Link Layer Support for Unified Radio Power Management in Wireless Sensor Networks

Kevin Klues, Guoliang Xing, Chenyang Lu

Department of Computer Science and Engineering

Washington University in St. Louis

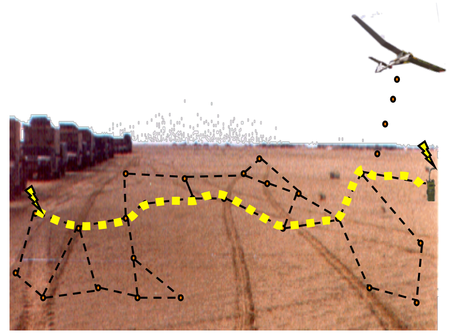
IPSN ‘07

April 27th, 2007

Wireless Sensor Network Applications

Habitat monitoring

Structural monitoring



Intruder Detection

Healthcare

- Limited power supplies: batteries, small solar panels
- Long lifetime requirements: months to tens of years

⇨ Effective power management policies are crucial

Problem

- Communication cost is high
  - ⇨ Explosion in the development of various radio power management protocols
- Protocols make different assumptions
  - ⇨ No single protocol is suited to the needs of every application
- Existing radio stack architectures are monolithic
  - ⇨ Hard to develop new protocols or tune existing ones to specific application requirements

Solution: UPMA

- Unified Radio Power Management Architecture
- Monolithic --> Composable radio stack architecture
- Pluggable power management policies
- Separation of power management features
- Cross layer in nature

This work focuses on providing link layer support towards the realization of such an architecture

Outline

- Radio Power Management Overview
- Design and Implementation
- Evaluation
- Summary
- Questions…

Radio Power Management

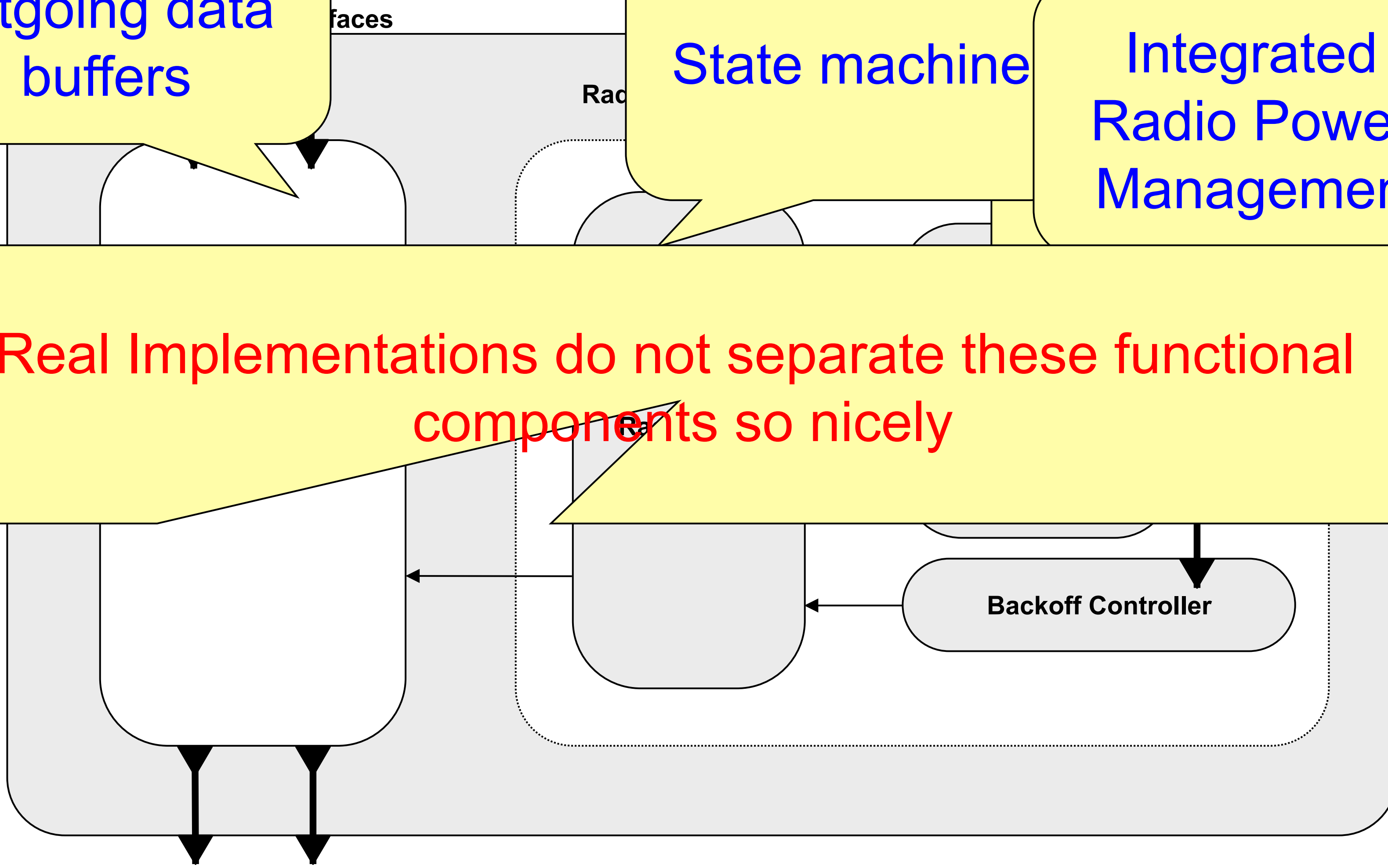
- Scheduled Contention Protocols (e.g. S-MAC)
  - Low sending cost
  - High reception cost
  - Time synchronization overhead
- Channel Polling Protocols (e.g. B-MAC / LPL)
  - High sending cost
  - Low reception cost
  - NO time synchronization required
- Backbone based protocols (e.g. PEAS)
  - Complement both of these types of protocols
  - Continuously active backbone
  - Sacrifice energy to provide improved performance

Traditional Core Radio Functionality

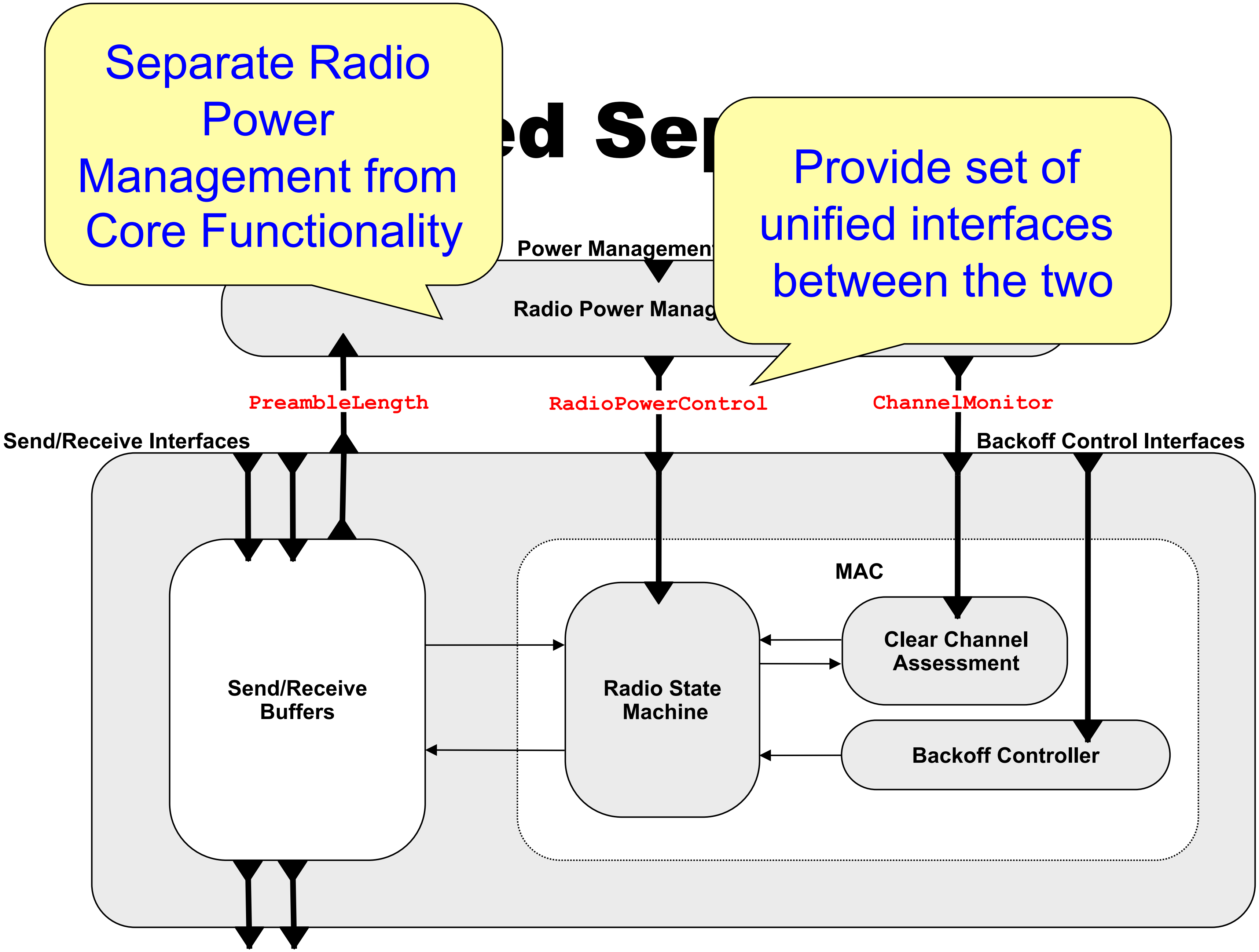
Incoming and Outgoing data buffers

State machine

Integrated Radio Power Management



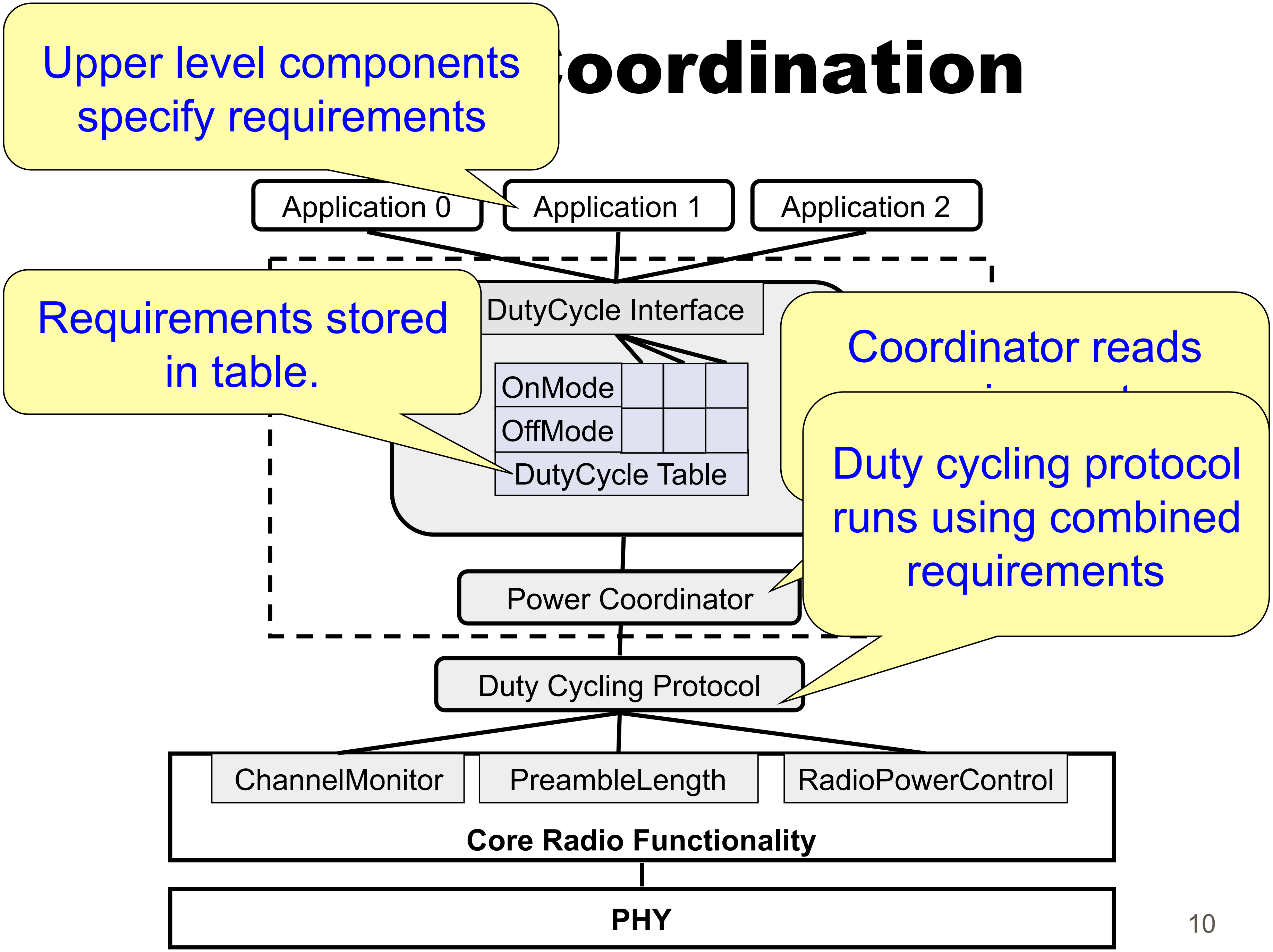
Real Implementations do not separate these functional components so nicely


Separate Radio Power Management from Core Functionality
ed Se
Provide set of unified interfaces between the two
Power Managemen
Radio Power Manag
PreambleLength
RadioPowerControl
ChannelMonitor
Send/Receive Interfaces
Backoff Control Interfaces
MAC
Send/Receive Buffers
Radio State Machine
Clear Channel Assessment
Backoff Controller

Unified Interfaces

- `RadioPowerControl`
  - Turn radio on and off
  - Required by ***all*** duty cycling protocols
- `ChannelMonitor`
  - Clear channel assessment (CCA) capabilities
  - Required by both ***contention*** and ***polling*** based protocols
- `PreambleLength`
  - Set preamble length
  - Required by ***polling*** based protocols

Coordination



The Power Coordinator

- No “one-size-fits-all” power coordinator
- Separates core logic of each upper level component from coordination with others
- Different coordination policies can be plugged in without modification of any protocols above or below it

Implementation

- Created instance for the CC1000 (mica2) and CC2420 (telos) radio stacks in tinyos-2.0

- Protocols implemented:
  - Polling based - Low Power Listening (LPL)
  - Contention based - Simple Synchronous Sleeping (SSS)
    - Basic Sleep Scheduler (BSS)
  - Backbone - PEAS

Implementation

- Coordination policies implemented
  - Simple OR policy
    - If *any* application wants the radio on, it will be turned on
    - Only if *all* applications want it off, will it be turned off
  - Policy combining PEAS with BSS
    - Nodes in backbone run BSS
    - Others sleep except when attempting to join the backbone

Efficiency: Memory Footprint

- Monolithic LPL vs. our LPL (mica2 implementation)

	Original LPL	New LPL
	RAM/ROM	RAM/ROM
SenderApp	383/11956	394/12350
ReceiverApp	705/15098	716/15560

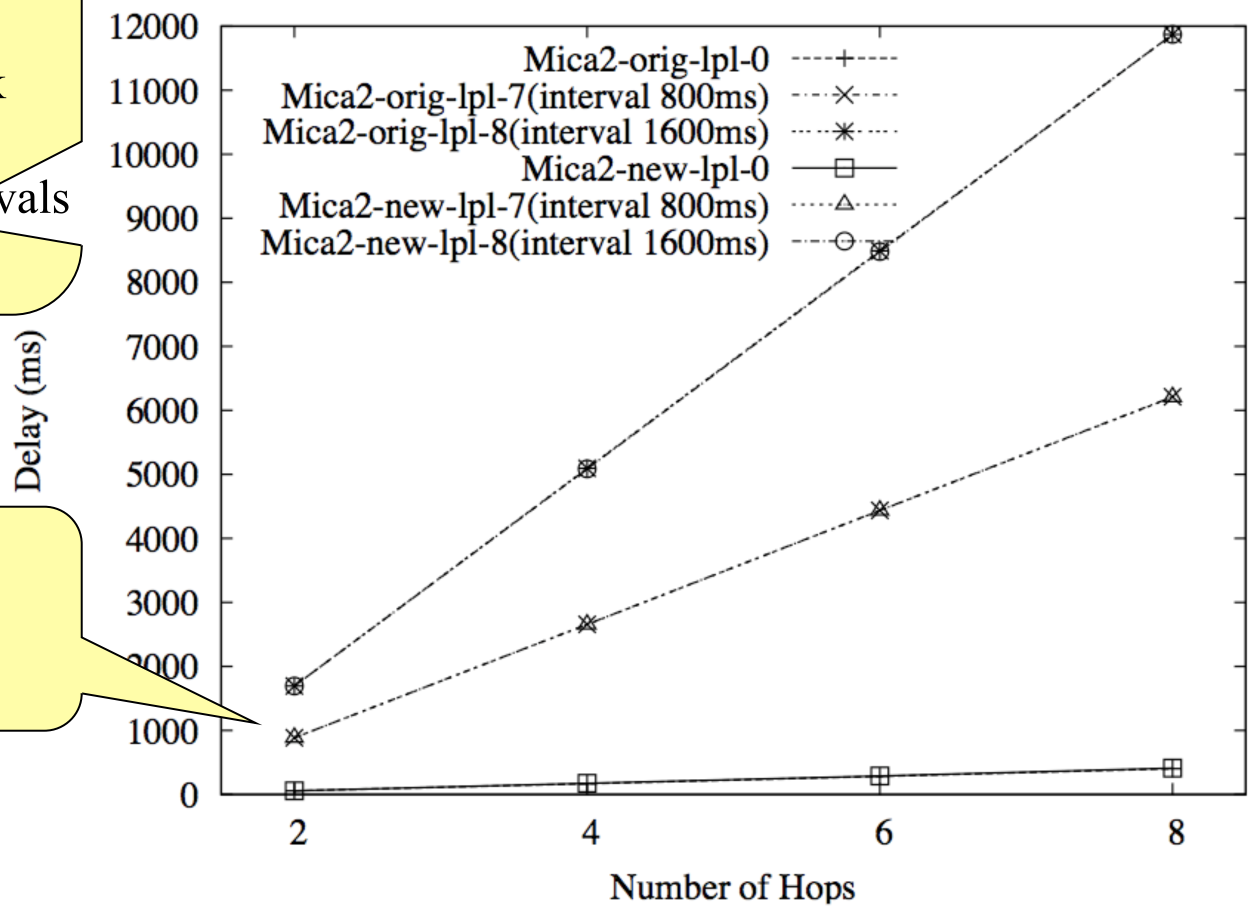
Bytes

- Compiled for TinyOS-2.0 with simple sending application
- Slight increase in code size due to addition of separate timer for performing LPL checks and flags for control logic

Efficiency: Latency

- Monolithic LPL vs. Our LPL (mica2 implementation)

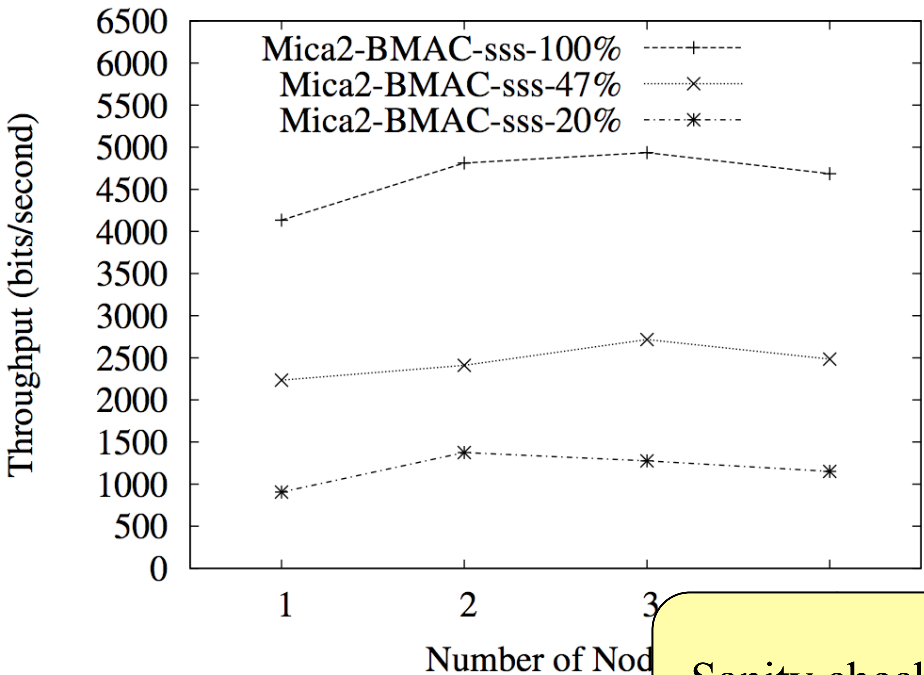
- 5 Nodes in chain
- First is both source and sink
- 20 messages @ 38 bytes
- Run at different check intervals

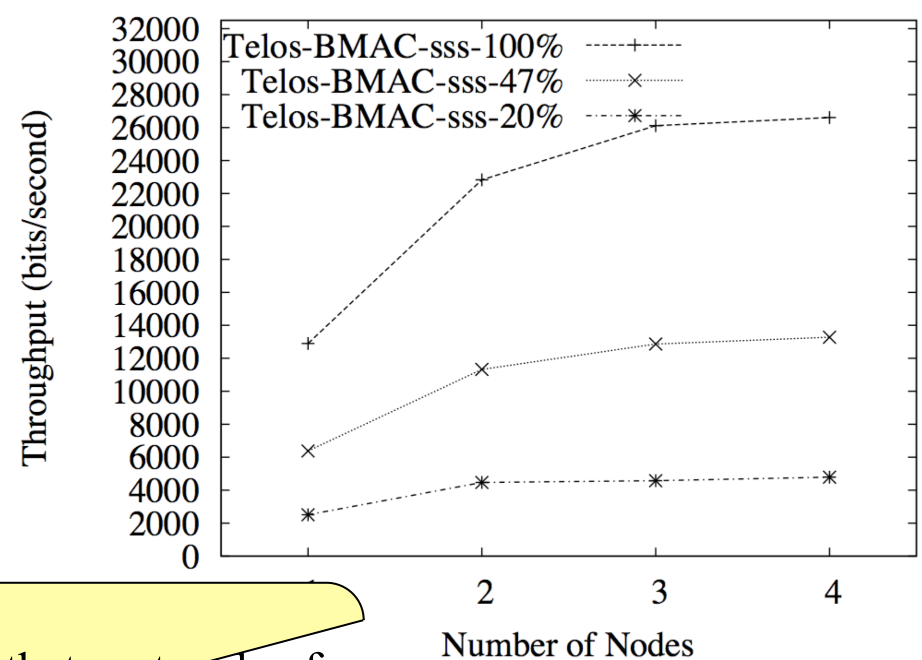


Identical performance independent of check interval

Portability

- Mica2 and Telos run same SSS implementation on top of respective radio stacks exposing unified interfaces
- The Telos radio can send at a much higher baud rate and thus has considerably higher throughput

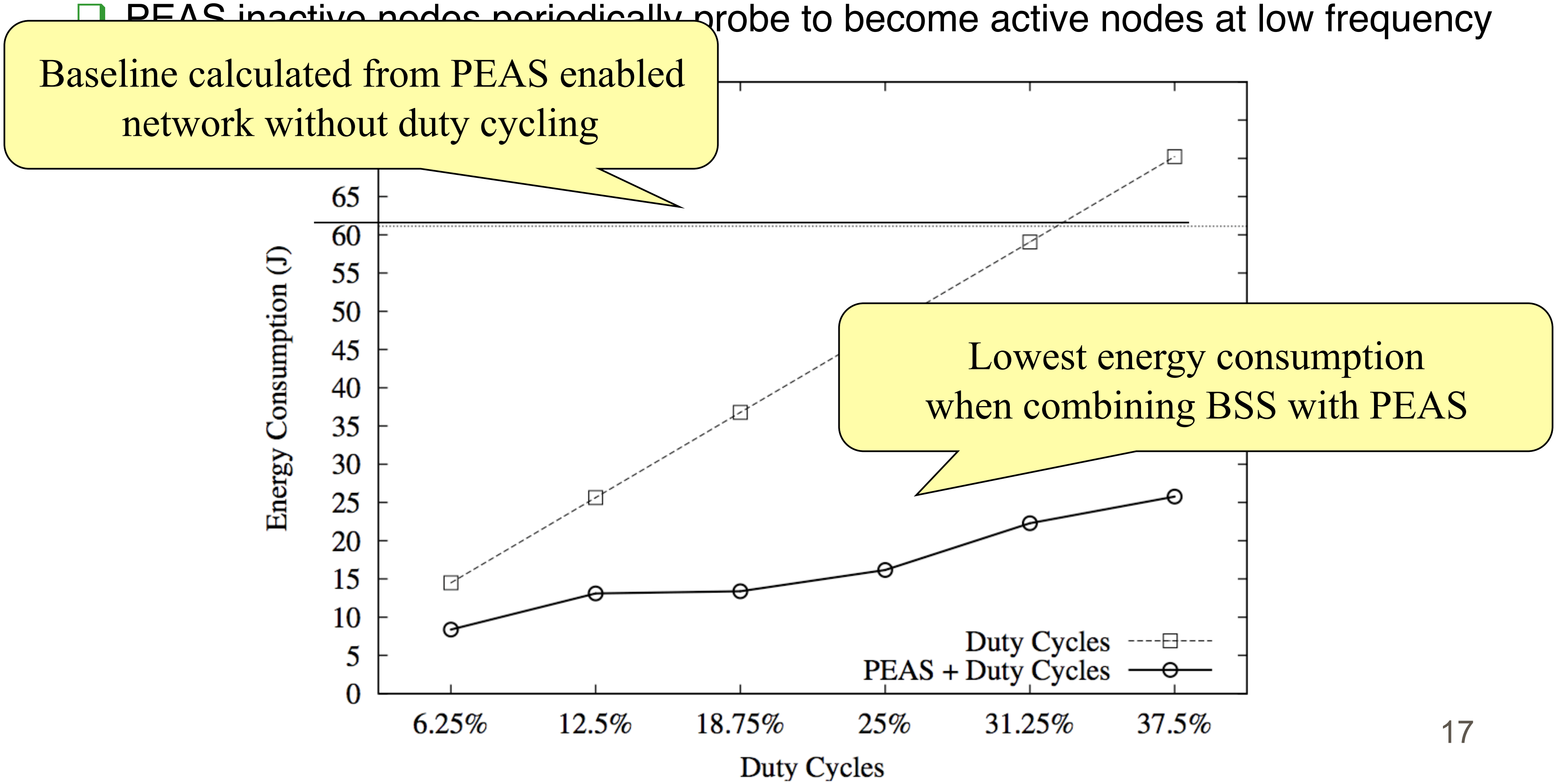




Sanity check that protocols of this type can be implemented

Coordination

- One base station receiver (Telos)
- Fifteen potential senders in 5x3 grid (Telos)
- Each sender runs PEAS as well as 1 of 6 different applications
- PEAS active nodes run BSS, sending periodically to the base station
- PEAS inactive nodes periodically probe to become active nodes at low frequency



Summary

- Different applications have different requirements
  - Throughput
  - Latency
  - Network lifetime
- Different power management protocols designed to meet these different requirements
- Link Layer support for UPMA allows:
  - Flexible integration of radio power management protocols
  - Composability of complementary protocols
- This is the first step towards a full realization of the entire UPMA architecture……

Questions?

?