MLA: MAC Layer Architecture

Octav Chipara

Original slides by Chenyang Lu

Challenges

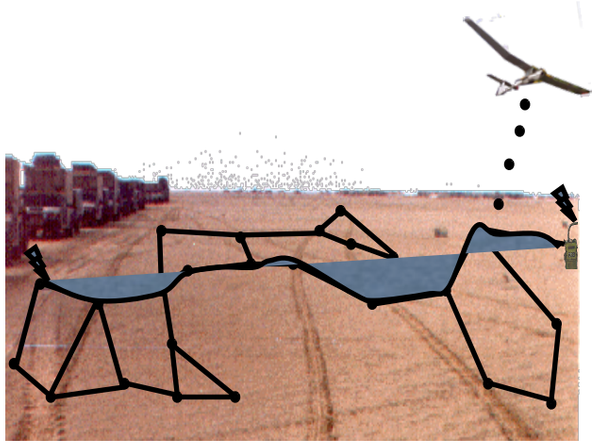
- **Power management is critical for wireless sensor networks**
  - Limited energy source
  - Lifetime from months to years
- **Gap between protocols and systems**
  - Significant advance in power management protocols
  - Significant challenges to integrate them in real systems
  - Minimum support for power management in OS

- **Need unified architectures for flexible power management!**

Diversity of MAC Protocols

- **Conflicting application requirements**
  - Energy
  - Latency
  - Throughput
- **Radio is a major consumer of energy**
- **Need different MACs to meet different requirements**

Habitat Monitoring



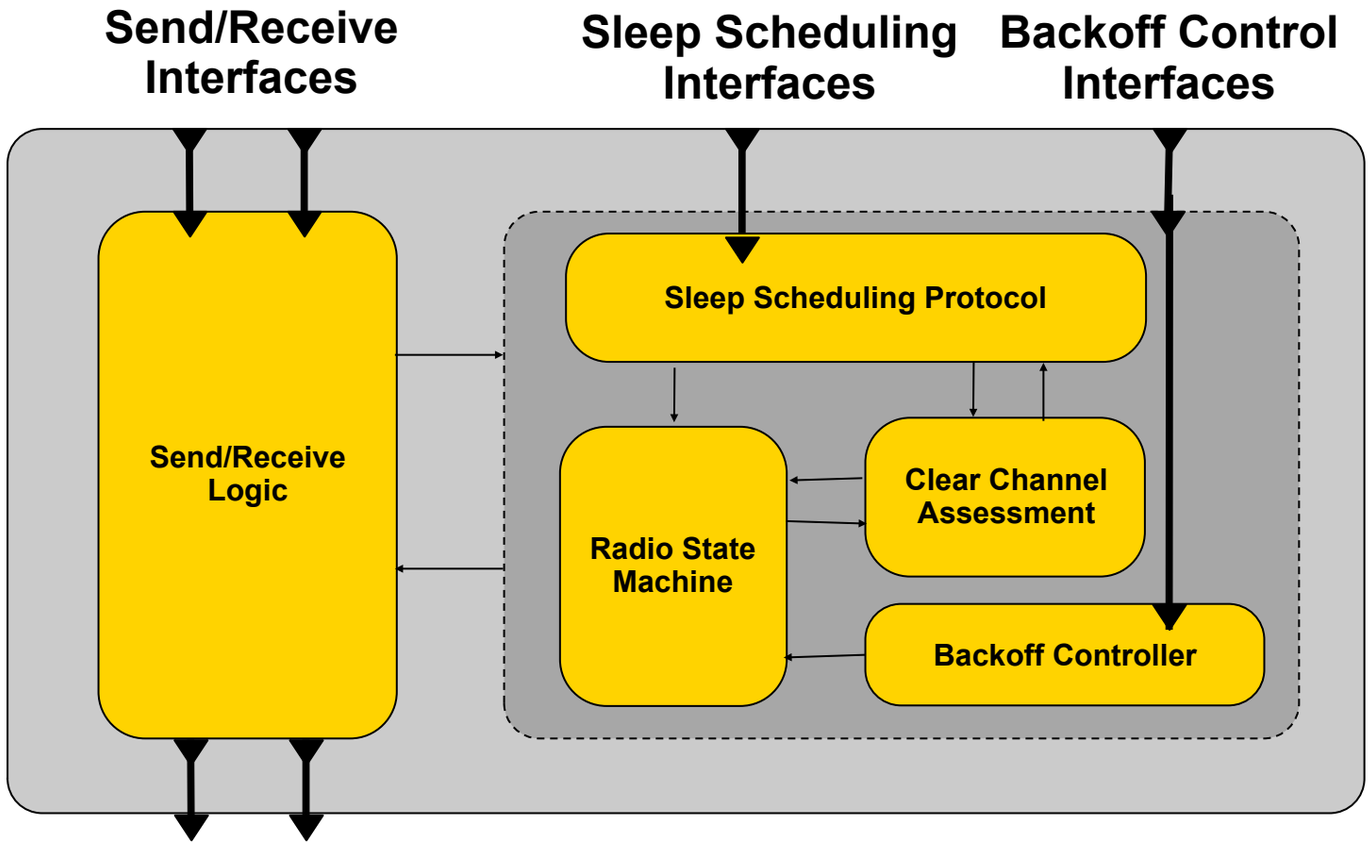
Tracking

Structural Health

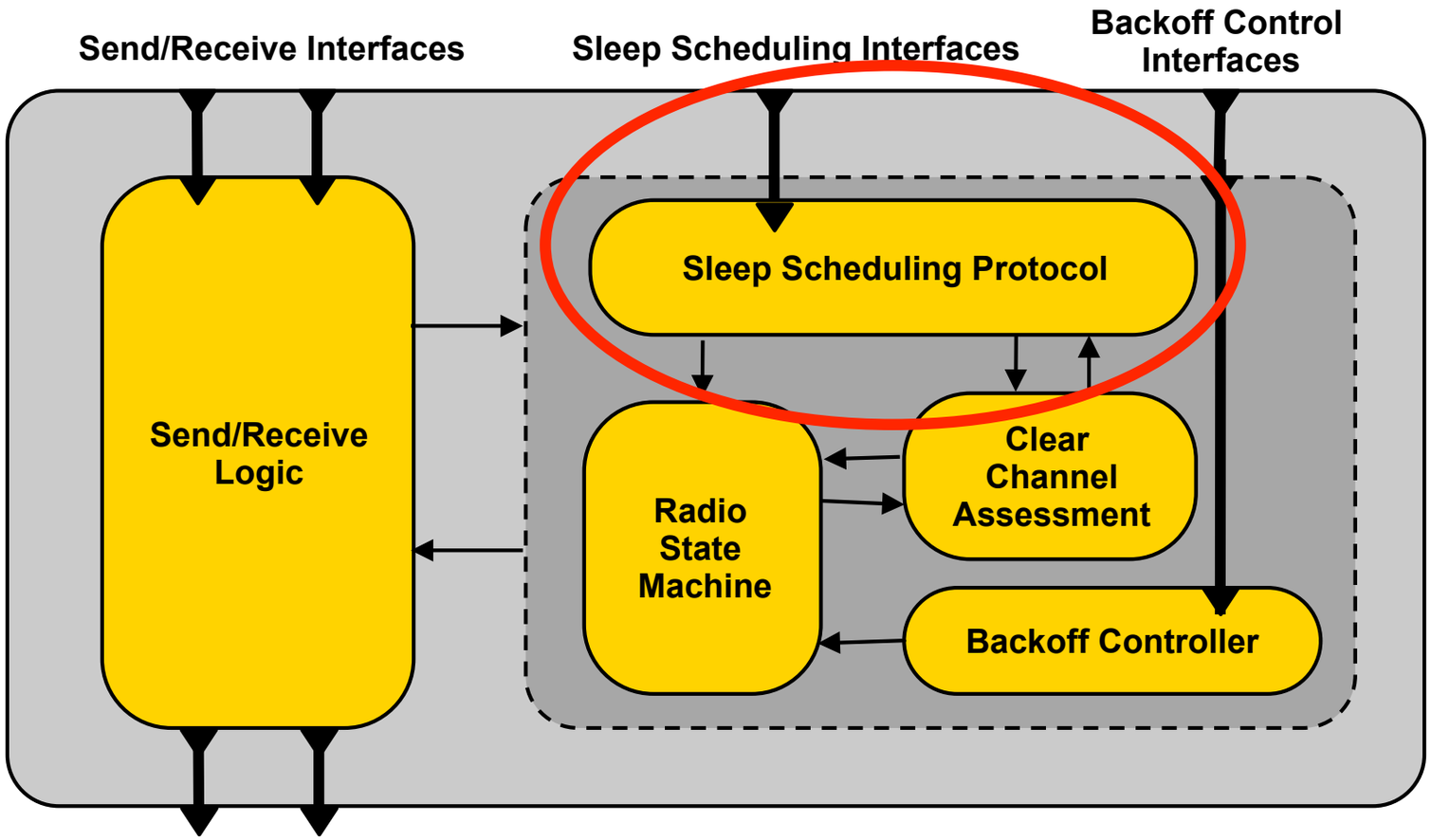
Health Care

Current Solution

- **Design a new MAC protocol as a monolithic stack**
  - S-MAC
  - BMAC
  - ZMAC
  - XMAC
  - WiseMAC
  - T-MAC
  - SCP
  - Funnel-MAC
  - Crankshaft
  - 802.15.4
  - DRAND
  - ……

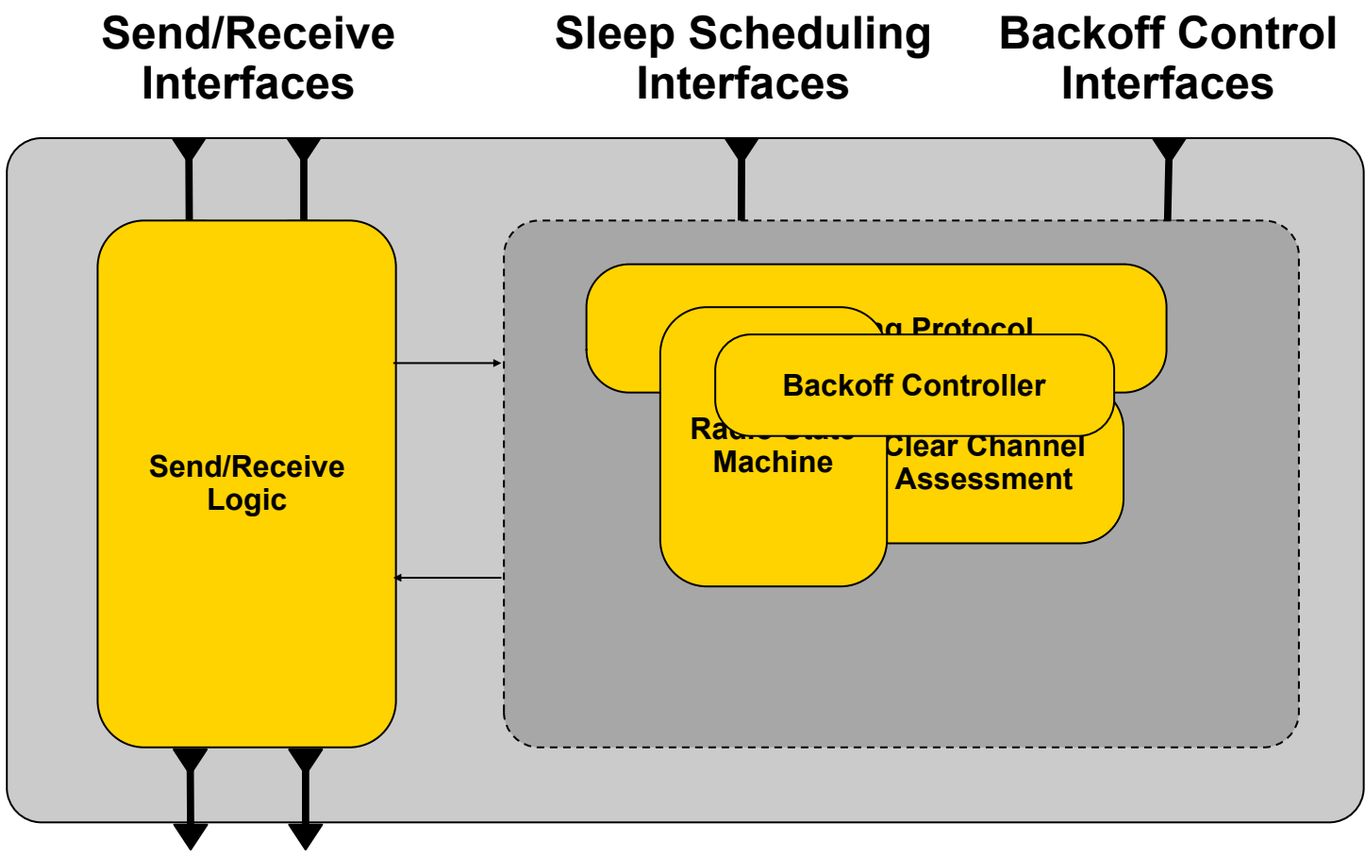


Problem with Current Solution



No separation between power management & core radio functionality

Problem with Current Solution



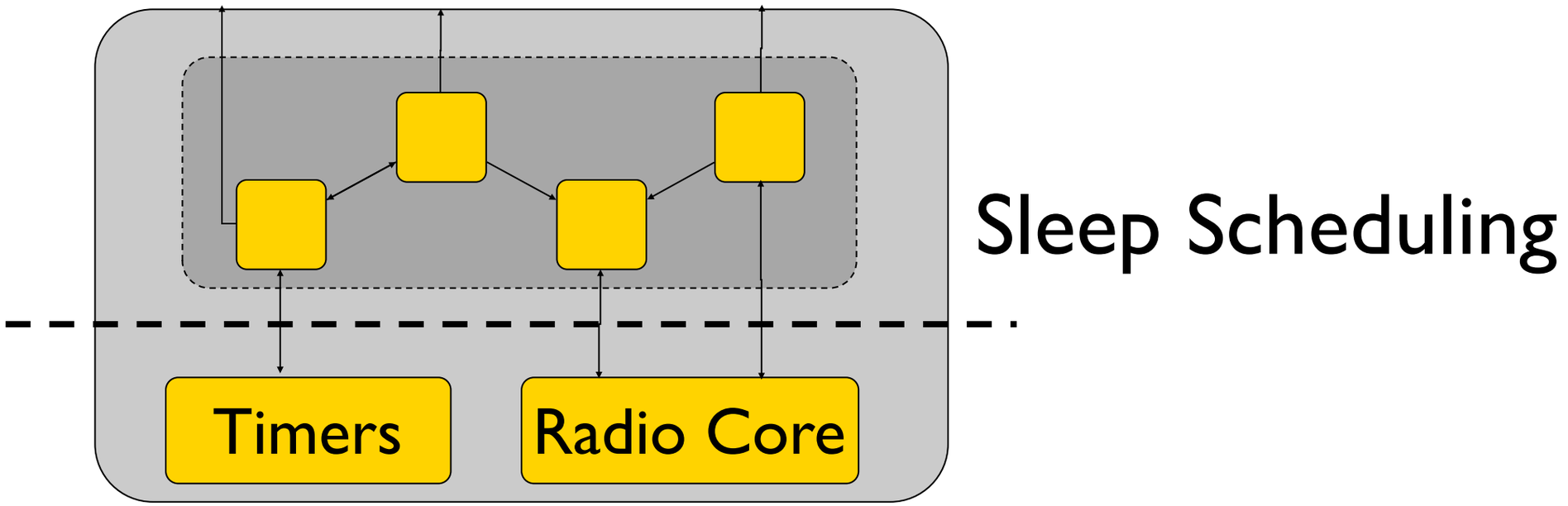
All features jumbled into one big monolithic implementation

Problem: Monolithic Radio Stack

- **Hard to develop new MAC protocols**
  - No clear separation of concerns
  - Need intimate knowledge of entire stack
- **Hard to maintain multiple MAC stacks as OS evolves**
- **Protocols not reusable across radio/processor platforms**

MLA: MAC Layer Architecture

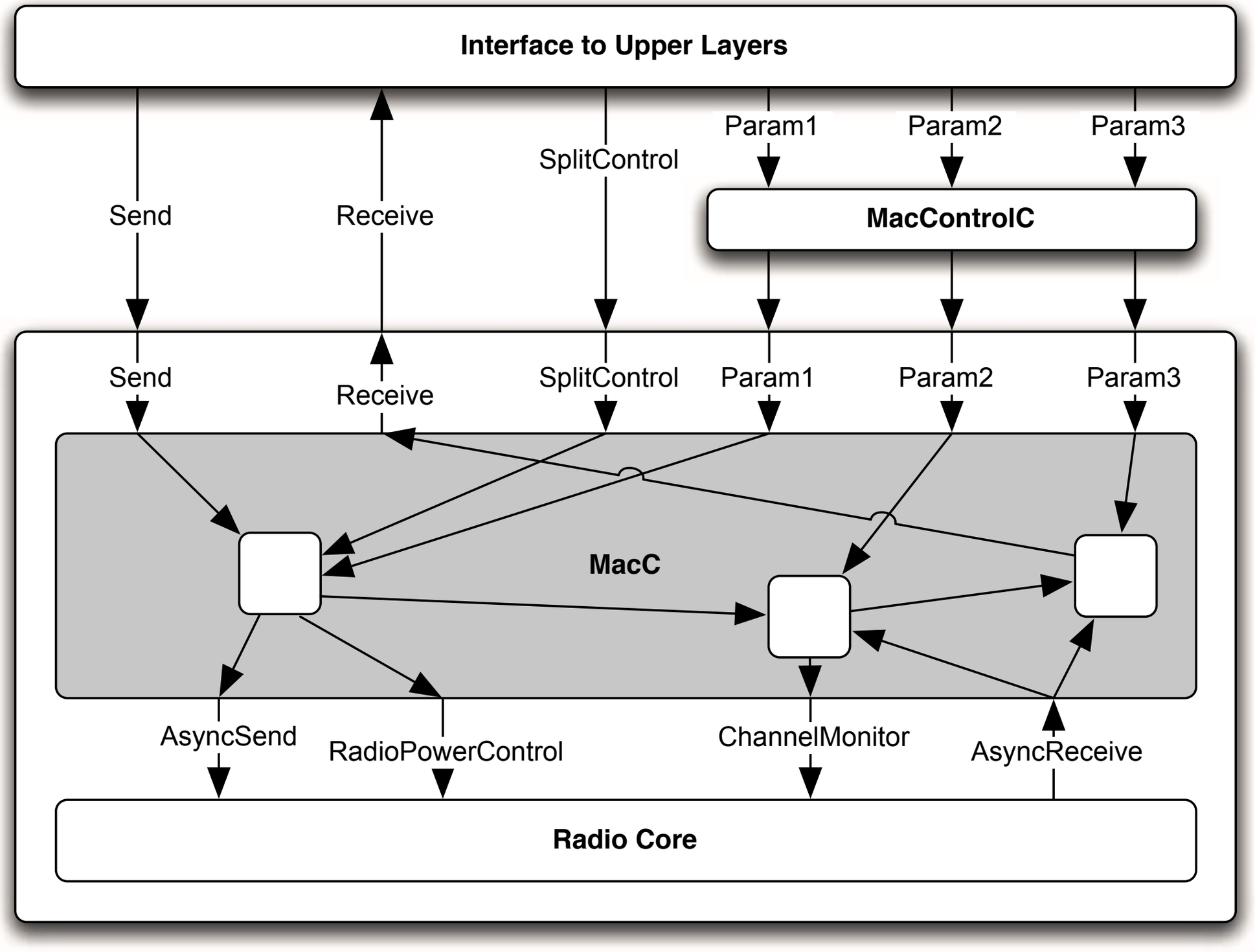
- **Separation of sleep sleeping from radio core**
- **Components for sleep scheduling protocols**
  - Reusable ➜ ease development & maintenance of protocols
  - Platform independent ➜reduce porting effort



MLA: MAC Layer Architecture

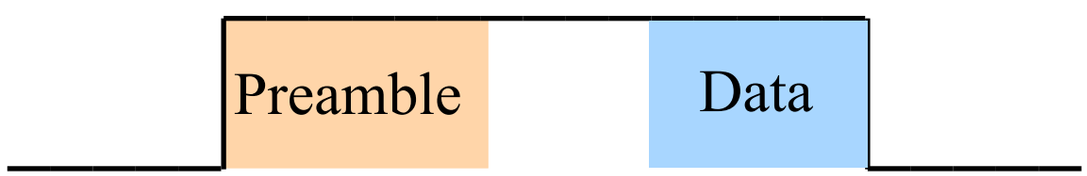
- **Components implement common features of MAC protocols**
  - Hardware-independent: portable across platforms
  - Hardware-dependent: portable interfaces, platform specific implementations
- **Simplifies porting to a new platform**
  - Re-implement hardware-dependent components
    - Once per platform
    - Hardware independent components stay the same
- **Support diverse MAC protocols**
  - CSMA (contention-based), TDMA (scheduling-based), Hybrid
- **Comparable efficiency to monolithic implementations**

MLA architecture

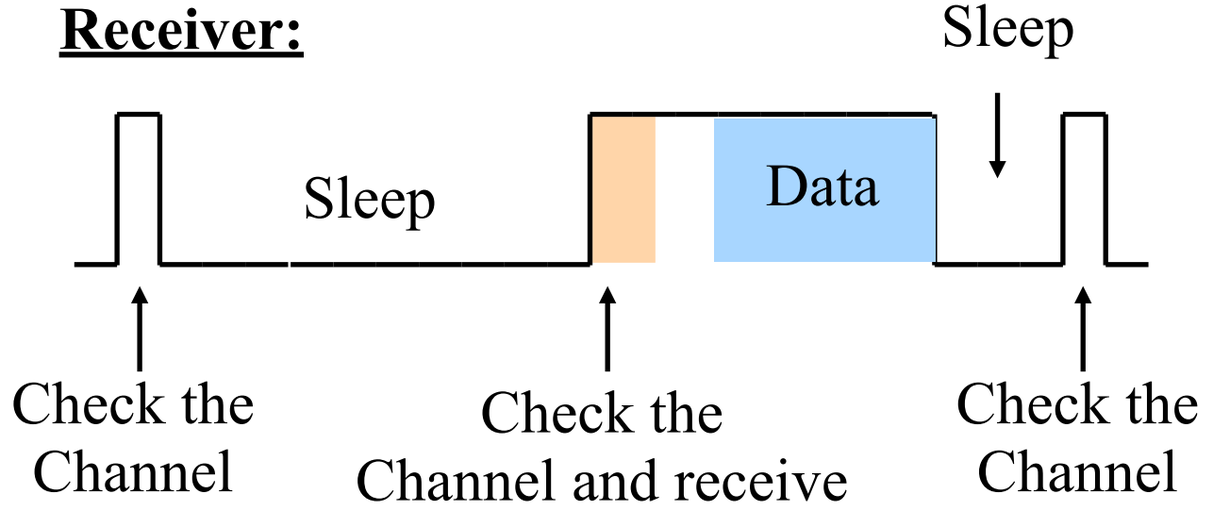


B-MAC: An Example Protocol

Sender:



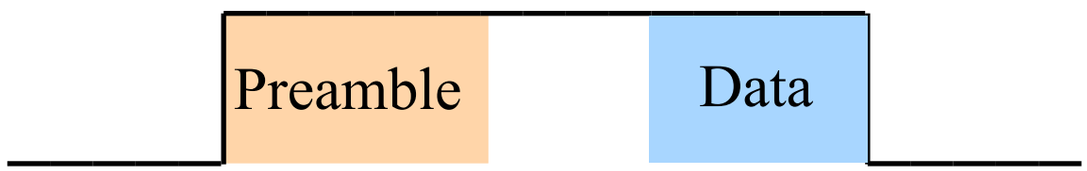
Receiver:



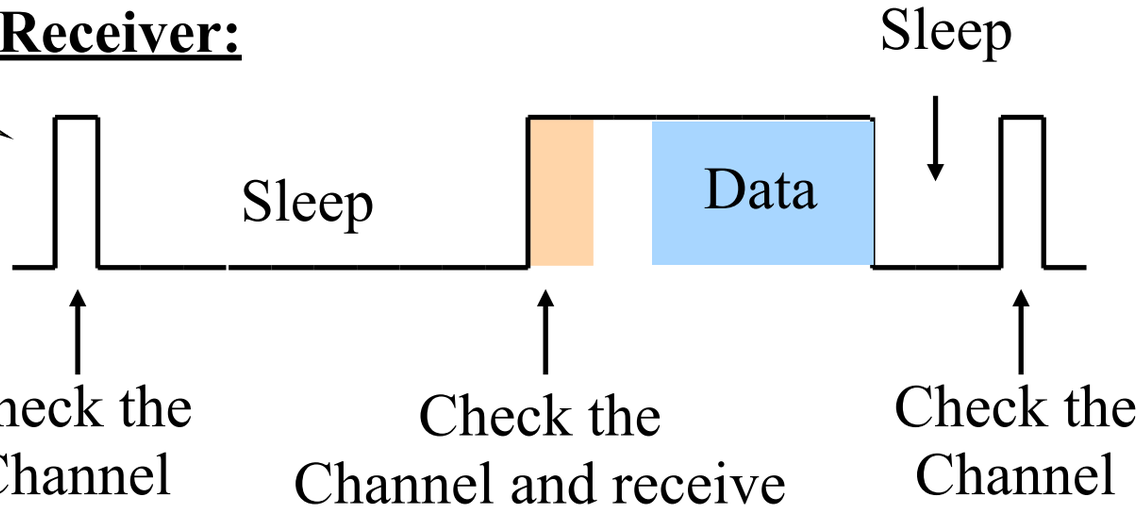
B-MAC

Receiver performs periodic CCA check

Sender:



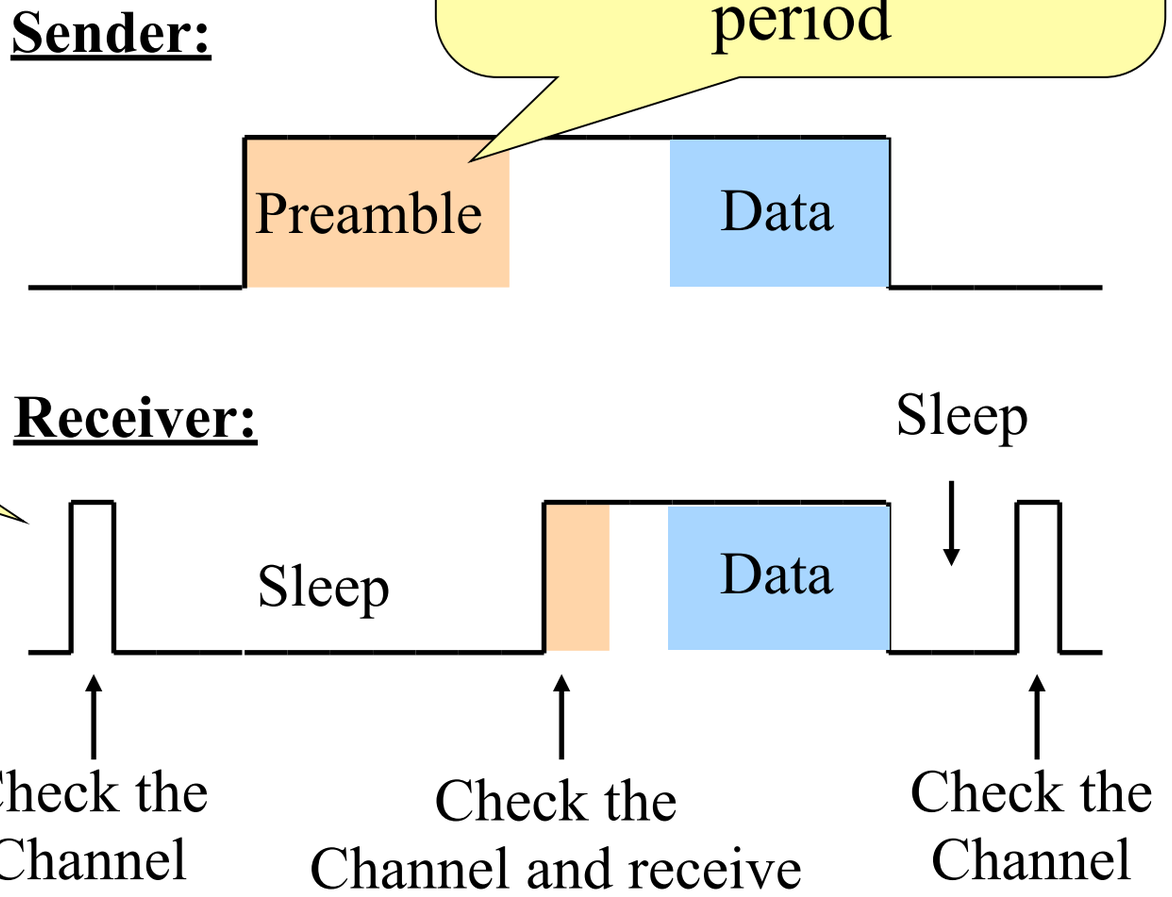
Receiver:



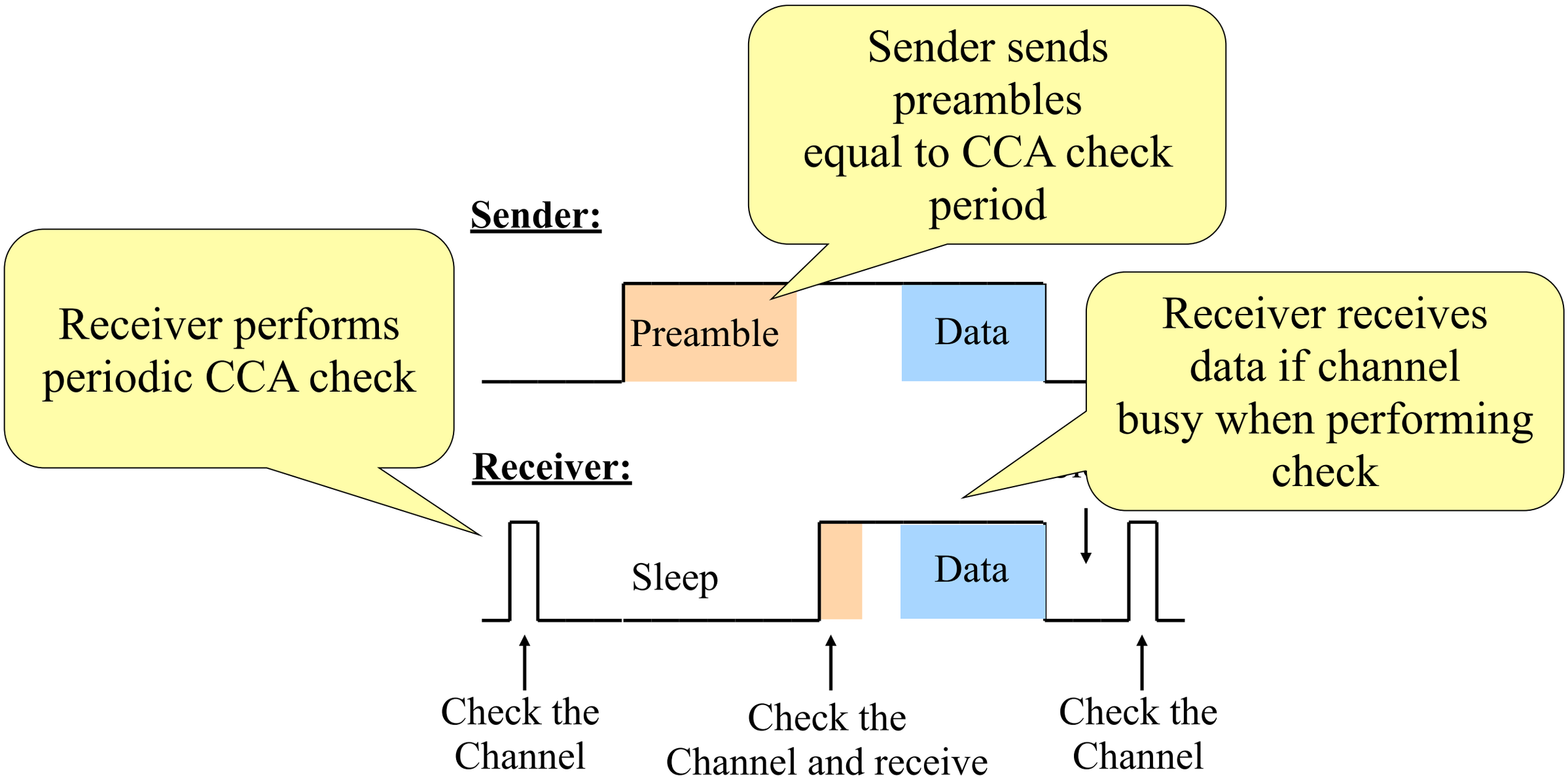
B-MAC

Receiver performs periodic CCA check

Sender sends preambles equal to CCA check period

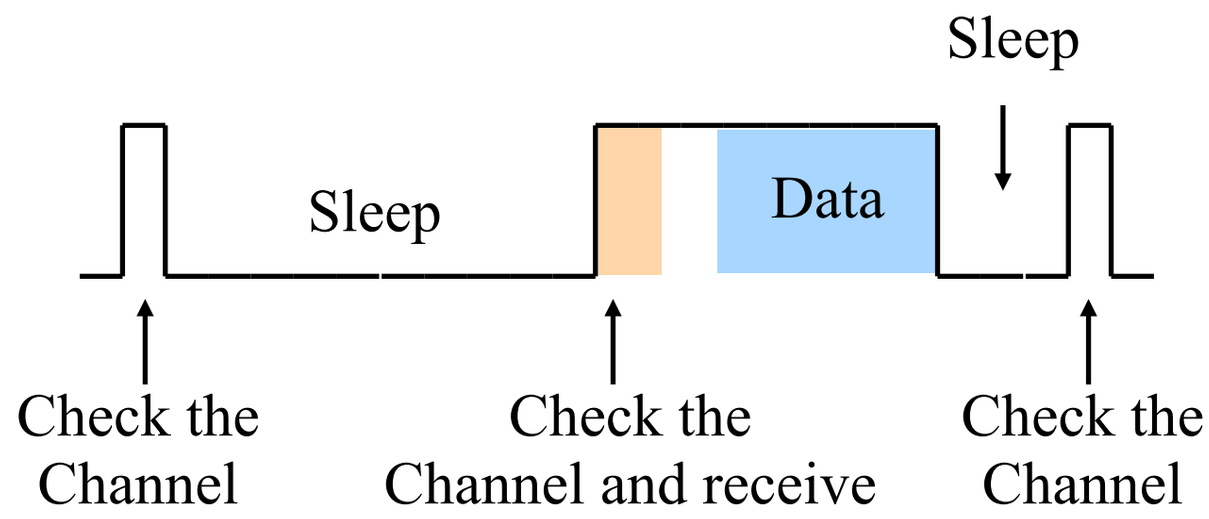


B-MAC


Sender sends preambles equal to CCA check period
Receiver performs periodic CCA check
Sender:
Preamble
Data
Receiver receives data if channel busy when performing check
Receiver:
Sleep
Data
Check the Channel
Check the Channel and receive
Check the Channel

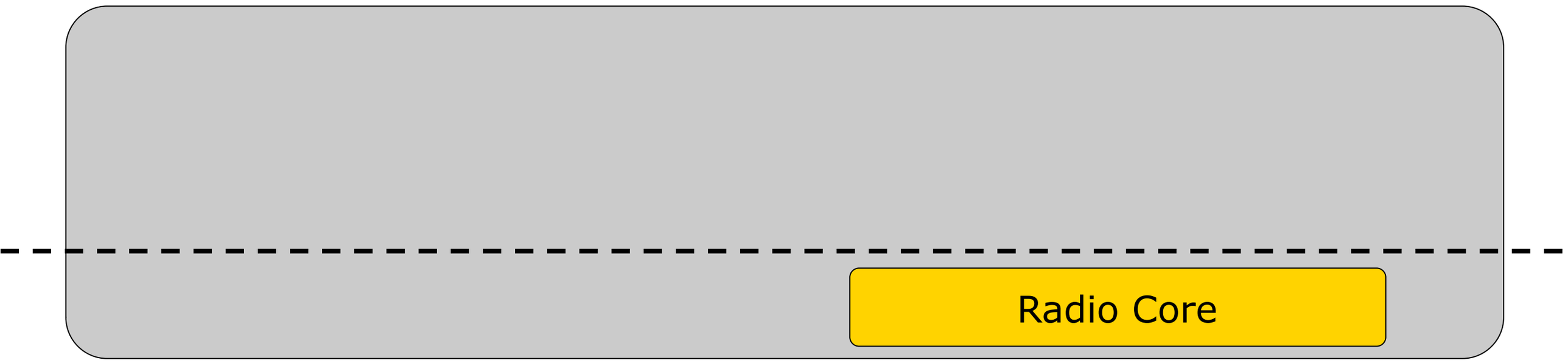
B-MAC: What Does It Need?

- Method of turning the radio on and off
- Method of checking the channel for radio activity (CCA)
- Periodic Timer to listen for radio activity
- A way of sending / receiving preambles
- A way of sending / receiving data



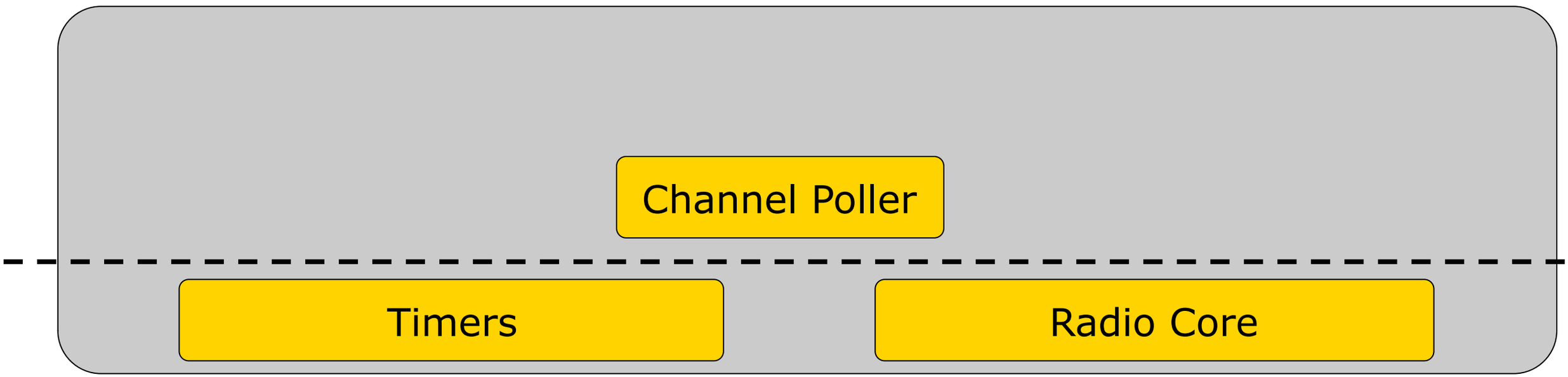
Breakdown of B-MAC

- **What does it need?**
  - Method of turning the radio on and off
  - Method of checking the channel for radio activity (CCA)



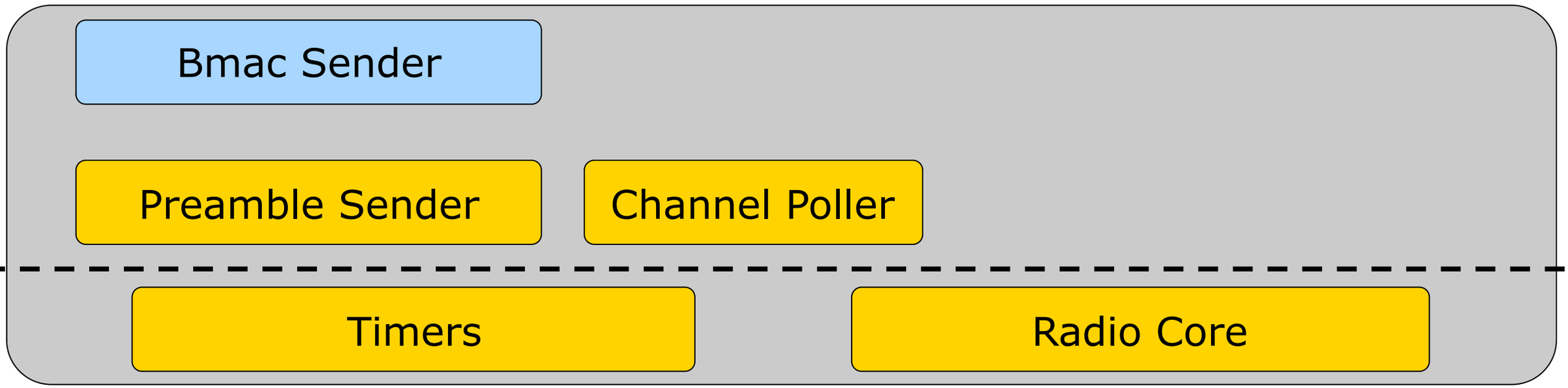
Breakdown of B-MAC

- **What does it need?**
  - Method of turning the radio on and off
  - Method of checking the channel for radio activity (CCA)
  - Periodic Timer to listen for radio activity



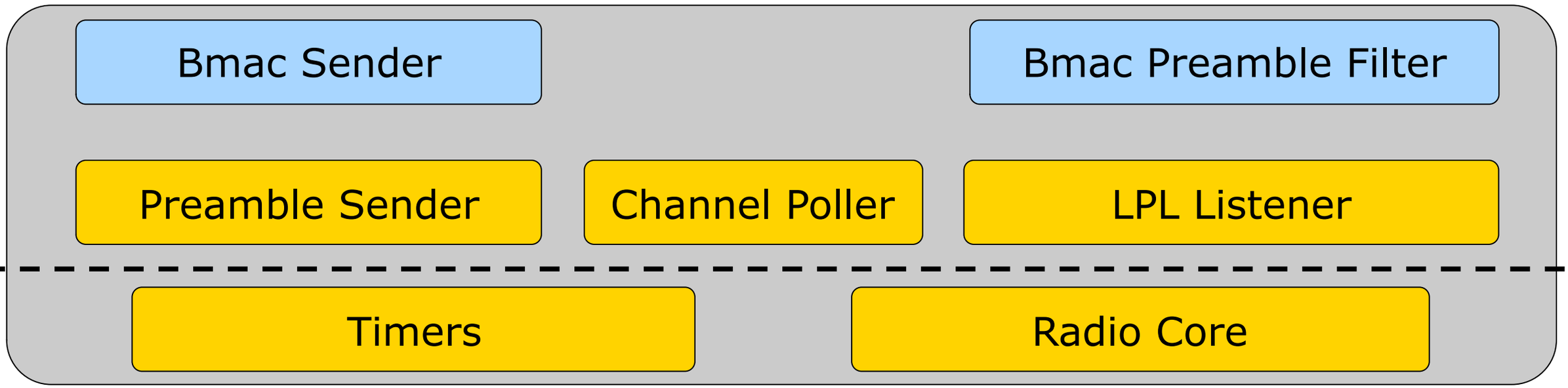
Breakdown of B-MAC

- **What does it need?**
  - Method of turning the radio on and off
  - Method of checking the channel for radio activity (CCA)
  - Periodic Timer to listen for radio activity
  - A way of sending preambles and data

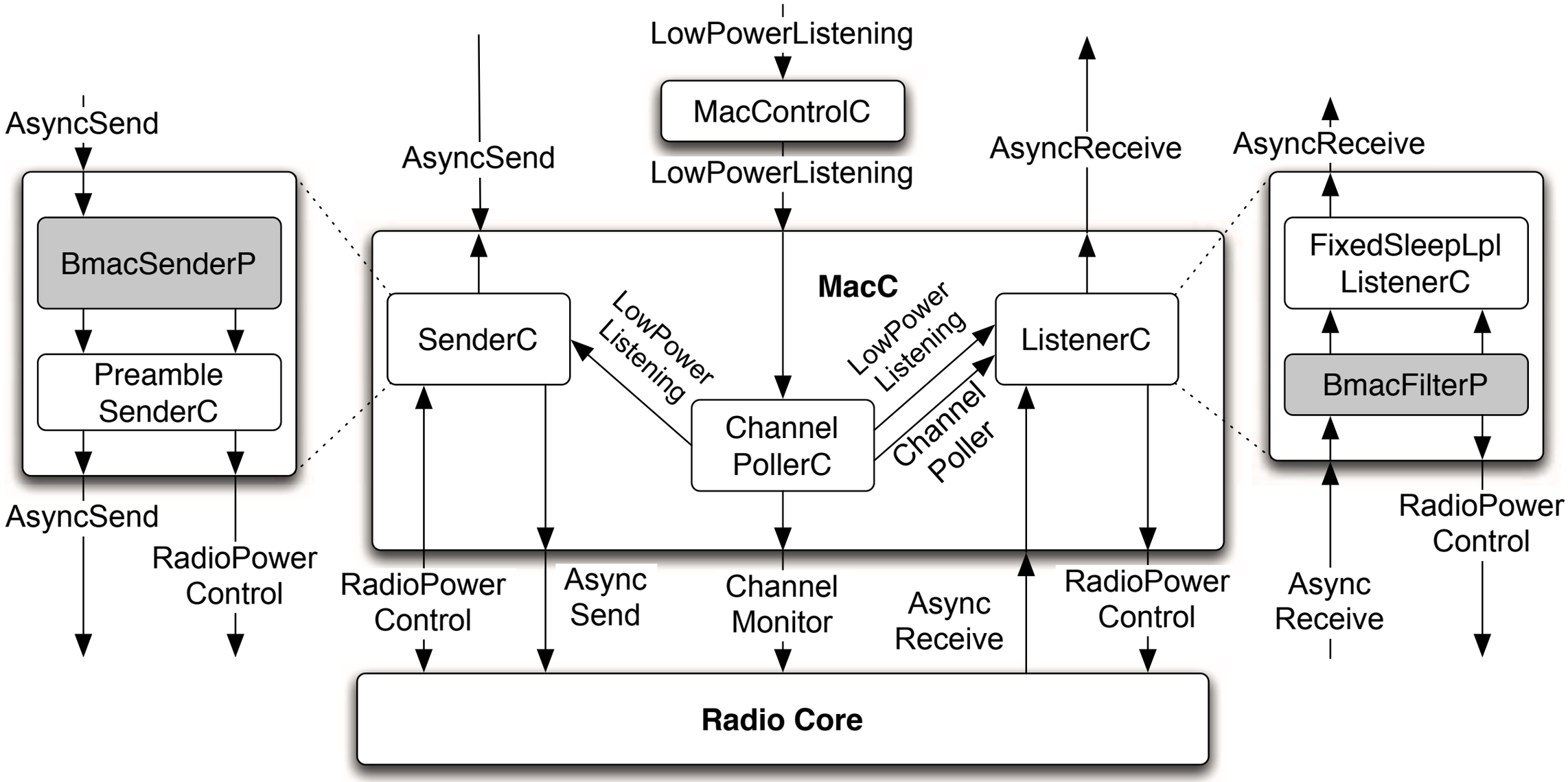


Breakdown of B-MAC

- **What does it need?**
  - Method of turning the radio on and off
  - Method of checking the channel for radio activity (CCA)
  - Periodic Timer to listen for radio activity
  - A way of sending preambles and data
  - A way of receiving data and filtering out preambles

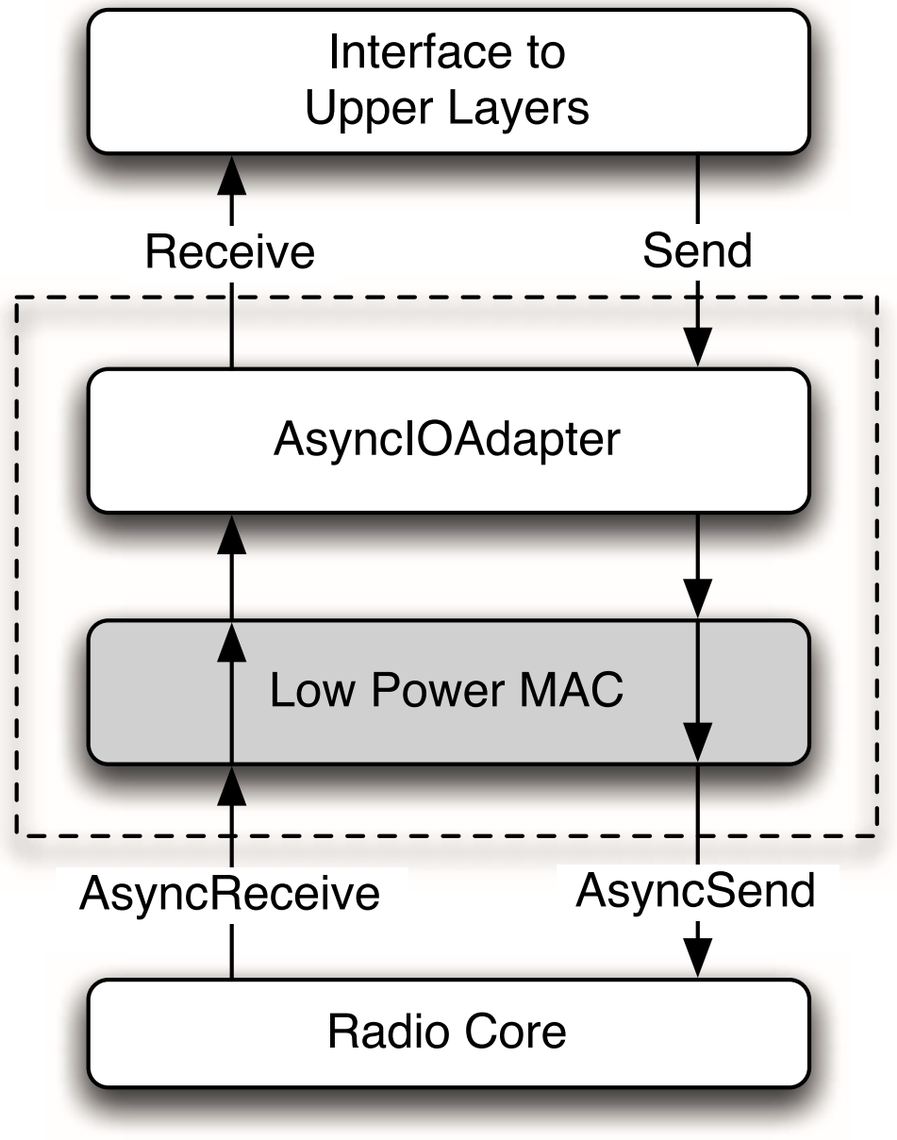


BMAC - details



Low-latency IO

- **Low-latency is essential for TDMA and contention-based protocols**
  - expose the async receive and sends from the radio layer
    - provides hooks for low-latency operation
  - the usual warnings about asynchronous context still apply



Component Library

CSMA Protocols

Hardware Independent	Hardware Dependent
Preamble Sender	Radio Core
LPL Listener	Local Time
Channel Poller	Alarm
Slot Handlers (TDMA/CSMA)	
Time Synchronization	
Low Level Dispatcher	
Asynchronous I/O Adapter	

Component Library

TDMA Protocols

Hardware Independent	Hardware Dependent
Preamble Sender	Radio Core
LPL Listener	Local Time
Channel Poller	Alarm
Slot Handlers (TDMA/CSMA)	
Time Synchronization	
Low Level Dispatcher	
Asynchronous I/O Adapter	

Component Library

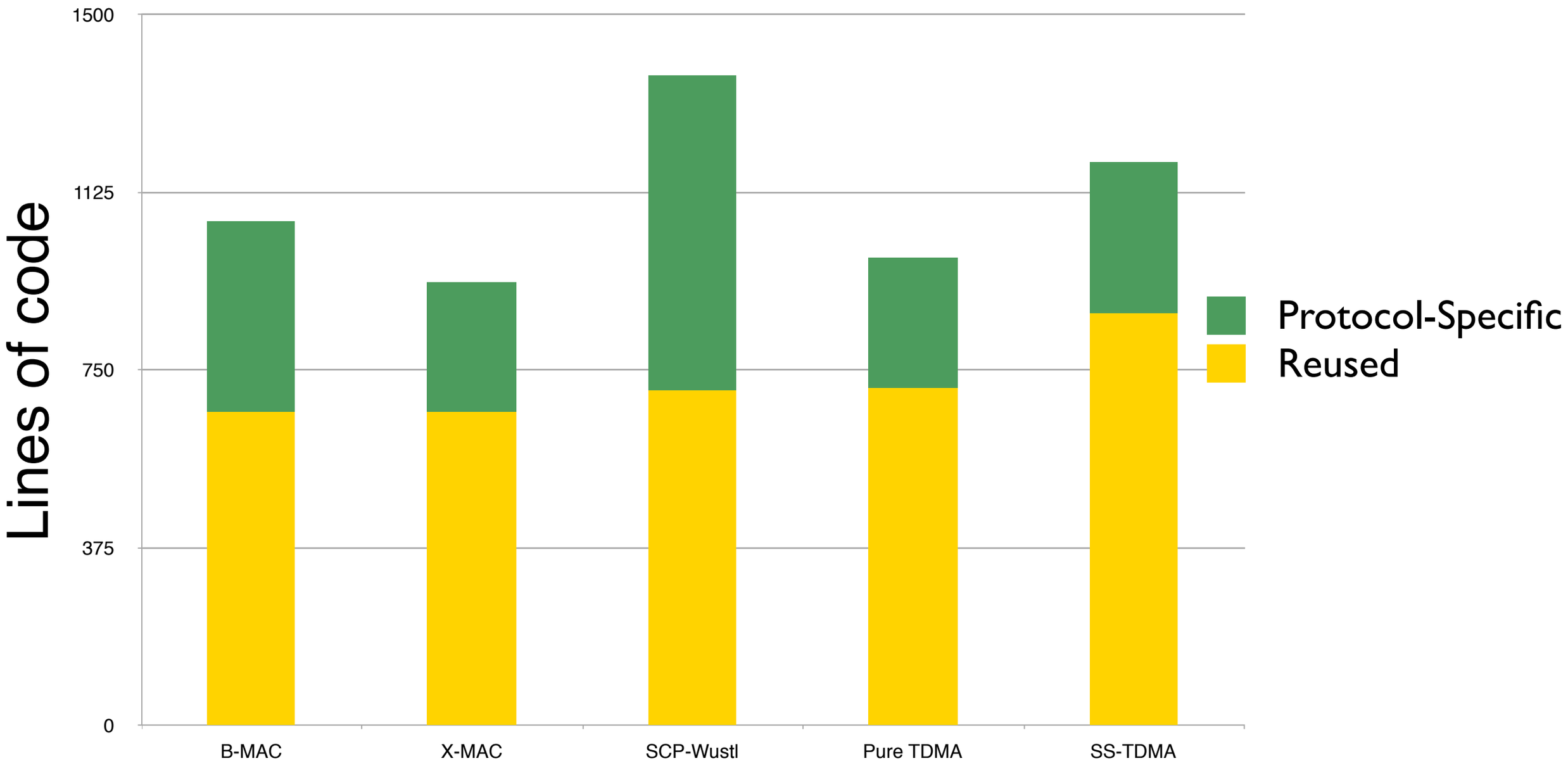
Hybrid Protocols

Hardware Independent	Hardware Dependent
Preamble Sender	Radio Core
LPL Listener	Local Time
Channel Poller	Alarm
Slot Handlers (TDMA/CSMA)	
Time Synchronization	
Low Level Dispatcher	
Asynchronous I/O Adapter	

Evaluation

- **All evaluations performed on TelosB motes in TinyOS 2.0.1**
- **Implemented 5 MAC protocols**
  - B-MAC, X-MAC, SCP-Wustl, Pure TDMA, SS-TDMA
- **Measure**
  - Reusability of components among protocols
  - Memory footprint compared to monolithic implementations
  - Throughput
  - Latency
  - Energy Consumption

Code Reuse

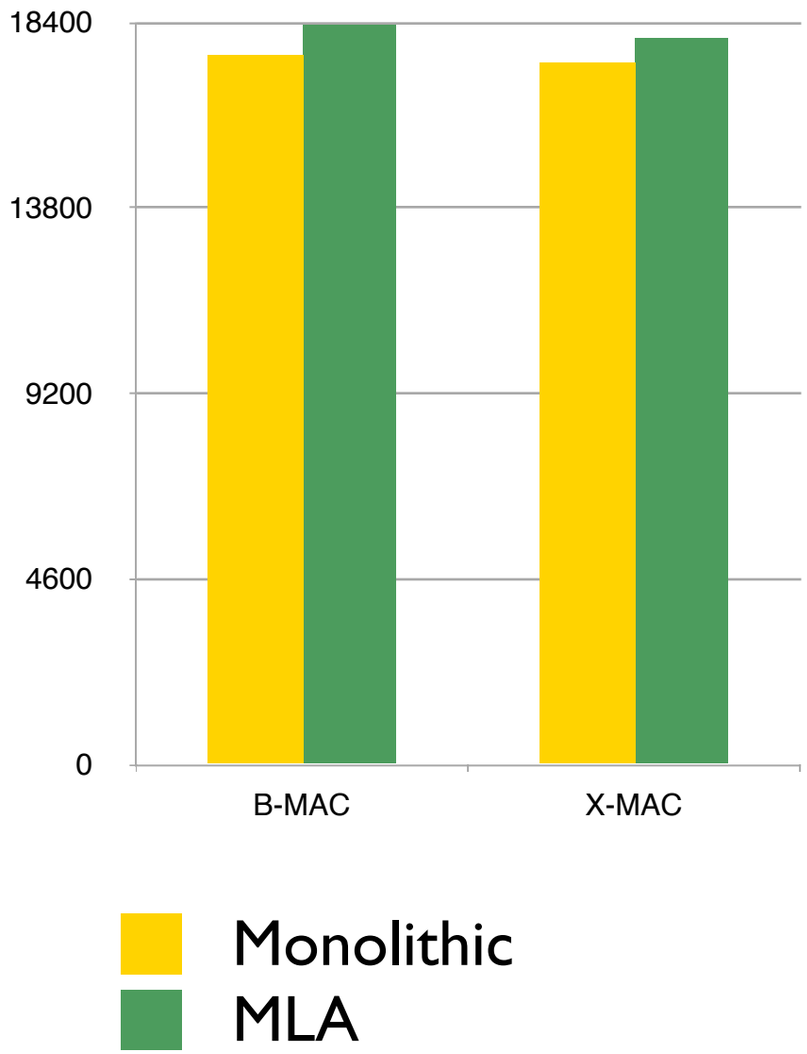

Lines of code
1500
1125
750
375
0
B-MAC
X-MAC
SCP-Wustl
Pure TDMA
SS-TDMA
Protocol-Specific
Reused

Reusability of Components

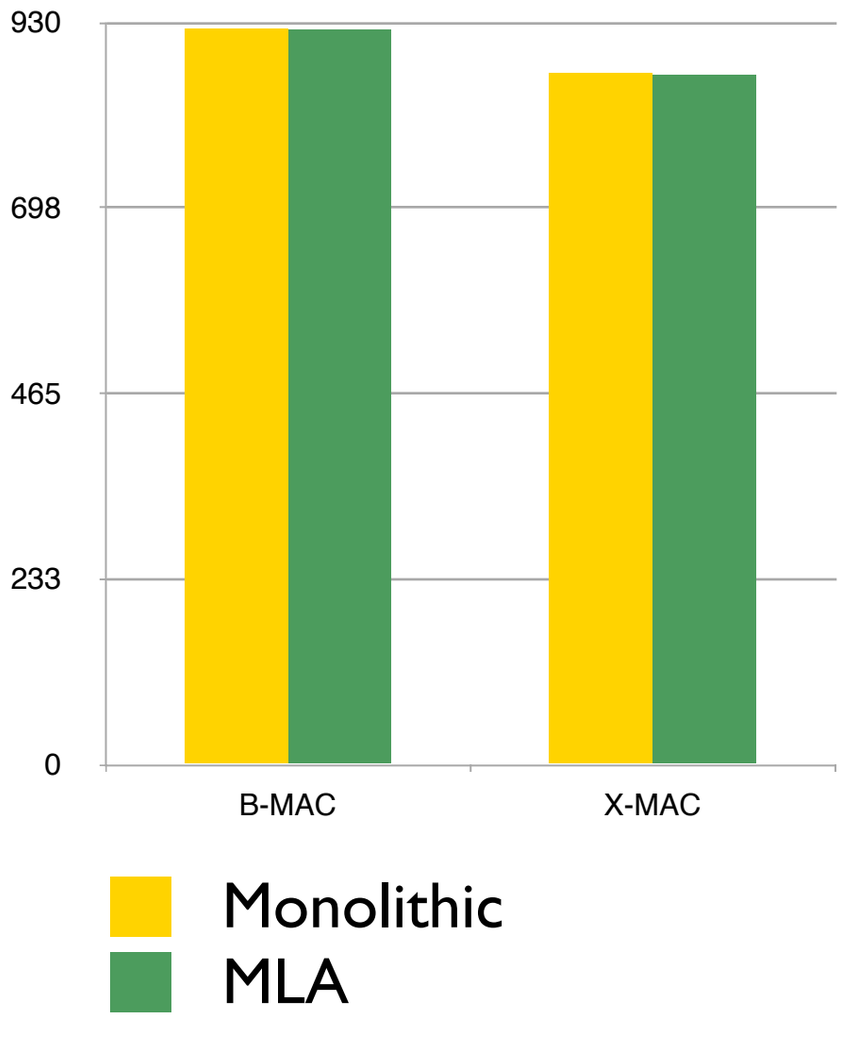
	B-MAC	X-MAC	SCP-Wustl	Pure-TDMA	SS-TDMA
Channel Poller	●	●	●		
LPL Listener	●	●	●		
Preamble Sender	●	●	●		
Time Synchronization			●	●	●
TDMA Slot Handler				●	●
CSMA Slot Handler					●
Low Level Dispatcher				●	●
Async I/O Adapter	●	●	●	●	●
Alarm	●	●	●	●	●
Local Time			●	●	●
Radio Core	●	●	●	●	●
Other Components	3	3	4	2	2
Reused Components	6	6	8	7	8

Memory Footprint (TelosB)

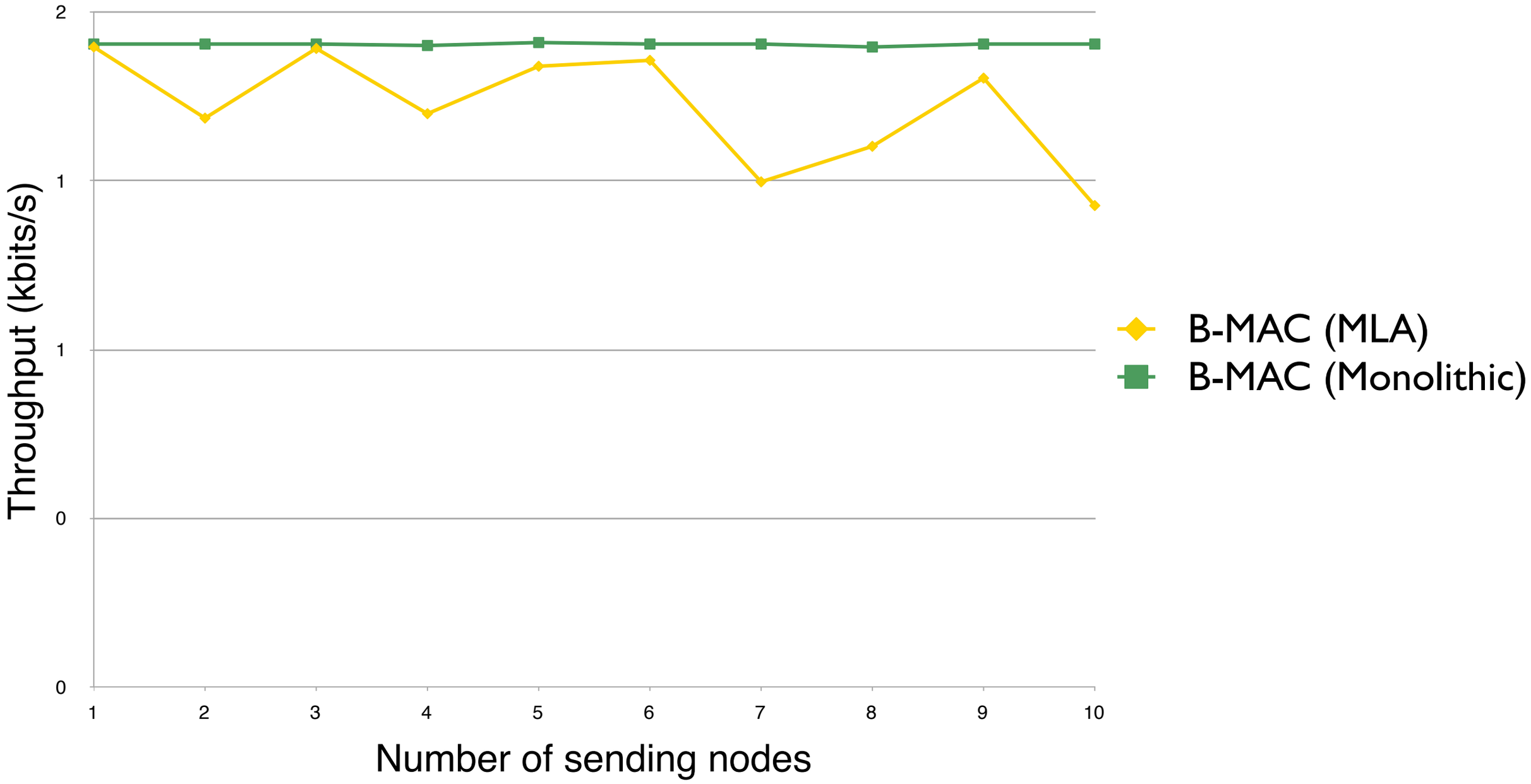
ROM Overhead



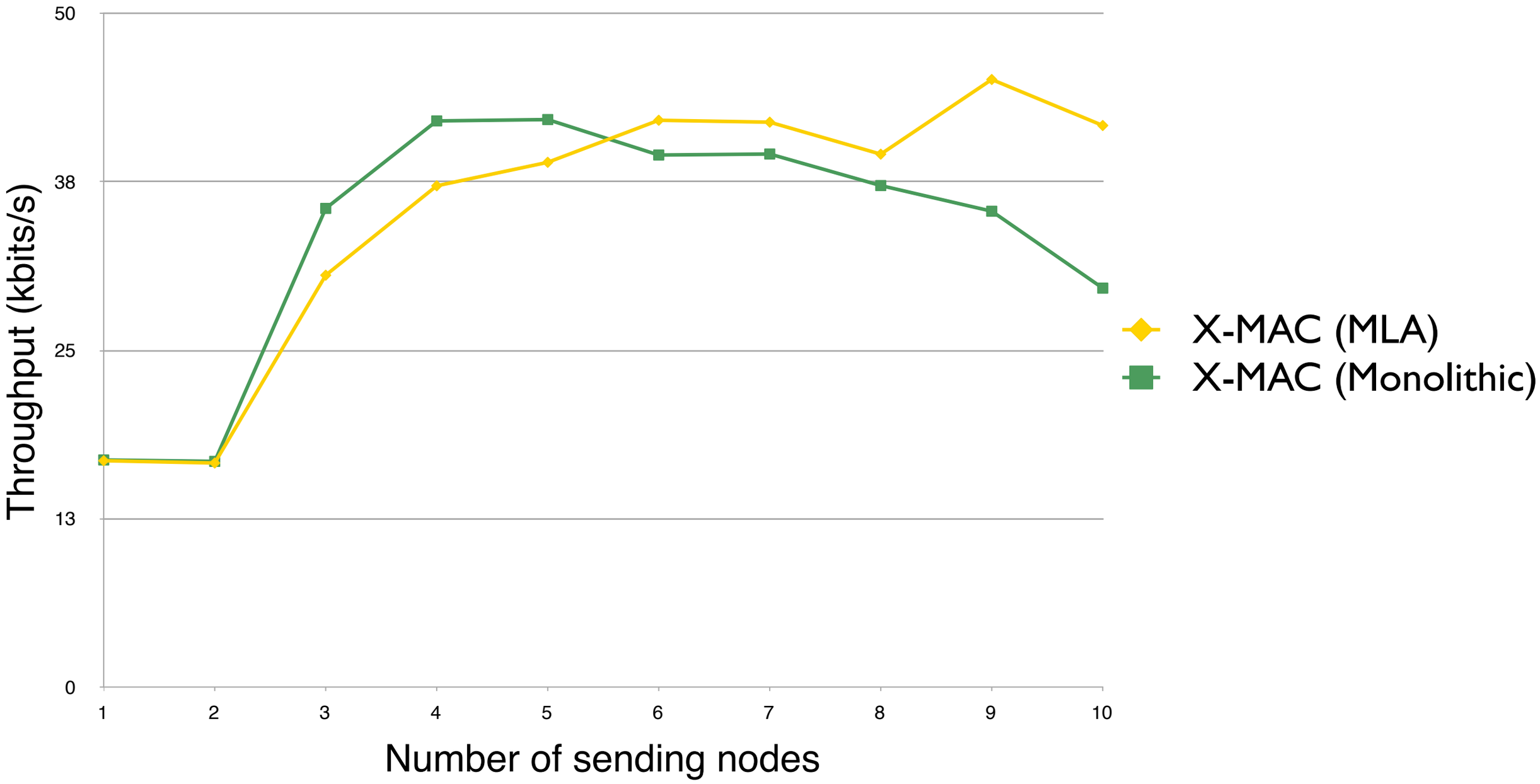
RAM Overhead



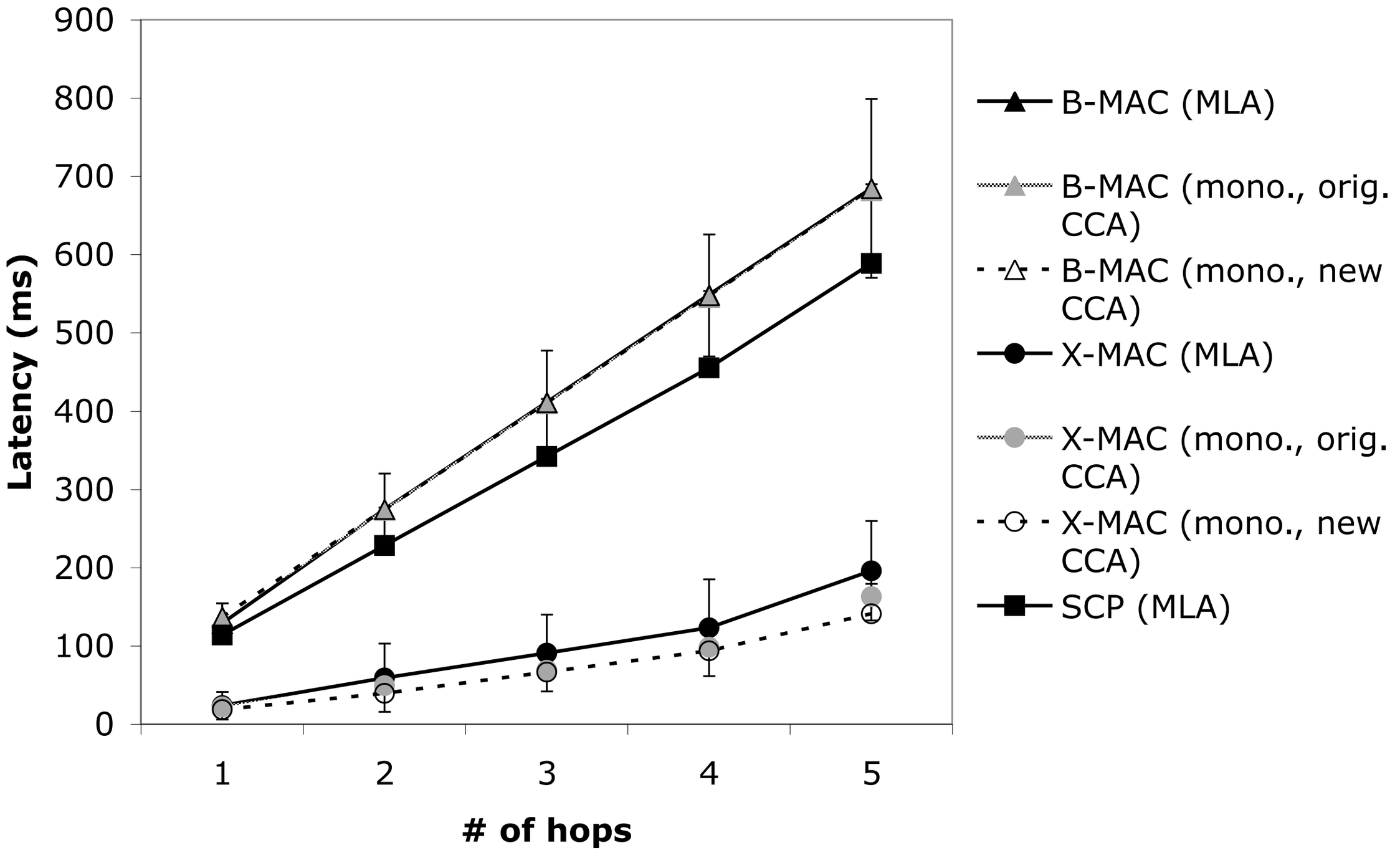
Throughput (BMAC)



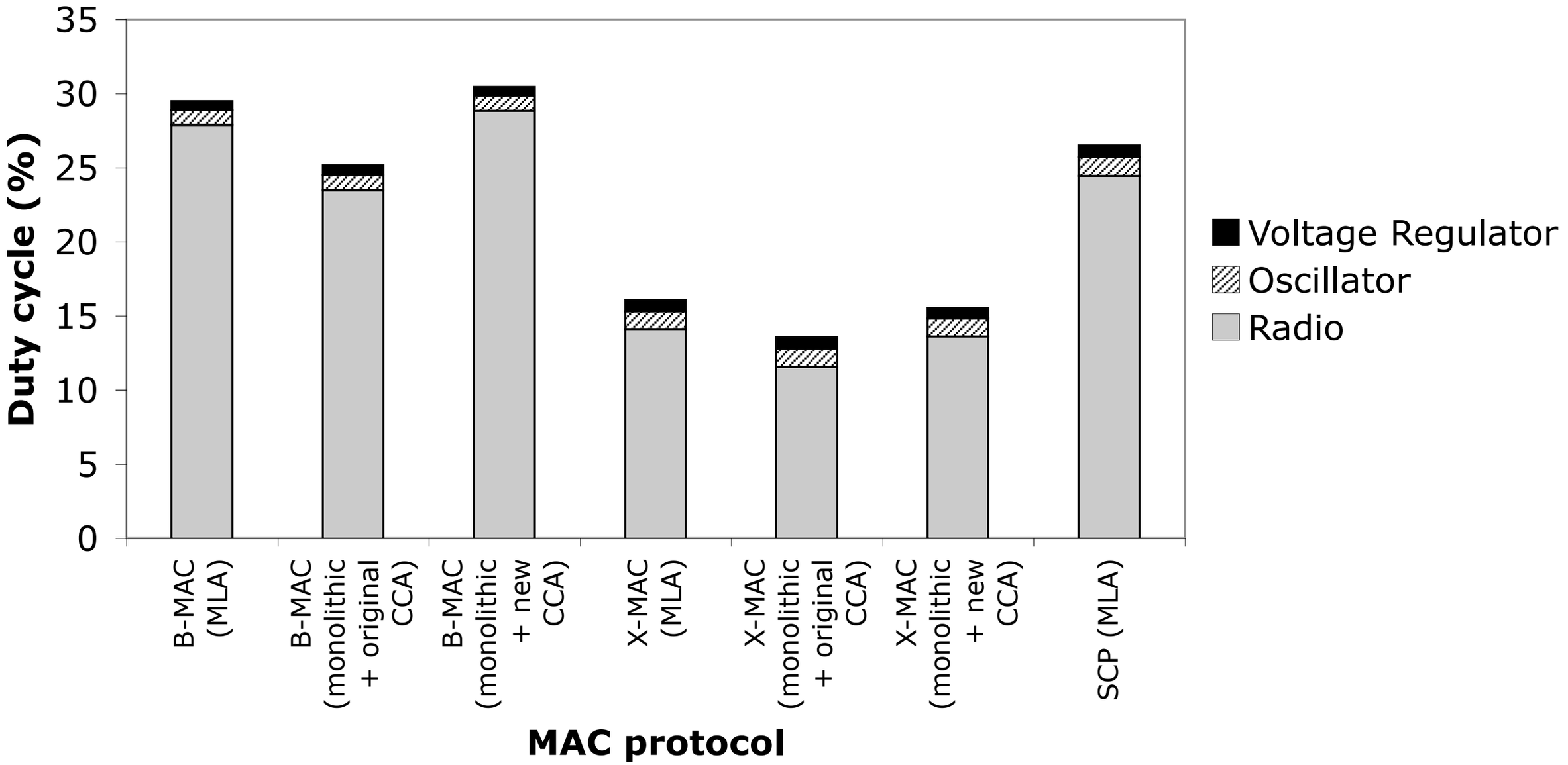
Throughput


50
38
25
13
0
Throughput (kbits/s)
1
2
3
4
5
6
7
8
9
10
Number of sending nodes
X-MAC (MLA)
X-MAC (Monolithic)

Latency


900
800
700
600
500
400
300
200
100
0
Latency (ms)
1
2
3
4
5
# of hops
B-MAC (MLA)
B-MAC (mono., orig. CCA)
B-MAC (mono., new CCA)
X-MAC (MLA)
X-MAC (mono., orig. CCA)
X-MAC (mono., new CCA)
SCP (MLA)

Duty cycle


Duty cycle (%)
35
30
25
20
15
10
5
0
B-MAC (MLA)
B-MAC (monolithic + original CCA)
B-MAC (monolithic + new CCA)
X-MAC (MLA)
X-MAC (monolithic + original CCA)
X-MAC (monolithic + new CCA)
SCP (MLA)
MAC protocol
Voltage Regulator
Oscillator
Radio

MLA: Summary

- **Component-based, low-power MAC architecture**
  - Increases flexibility
  - Simplifies development
  - Reduces porting effort

- **Provides evidence contrary to the existing philosophy that radio stacks must be monolithic to be efficient**
  - Bridge the gap between algorithms/protocols and systems.

- **Code: tinyos-2.x-contrib/wustl/upma**

Solve the Real Problems

- **Hard to develop new MAC protocols?**
  - RI-MAC (SenSys'08) built on top of MLA
  - More built on MLA
- **Hard to maintain multiple MAC stacks as OS evolves?**
  - Upgrading MLA for TinyOS 2.0.1->2.0.2->2.1 took several hours
  - Multiple MAC protocols survived upgrade without any change!
- **Protocols not reusable across radio/processor platforms?**
  - Supports both Telos and MicaZ
- **TinyOS 2.1 version available from TinyOS "contrib" CVS**

References

- K. Klues, G. Hackmann, O. Chipara and C. Lu, A Component-Based Architecture for Power-Efficient Media Access Control in Wireless Sensor Networks, SenSys'07.
- K. Klues, G. Xing and C. Lu, Link Layer Support for Unified Radio Power Management in Wireless Sensor Networks, IPSN'07.