/CTS/ACK – has duration fields in it
- If ACK is not received, increase the transmission time, retransmit. ACK will be also be updated.

- Difference between 802.11 & S-MAC
  - Medium is reserved upfront for the whole transmission in S-MAC

Drawbacks of S-MAC

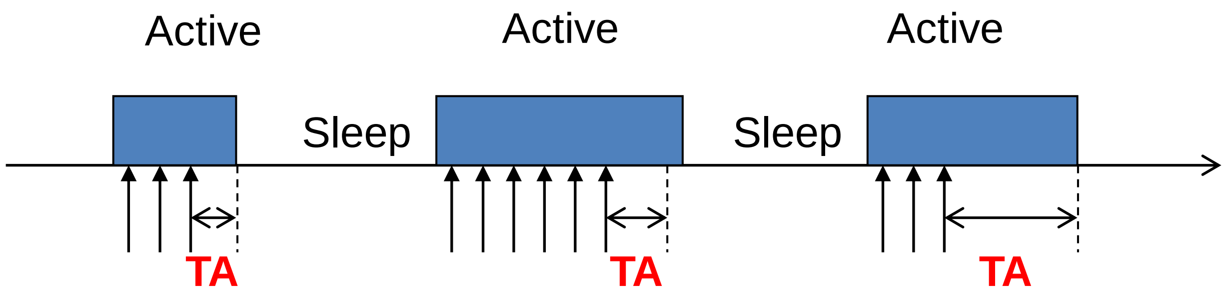
- Active (Listen) interval – long enough to handle to highest expected load
  - If message rate is less – energy is still wasted in idle-listening
- **S-MAC fixed duty cycle – is NOT OPTIMAL**

T-MAC: Timeout-MAC

- Listen for a short time after awake period.
- Sleeps if IDLE.
- Improves on S-MAC by shortening the awake period if IDLE.
- For variable payloads, T-MAC uses ~20% of energy used in S-MAC.

T-MAC

- Adaptive duty cycle:



- A node is in active mode until ***no activation event occurs for time TA (Traffic Aware)***
  - Periodic frame timer event, receive, carrier sense, send-done, knowledge of other transmissions being ended
- Communication ~= S-MAC
- Frame schedule maintenance ~= S-MAC

T-MAC:

Contention Interval

- ❑ Waiting/listening for a random time within a fixed contention interval (unlike exponential back-off in 802.11)

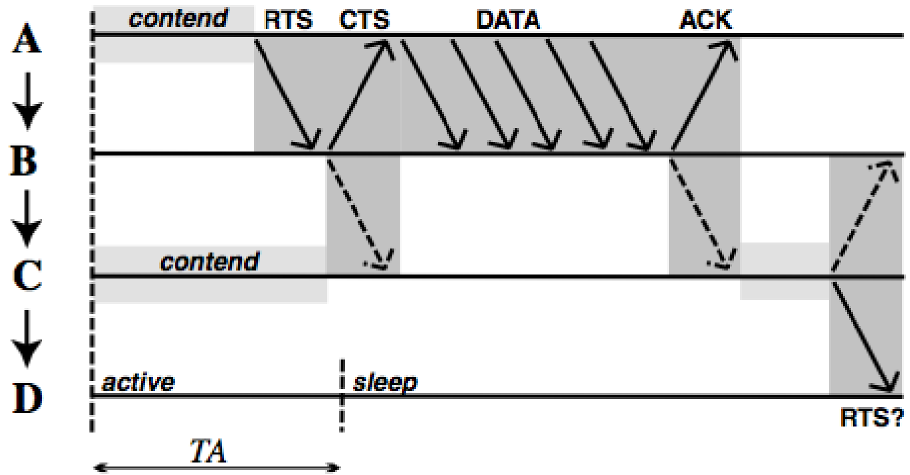
- ❑ Assumptions:
  - ❖ load is always high, does not vary

T-MAC: Choosing TA

- Requirement: a node should not sleep while its neighbors are communicating, potential next receiver
- **TA > C+R+T**
  - C – contention interval length;
  - R – RTS packet length;
  - T – turn-around time, time between the end of RTS and start of CTS;
- TA = 1.5 * (C+R+T);

T-MAC: Overhearing Avoidance

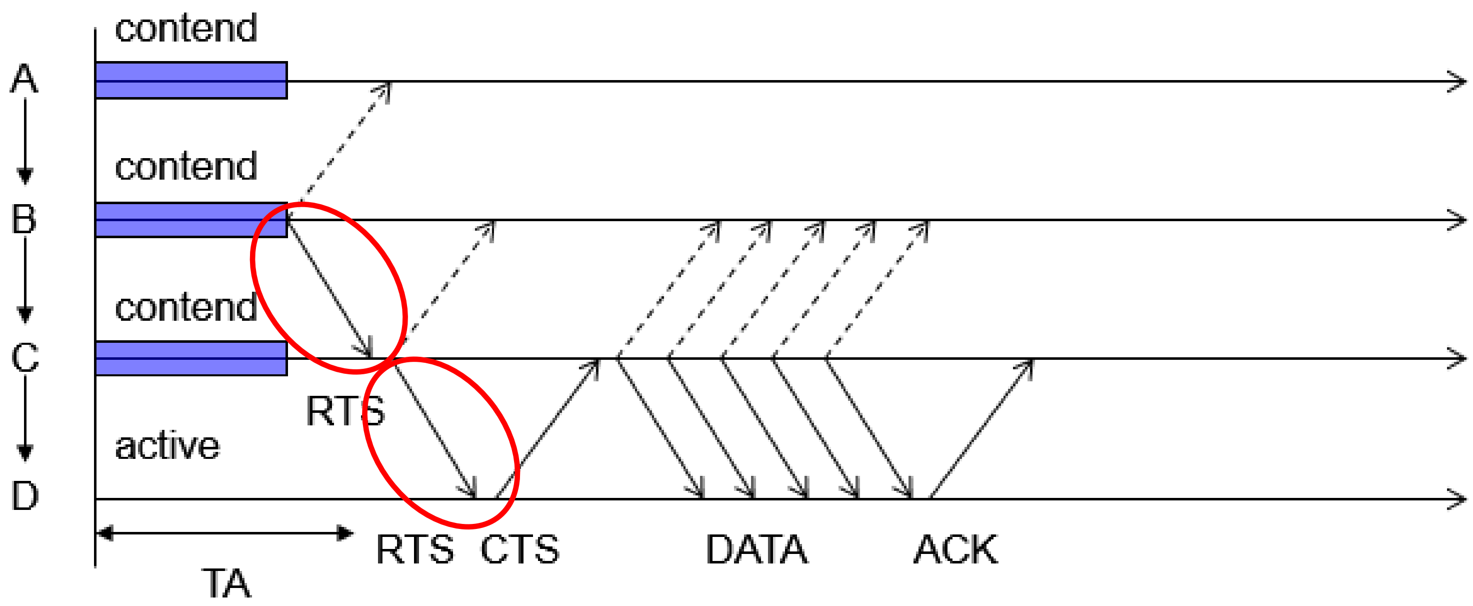
- ~= S-MAC
- But implemented as an option in T-MAC
- Node – goes to sleep after overhearing RTS/CTS of other nodes communication
  - Miss other RTS/CTS transmissions
  - Disturb the medium while waking up
  - Overhearing avoidance should not used when maximum throughput is required



T-MAC: Solution

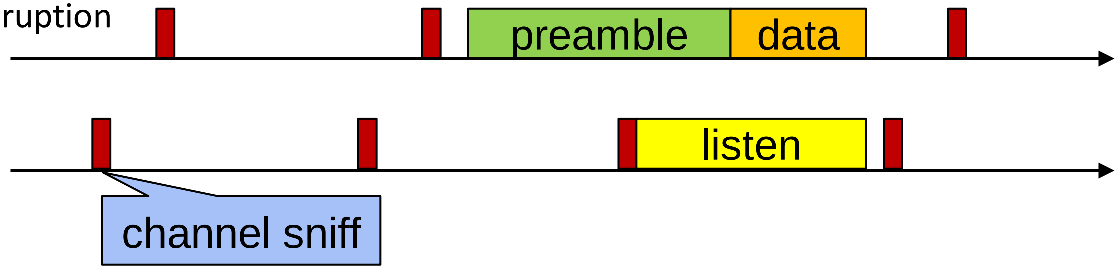
Full-Buffer Priority – suitable for unidirectional flows

- ❑ Buffer – almost full – prefer sending than receiving
- ❑ Receive RTS, send its own RTS back instead of CTS
  - ❖ E.g. C gets RTS from B, it sends its RTS and gets CTS from D. Then C's data packets can be sent before B's packets.



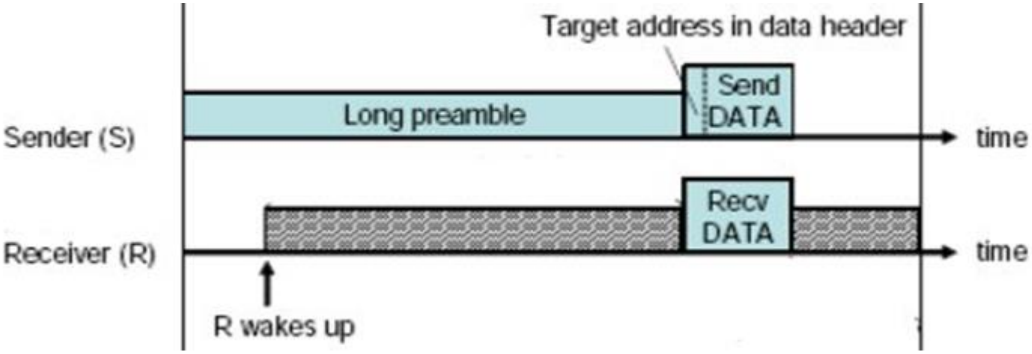
LPL: Low-Power Listening

- Goal: minimize listen cost
- RX: CCA to sample, ON to receive packet
  - Extremely short CCA (500 us)
- Tx uses short ***chirp*** messages as ***preamble***
  - Addressing → minimizes overhearing cost
  - Data time → receive cost independent of sample period
- Nodes wake up for a short period and check for channel activity.
  - Return to sleep if no activity detected.
- If a sender wants to transmit a message, it sends a long preamble to make sure that the receiver is listening for the packet.
  - preamble has the size of a sleep interval
- Very robust
  - No synchronization required
  - Instant recovery after channel disruption
- Example: B-MAC

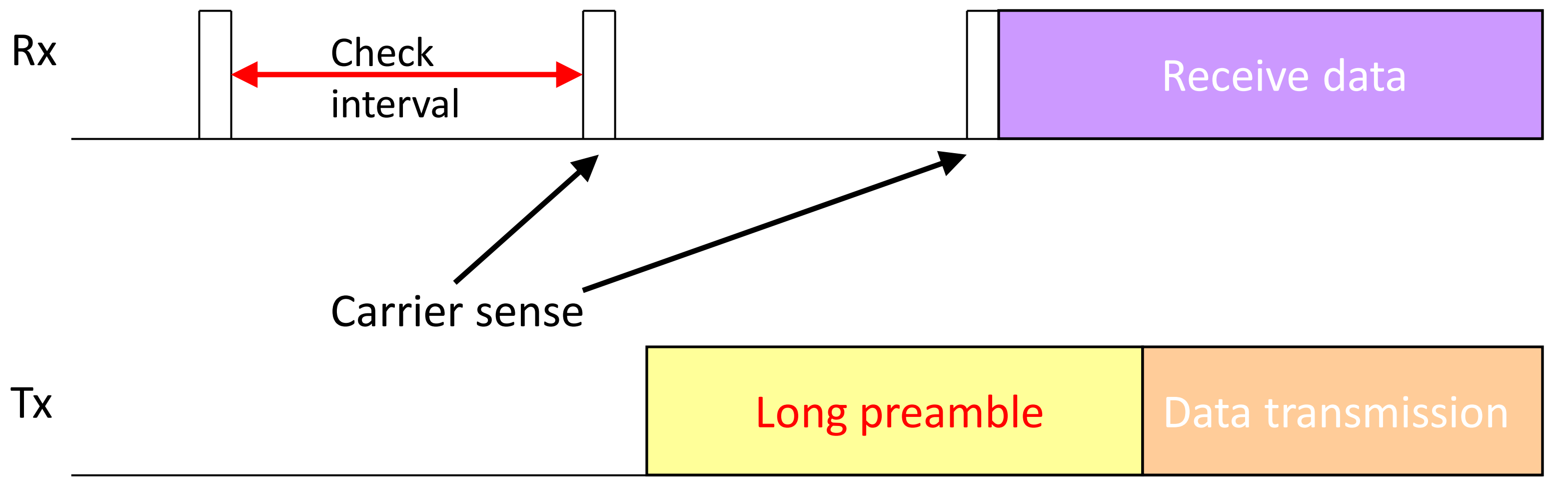


B-MAC

- Uses local schedules
- Send preamble that is slightly longer than the sleep period.
- Long preamble assures that the neighbor will receive packet.
- Suffers from overhearing problem
- Transmitting node precedes data packet with preamble slightly longer than sleep period of receiver.
- During awake period, node samples medium & if a preamble is detected it remains awake to receive the data.

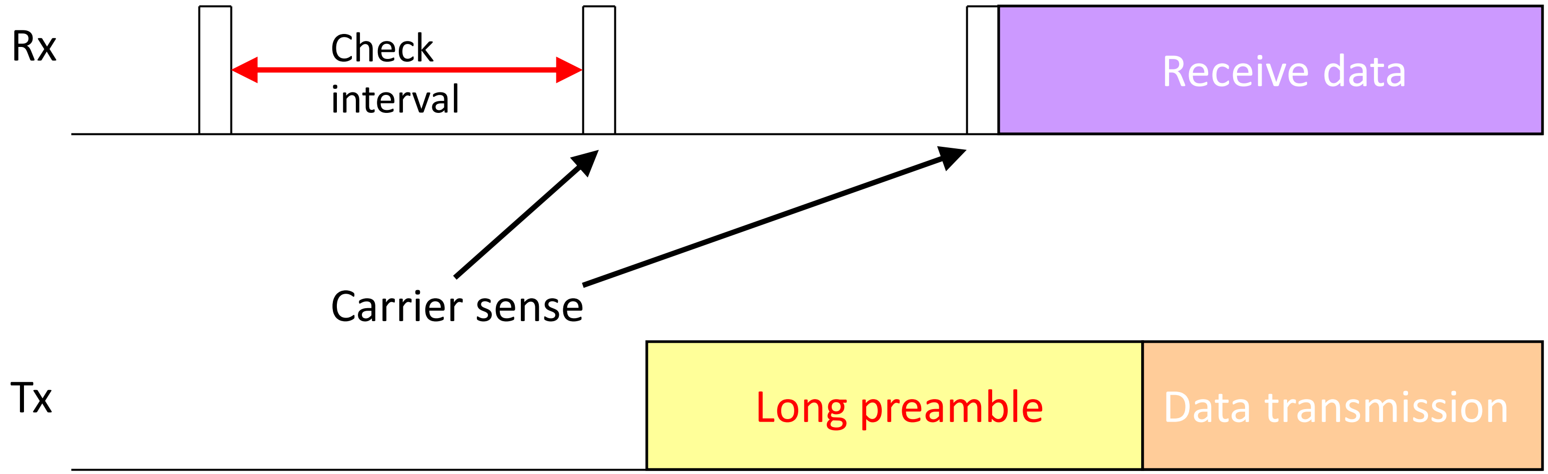


B-MAC Protocol



- Shift most burden to the sender
- Sender uses a long preamble before each packet to wake up the receiver.
- Data transmission can use RTS/CTS or some other strategy.

Key challenges of B-MAC



- ❑ *Check Interval* has to be very short to ensure reasonable length preamble
- ❑ Carrier sense duration also has to be very short to ensure receiver does not spend too much energy.
- ❑ Carrier sense has to be very accurate to reduce:
  - ❖ latency of transmission
  - ❖ energy consumption at sender

How to Fix Some Issues in B-MAC

- X-MAC: *A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks*
  (M. Buettner, G. V. Yee, E. Anderson, R. Han)

- **B-MAC = CSMA + LPL + Noise Floor Estimation + Explicit ACK**

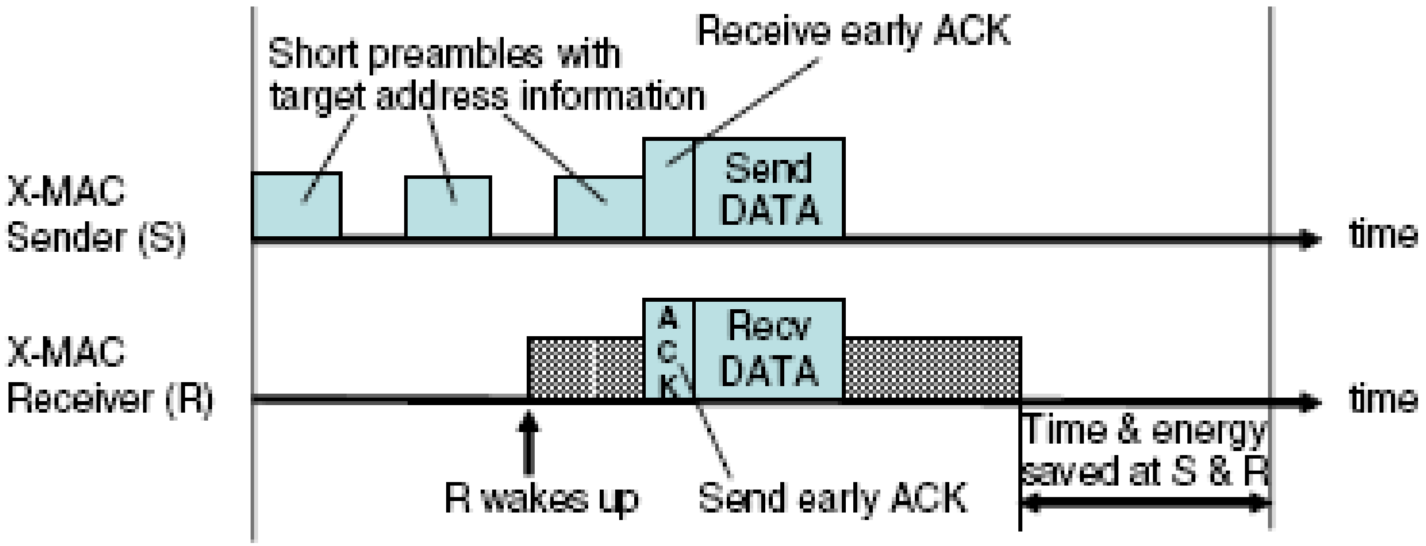
- **X-MAC = B-MAC + Early ACK + Encoded preamble (???)**

X-MAC

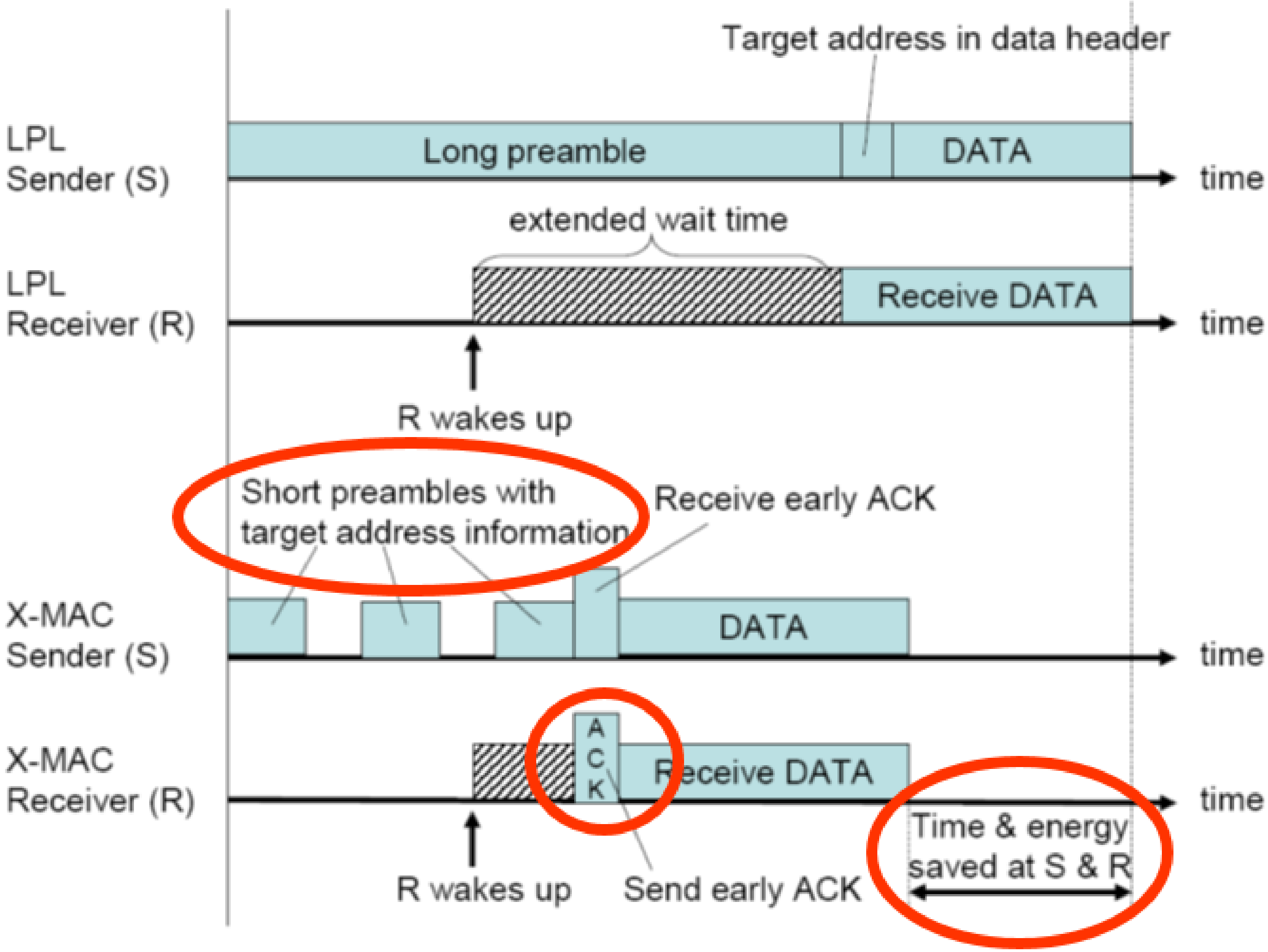
- Short preamble
  - Reduce latency and reduce energy consumption
- Target in preamble
  - Minimize overhearing problem.
- Strobed preamble
  - Reduces latency for the case where destination is awake before preamble completes.
- Reduces per-hop latency and energy
- Dynamic duty-cycle algorithm

X-MAC

- ❑ Strobed preamble
  - ❖ Allowing interruption and wake up faster
- ❑ Short preamble
  - ❖ Embedded with address information of the target



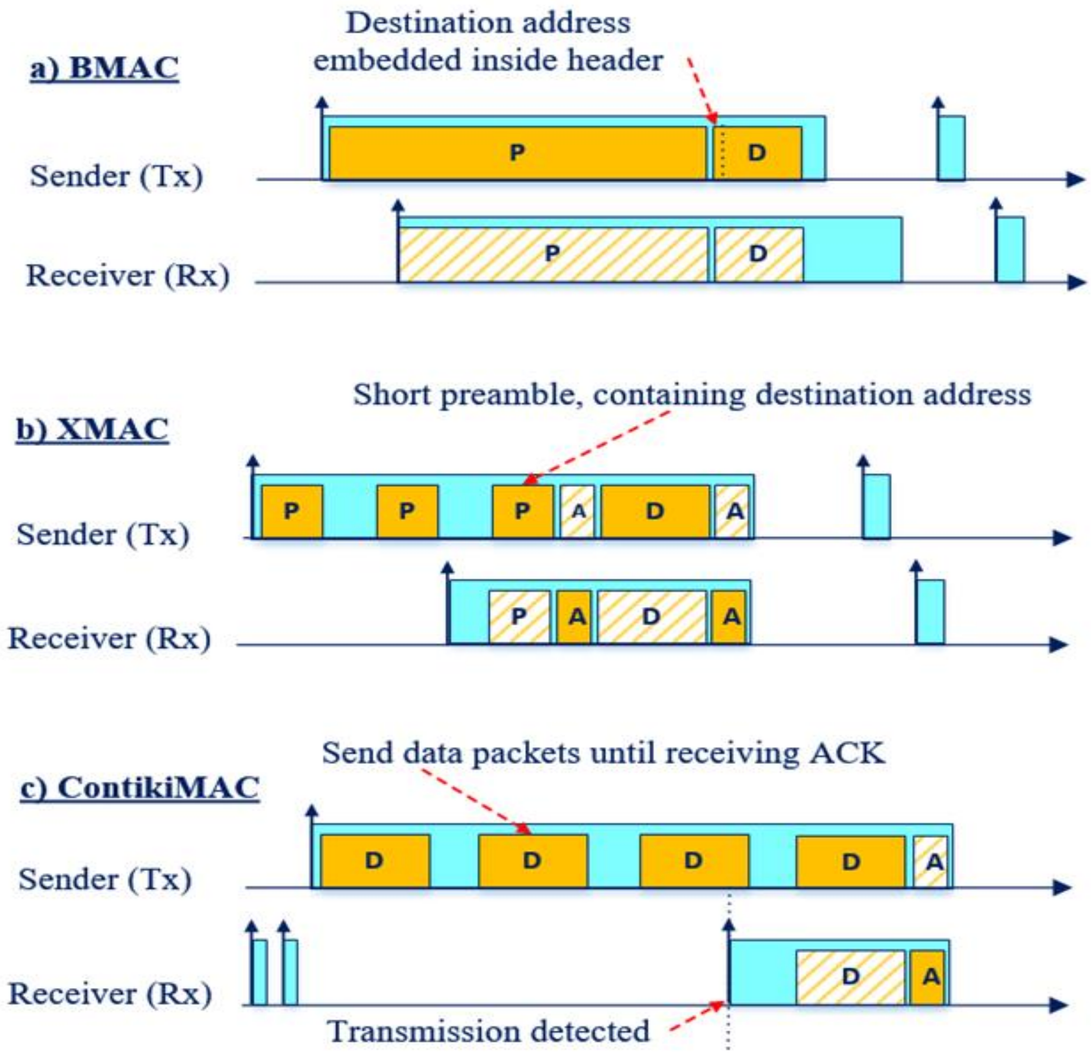
X-MAC vs LPL: Operations



X-MAC: Benchmarking parameters

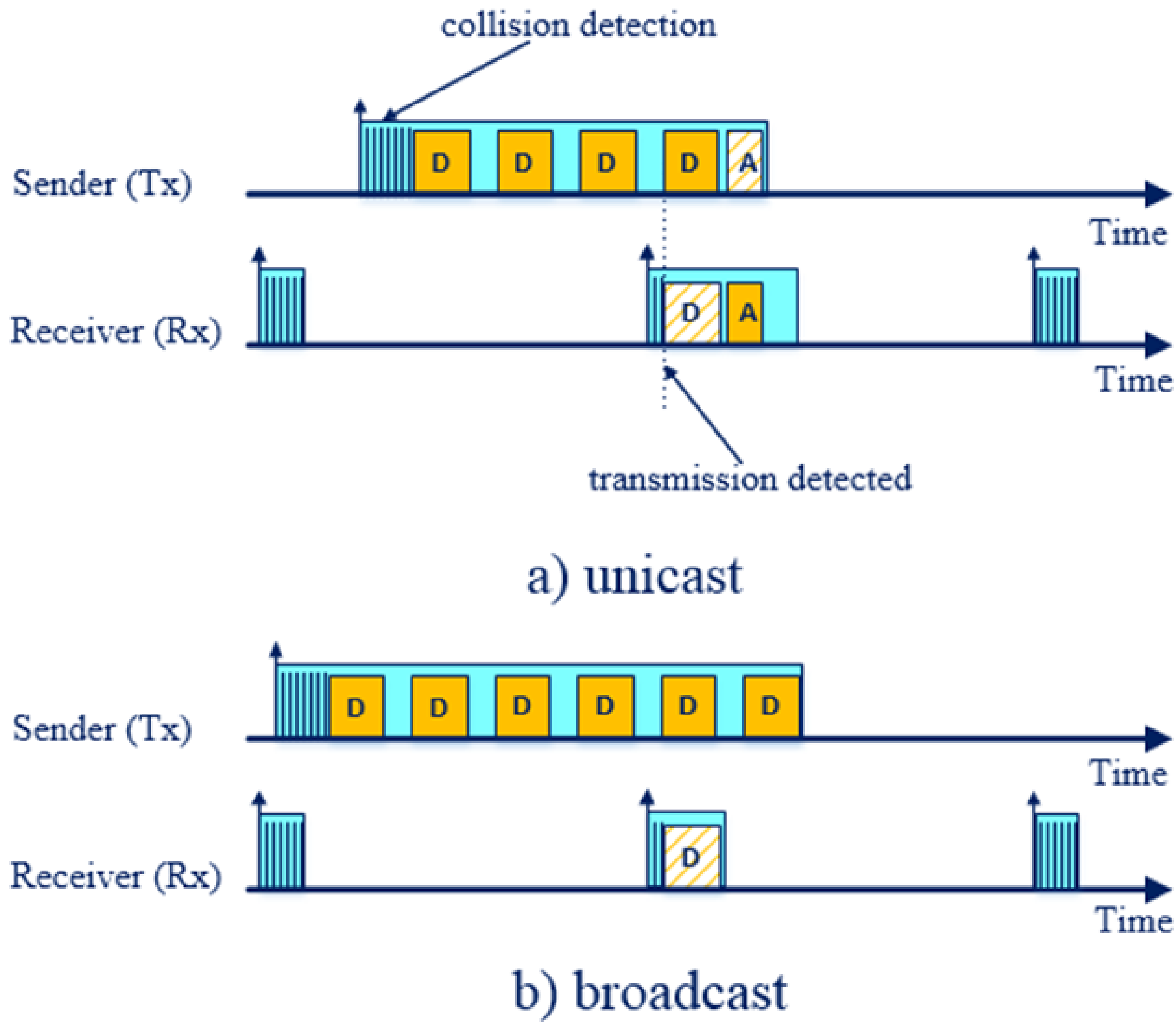
- Design goals of the X-MAC protocol for duty-cycled WSNs:
  - Energy-efficiency
  - Low loss % of packets
  - Low latency for data
  - High throughput for data
  - Duty cycles
- Small pauses between preamble packets permit the target receiver to send an ACK
- Truncating preamble saves energy at transmitter and receiver and allows for lower latency.
- Non-target receivers which overhear the strobed preamble can go back to sleep immediately
- X-MAC's strobed preamble approach outperforms traditional LPL.

ContikiMAC

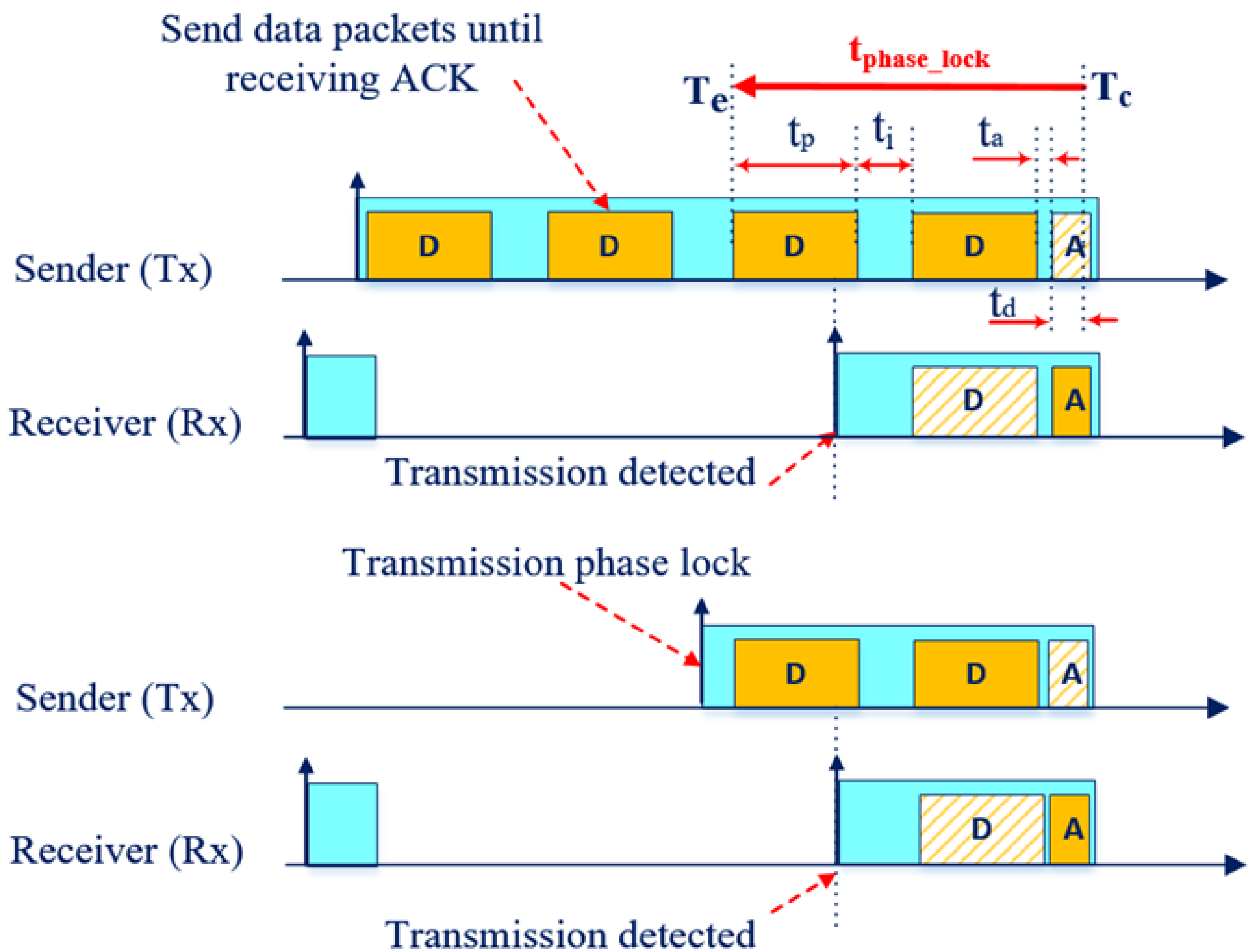


a) BMAC
Destination address embedded inside header
Sender (Tx)
P
D
Receiver (Rx)
P
D
b) XMAC
Short preamble, containing destination address
Sender (Tx)
P
P
P
A
D
A
Receiver (Rx)
P
A
D
A
c) ContikiMAC
Send data packets until receiving ACK
Sender (Tx)
D
D
D
D
A
Receiver (Rx)
D
A
Transmission detected

ContikiMAC: Unicast and Broadcast

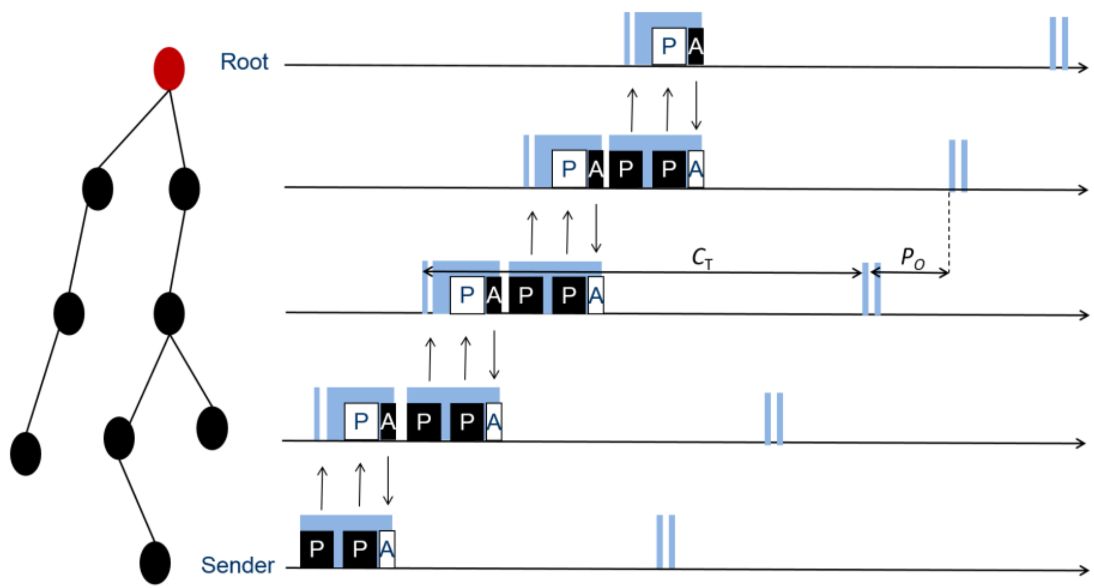


ContikiMAC: Phase Lock



RAWMAC: extension of ContikiMAC

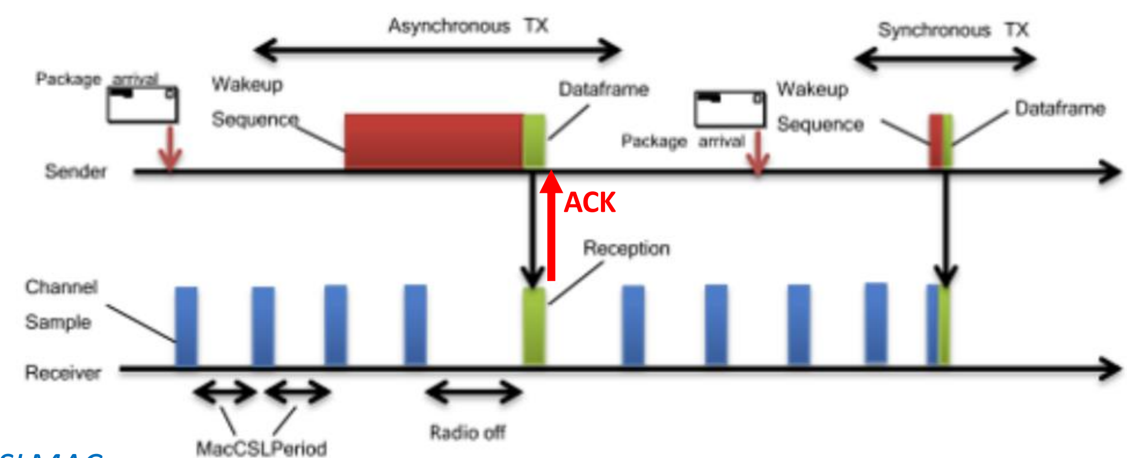
- Key idea: extend ContikiMAC by aligning wake-up phase to the parent in RPL tree
  - Phase offset is configured with P_0 parameter (optimal value should be chosen carefully)
  - Including mechanism to mitigate clock drifts



https://github.com/sics-iot/calipso-integrated

Coordinate Sampled Listening: CSL

- TX node sends wakeup frames before a data packet is sent. Information on the time at which the data packet will be sent is included in these wakeup frames.
- If a RX node is awake when a wakeup frame is being transmitted and it can successfully retrieve the timing information when the data packet is going to be sent, it goes to sleep till the time that the data packet will be sent.



https://github.com/Daparrag/Contiki3.0-CSLMAC

Coordinate Sampled Listening: CSL

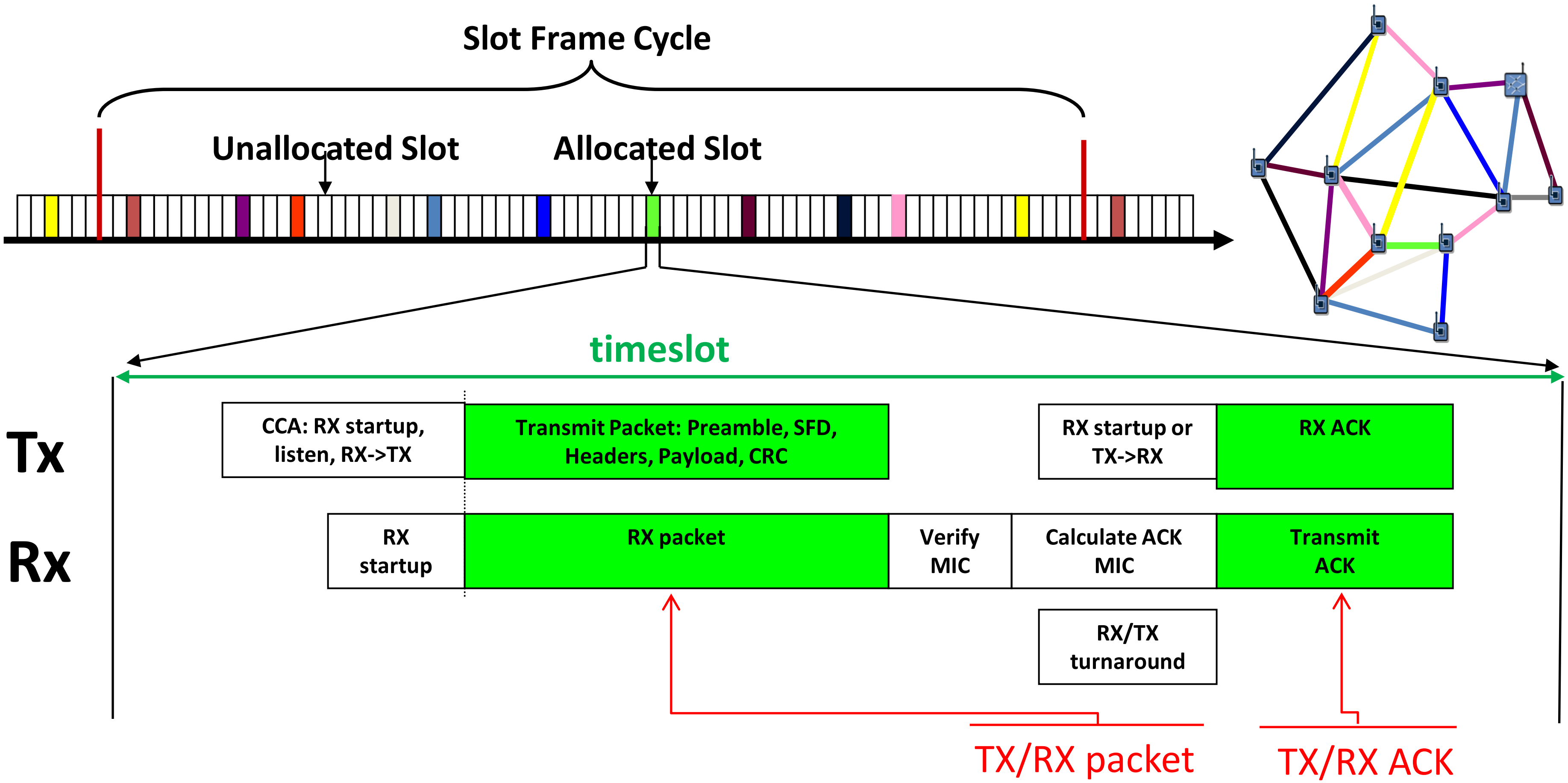
- If however, RX is not awake and ***does not*** receive a wakeup frame, it goes to sleep. RX wakes up periodically to check for wakeup frames. If RX successfully receives a data packet from TX (after receiving the wakeup frame), it can send an ACK frame that also contains the periodicity at which the RX wakes up.
- This information can be used at the TX to estimate the next time that TX will be awake. At this point, TX can then decide to send the wakeup frames at the expected time for the RX to be awake.

Protocol	RDC	TOTAL-PRR	Packets sent	Packet received	Channel Sampling
CSL-MAC	14%	0.5384	52	28	8hz
XMAC	7.49%	0.0961	52	5	
ContikiMAC	7.55%	0.2115	52	11	

TSCH

- Time Slotted, Channel Hopping (TSCH) technology is the basis for the wireless network of two industrial standards
  - HART Foundation (over 200 companies worldwide): WirelessHART- published 9/07
  - ISA (over 30,000 members worldwide): ISA100 Committee, ISA100.11a working group- in working group draft
- TSCH has been implemented by multiple companies on multiple 2.4 GHz IEEE std. 802.15.4 platforms

TSCH: Timeslot Access



Devices are configured with a slot-frame and timeslots to communicate with each other.

Timeslot Basics

- All devices in the same network synchronize slotframes
- All timeslots are contained within a slotframe cycle
- Timeslots repeat in time: the slotframe period
- Device-to-device communication within a timeslot includes packet Tx/Rx & ACK Tx/Rx
- Configurable option for CCA before transmit in timeslots

Timeslot Operation In Devices

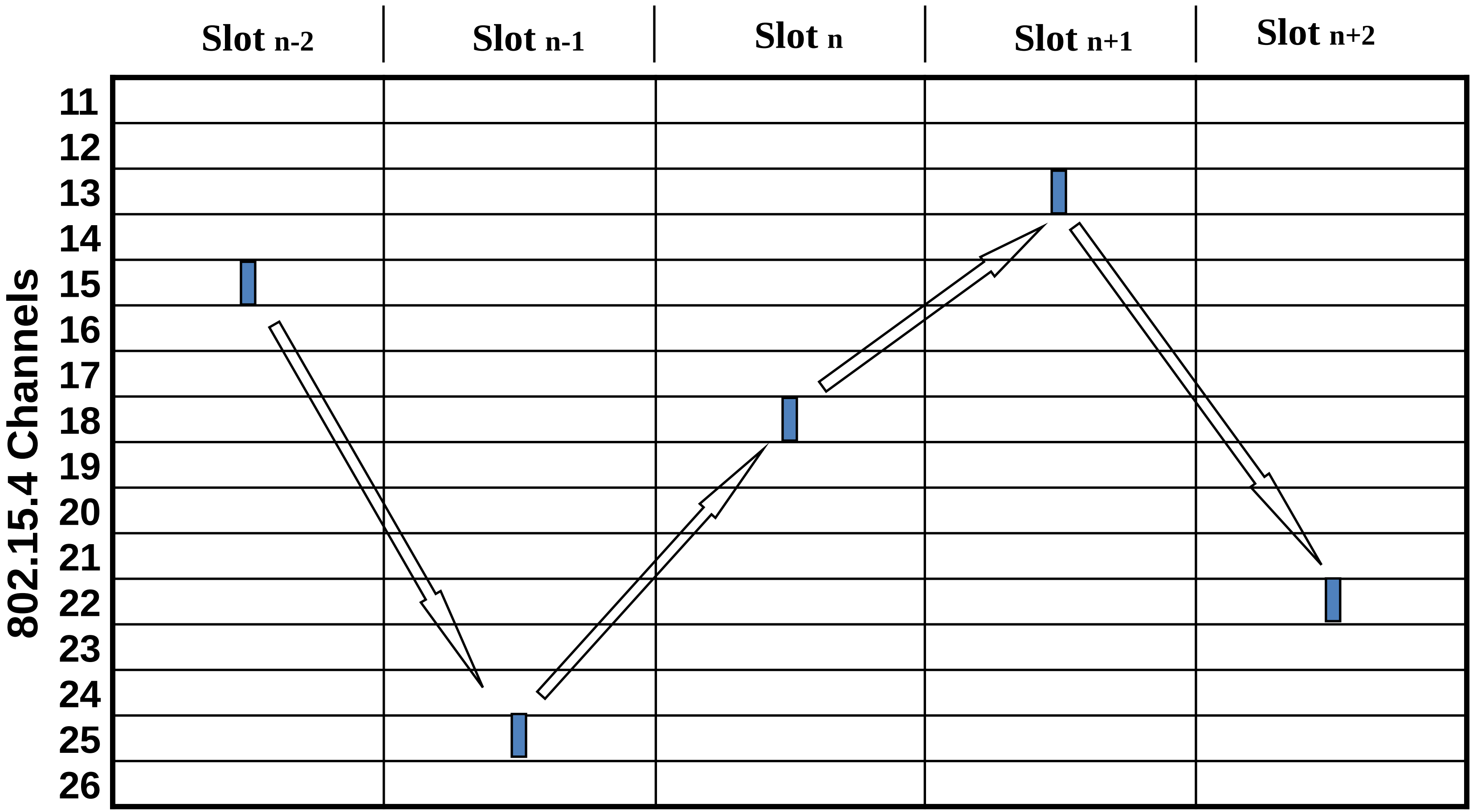
Devices use timeslots to:

- Schedule when they wakeup to transmit or listen
- Keep time synchronized
  - Specification on time difference tolerances
  - Time synchronization mechanisms
- Time the sequence of operations
  - Allow the source and destination to set their frequency channel
  - Listening for a packet
  - Sending a packet
  - Listening for an ACK
  - Generating an ACK
- Synchronizes channel hops
- Provide time to higher layers

TSCH: Link Types

- ❑ Dedicated Link – assigned to one device for transmission and to one or more devices for reception
  - ❖ A dedicated broadcast link is assigned to all devices for reception
- ❑ Shared Link – assigned to more than one device for transmission
  - ❖ ACK failures detect collisions
  - ❖ A slot based back-off algorithm resolves collisions

TSCH: Channel Hopping

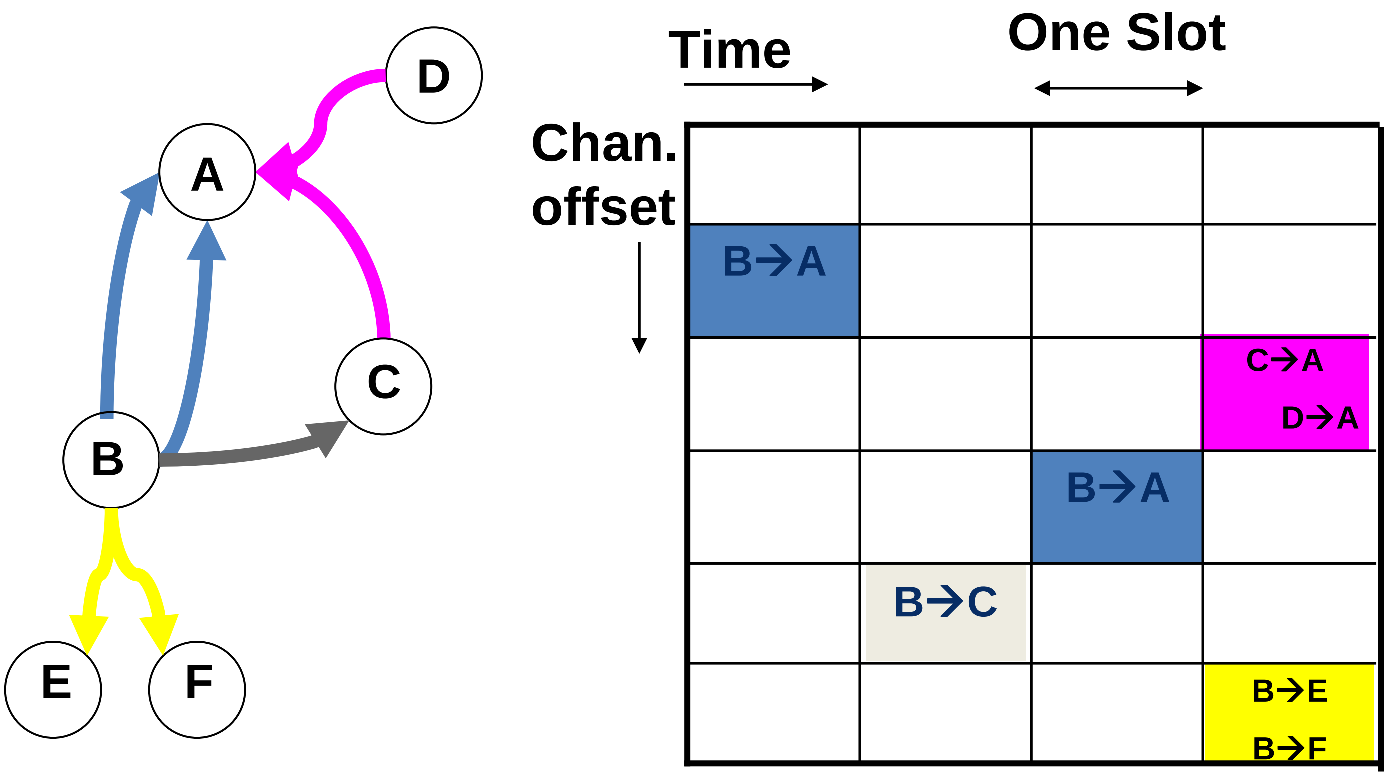


- Combined with timeslot access to enhance reliability

TSCH: Channel Hopping

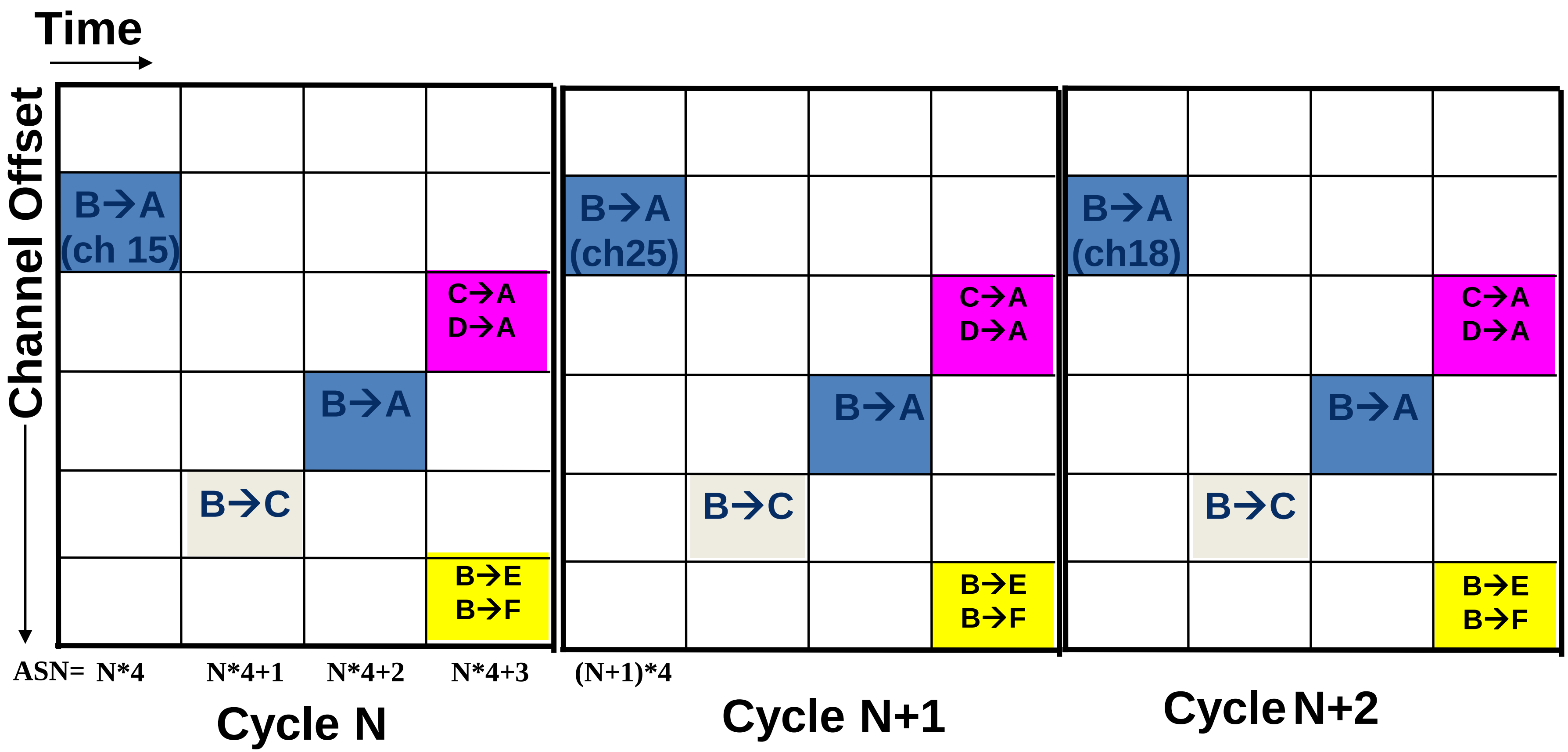
- Mitigate Channel Impairments
  - Channel hopping adds frequency diversity to mitigate the effects of interference and multipath fading
- Increase Network Capacity
  - One timeslot can be used by multiple links at the same time
- More secured

Link = (Timeslot , Channel Offset)



- The two links from B to A are dedicated
- D and C share a link for transmitting to A
- The shared link does not collide with the dedicated links

TSCH: Channel Hopping



- Each link rotates through **k** available channels over **k** cycles.
  - Ch # = Chan Hopping Seq. Table ((ASN + Channel Offset) % Number_of_Channels)
- Blacklisting can be defined globally and locally.

TSCH: operating frequency

- The channel offset is translated in an operating frequency f

$$f = F\{(\mathrm{ASN} + chOf) \mod n_{ch}\}$$

- ASN (Absolute Slot Number): total # of slots that elapsed since the network was deployed (5 bytes):

$$ASN = (k \cdot S + t)$$

where:

- $\boldsymbol{S}$ is the slotframe size, k the slotframe cycle
- $\boldsymbol{n_{ch}}$ *:* number of used channels (16)
- ***t****:* slot in the slotframe
- ***chOf****:* channel offset

$\boldsymbol{F}$ is implemented as a look-up-table containing the sets of available channels

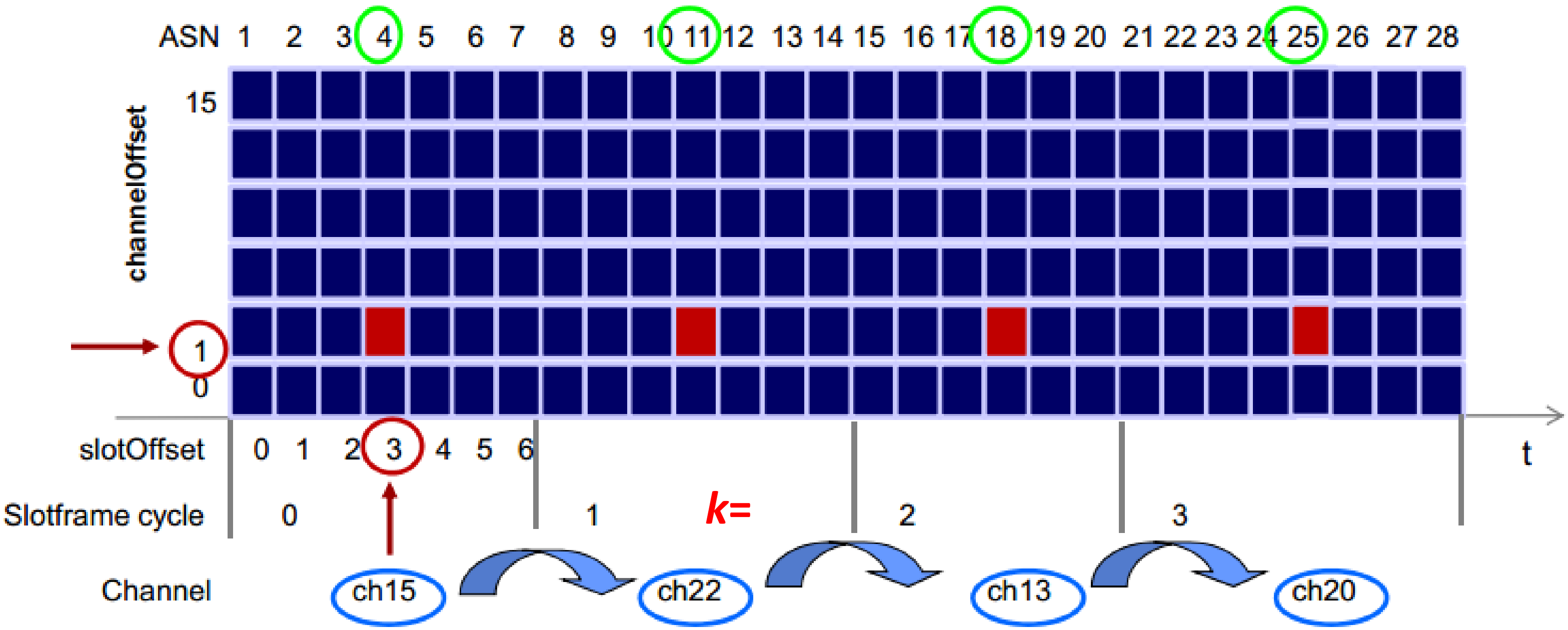
TSCH: frequency

$$f = F\{(\mathrm{ASN} + chOf) \mod n_{ch}\}$$

***n_{ch}*=16** (channel 11-26)

Table I. FrequencyTranslation

k	ASN	$chOf$	f
0	4	1	5
1	11	1	12
2	18	1	3
3	25	1	10

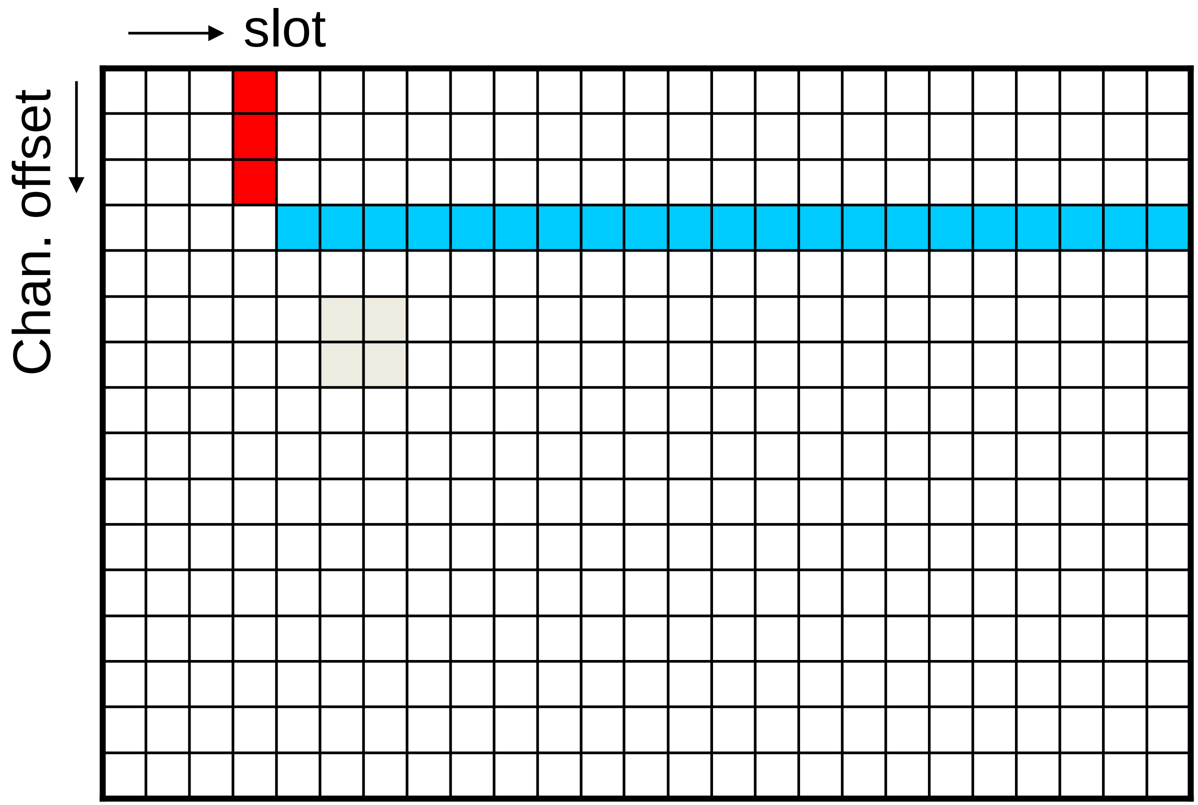


TSCH: Scheduling

- The standard only explains how the MAC layer executes a schedule it does not specify how such a schedule is built
- Centralized Scheduling
  - a *manager node* is responsible for building and maintaining the network schedule
- Distributed Scheduling
  - No central entity
  - Each node decides autonomously

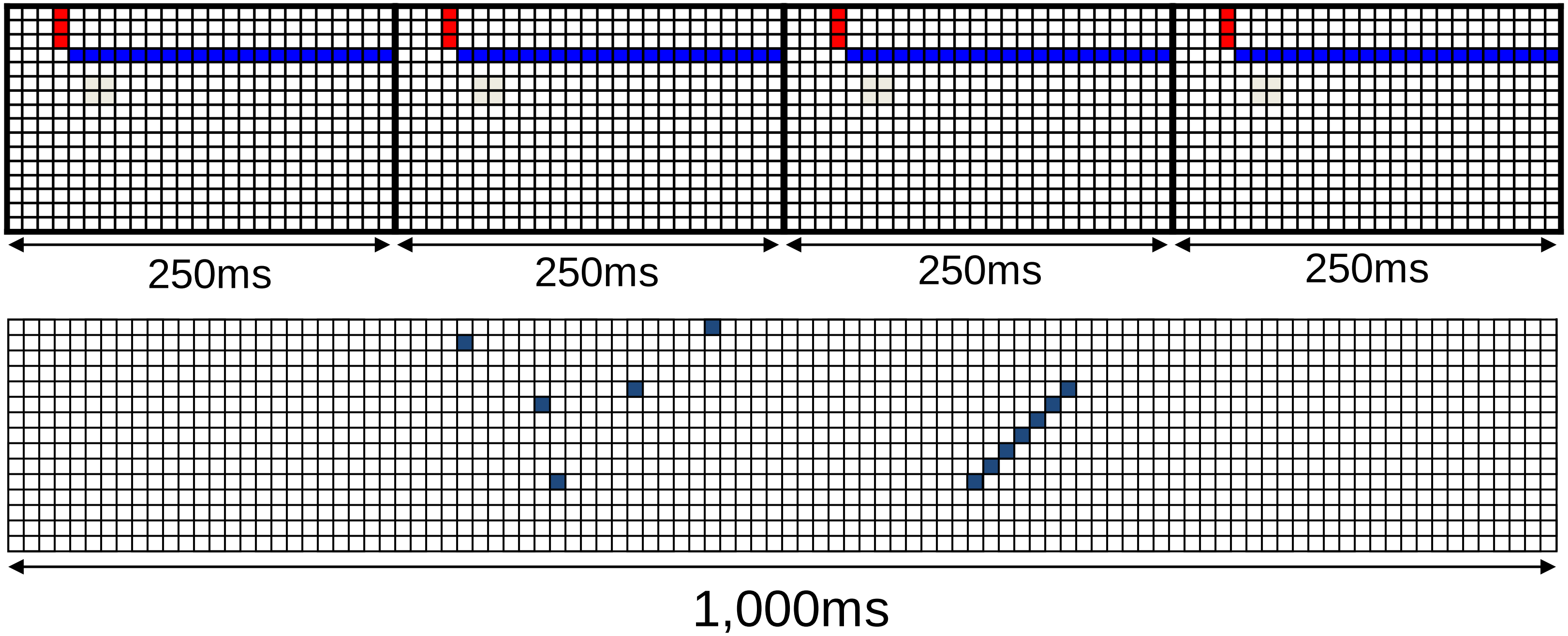
Non-conflicting Timeslot assignment

- Devices with multiple radios can be given one or more offsets.
- Devices can be given one or more slots in a particular slotframe.
- Devices with management ability can be given a block of (slot,offset)s



Non-conflicting timeslot assignment

- Multiple slotframes with different lengths can operate at the same time.
- 4 cycles of the 250ms slotframe are shown, along with a 1000ms slot frame
- There are never collisions if the 1000ms slot frame uses only the empty slots of the 250 ms slot frame



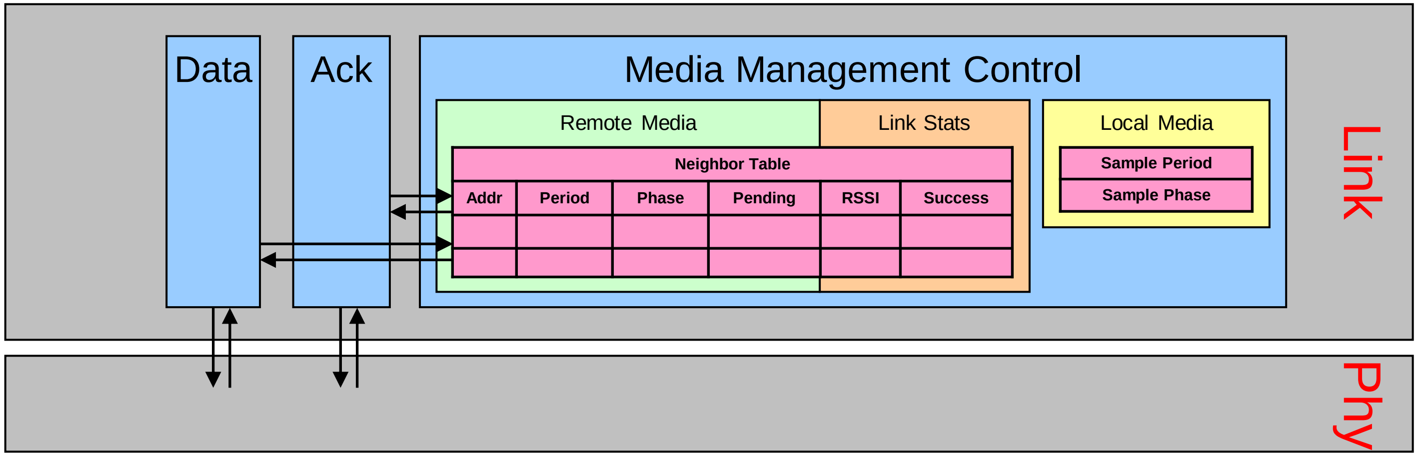
Examples of TSCH Capability

- ❑ Data collection
  - ❖ 100 pkt/s per access point channel using 10 ms slots*
  - ❖ 1600 pkt/s (16*100) network capacity with no spatial reuse of frequency
- ❑ Radio duty cycle (power consumption)
  - ❖ Near theoretical limit for networks with moderate to high traffic
  - ❖ ~0.02% for very low traffic networks
- ❑ Latency
  - ❖ 10ms / PDR (Packet Delivery Rate) per hop: best case
  - ❖ Statistical, but well modeled

* *10 ms slots are an example – the standard can define a range of slot sizes that can be selected for use*

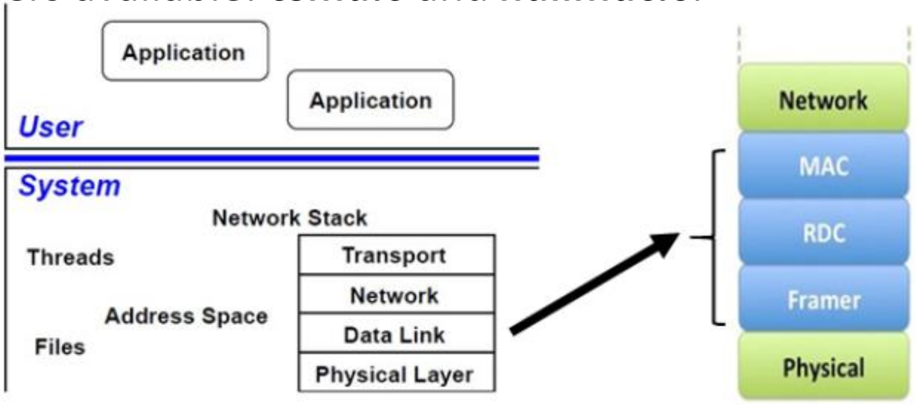
Link Layer – Abstraction

- ❑ Media Management Control
  - ❖ Local Media: Sample Period + Phase
  - ❖ Remote Media: Sample Period + Phase + Pending
- ❑ Neighbor Connectivity Statistics
  - ❖ Signal Quality (RSSI)
  - ❖ Success Rates (windowed average)



Link layer in Contiki OS

- Framer layer is not a regular layer implementation; it is actually a collection of auxiliary functions that are called for creating a frame with data to be transmitted and parsing of data being received. There are two types of Framer layers: ***framer-802154.c*** and ***framer-nullmac.c***.
- Radio Duty-Cycle (RDC) layer takes care of the **sleep period** of nodes. This is the most important layer for deciding exactly when the packets will be transmitted and it is responsible for making sure that the node is awake when packets are to be received. Examples of RDC layers that are implemented: ***contikimac.c, xmac.c, lpp.c, nullrdc.c*** and ***sicslowmac.c***.
- Finally, the MAC layer takes care of addressing and retransmission of lost packets. We have only two types of MAC layers available: ***csma.c*** and ***nullmac.c***.

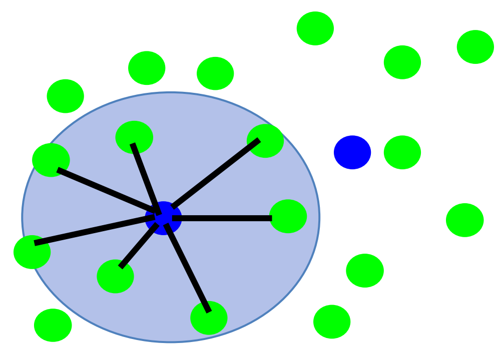


Open Issues for Research and Industry

- Sampled vs Synchronized
  - Is time synchronization required for the network to work at all?
  - Or is it an optimization
- Multi-channel usage
  - Frequency agility
  - Increased diversity
  - Impact on prediction, link assessment
- Distributed vs. Centralized
  - How much is determined locally vs network wide
- Fundamental lower bounds
  - energy is required to maintain structure
  - Where is the optimal point
- Visibility
  - How does network level inform link of relevance
  - How does link level inform network of status

Connectivity

- Much of the CS work on network protocols
  - Routing, cluster head formation, topology formation, …
  - assumes a unit disk model
    - If Distance < R, Connectivity = 1, otherwise 0



  - Empirical models based on fading, signal-to-noise ratio (SNR), modulation, and coding.
    - PRR (packet receive rate) for SNR (g), frame size (f)…

$$p = (1 - \frac{1}{2}\exp^{-\frac{\gamma}{2}\frac{1}{0.64}})^{8f}$$

For Mica2 with FSK CC1000

    - Nakagami and Rayleigh Fading

$$PL(d) = PL(d_0) + 10nlog_{10}(\frac{d}{d_0}) + X_\sigma$$

PRR – Packet Reception Rate

- Calculated based on packet receipts
  - ~RSSI/LQI
- TI CC2420:
  - IEEE 802.15.4 compliant
  - 2400 – 2483.5 MHz RF tranceiver
  - O-QPSK Direct Sequence Spread Spectrum (DSSS)
  - 250 kpbs data rate, 2 Mchips/s
  - 0 dBm (1 mW) transmit power
  - -95 dBm receive sensitivity
  - 30/45 dB adjacent channel rejection
  - 53/54 dB alternate channel rejection
  - PIFA PCB antenna

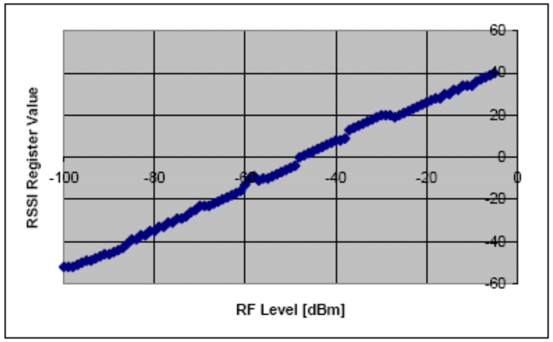
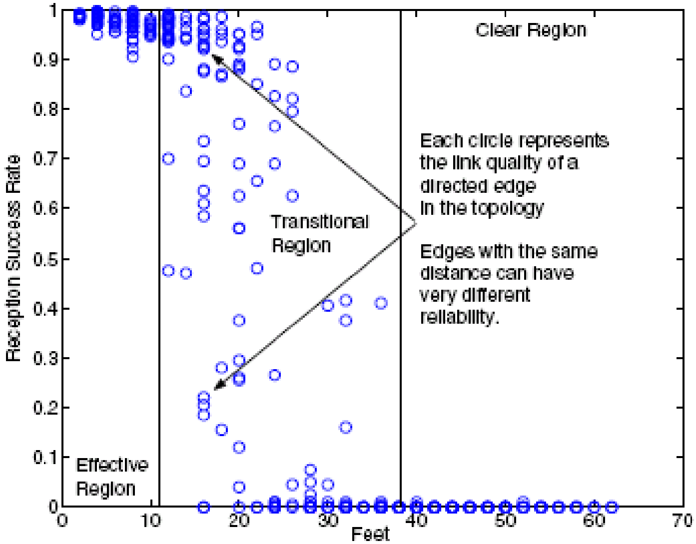


Figure 27. Typical RSSI value vs. input power



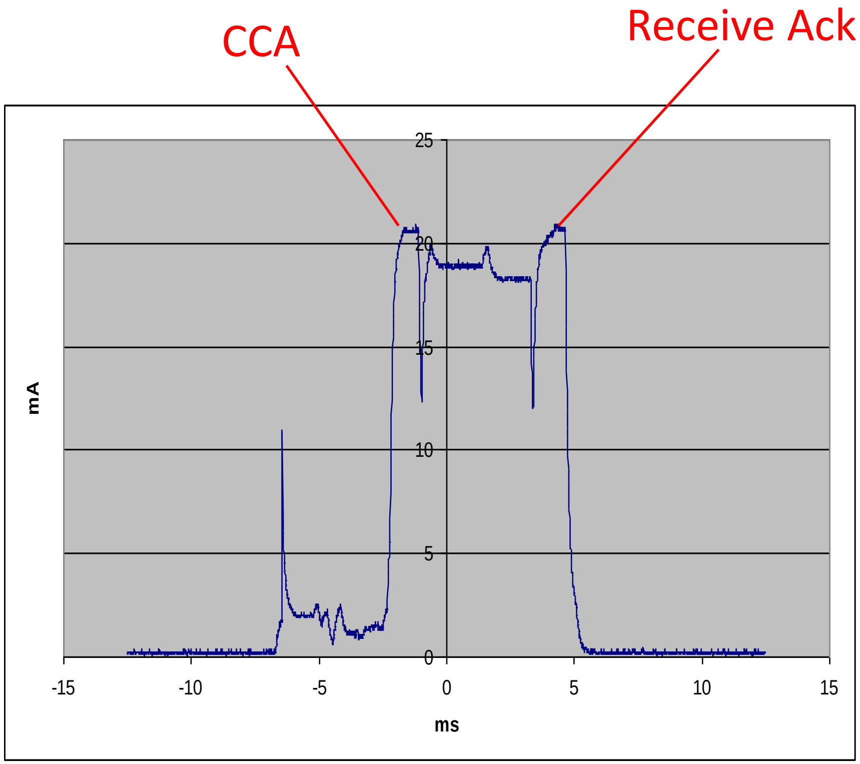
(a) Reception probability of all links in a network with a line topology.

Issue 1: Listen

- MACs deal with mitigating high contention
  - In low duty cycle networks, contention is low regardless
  - Can occur due to highly correlated behavior
    - Example: all node sample periodically
    - Separate sample from report, shift phase
- WSN packets are not the only thing on the air
- CCA essential for determining noise also
  - Don’t transmit over a noisy (or busy) channel
  - Required even with TDMA techniques
- Low-Power Wireless Packets are small
  - IEEE 802.15.4 limit 127 bytes
  - PRR drops rapidly with frame size!
  - Handshake is extremely energy expensive
- In a mesh, hidden terminals and exposed terminals are EVERYWHERE

Issue 2: Link Level Retransmission

- There will be packet loss, even on good links
- End-to-end **PRR(*n*** hops) = **PRR*ⁿ***
- Link-level ACK are essential.
- Provides ability to estimate the quality of the link!



Issue 3: Asymmetric Links

- Asymmetric Links are common
  - Non-isotropic antenna, propagation, multipath
  - Variations in transmit power or receive sensitivity among nodes
  - Variations in noise level at receiver and transmitter
- Cannot assume the reverse link is good
  - Verify it!
  - Continuously

Issue 4: Variation in Time

- Link quality varies in time due to many factors
- Changing physical environment.
- Changing RF noise
- Changing traffic from other nodes
  - Essentially additional noise, but right in the channel
- Cannot expect to determine connectivity in advance and just use it.

Issue 5: Reception $\not\Rightarrow$ Link

- Reception of a packet from a node does not imply the link is good.
- Will infrequently receive good packets even on a bad link.
- Many nodes far away => will frequently receive a packet from one of them.
- Neighbor table cannot contain an entry for everyone that a node has heard from!
- Must track only a small important subset?
- Capture Effect
- A node will receive packets with low signal strength (at least if the noise is low)
- If a strong packet appears at the receiver while it is in the middle of processing a weak packet
  - Both will be lost
- And collision $\not\Rightarrow$ loss
  - If a weak packet arrives while receiving a strong packet, the strong packet will be received if the weak does not exceed the SNR threshold.

Issue 6: Determining Link Quality

- RSSI, LQI, and Packet Sequence numbers tell receiver about the inbound link.
- Need to reply to the sender for it to know if that is a good link.
- The 2-way exchange provides way to filter out asymmetric links.
- Link ACKs inform sender about outbound link.

- The more channels you track the less you know about each
- Need a simple metric for network layer to use

Summary

- Many of the best protocols are opportunistic
  - Use whatever connectivity occurs
- Topology determination and route selection is a constant and gentle process
  - Passive monitoring wherever possible
  - Use every piece of information available to track quality
- Concentrate link estimation on the few important candidates.
- Additional network density helps reliability
  - If the media management is done right