# ULTRA-LOW DUTY CYCLE MAC WITH SCHEDULED CHANNEL POLLING

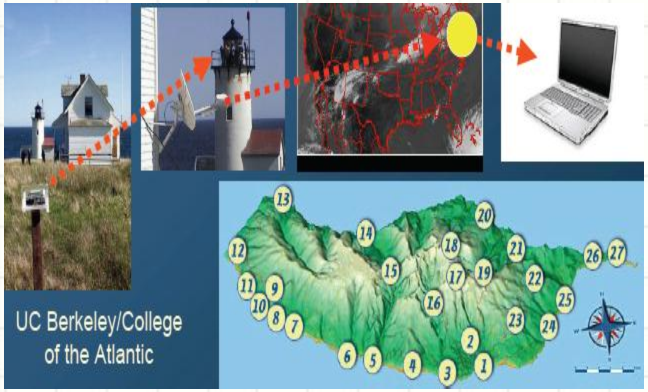
Wei Ye, Fabio Silva
John Heidemann

Present By: Eric Wang

# Outline

- Introduction
- Design of SCP-MAC
- Lower Bound of Energy Performance with Periodic Traffic
- Protocol Implementation
- Experimental Evaluation
- Related Work

# WSN applications



- Sensor deployments for months or years
- Energy is crucial
- Idle listening waste

# Channel Polling & Duty Cycle

- Channel Polling
  - Nodes wake up very briefly to check channel activity without actually receiving data
  - Sender sends a long preamble to guarantee intersecting with a polling
- Duty Cycle
  - Ratio between listen time and a full listen/sleep interval

# LPL

- Low Power Listening
  - WiseMAC, B-MAC
- Very Brief channel polling activity
- Long Preambles
- Save energy particularly during low network utilization
- Synchronization cost
  - Scheduling
  - Long Preambles
  - Limit Duty Cycle 1-2%

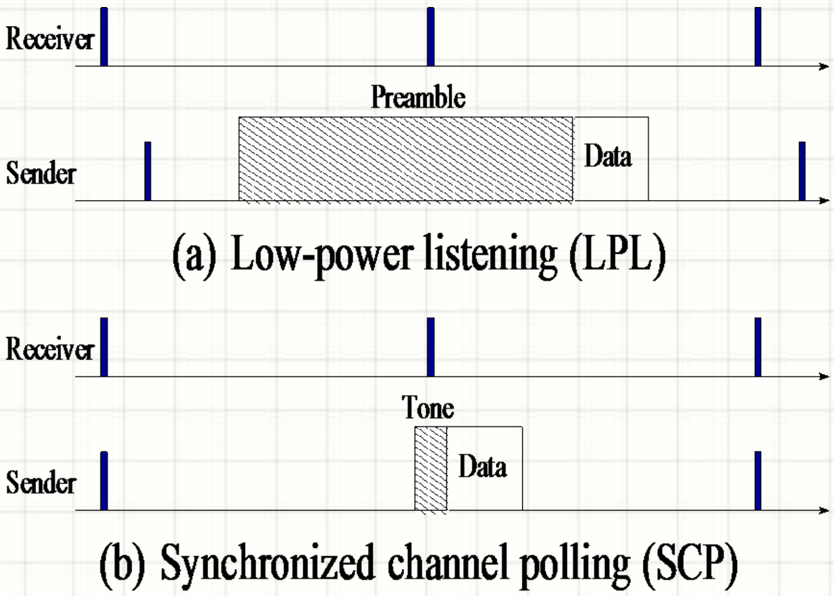
# SCP – Scheduled Channel Polling vs. LPL

| **Low-power Listening (LPL)** | **Scheduled Channel Polling (SCP)** |
|---|---|
| Efficiency gained at much big cost for senders | Optimal configurations for synchronizing channel polling, minimized energy cost |
| Very sensitive to tuning for neighborhood size and traffic rate | Adapt well to variable traffic, broadcast, unicast under bursty traffic rates |
| Hard to adapt to newer radios like 802.15.4 due to limited preamble size | Can operate effectively on new radios |

# Outline

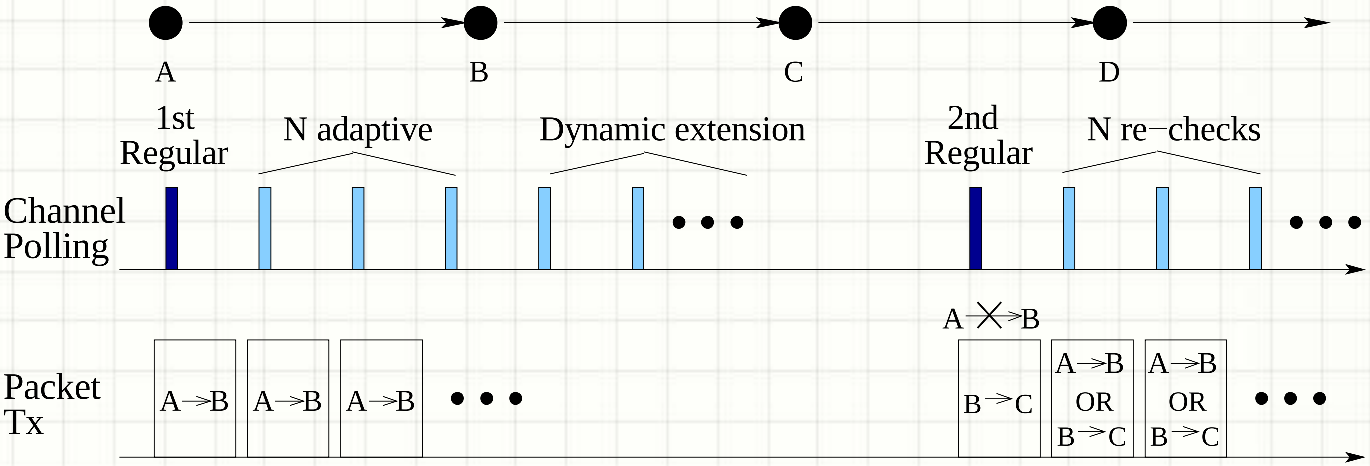
- Introduction
- Design of SCP-MAC
- Lower Bound of Energy Performance with Periodic Traffic
- Protocol Implementation
- Experimental Evaluation
- Related Work

# Synchronized Channel Polling



- Synchronization
  - Short wake-up tones
  - Reduces overhearing cost
  - Effective for broadcast and unicast
- Reduce Sync Cost
  - Broadcasts Sync schedules to neighbors by SYNC packet
  - Piggyback information

# Adaptive Channel Polling and Multi-hop Streaming



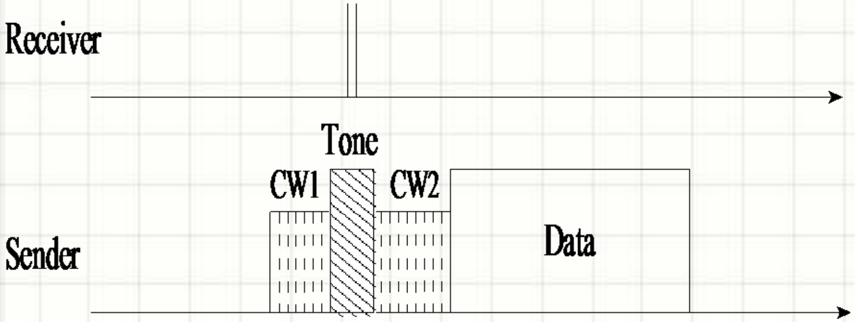
Basic idea: Add additional high-frequency polling slots to nodes on the path.

Short burst: Apply strategy to next node before sending

Long burst: Apply strategy to all nodes on the path before sending

# Other Optimizations

**Two-Phase Contention**



Carrier sense in CW1

if IDLE: send a tone, enter CW2

else: waits for receiving

if CW2 also IDLE: send data

**Overhearing Avoidance Based on Headers**

- RTS/CTS enabled:
  - Same as S-MAC
- RTS/CTS disabled:
  - Receiver checks destination after receiving MAC header
  - If not to itself, stop receiving and go back to sleep

# Outline

- Introduction
- Design of SCP-MAC
- Lower Bound of Energy Performance with Periodic Traffic
- Protocol Implementation
- Experimental Evaluation
- Related Work

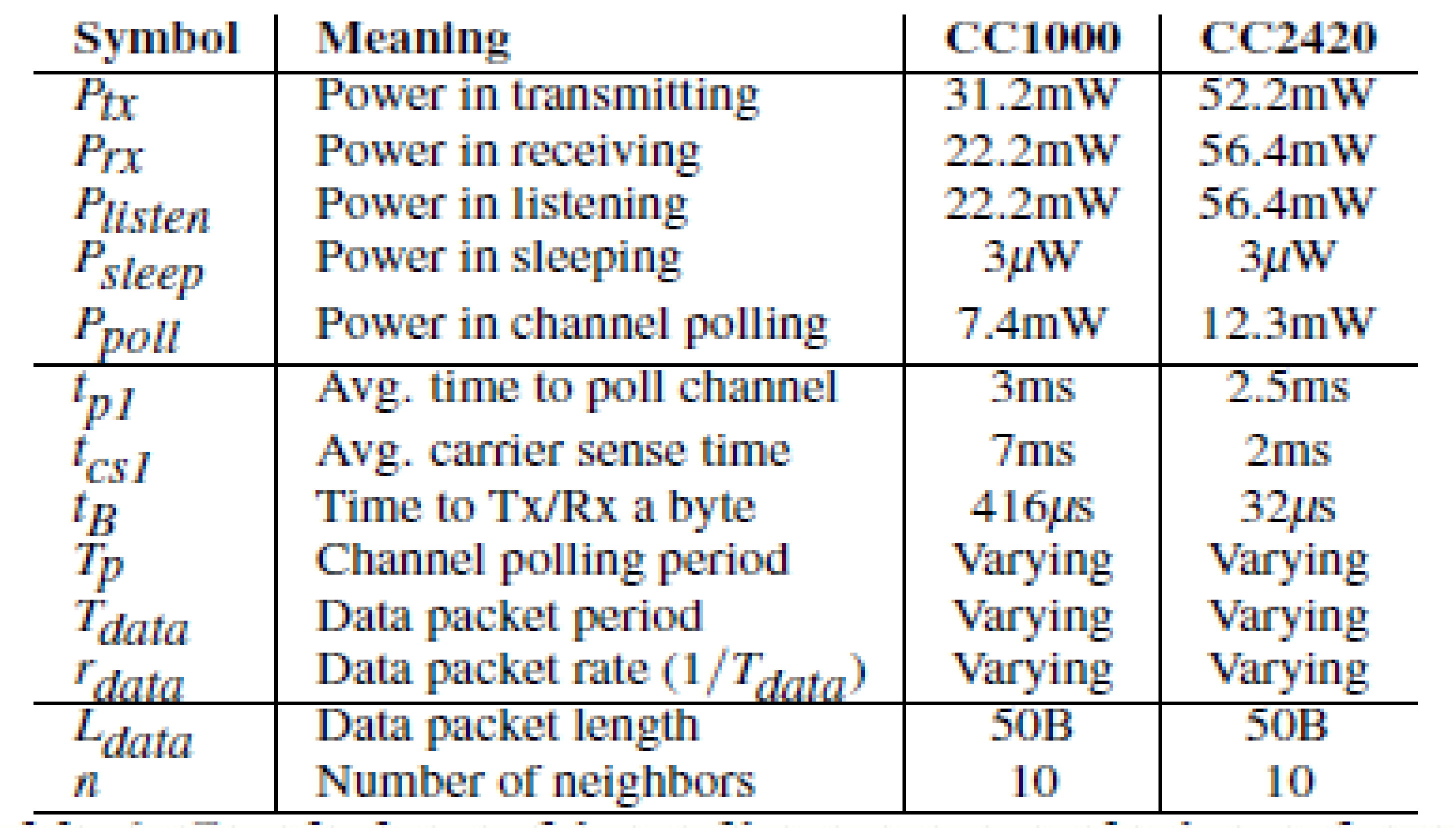
# Models and Metrics

- Analysis focuses on the energy consumption by the radio, and does not model other components, such as the CPU or sensors.
- Four stable radio states: transmitting, receiving, listening, and sleeping. $P_{tx}$, $P_{rx}$, $P_{listen}$ *and* $P_{sleep}$
- Expected energy consumption, per node:

$$E = E_{cs}+E_{tx}+E_{rx}+E_{poll}+E_{sleep}$$

$$= P_{listen}t_{cs}+P_{tx}t_{tx}+P_{rx}t_{rx}+P_{poll}t_{poll}+P_{sleep}t_{sleep}$$

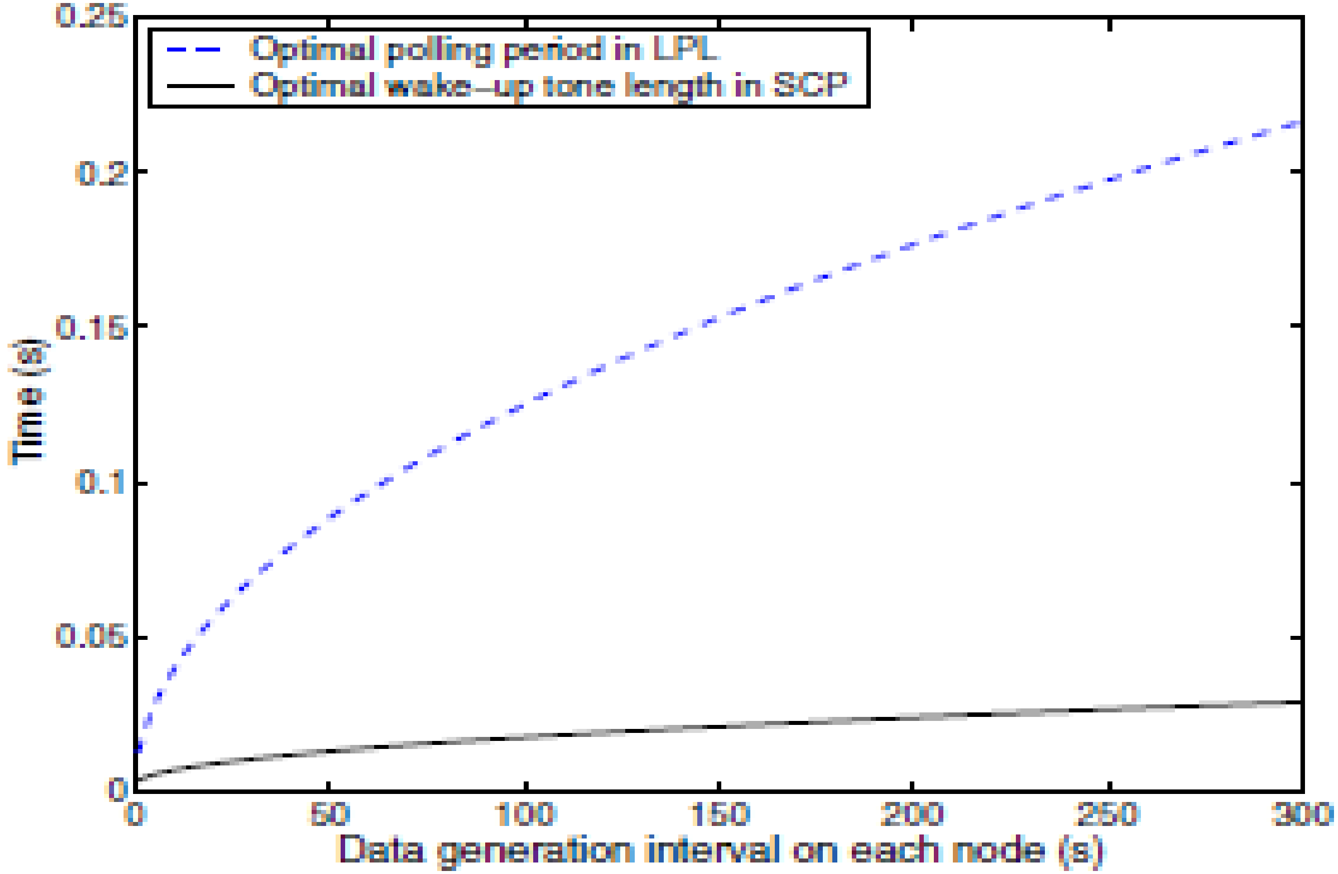
# Models and Metrics



| Symbol | Meaning | CC1000 | CC2420 |
|---|---|---|---|
| $P_{tx}$ | Power in transmitting | 31.2mW | 52.2mW |
| $P_{rx}$ | Power in receiving | 22.2mW | 56.4mW |
| $P_{listen}$ | Power in listening | 22.2mW | 56.4mW |
| $P_{sleep}$ | Power in sleeping | 3$\mu$W | 3$\mu$W |
| $P_{poll}$ | Power in channel polling | 7.4mW | 12.3mW |
| $t_{p1}$ | Avg. time to poll channel | 3ms | 2.5ms |
| $t_{cs1}$ | Avg. carrier sense time | 7ms | 2ms |
| $t_B$ | Time to Tx/Rx a byte | 416$\mu$s | 32$\mu$s |
| $T_p$ | Channel polling period | Varying | Varying |
| $T_{data}$ | Data packet period | Varying | Varying |
| $r_{data}$ | Data packet rate ($1/T_{data}$) | Varying | Varying |
| $L_{data}$ | Data packet length | 50B | 50B |
| $n$ | Number of neighbors | 10 | 10 |

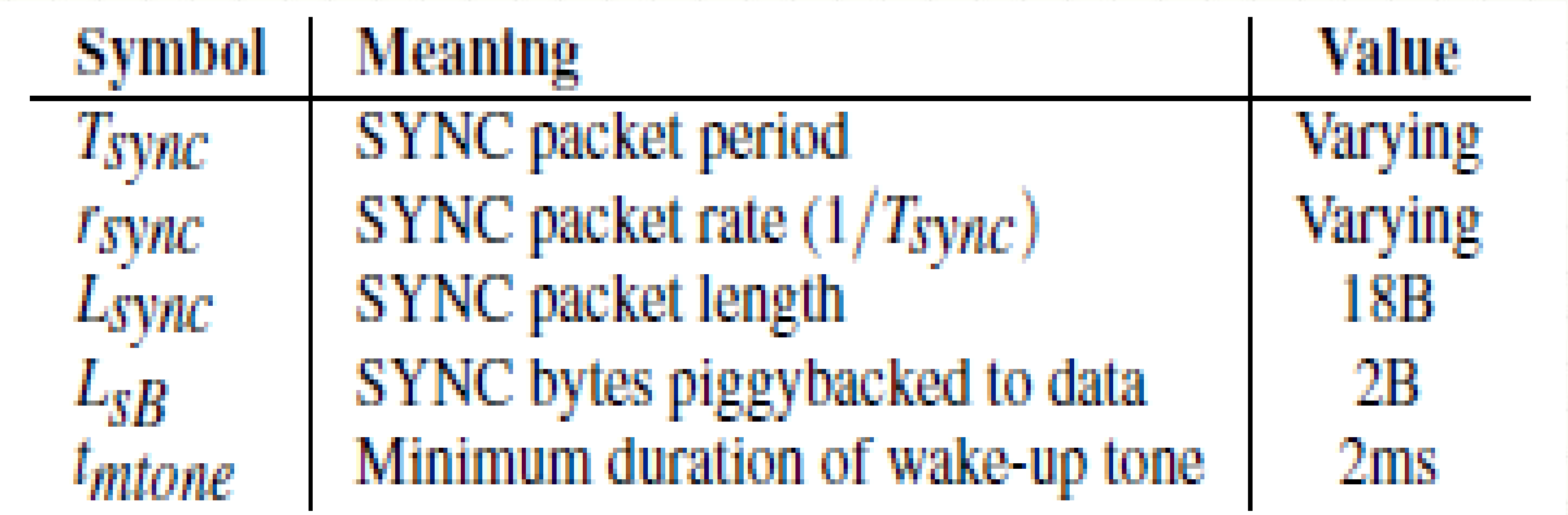
Symbols used in radio energy analysis, and typical values for the Mica2 radio (CC1000) and an 802.15.4 radio (CC2420)

# Asynchronous Channel Polling: LPL



Optimal channel polling period in LPL (dotted), and wakeup-tone length in SCP (solid), given neighborhood size of 10

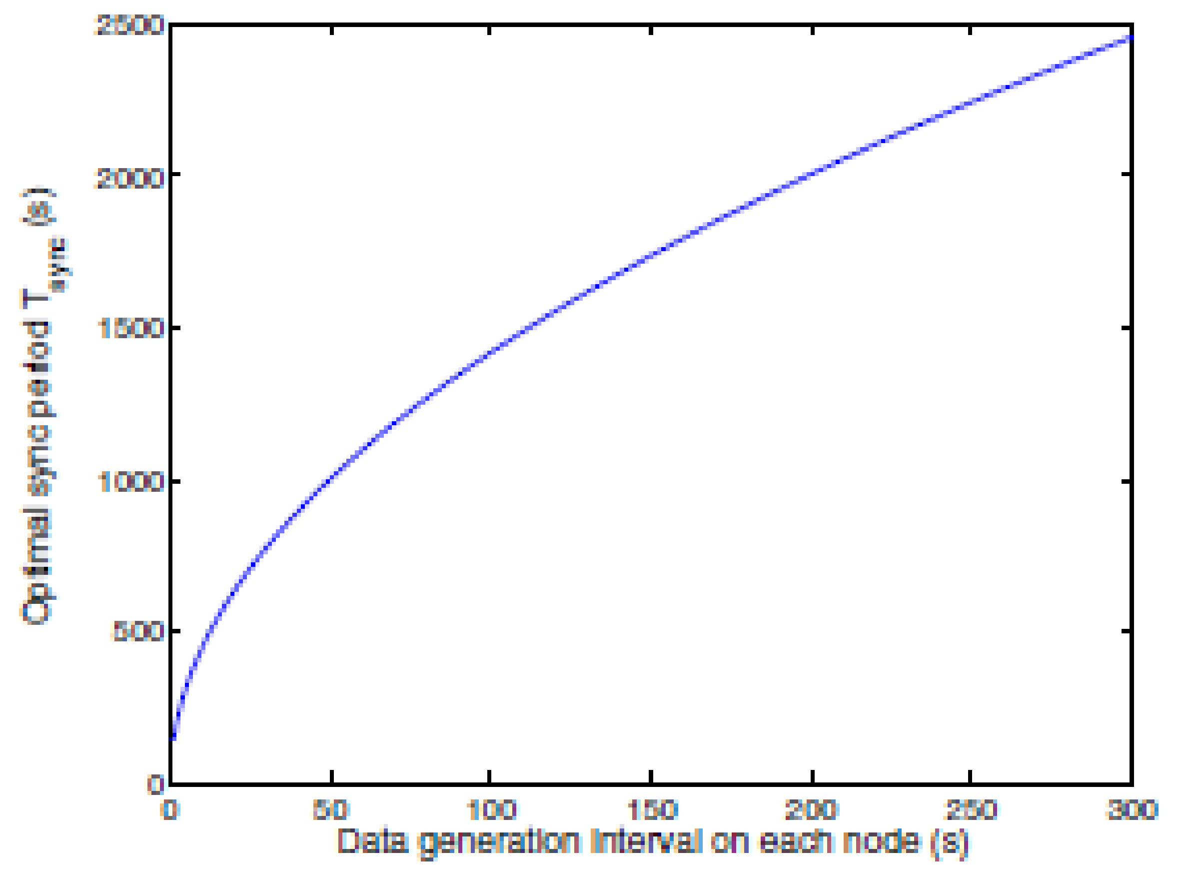
# Scheduled Channel Polling: SCP



| Symbol | Meaning | Value |
|---|---|---|
| $T_{sync}$ | SYNC packet period | Varying |
| $r_{sync}$ | SYNC packet rate ($1/T_{sync}$) | Varying |
| $L_{sync}$ | SYNC packet length | 18B |
| $L_{sB}$ | SYNC bytes piggybacked to data | 2B |
| $t_{mtone}$ | Minimum duration of wake-up tone | 2ms |

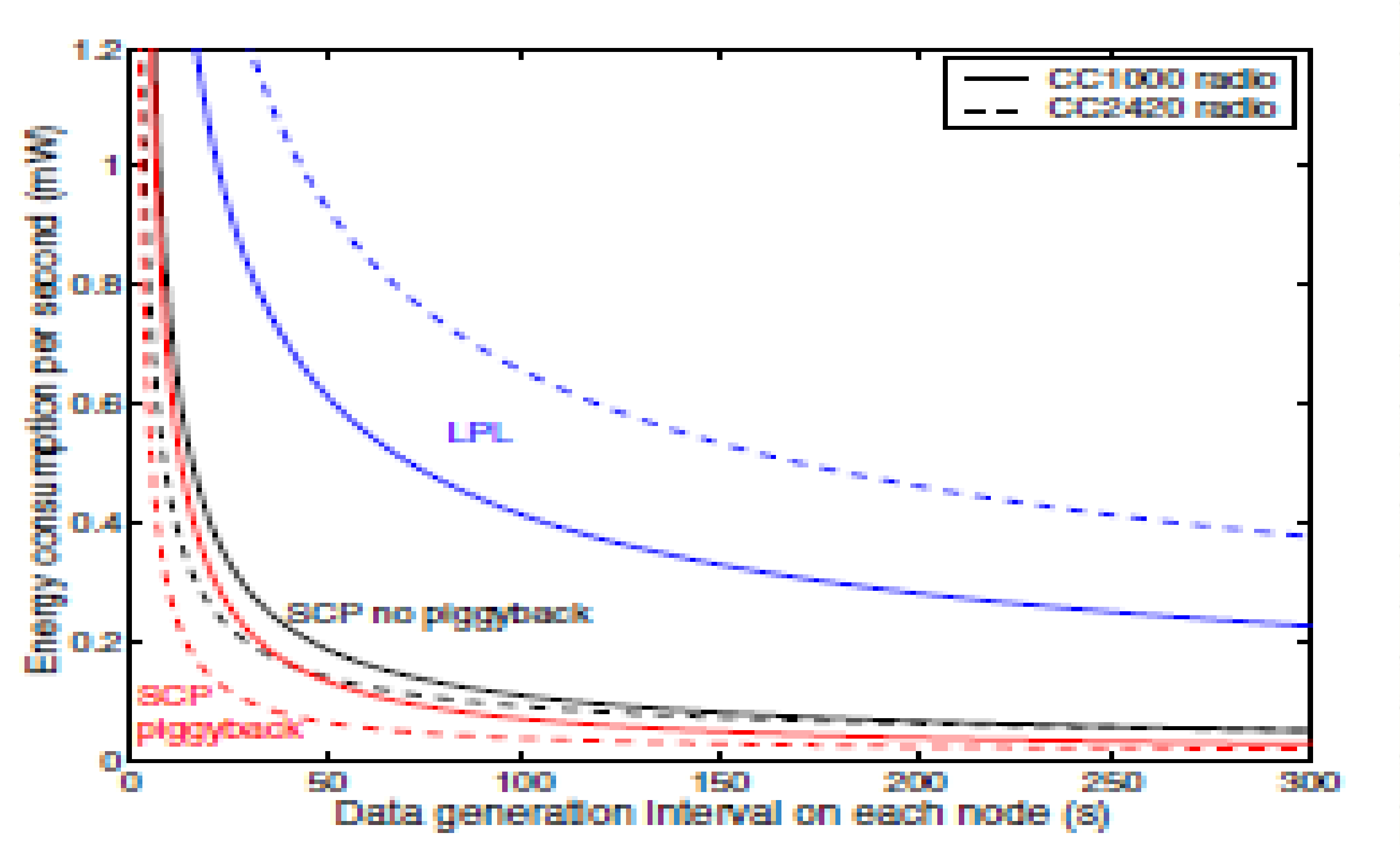
Additional parameters in SCP-MAC

# Scheduled Channel Polling: SCP



Optimal SYNC period for SCP-MAC

# Scheduled Channel Polling: SCP



Analysis of optimal energy consumption for LPL and SCP with and without piggyback for CC1000 (solid lines) and CC2420 (dashed)

# Outline

- Introduction
- Design of SCP-MAC
- Lower Bound of Energy Performance with Periodic Traffic
- Protocol Implementation
- Experimental Evaluation
- Related Work

# Protocol Implementation

- Implement SCP-MAC in TinyOS over the Mica2 motes with the CC1000 radio
- Describe the preliminary port to MicaZ motes with the CC2420 radio supporting IEEE802.15.4

# Software Architecture

- Break MAC functionality into four layers:
  - PHY: physical layer
  - CSMA layer
  - LPL layer
  - SCP layer
- Several parameters and options at compile time:
  - RTS/CTS handling
  - Overhearing avoidance
  - Adaptive channel polling

# Physical Layer

- PHY layer (bottom of the stack)
  - Handles radio states
  - Interacts directly with radio, sending bytes/packets
  - Buffers all bytes, pass entire packet to MAC
  - Carrier sense, wakeup tone, CRC, time-stamping
  - Record time spent in each radio state
  - MAC-independent

# CSMA layer

- Basic CSMA protocol
  - Providing common service to LPL and SCP
  - Includes preamble length parameter
  - Perform carrier sense and random backoff
  - For unicast, RTS/CTS disable/enable (compile time)
  - Optional retransmission and overhearing avoidance

# LPL layer

- Major purpose: periodically poll the channel and send radio to sleep when no activity
  - Adjust preamble length to ensure intersect with polling frequency
  - Coordinates concurrent polling and transmission
  - Exports interface to query and adjust channel polling times, to support SCP.

# SCP layer

- Implement above LPL
- Uses basic LPL to bootstrap schedules with SYNC packets
- Coordinates packet transmission timing.
- Implements the randomized contention widow before wakeup tone transmission

# Interaction with TinyOS

- Implement a new timer in TinyOS to add support for dynamically adjusting timer values and asynchronous, low-jitter triggers
- Timer implementation is based on the 8-bit hardware counter on Mica2
- Runs independently from the CPU, allowing the CPU to sleep when no other activity is present
- Each timer event is about 0.4% of the cost of a channel poll

# Efficient piggybacking of synchronization information

- To minimize the cost of synchronizations, we wish to avoid explicit SYNC packet, could piggyback sync information in broadcast packets
  - Sender: Reuse the address field to piggyback schedule information
  - Receiver: Extract sync information and perform as scheduled.
  - It’s for free!! And no affect to upper layer operation

# Port to IEEE 802.15.4 Radio

- SCP-MAC to run on the 802.15.4 radios found on the MicaZ hardware.
- Challenges:
  - CC2420 is a packet-level radio, and the microcontroller cannot get byte-level access.
  - Potentially affects the accuracy of time synchronization.
  - CC2420 limits the preamble length to 16 bytes with a default length of 4 bytes.

# Port to IEEE 802.15.4 Radio

To implement long preambles:

- Sequentially send multiple wakeup packets back to back.
- Ensure that a receiver does not miss the “preamble” even if its channel polling time falls in a gap between the wakeup packets.

*To reduce these gaps: pre-load the wakeup packet into the radio buffer before carrier sense, then resend the same packet from the buffer multiple times to make up a long preamble*

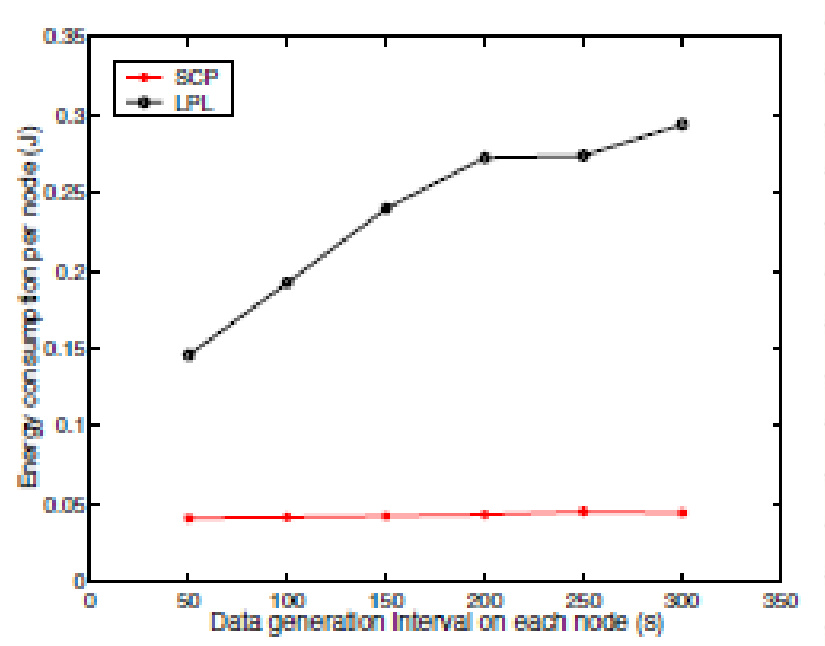
# Outline

- Introduction
- Design of SCP-MAC
- Lower Bound of Energy Performance with Periodic Traffic
- Protocol Implementation
- Experimental Evaluation
- Related Work

# Optimal setup with periodic traffic

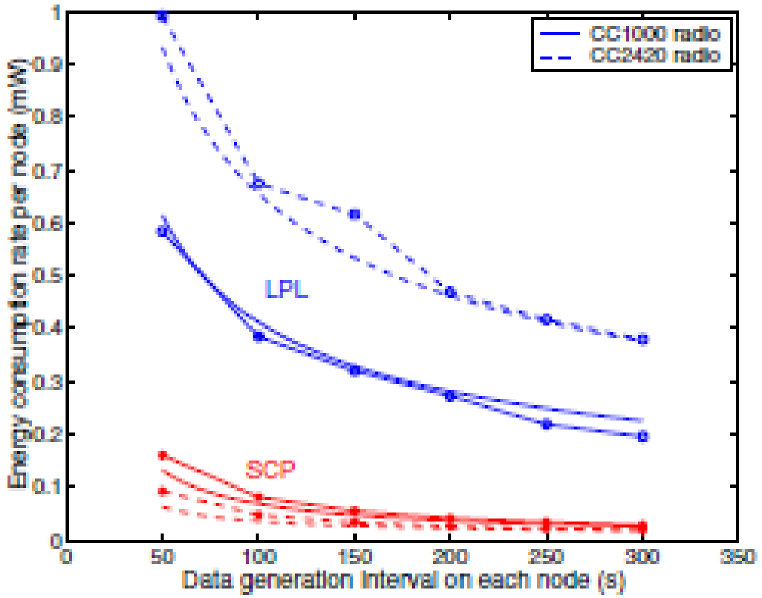
- Comparing the energy performance of SCP and LPL under optimal configuration with completely periodic.
- Adaptive channel polling and overhearing avoidance are turned off.
- MAC parameters vary based on network size and data rate.
- Placing 10 nodes in a single hop network.
- Each node periodically generates a 40B message (not including preamble).
- Each node’s message generation interval from 50–300s.
- Run each experiment for 5 message periods, generating 50 total messages over each experiment.

# Optimal setup with periodic traffic



Mean energy consumption (J) for each node as traffic rate varies (assuming optimal configuration and periodic traffic)

# Optimal setup with periodic traffic



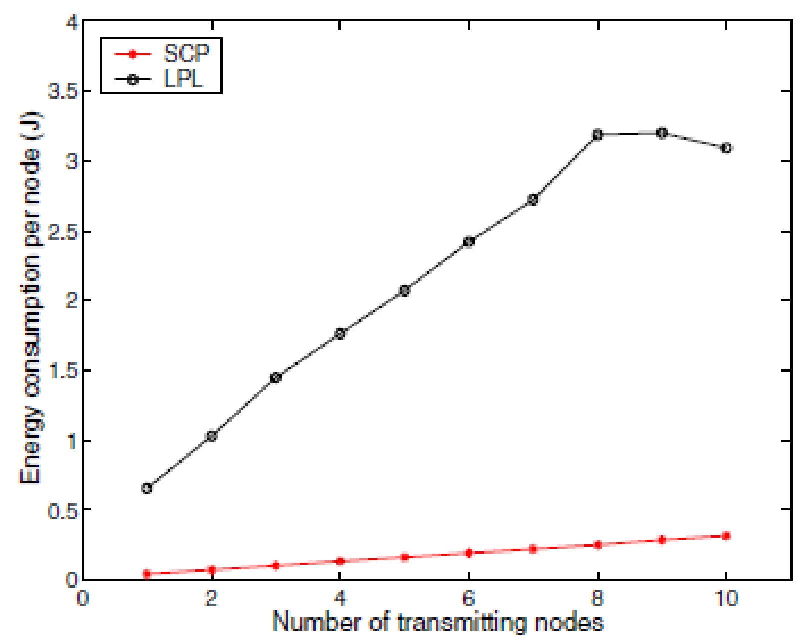
Energy consumption of LPL increases on faster radio, while SCP decreases energy Consumption.

Mean energy consumption rate (J/s or W) for each node as traffic rate varies. The radios are the CC1000 (solid lines) and CC2420 (dashed)

# Performance with unanticipated traffic

- In many applications the traffic load is less predictable (fire detection in forests).
- Tuning LPL and SCP for a 0.3% duty cycle, polling every second.
- All other parameters match the prior experiment.
- Each node generates 20 100B long messages

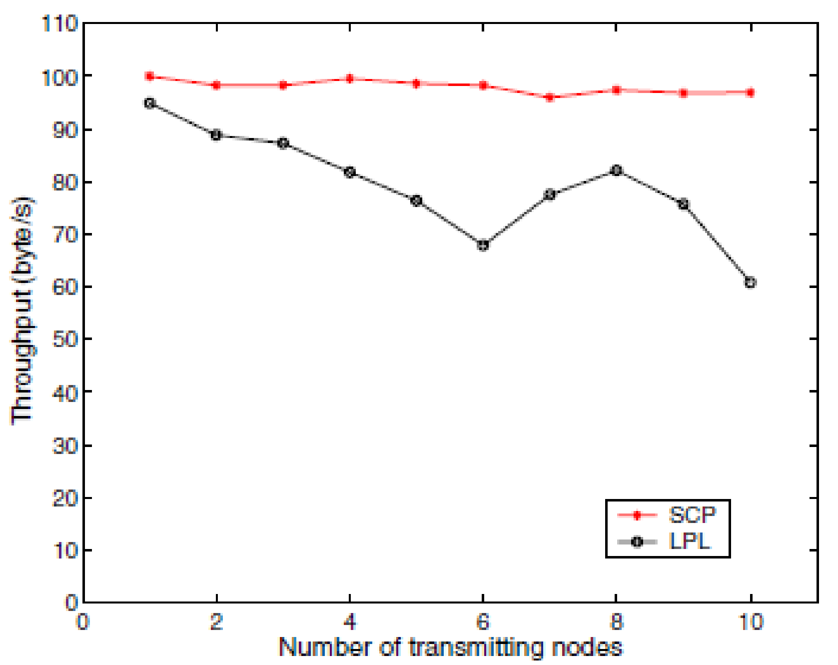
# Performance with unanticipated traffic



Energy consumptions on heavy traffic load with very low duty cycle configurations

LPL consumes about 8 times more energy than SCP to transmit equal amount of data. This is due to long preambles of LPL.

# Performance with unanticipated traffic



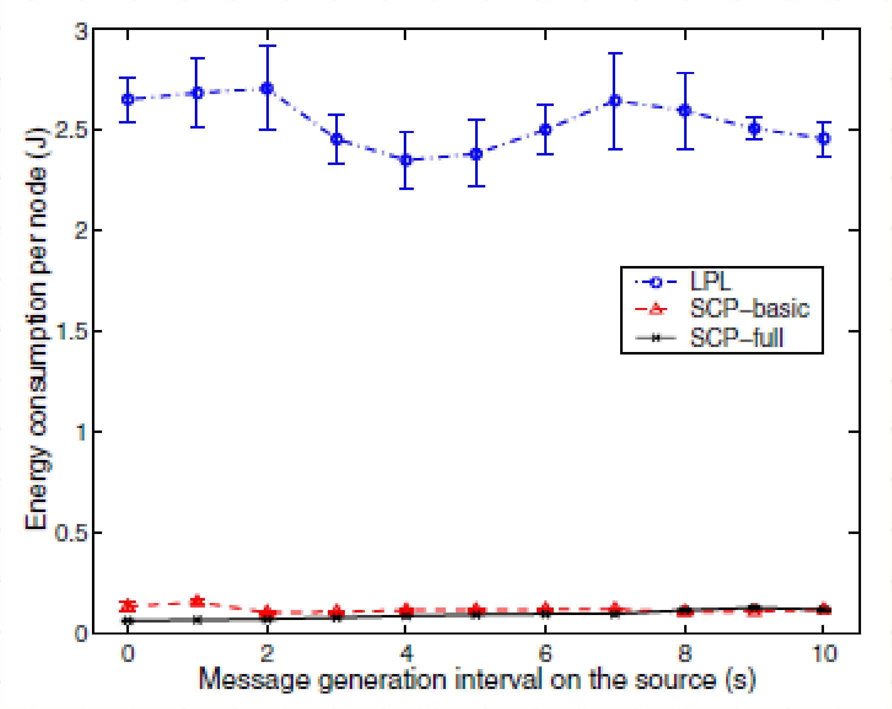
Throughput on heavy traffic load with very low duty cycle configuration

More effective two-phase contention

# Performance in a Multi-hop Network

- 9-hop linear network with 10 nodes.
- Adaptive channel polling is designed to reduce latency.
- All packets are sent as unicast without RTS/CTS.
- Acknowledgments with up to three retries.

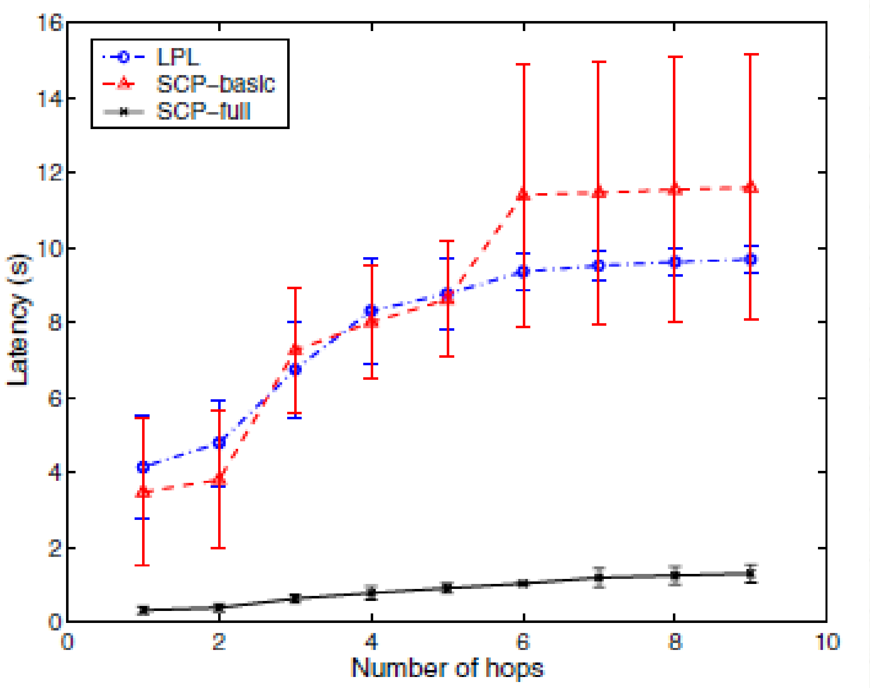
# Performance in a Multi-hop Network



Mean energy consumption per node for multi-hop experiments (20 packets over 9 hops)

SCP-basic: SCP without adaptive channel polling.
SCP-full: SCP with adaptive channel polling.
LPL consumes 20-40 times more energy
Than SCP full.
Reason: Long preambles

# Performance in a Multi-hop Network



Mean packet latency over 9 hops at the heaviest load

SCP-basic and LPL have comparable latency.
Power of adaptive polling!

# Outline

- Introduction
- Design of SCP-MAC
- Lower Bound of Energy Performance with Periodic Traffic
- Protocol Implementation
- Experimental Evaluation
- Related Work

# Related Work

- Power-save mode in IEEE 802.11 synchronizes wakeup times of nodes in a single-hop network.
- S-MAC developed a fully distributed algorithm to synchronize the wakeup schedules of nodes in a multi-hop network.
- T-MAC improves S-MAC by reducing the wakeup duration controlled by an adaptive timer.
- WiseMAC can reduce the preamble length after an initial unicast packet with a long preamble.

# Related Work

TDMA, second class of MAC protocols.

- LEACH and BMA.
- LMAC and TRAMA.
- ZMAC, proposed a hybrid protocol to combine TDMA with CSMA.

# Questions?