Institute for the Wireless Internet of Things
at Northeastern University

EECE 5155
Wireless Sensor Networks (and The Internet of Things)

Prof. Francesco Restuccia
Email: frestuc@northeastern.edu

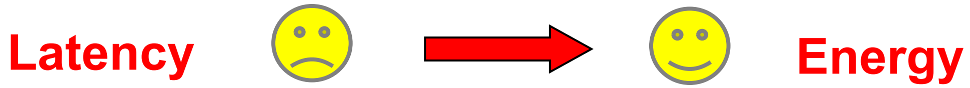
Contention-Based Mac Protocols: S-MAC

S-MAC: Sleep MAC

W. Ye, J. Heidemann, D. Estrin, , "Medium Access Control with Coordinated Adaptive Sleeping for Wireless Sensor Networks," IEEE/ACM Trans. on Networking, June 2004.

- **Problem:** "Idle Listening" consumes significant energy
- **Solution:** Periodic listen and sleep

- During sleeping, radio is turned off
- Reduce duty cycle to ~ 10% (Listen for 200ms and sleep for 2s)



S-MAC

- Each node goes into periodic sleep mode during which it switches the radio off and sets a timer to awake later
- When the timer expires it wakes up and listens to the channel, to see if any other node wants to talk to it
- Requires a **periodic synchronization among nodes** to take care of any type of clock drift

Periodic Sleep and Listen

- All nodes are free to choose their own listen/sleep schedules

- To reduce control overhead, only **neighboring nodes are synchronized together**

- Preferably, they listen at the same time and go to sleep at the same time

Synchronization

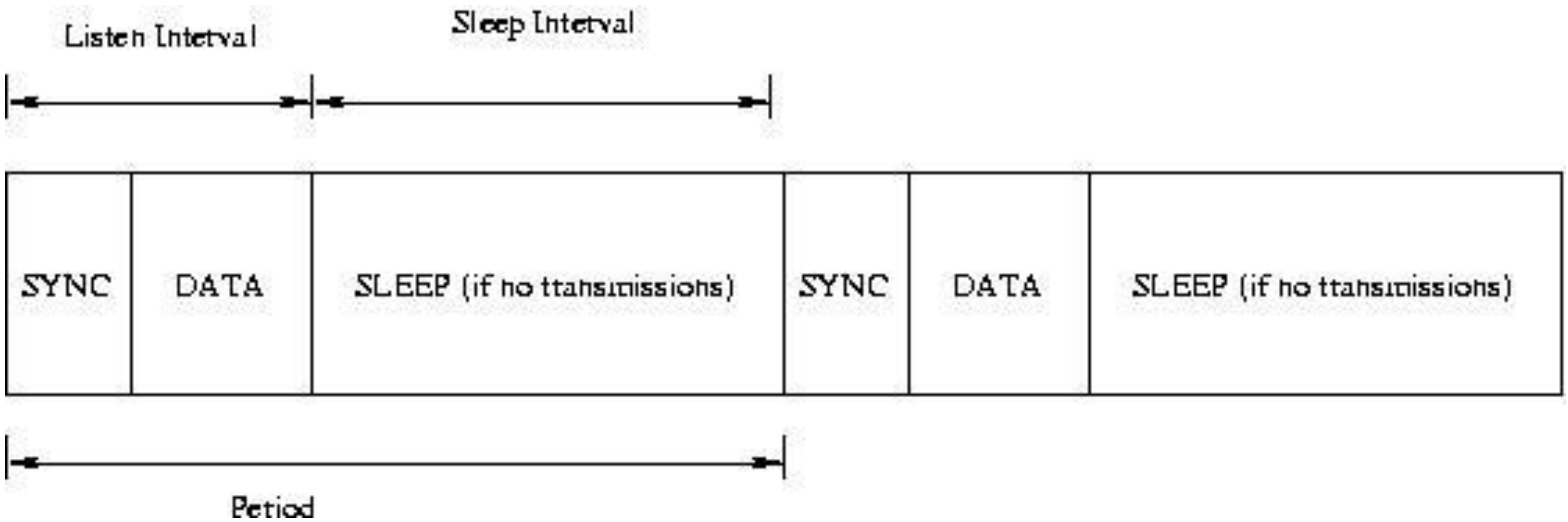
- SYNC packets are exchanged periodically to maintain schedule synchronization

SYNC PACKET

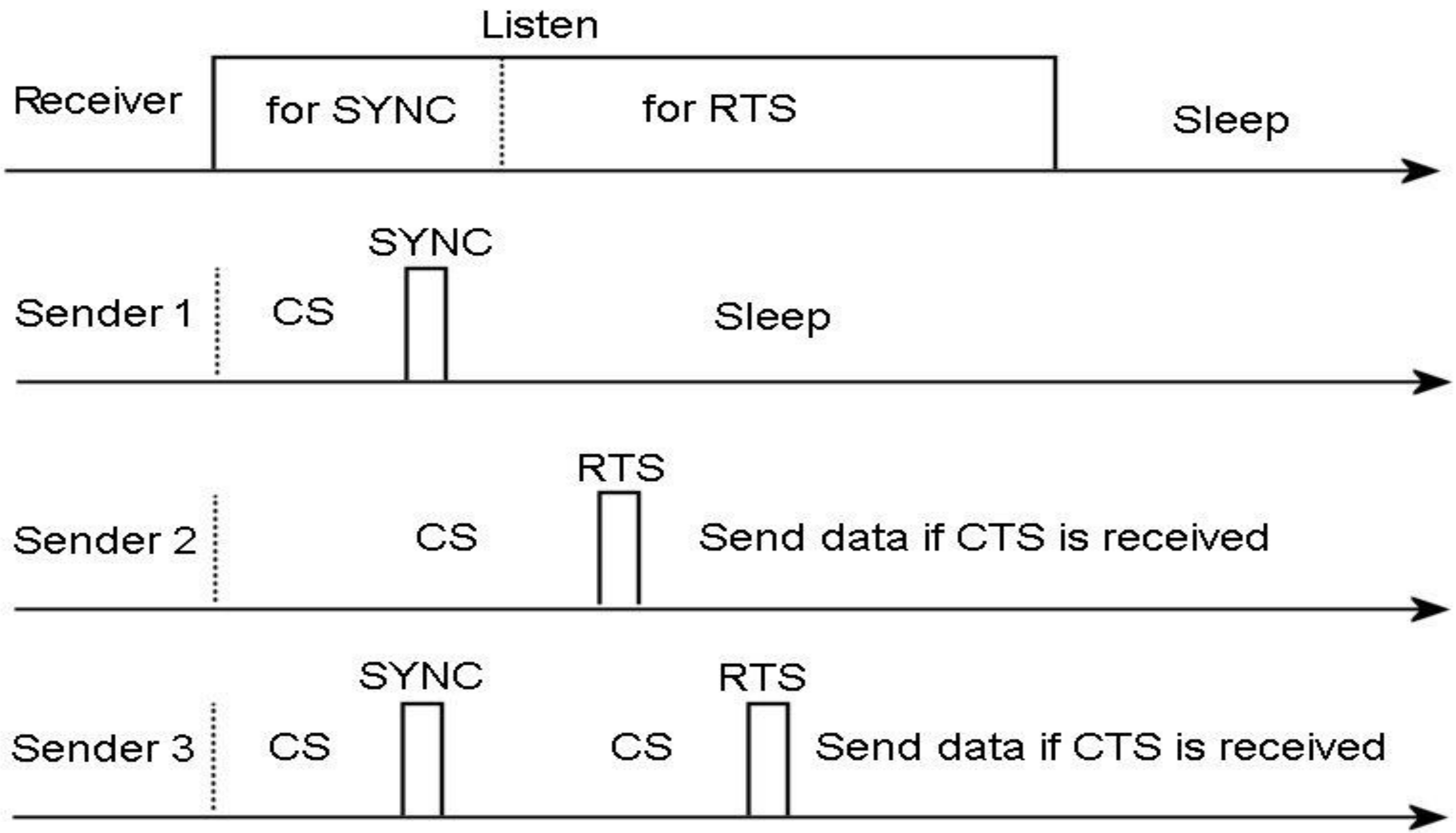
Sender Node ID	Next Sleep Time

- SYNCHRONIZATION PERIOD: Period for a node to send a SYNC packet
- Receivers will adjust their timer counters immediately after they receive the SYNC packet

Periodic Listen and Sleep



Maintaining Synchronization

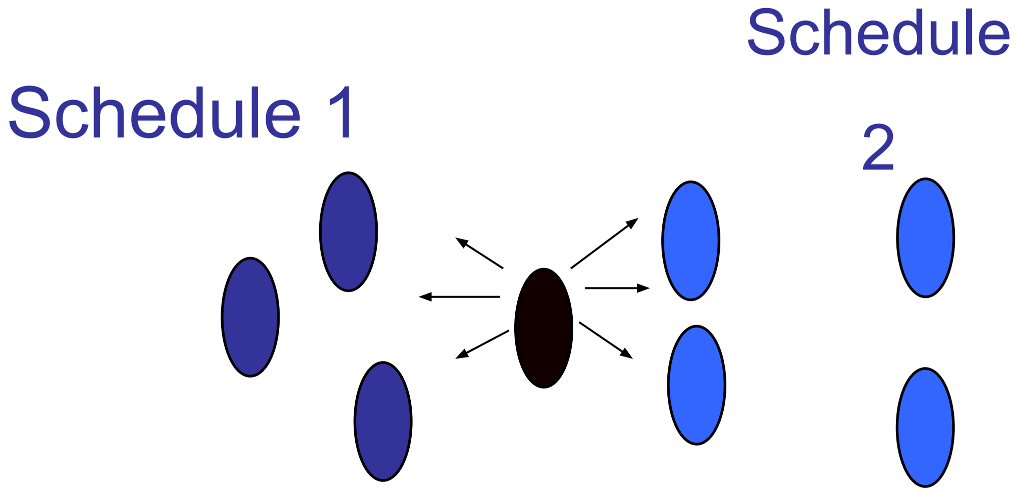


Choosing and Maintaining Schedules

- Each node maintains a schedule table that stores schedules of all its known neighbors
- For initial schedule, do:
  1. A node first listens to the medium for a certain amount of time (at least the synchronization period)
  2. If it does not hear a schedule from another node, it randomly chooses a schedule and broadcasts its schedule with a SYNC packet immediately
  3. This node is called a Synchronizer
  4. If a node receives a schedule from a neighbor before choosing its own schedule, it just follows this neighbor's schedule, i.e. becomes a Follower, waits for a random delay and broadcasts its schedule

Coordinated Sleeping

- In a large network, we cannot guarantee that all nodes follow the same schedule
- The node on the border will follow both schedules
- When it broadcasts a packet, it needs to do it twice, first for nodes on schedule 1 and then for those on schedule 2



Collision Avoidance

- S-MAC is based on **contention**, i.e., if multiple neighbors want to talk to a node at the same time, they will try to send when the node starts listening

- Similar to IEEE 802.11, i.e. use RTS/CTS mechanism to address the hidden terminal problem

- Perform carrier sense before initiating a transmission

Collision Avoidance

- If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again
- Broadcast packets are sent without using RTS/CTS
- Unicast data packets follow the sequence of RTS/CTS/DATA/ACK between the sender and receiver
- Duration field in each transmitted packet indicates how long the remaining transmission will be so if a node receives a packet destined to another node, it knows how long it has to keep silent
- The node records this value in network allocation vector (NAV) and sets a timer for it

Collision Avoidance

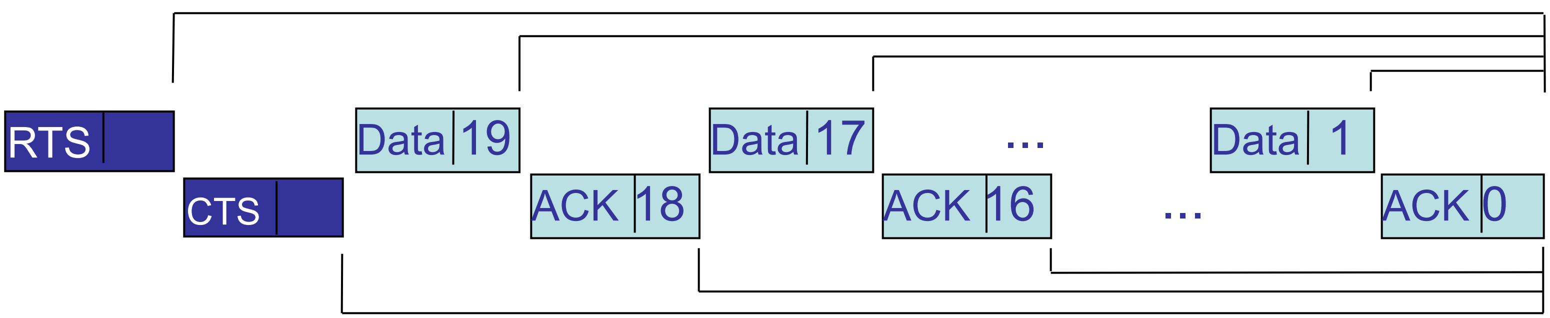
- When a node has data to send, it first looks at NAV
- If this value is not zero, then medium is busy (virtual carrier sense)
- The medium is determined as free if both virtual and physical carrier sense indicate the medium is free
- All immediate neighbors of both the sender and receiver should sleep after they hear RTS or CTS packet until the current transmission is over

Message Passing Feature

- Long messages are broken down into smaller packets and sent continuously once the channel is acquired by RTS/CTS handshake
- Increases the sleep time, but leads to **fairness** problems

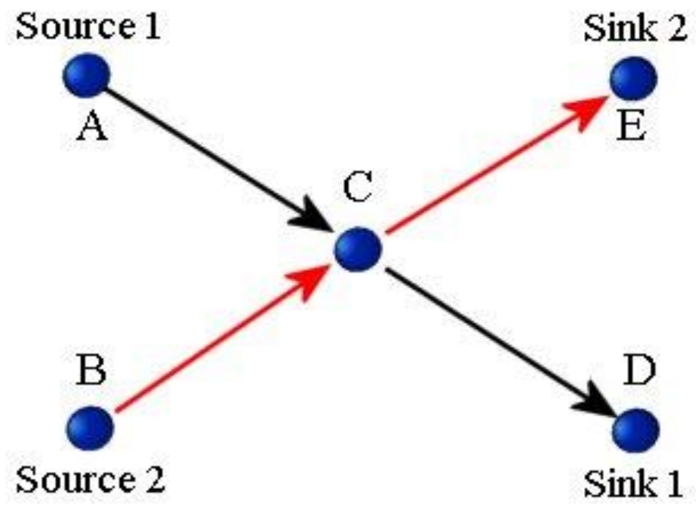
Msg Passing

- S-MAC message passing



S-MAC – Performance Evaluation

- Topology: Two-hop network with two sources and two sinks
- Sources periodically generate a sensing message which is divided into fragments
- Traffic load is changed by varying the inter-arrival period of the messages: (for inter-arrival period of 5s, message is generated every 5s by each source. Here it varies between 1-10s)

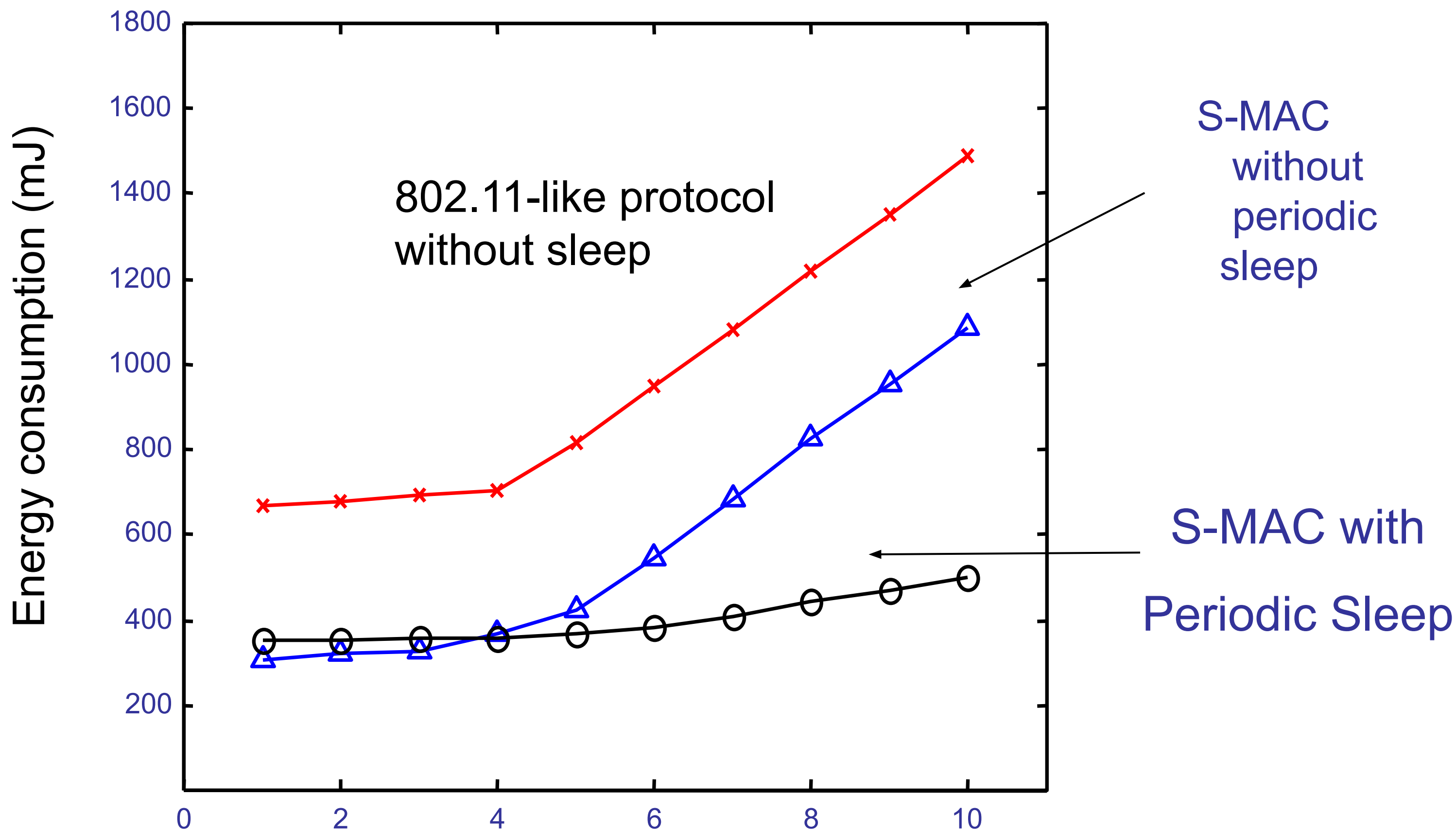


S-MAC – Example

- In each test, there are 10 messages generated on each source node
- Each message has 10 fragments, and each fragment has 40 bytes (200 data packets to be passed from sources to sinks)
- The total energy consumption of each node is measured for sending this fixed amount of data

Experiments

Average energy consumption in the source nodes A&B



Message inter-arrival period (traffic load) (second)
(small value → heavy traffic load)

Experiments

- S-MAC consumes much less energy than 802.11-like protocol without sleeping
- At heavy load, idle listening rarely happens, energy savings from sleeping is very limited.

- At light load, periodic sleeping plays a key role

Let's think...

What are the pros and cons of S-MAC?

How would you improve it?

Think-Share!

S-MAC - Conclusions

- A mainly static network is assumed
- Trades off latency for reduced energy consumption
- Redundant data is still sent with increased latency
- Increased collision rate due to sleep schedules

Contention-Based Mac Protocols: B-MAC

B-MAC

J. Polastre, J. Hill, D. Culler, "Versatile Low Power Media Access for WSNs", Proc. of ACM SenSys, Nov. 2004.

- What is B-MAC?
  - A configurable MAC protocol for WSNs
  - Small core and factoring out higher-level functionality
  - An adaptive bidirectional interface for WSN applications
- It's a traditional MAC protocol for WSNs
  - Work reasonably well for a large set of traffic workloads (applications)
  - Create a flexible set of functionalities able to provide solutions to a set of goals

B-MAC Goals

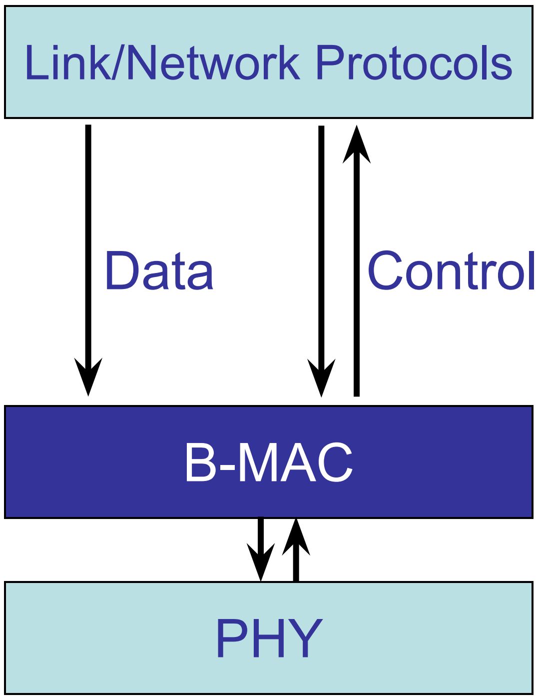
- Goals
  - Low Power Operation
  - Effective Collision Avoidance
  - Simple Implementation, Small Code and RAM Size
  - Efficient Channel Utilization
  - Reconfigurable by Network Protocols
  - Tolerant to Changing RF/Networking Conditions
  - Scalable to Large Numbers of Nodes

Classical vs. Minimalistic

- S-MAC
  - A classic approach
  - User pre-configures duty cycle
  - Applications rely on S-MAC to adjust its operation as the environment evolves
- B-MAC
  - A minimalistic approach
  - Small core of medium access functionalities
  - RTS/CTS, ACKs, etc. are considered higher layer functionalities (services)
  - Uses Clear Channel Assessment (CCA) and packet backoffs for channel arbitration, link layer acknowledgments for reliability
  - Uses Low Power Listening (LPL)
  - More flexible and more tunable

B-MAC: Principles

- Reconfigurable parameters:
  - Backoff/Timeouts
  - Duty Cycle
  - Optional ACKs
- Flexible control
- Feedback to higher protocols
  - Model of operation
  - Upward costs (e.g., link quality)
- Minimal implementation
- Minimal state



B-MAC Tiny OS Interface

- Interfaces for flexible control of B-MAC by higher layer services
- Allow services to toggle CCA and ACKs, set backoffs on a per message basis, and change the LPL mode for transmit and receive

```
interface MacControl {
  command result_t EnableCCA();
  command result_t DisableCCA();
  command result_t EnableAck();
  command result_t DisableAck();
  command void* HaltTx();
}

interface MacBackoff {
  event uint16_t initialBackoff(void* msg);
  event uint16_t congestionBackoff(void* msg);
}

interface LowPowerListening {
  command result_t SetListeningMode(uint8_t mode);
  command uint8_t GetListeningMode();
  command result_t SetTransmitMode(uint8_t mode);
  command uint8_t GetTransmitMode();
  command result_t SetPreambleLength(uint16_t bytes);
  command uint16_t GetPreambleLength();
  command result_t SetCheckInterval(uint16_t ms);
  command uint16_t GetCheckInterval();
}
```

B-MAC Important Design Aspects

- Clear Channel Assessment (CCA)
- Low Power Listening (LPL)
- Packet backoffs
- Link layer acknowledgments

Clear Channel Assessment (CCA)

- Effective collision avoidance

- Find out whether the channel is idle
  - If too pessimistic: waste bandwidth
  - If too optimistic: more collisions

- Key points:
  - Ambient noise may change significantly depending on the environment
  - Packet reception has fairly constant channel energy
  - What is noise? What is signal?
  - **How can we be certain that the channel is free?**

- Automatic gain control in software to estimate the noise floor

Noise Floor Estimation

- Take a number of received signal strength indicator (RSSI) samples when the channel is assumed to be free/idle
- RSSI samples are entered into a FIFO queue
- Median of the queue is added to an exponentially weighted moving average with decay *a*
- Median is used as a simple low pass filter to add robustness to the noise floor estimate

$$A_t = a * S_t + (1 - a) * S_t-1$$

where *a* is assumed to be 0.06 and FIFO queue size of 10

- Once a good estimate of the noise floor is established, a request to transmit a packet starts the process of monitoring the received signal from the radio

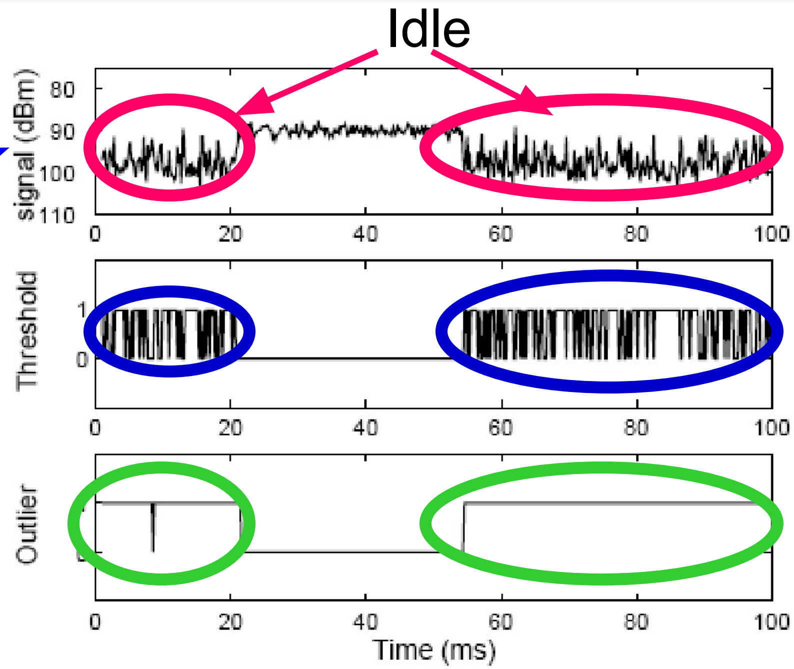
Single-Sample Thresholding vs Outlier Detection

- Common approach: take single sample, compare to noise floor
  - Large number of false negatives → lower effective channel BW
- B-MAC: search for outliers
  - If a sample has significantly lower energy than the noise floor during the sampling period, then the channel is clear

If 5 samples are taken and no outlier is found, the channel is busy

CCA vs. Threshold Techniques

Signal Strength Indicator (RSSI) from transceiver



- A packet arrives between 22 and 54ms. The middle graph shows the output of a thresholding CCA algorithm (1: channel clear, 0: channel busy)
- Bottom shows the output of an outlier detection algorithm
- Threshold: waste channel utilization
- CCA: Fully utilize the channel since a valid packet could have no outlier significantly below the noise floor

Clear Channel Assessment - Recap

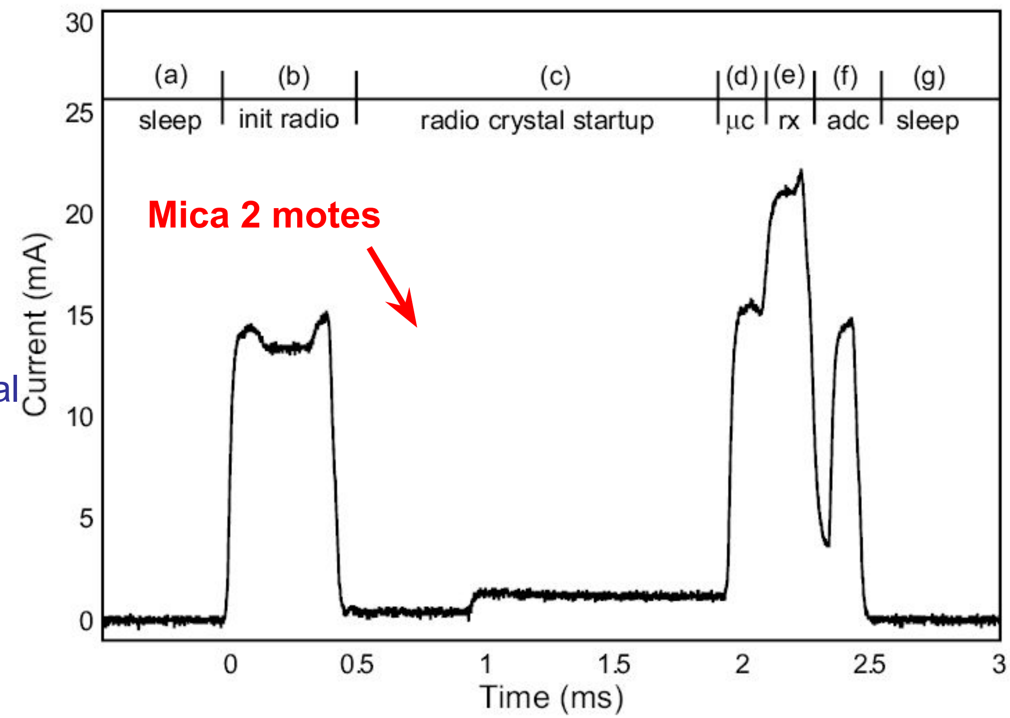
- Before transmission – take samples of the channel

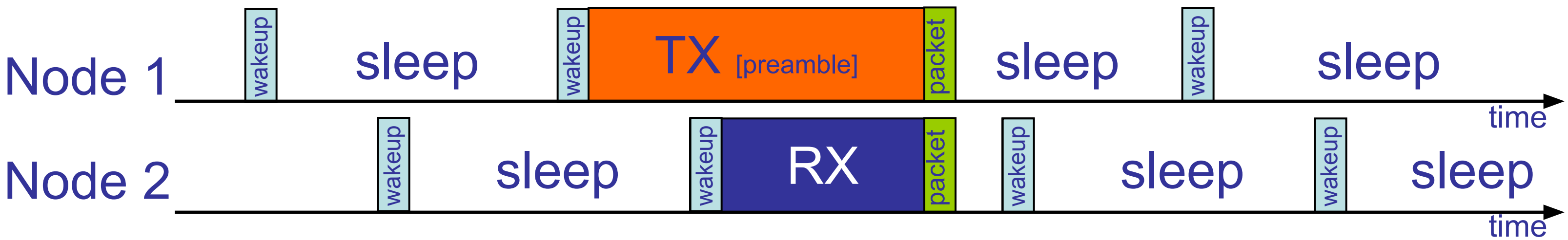
- If five samples are taken, and no outlier found => channel busy, take a random backoff

- Noise floor updated when the channel is known to be clear, i.e., just after packet transmission

Low Power Listening (LPL)

- Periodic Channel Sampling
  - Energy Cost = RX + TX + Listen
  - Goal: **minimize idle listening**
- Periodically
  - Wake up, sample channel, sleep
- Properties
  - Wake-up time fixed
  - “Check Time” between wakeups variable
  - Preamble length matches wakeup interval
- Overhear all data packets
  - Duty cycle depends on number of neighbors and traffic





Low Power Listening (LPL) - 2

- Goal: Minimize "Listen Cost"
- Principles
  - Node periodically wakes up, turns radio on and checks activity on the channel
  - If energy/activity on the channel is detected, node powers up and stays awake for the time required to receive the incoming packet
- Node goes back to sleep
  - If the complete packet is received
  - After a timeout (if no packet received (a false positive))
- Preamble length matches channel checking period
  - No explicit synchronization required
- Noise floor estimation used to detect channel activity during LPL

Check Interval for Channel Activity

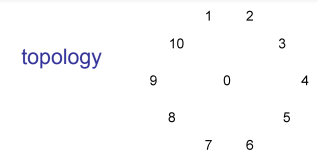
- To reliably receive data, the preamble length is matched to the interval that the channel is checked for activity
- If the channel is checked for every 100 ms, the preamble must be at least 100 ms long for a node to wake up, detect activity on the channel, receive the preamble and then receive the message
- Interval between LPL samples is maximized so that the time spent sampling the channel is minimized
  - Transmit mode ~~ Preamble length
  - Listening mode ~~ Check interval

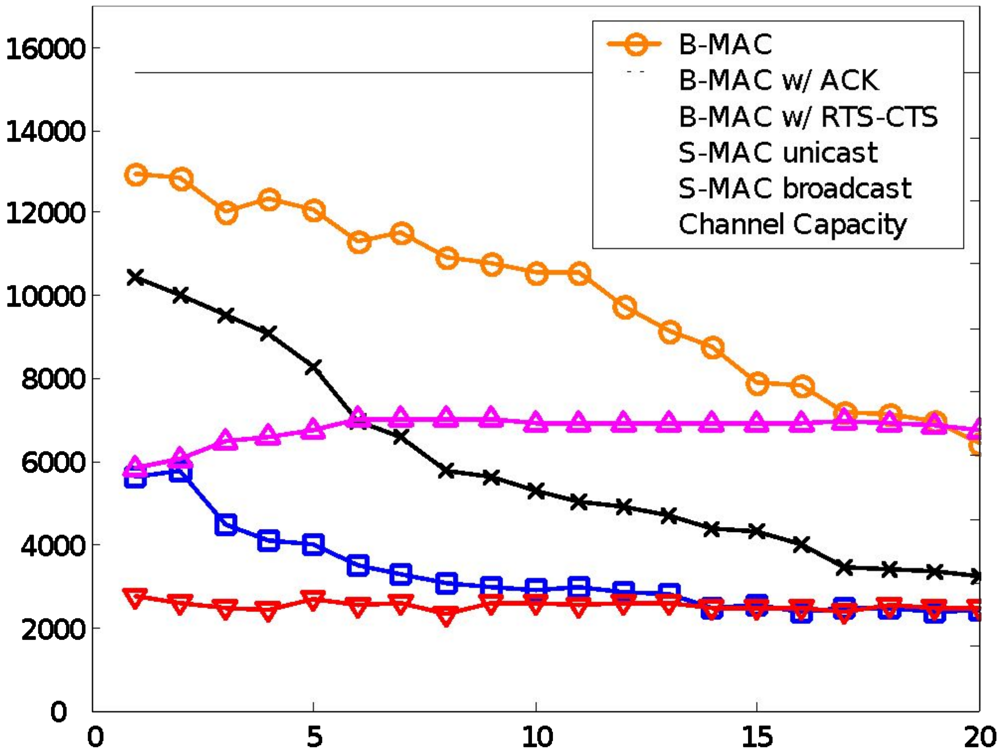
LPL Check Interval

- Sampling rate (traffic pattern) defines optimal check interval
- Check interval
  - Too small: energy wasted on idle listening
  - Too large: energy wasted on transmissions (long preambles)
- In general, it is better to have larger preambles than to check more often!

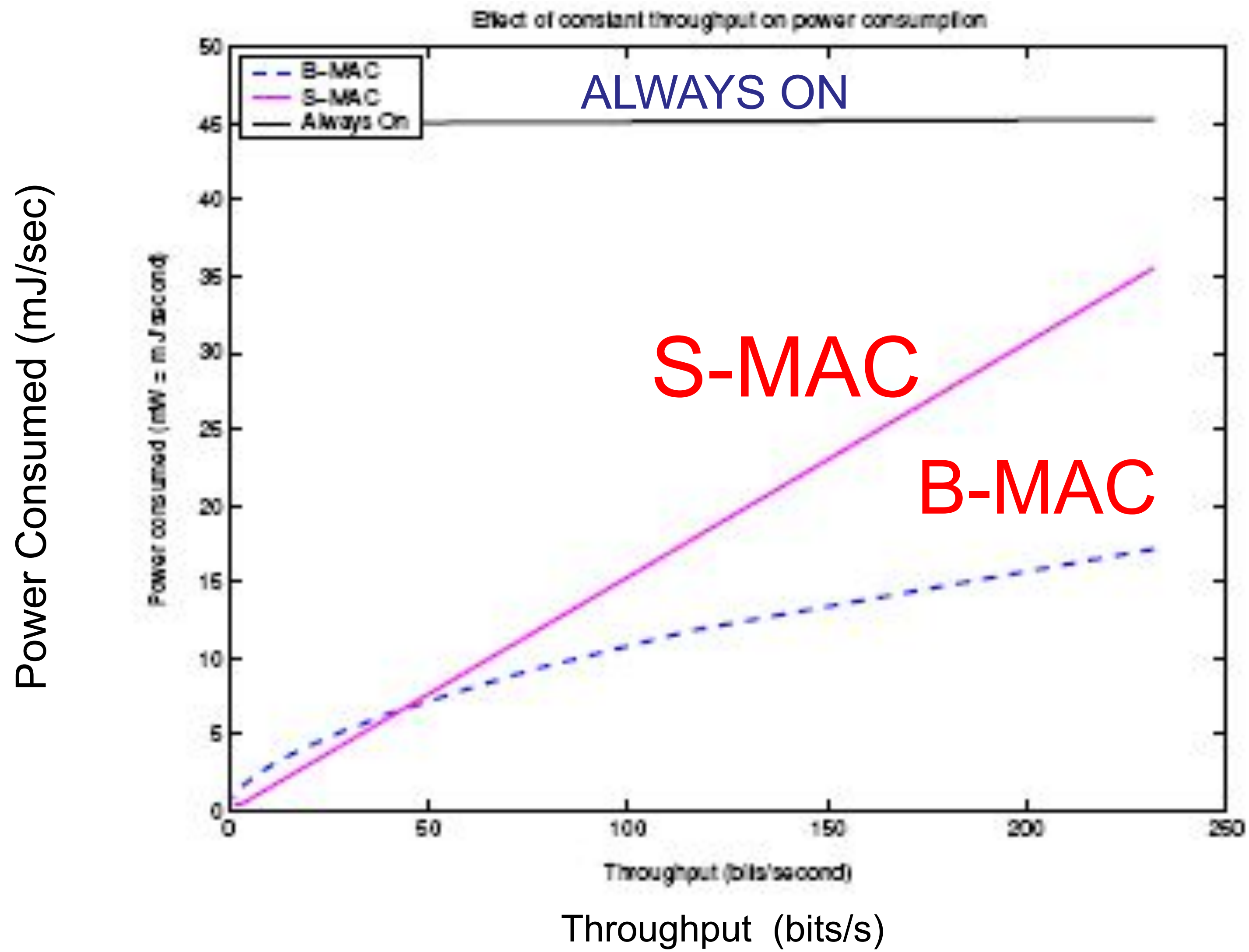
S-MAC and B-MAC: Comparison

- Experimental Setup:
  - *n* nodes send as quickly as possible to saturate the channel
  - Shown: throughput as a function of the number of nodes
- B-MAC has about 4.5 time higher throughput than S-MAC-unicast
  - Not as much when ACK or RTS/CTS is used
  - Differences less pronounced as # of nodes increases
  - B-MAC has CCA, thus it backs off less frequently



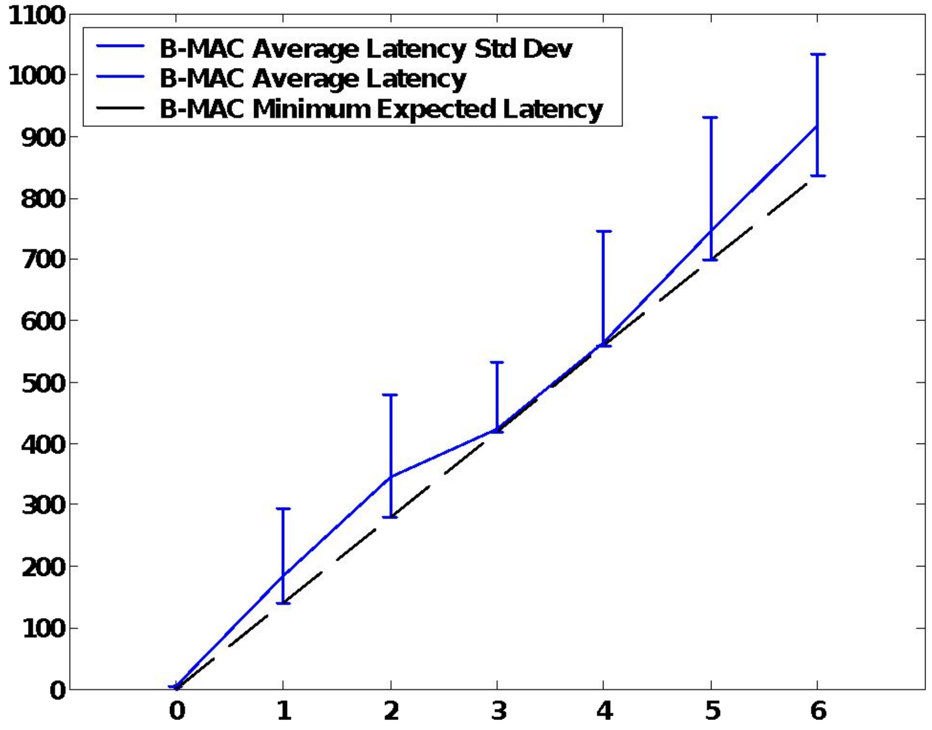


Throughput vs Power Consumption

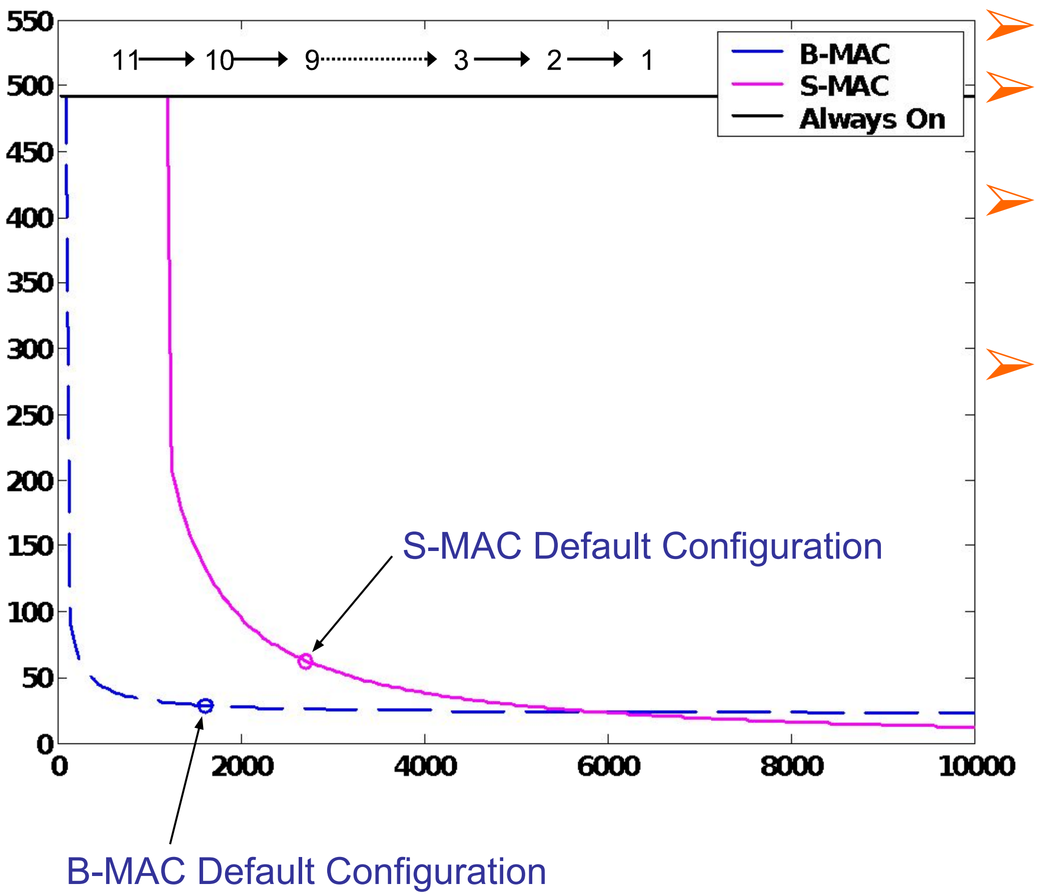


Tradeoffs: Latency vs Reliability

- **Reliability**
  - 98.5% of all packets delivered
  - Some nodes 100% delivery
- …but communication links are volatile
  - Retransmissions required
  - After **5** retries, give up and pick a new parent
- Actual latency
  - Retransmission delay
  - Contention delay (infrequent)

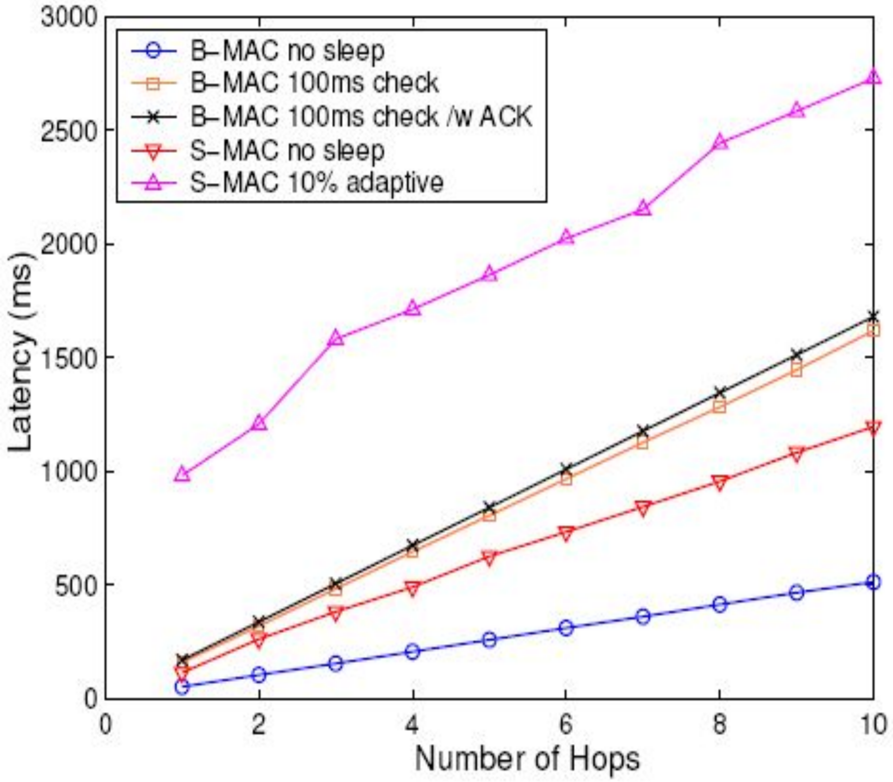


Tradeoffs: Latency for Energy



- 10-hop network
- Source sends a 100-byte packet every 10 seconds
- SMAC: 10% Duty Cycle
  BMAC: choose optimal check interval
- Poor SMAC again performs worse...
  - Reason: sync packets, probability of multiple schedules---less time to sleep

Latency



B–MAC no sleep
B–MAC 100ms check
B–MAC 100ms check /w ACK
S–MAC no sleep
S–MAC 10% adaptive
Latency (ms)
Number of Hops
3000
2500
2000
1500
1000
500
0
0
2
4
6
8
10

Let’s think...

Can you compare S-MAC vs B-MAC?

Think-Share!

Comparison of S-MAC and B-MAC

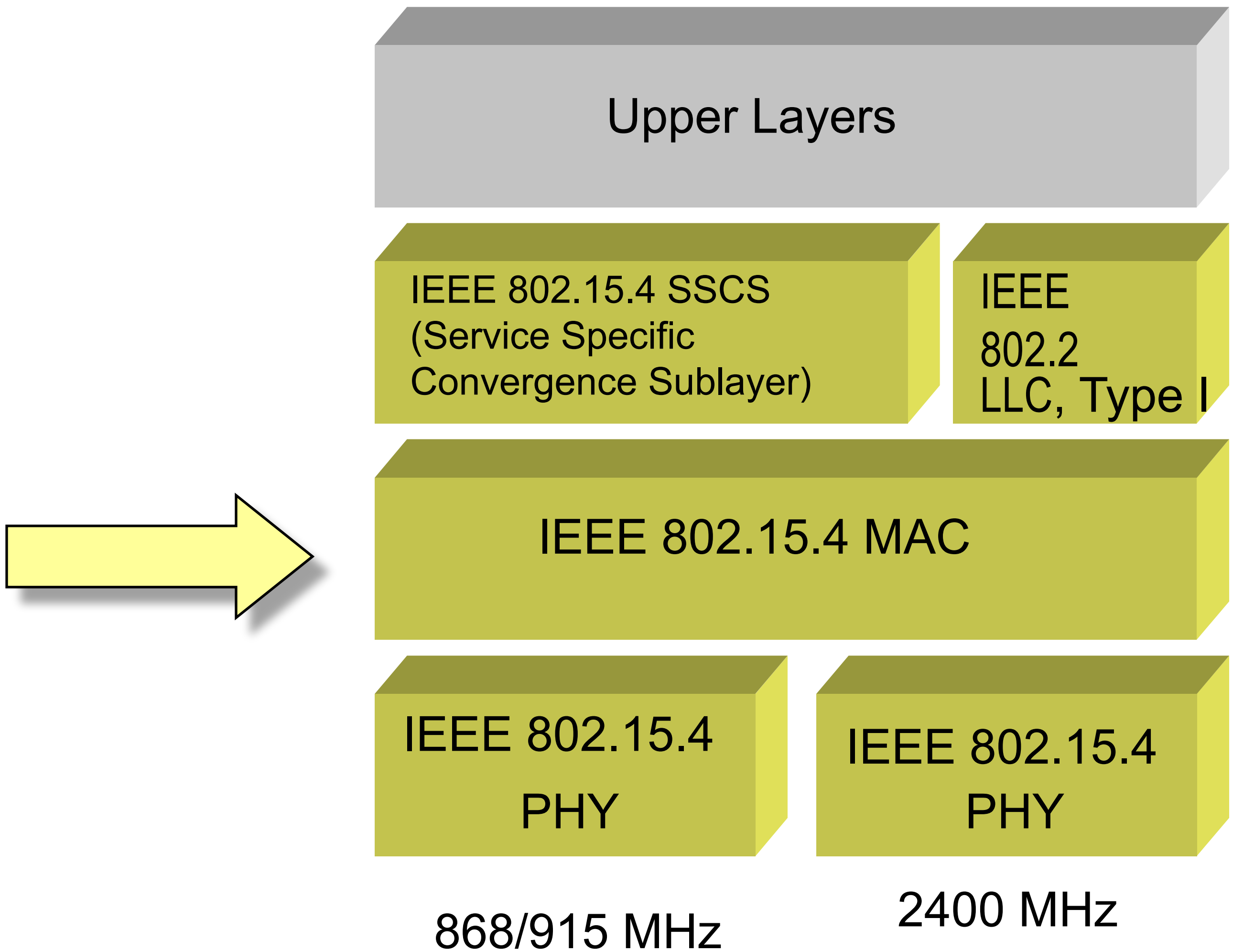
	S-MAC	B-MAC
Collision avoidance	CSMA/CA	CSMA
ACK	Yes	Optional
Message passing	Yes	No
Listen period	Pre-defined	Pre-defined
Listen interval	Long	Very short
Schedule synchronization	Required	Not required
Packet transmission	Short preamble	Long preamble
Code size	6.3KB	4.4KB (LPL & ACK)

IEEE 802.15.4
MAC LEVEL

Content

- Overview
- Topologies
- Superframe structure
- Frame formatting
- Data service
- Management service
- Interframe spacing
- CSMA procedure

802.15.4 Architecture



What IEEE 802.15.4 Aims at

- Extremely low cost
- Ease of installation
- Reliable data transfer
- Short range operation
- Reasonable battery life

MAC Overview

- Star and peer-to-peer topologies
- Association
- CSMA-CA channel access mechanism
- Packet validation and message rejection
- Optional guaranteed time slots (GTS)
- Guaranteed packet delivery
- Facilitates low-power operation
- Security

IEEE 802.15.4 Device Classes

- Full function device (FFD)
  - Any topology
  - Capable to be a PAN coordinator
  - Talks to any other device
  - Implements complete protocol set
- Reduced function device (RFD)
  - Limited to star topology or end-device in a peer-to-peer network.
  - Cannot become a PAN coordinator
  - Very simple implementation
  - Reduced protocol set

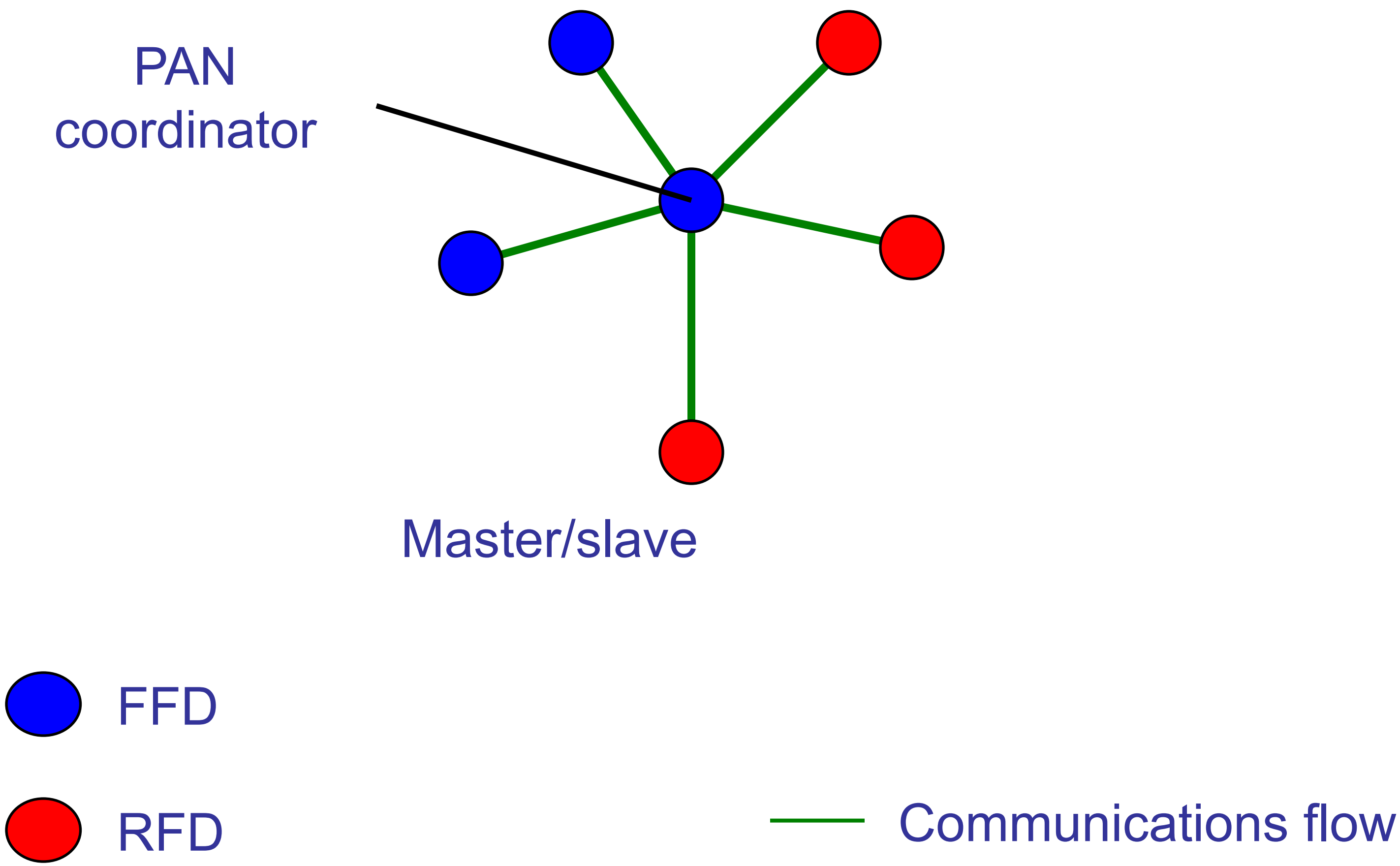
IEEE 802.15.4 Definitions

- **Network Device:** An RFD or FFD implementation containing an IEEE 802.15.4 medium access control and physical interface to the wireless medium.
- **Coordinator:** An FFD with network device functionality that provides coordination and other services to the network.
- **PAN Coordinator:** A coordinator that is the principal controller of the PAN. A network has exactly one PAN coordinator.

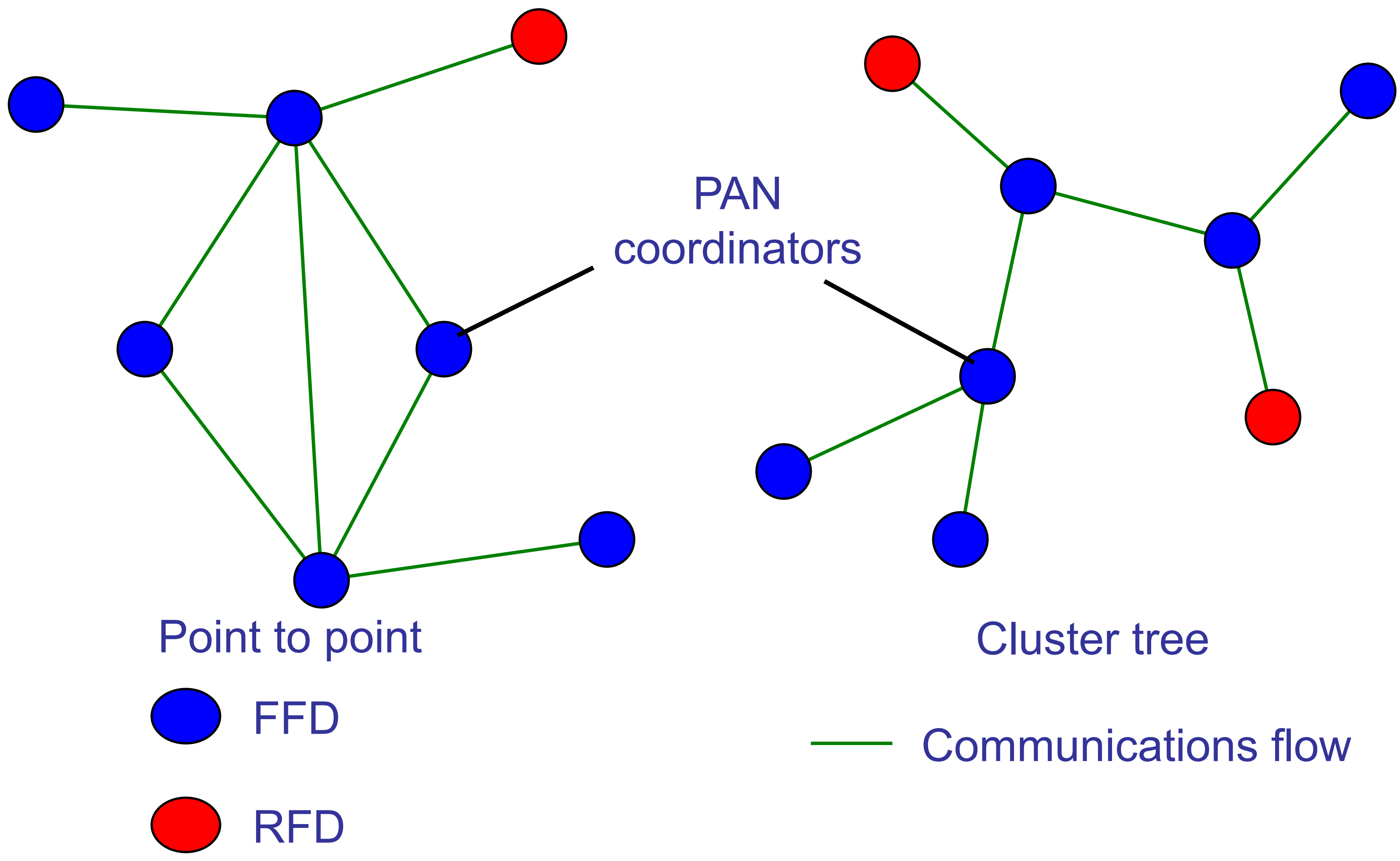
Low-Power Operation

- Duty-cycle control using superframe structure
  - Beacon order and superframe order
  - Coordinator battery life extension
- Indirect data transmission
- Devices may sleep for extended period over multiple beacons
- Allows control of receiver state by higher layers

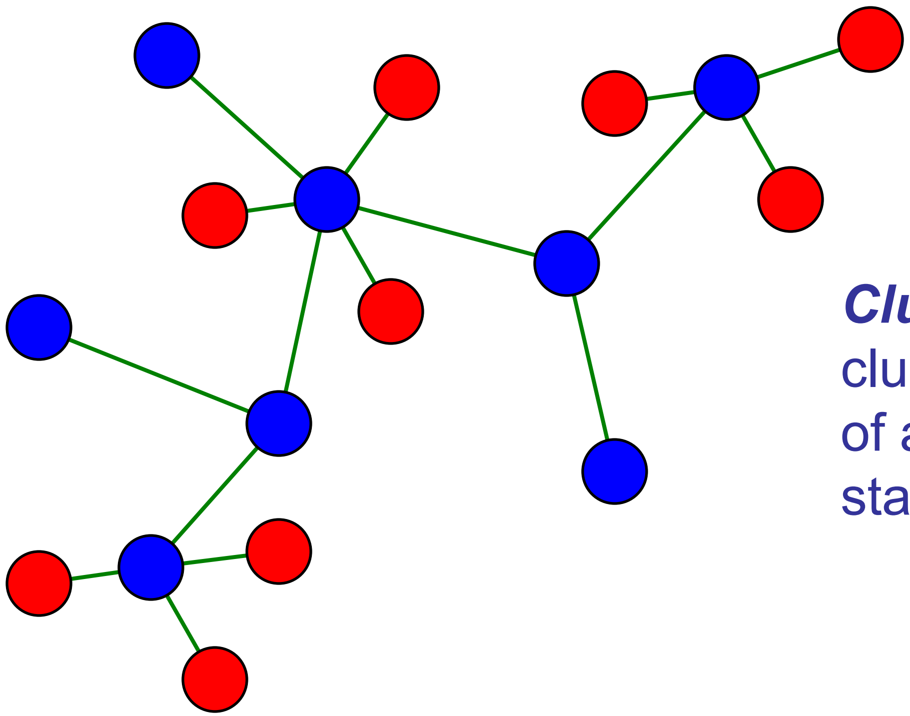
Star Topology



Peer-Peer Topology



Combined Topology



Clustered stars - for example, cluster nodes exist between rooms of a hotel and each room has a star network for control.

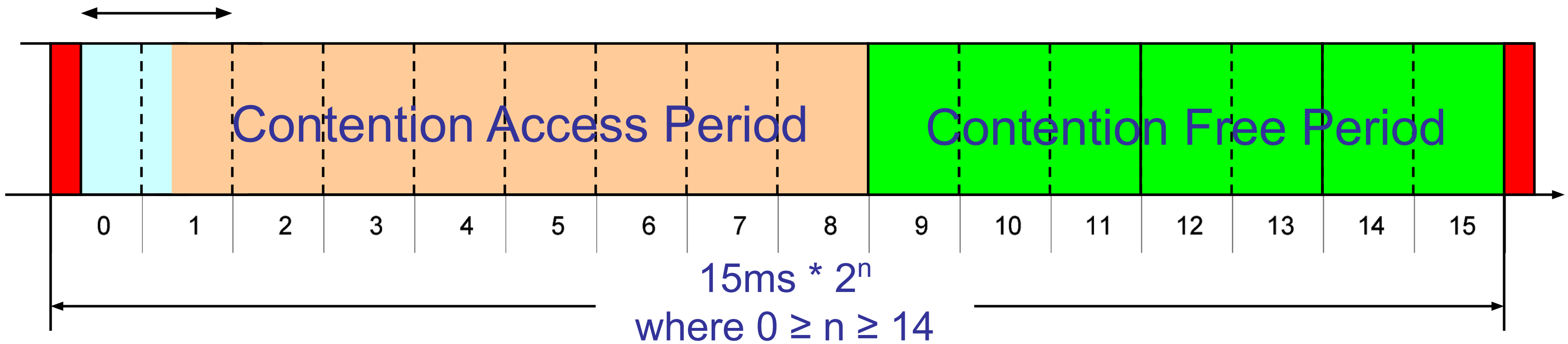
FFD

RFD

Communications flow

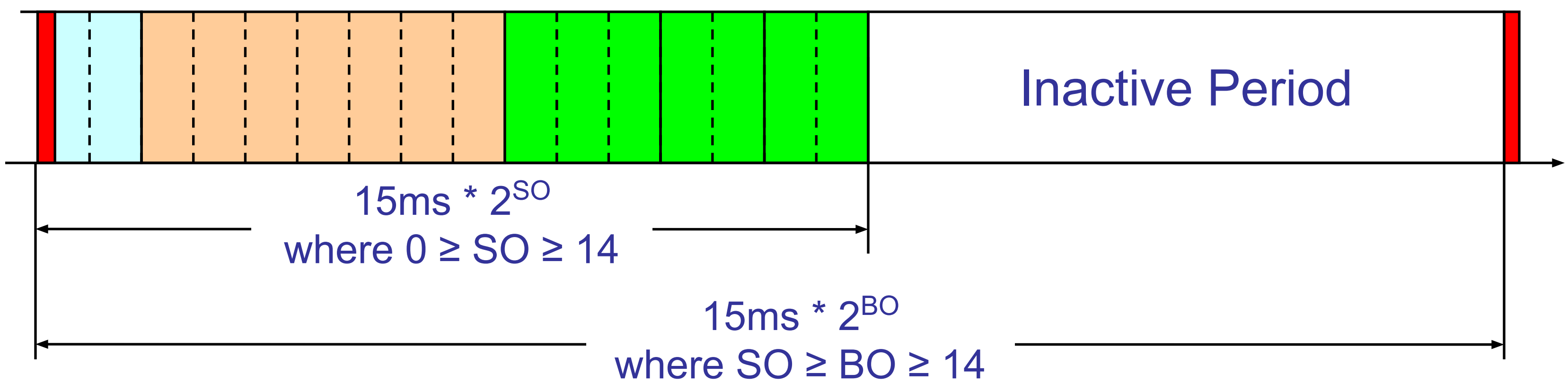
Optional Frame Structure

Battery life extension



Network beacon	(red)	Transmitted by PAN coordinator. Contains network information, frame structure and notification of pending node messages.
Beacon extension period	(light blue)	Space reserved for beacon growth due to pending node messages
Contention period	(peach)	Access by any node using CSMA-CA
Guaranteed Time Slot	(green)	Reserved for nodes requiring guaranteed bandwidth [n = 0].

Optional Frame Structure



SO = Superframe order
BO = Beacon order

- Superframe may have optional inactive period (duty cycle)

General MAC Frame Format

Octets:2	1	0/2	0/2/8	0/2	0/2/8	variable	2
Frame control	Sequence number	Destination PAN identifier	Destination address	Source PAN identifier	Source address	Frame payload	Frame check sequence
		Addressing fields					
MAC header						MAC payload	MAC footer

Bits: 0-2	3	4	5	6	7-9	10-11	12-13	14-15
Frame type	Sequrity enabled	Frame pending	Ack. Req.	Intra PAN	Reserved	Dest. addressing mode	Reserved	Source addressing mode

Frame control field

Beacon Frame Format

Octets:2	1	4 or 10	2	variable	variable	variable	2
Frame control	Beacon sequence number	Source address information	Superframe specification	GTS fields	Pending address fields	Beacon payload	Frame check sequence
MAC header			**MAC payload**				**MAC footer**

Bits: 0-3	4-7	8-11	12	13	14	15
Beacon order	Superframe order	Final CAP slot	Battery life extension	Reserved	PAN coordinator	Association permit

MAC Command Frame

Octets:2	1	4 to 20	1	variable	2
Frame control	Data sequence number	Address information	Command type	Command payload	Frame check sequence
MAC header			**MAC payload**		**MAC footer**

- Command Frame Types
  - Association request
  - Association response
  - Disassociation notification
  - Data request
  - PAN ID conflict notification
  - Orphan Notification
  - Beacon request
  - Coordinator realignment
  - GTS request

Data Frame Format

Octets:2	1	4 to 20	variable	2
Frame control	Data sequence number	Address information	Data payload	Frame check sequence
MAC header			**MAC Payload**	**MAC footer**

Acknowledgement Frame Format

Octets:2	1	2
Frame control	Data sequence number	Frame check sequence
MAC header		**MAC footer**

Data Service

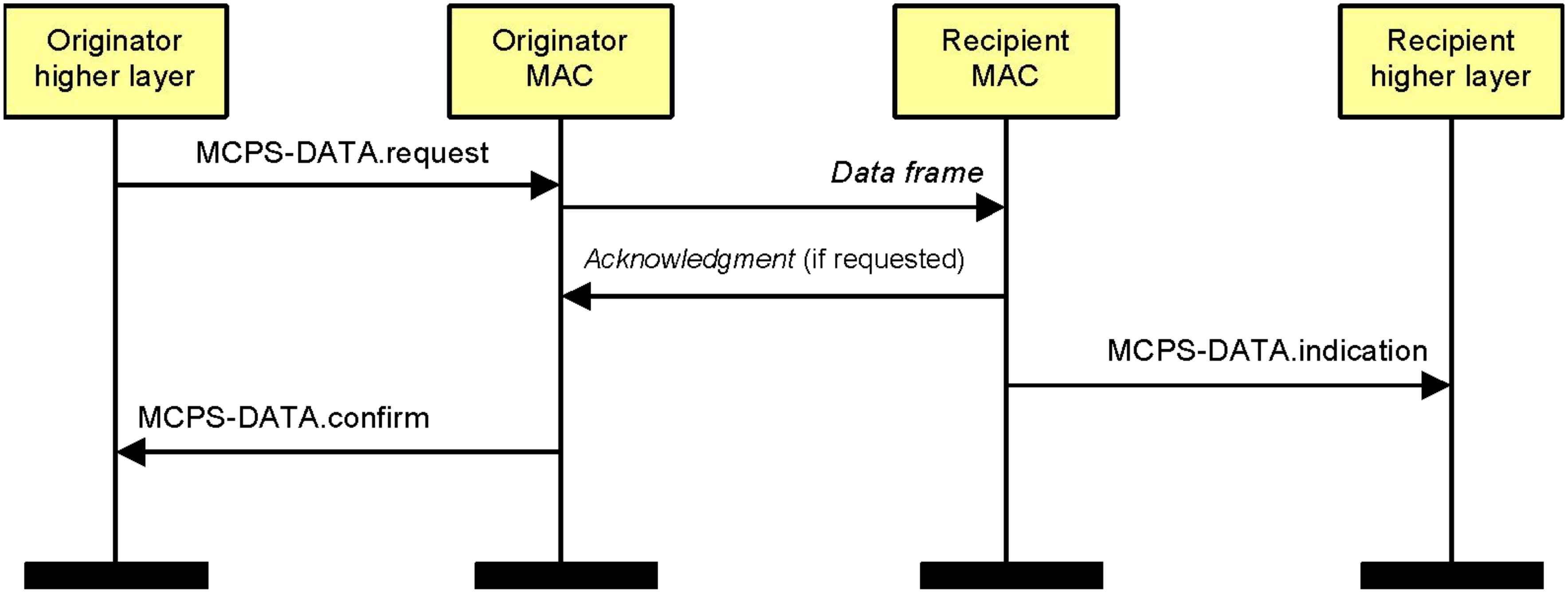
- Data transfer to neighboring devices
  - Acknowledged or unacknowledged
  - Direct or indirect
  - Using GTS service
- Maximum data length (MSDU) *aMaxMACFrameSize* (102 bytes)

MAC Data Primitives

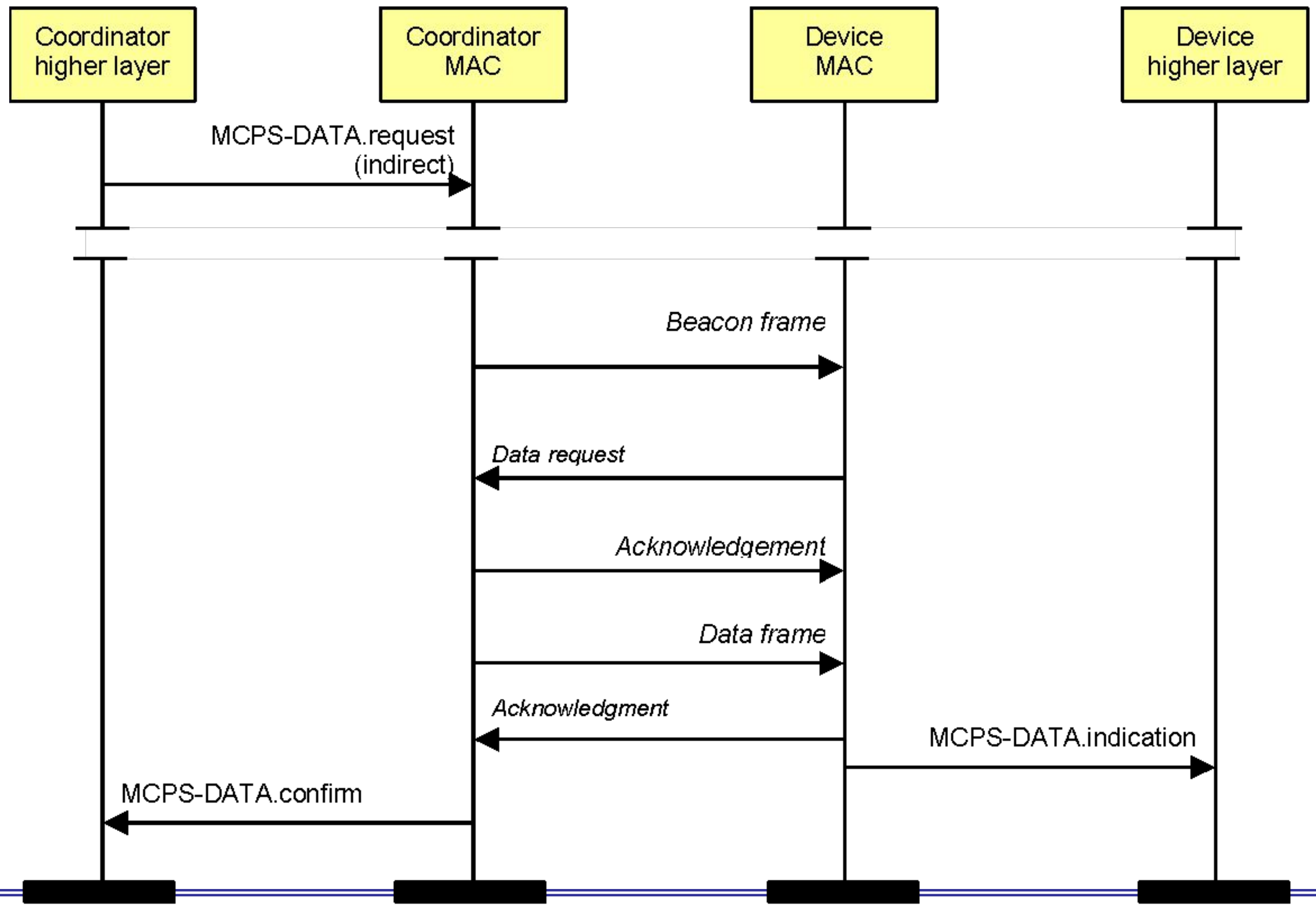
Primitive	Request	Confirm	Indication	Response
MCPS-DATA	Required	Required	Required	
MCPS-PURGE	Optional for RFD	Optional for RFD		

Data Transfer

Message Sequence Diagram



Indirect Data Transfer
Message Sequence Diagram



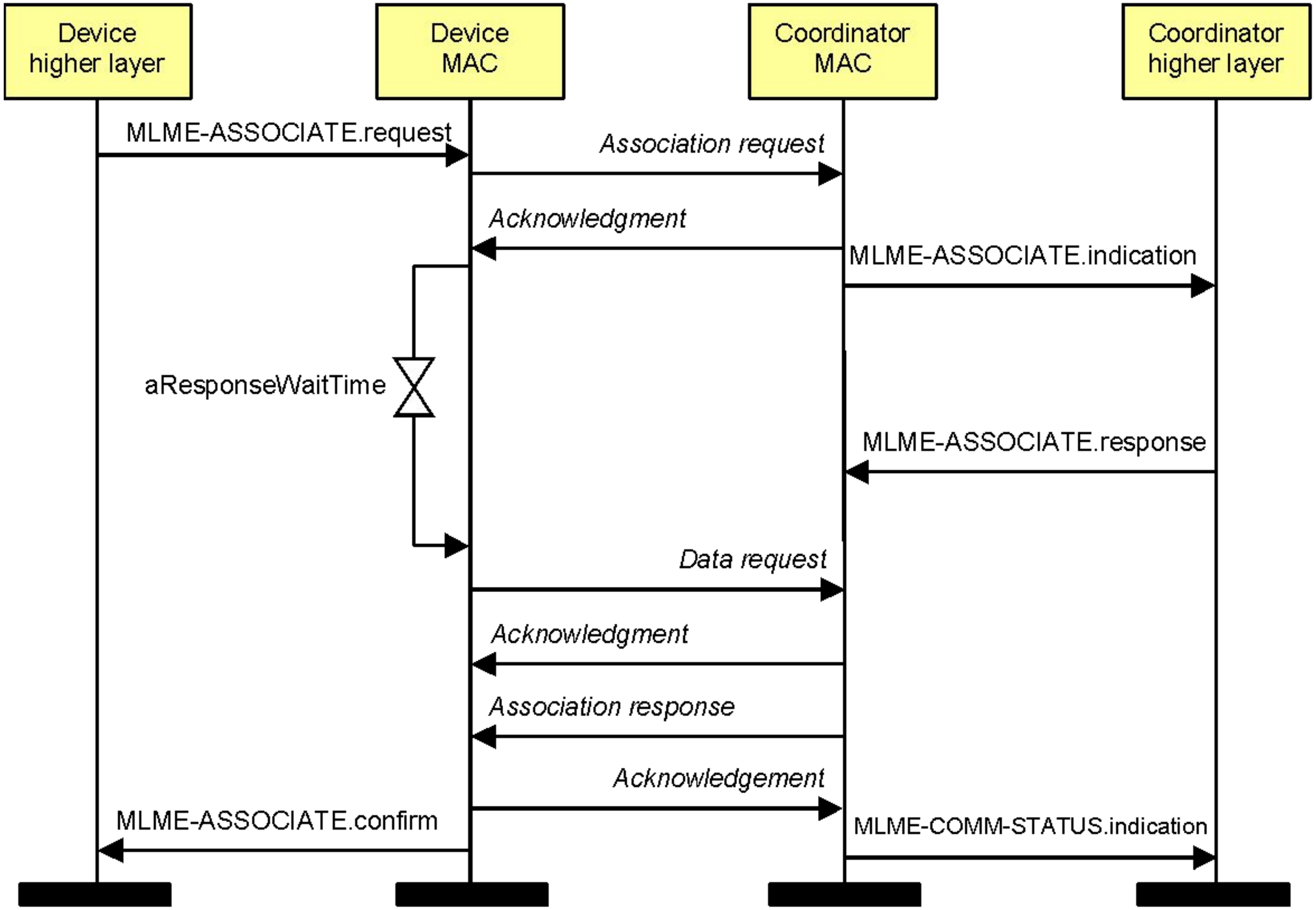
Management Service

- Association / disassociation
- GTS allocation
- Message pending
- Node notification
- Network scanning/start
- Network synchronization/search

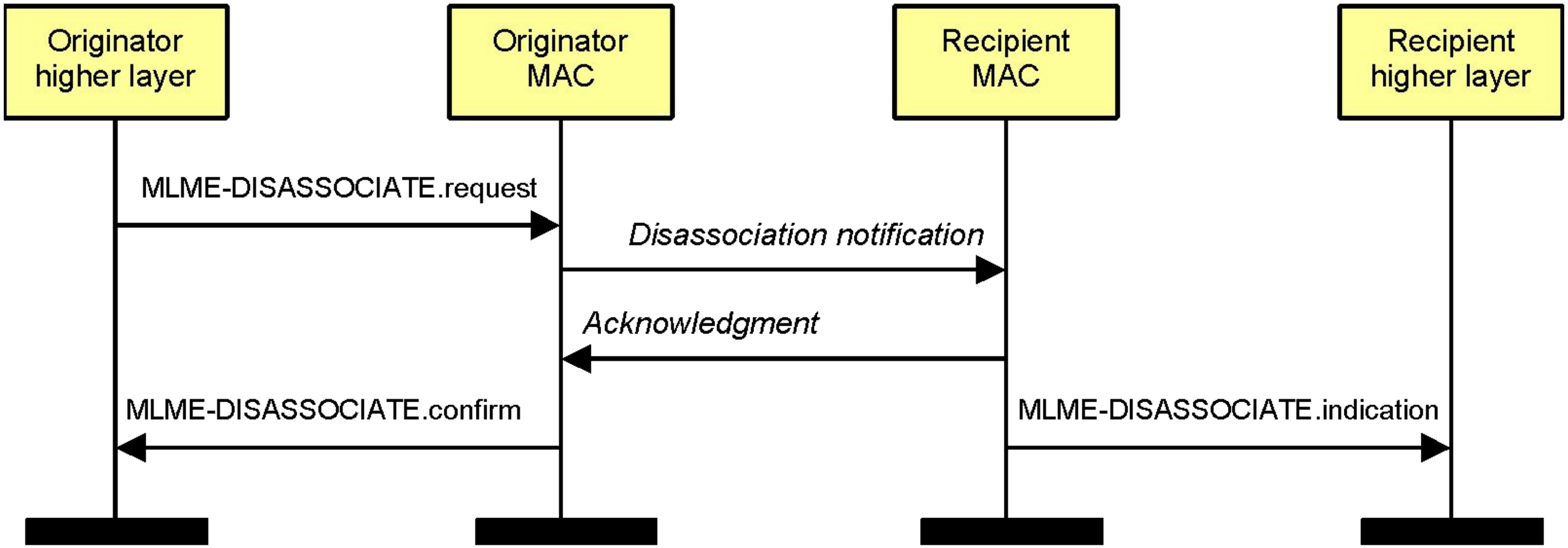
MAC Management Primitives

Primitive	Request	Confirm	Indication	Response
MLME-GET	Required	Required		
MLME-SET	Required	Required		
MLME-ASSOCIATE	Required	Required	Optional for RFD	Optional for RFD
MLME-DISASSOCIATE	Required	Required	Required	
MLME-GTS	Optional for RFD	Optional for RFD	Optional for RFD	
MLME-BEACON-NOTIFY			Required	
MLME-POLL	Required	Required		
MLME-COMM-STATUS			Required	
MLME-ORPHAN			Optional for RFD	Optional for RFD
MLME-SCAN	Required	Required		
MLME-START	Optional for RFD	Optional for RFD		
MLME-RX-ENABLE	Required	Required		
MLME-SYNC	Required			
MLME-SYNC-LOSS			Required	
MLME-RESET	Required	Required		

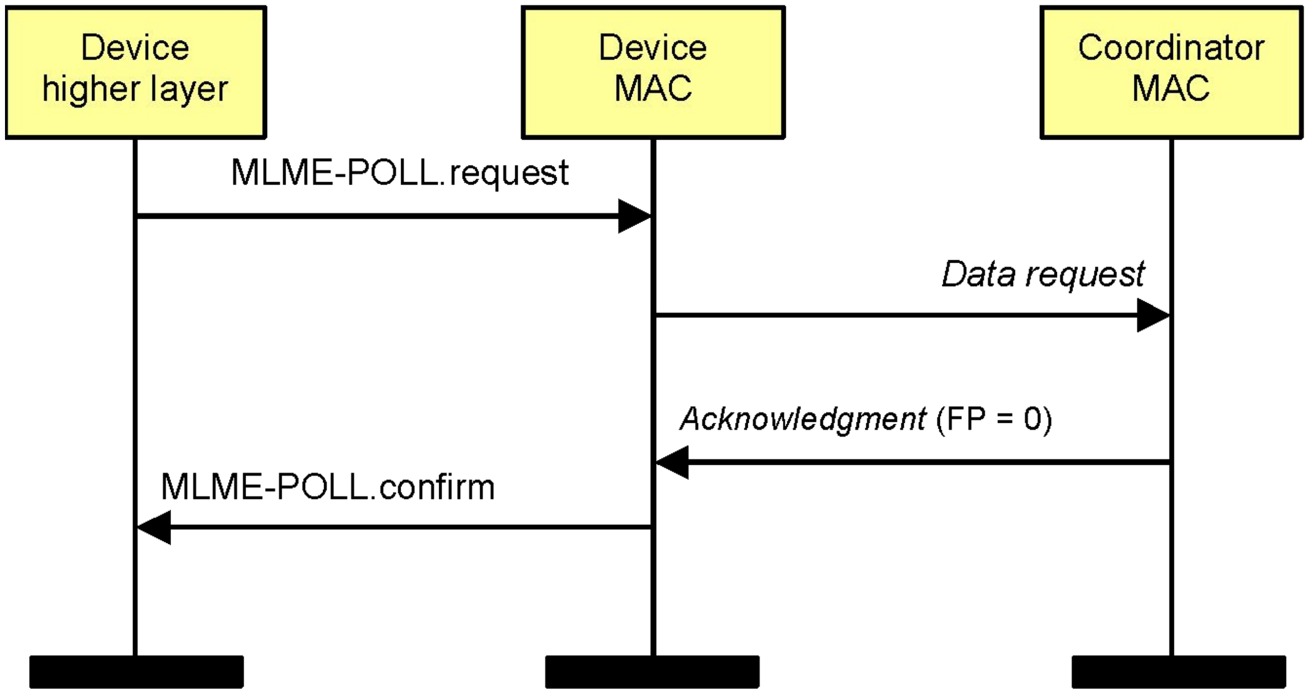
Association
Message Sequence Diagram



Disassociation
Message Sequence Diagram

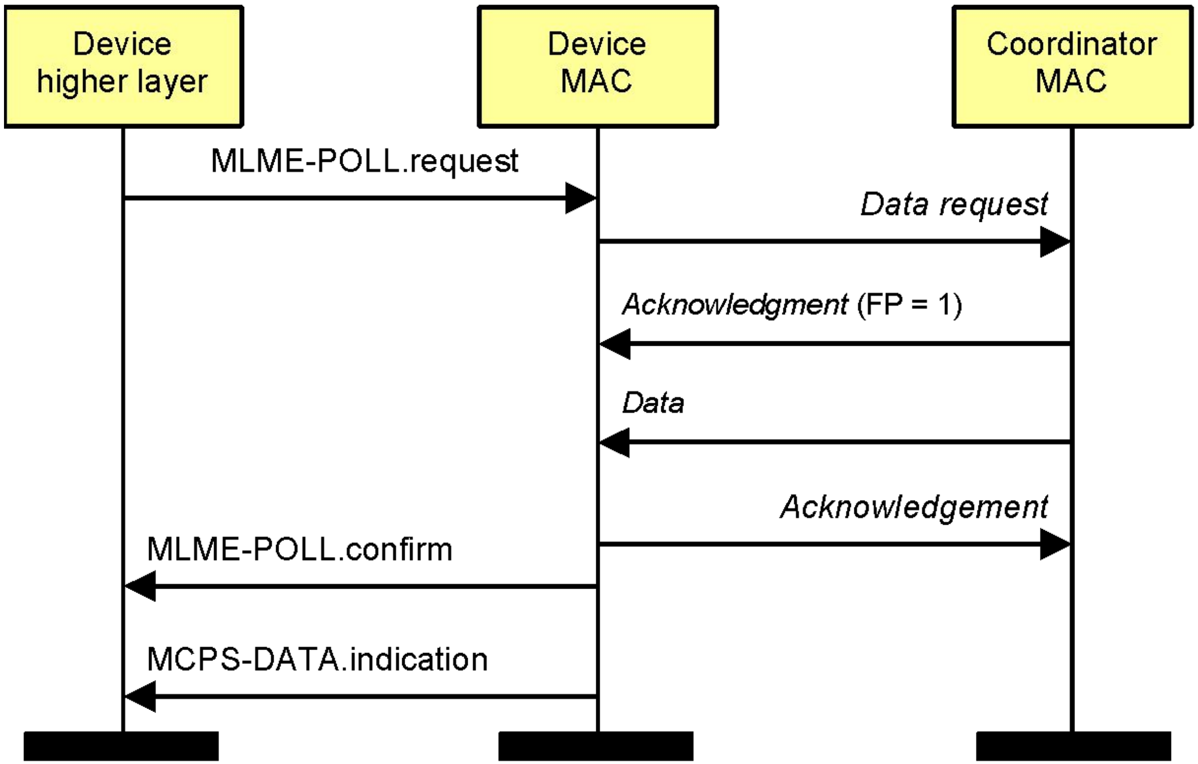


Data Polling
Message Sequence Chart



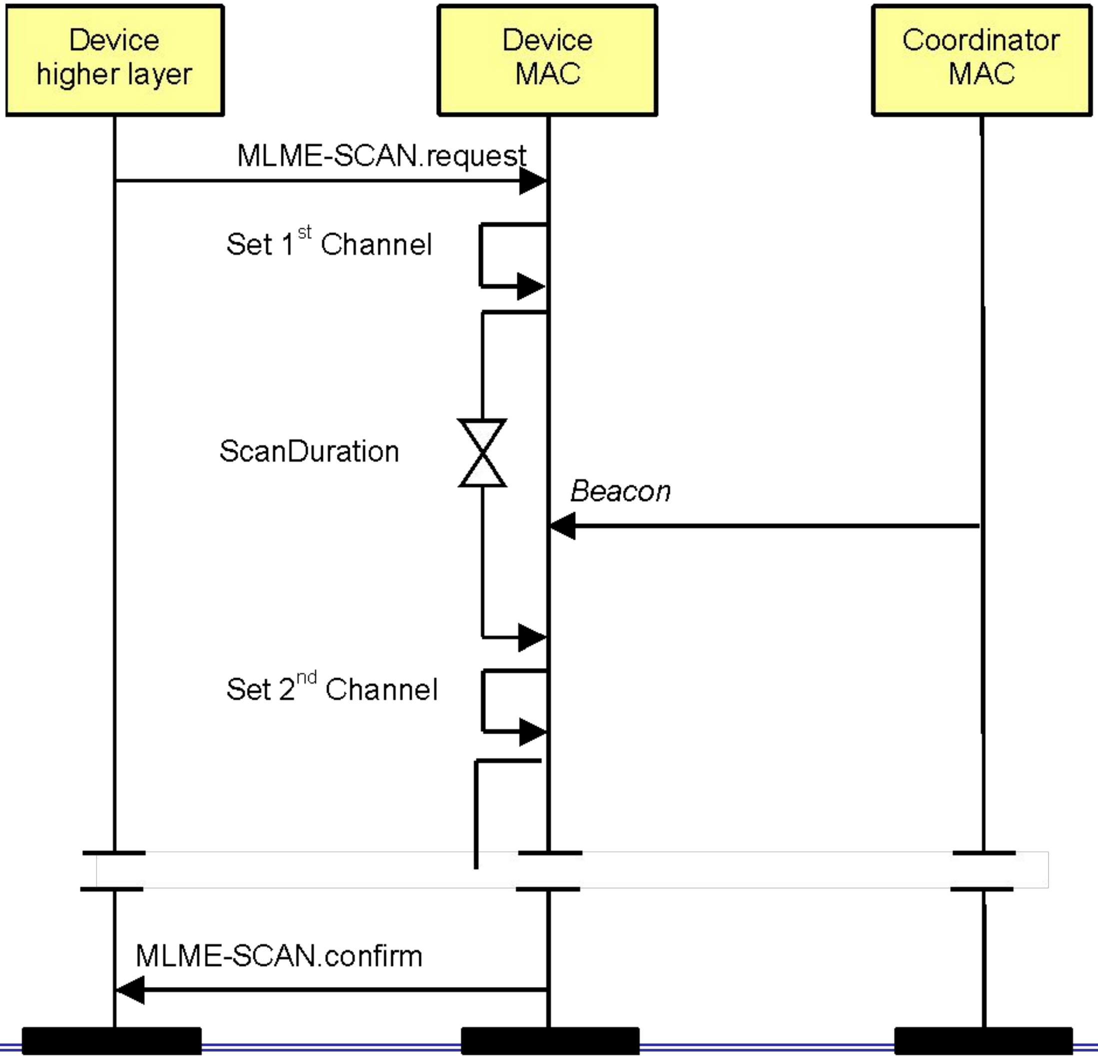
No data pending at the coordinator

Data Polling
Message Sequence Chart

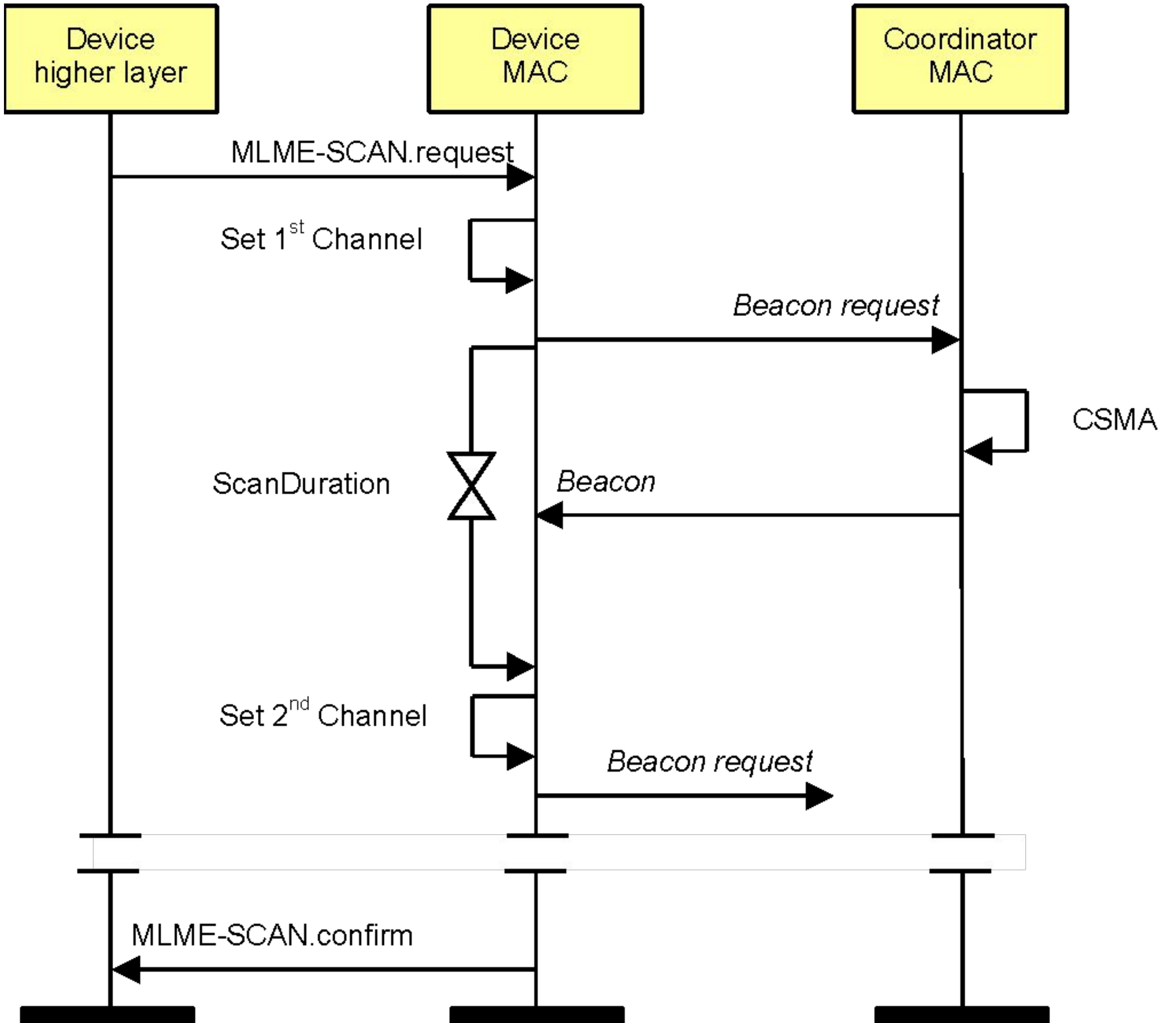


Data pending at the coordinator

Passive Scan

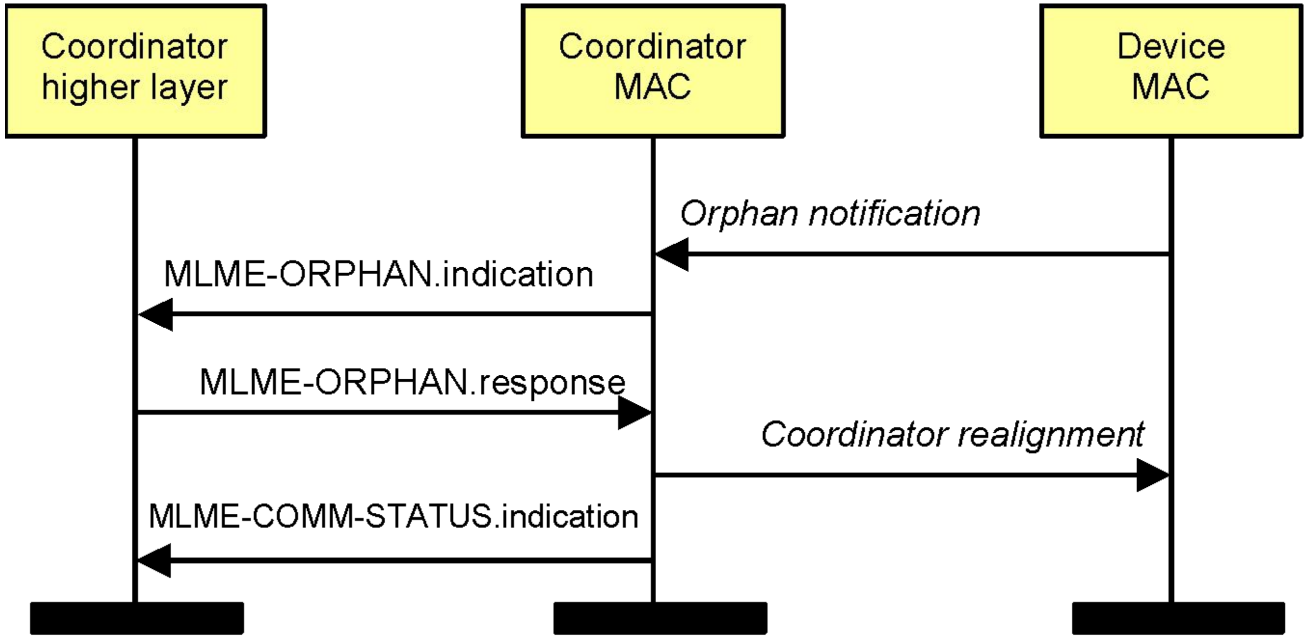


Active Scan

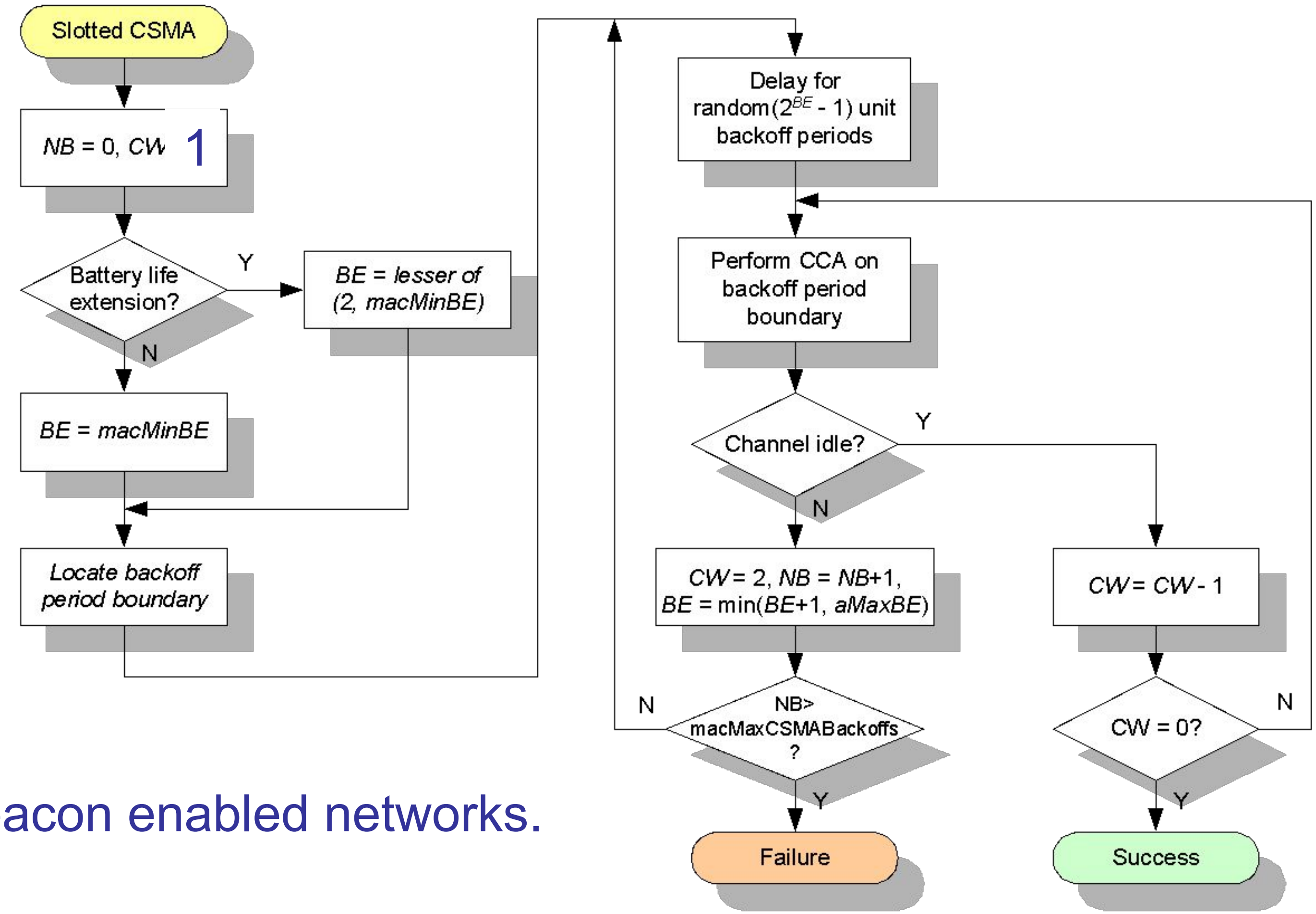


Device higher layer
Device MAC
Coordinator MAC
MLME-SCAN.request
Set 1st Channel
Beacon request
CSMA
ScanDuration
Beacon
Set 2nd Channel
Beacon request
MLME-SCAN.confirm

Orphaning
Message Sequence Diagram



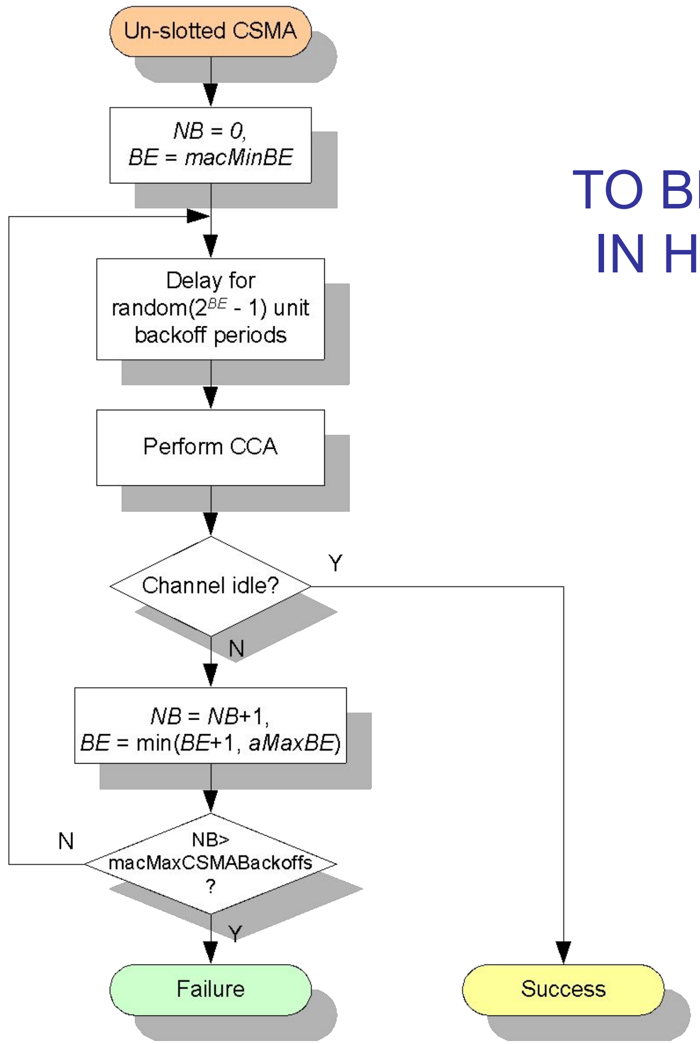
Slotted CSMA Procedure



Used in beacon enabled networks.

Un-slotted CSMA Procedure

Used in non-beacon networks.



TO BE EVALUATED IN HOMEWORK 1 !!!!