CPSL
Cyber-Physical Systems Laboratory

Self-Adapting MAC Layer for Wireless Sensor Networks

Mo Sha, Rahav Dor, Gregory Hackmann, Chenyang Lu
Cyber-Physical Systems Laboratory
Washington University in St. Louis

Tae-Suk Kim, Taerim Park
Samsung Advanced Institute of Technology
Samsung Electronics

Washington University in St. Louis

Motivation

- Numerous MAC protocols exist in the literature
  - CSMA/CA vs. TDMA
  - Sender-initiated vs. receiver-initiated

- **None** remains optimal under
  - changes in ambient wireless environment;
  - changes in network traffic;
  - changes in QoS requirement in multiple dimensions.

Wireless Health Application

Long battery lifetime

Resilient to interference

Low latency

⋮

Problem Formulation

- Given current network load (inter-packet interval)
- Given current wireless noise level (signal strength)
- Given user's preferred order of QoS attributes
  - Energy consumption, reliability, and latency

- Goal: select MAC protocol with optimal QoS in the specified order of the attributes

SAML: Self-Adapting MAC Layer

One-protocol-fit-all MAC

↓

Self-Adapting MAC Layer (SAML)

→ Provision multiple MACs in an efficient manner.

→ Select and activate the optimal protocol under the current load, condition and requirements.

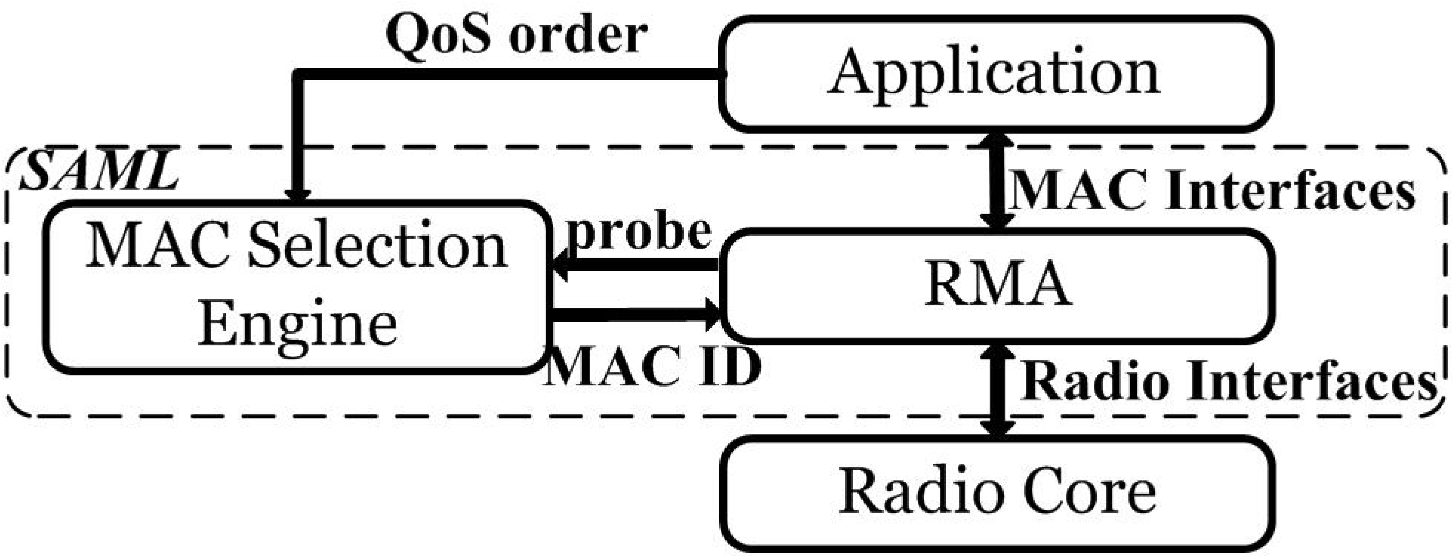
Network Model

CPSL
Cyber-Physical Systems Laboratory

- Star network
  - Hub works as the master
  - Sensors works as slaves
  - Communicate via 802.15.4 radio
- Hub: a smart phone
  - Hold multiple MAC protocols
  - Select MAC protocols
  - Coordinate network-wide MAC switch

- Sensors
  - Hold multiple MAC protocols
  - Follow hub's MAC decision to switch protocols

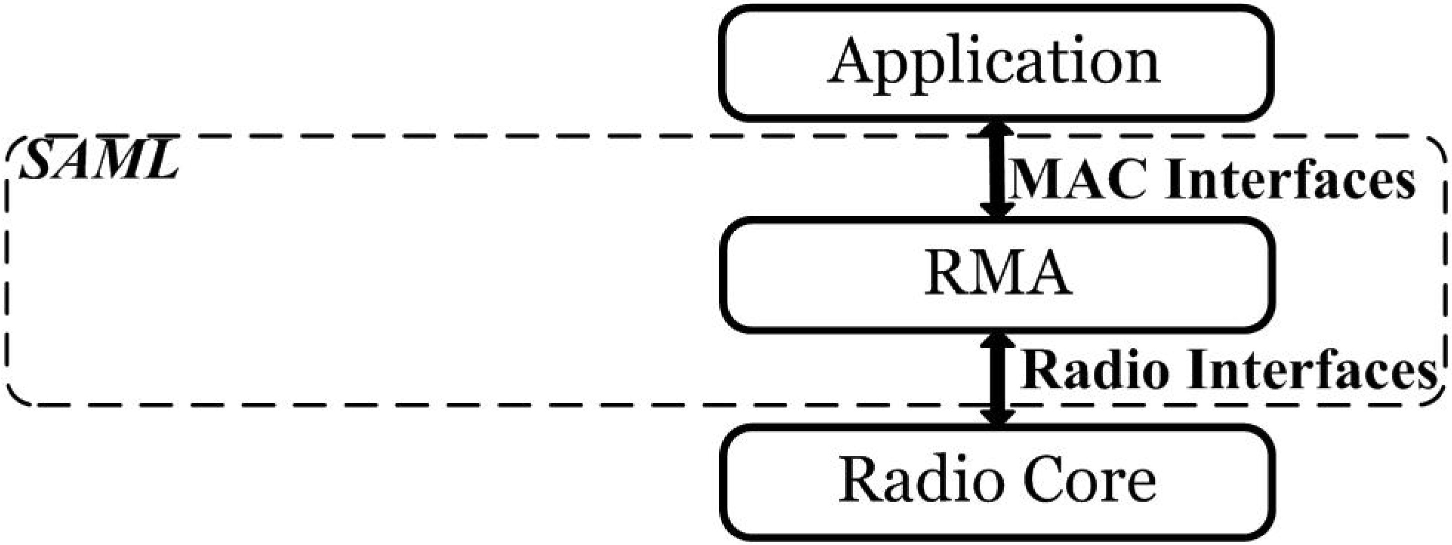
System Architecture: Hub



- **RMA (Reconfigurable MAC Architecture)** supports dynamic switching among different MACs
- **MAC Selection Engine** selects MAC protocols based on application's preference

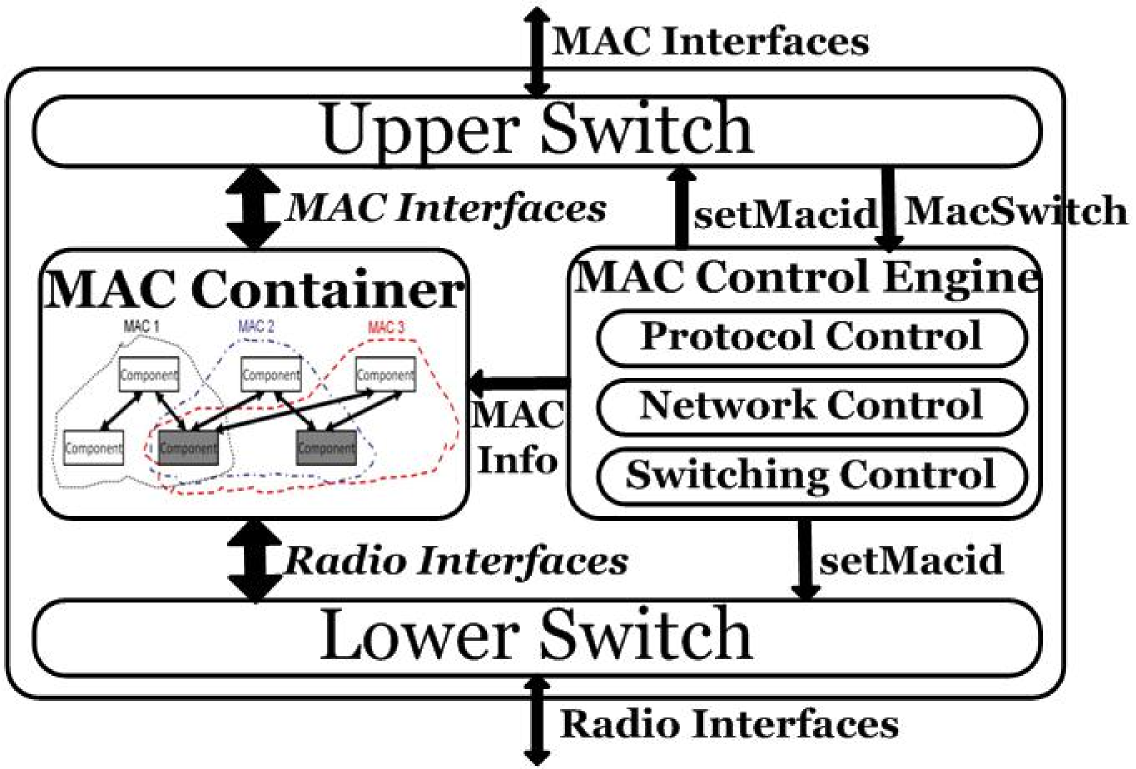
System Architecture: Sensors

Cyber-Physical Systems Laboratory



- **RMA (Reconfigurable MAC Architecture)** supports dynamic switching among different MACs

RMA



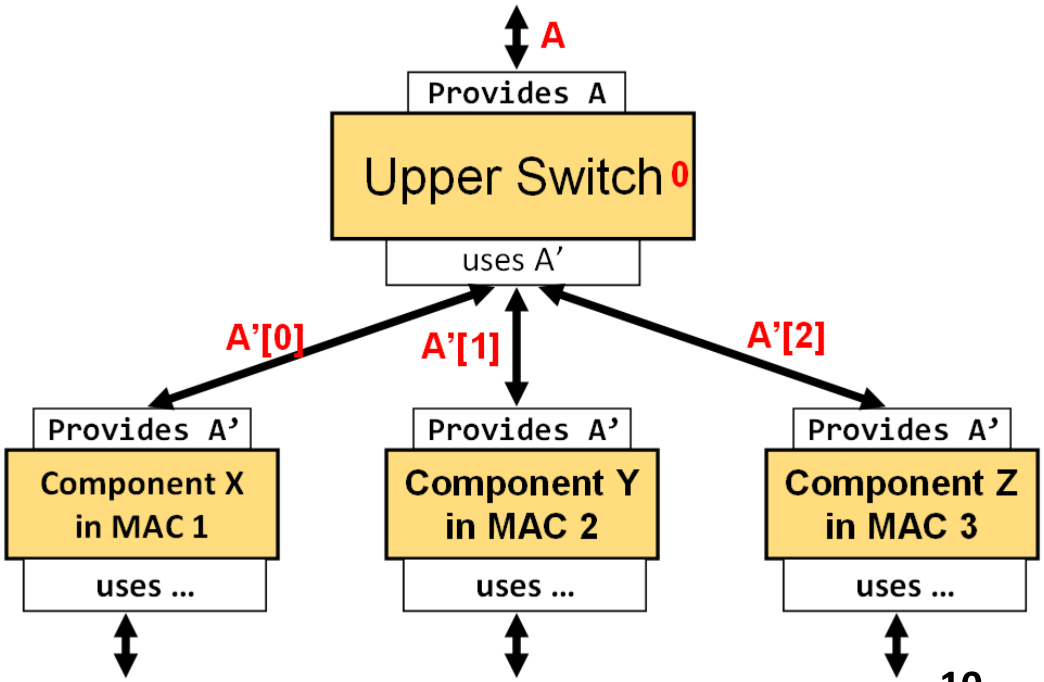
- **MAC Container** stores the MACs available at runtime
- **Upper Switch** & **Lower Switch**: unified interfaces to applications & radio core
- **MAC Control Engine** controls the ID of the active MAC, maintains the neighborhood table, and manages protocol

MAC Container

- Designed to hold reusable MAC components
- Re-wired on the fly to construct various MACs
- Realization in TinyOS
  - Build on components from MAC Layer Architecture (MLA) developed by CPSL [SenSys'07]
  - TinyOS compiler only creates one instance for each component
  - Add wrapper to shared component to avoid conflicts
  - Wrapper stores ID of the current active MAC protocol

Switches

- Provide uniform interfaces to the layer above and below the MAC layer
  - Upper switch: start/stop MAC, CCA/backoff control, send, receive
  - Use a select signal to determine which MAC is going to respond
- Realized in TinyOS/nesC via parameterized wiring

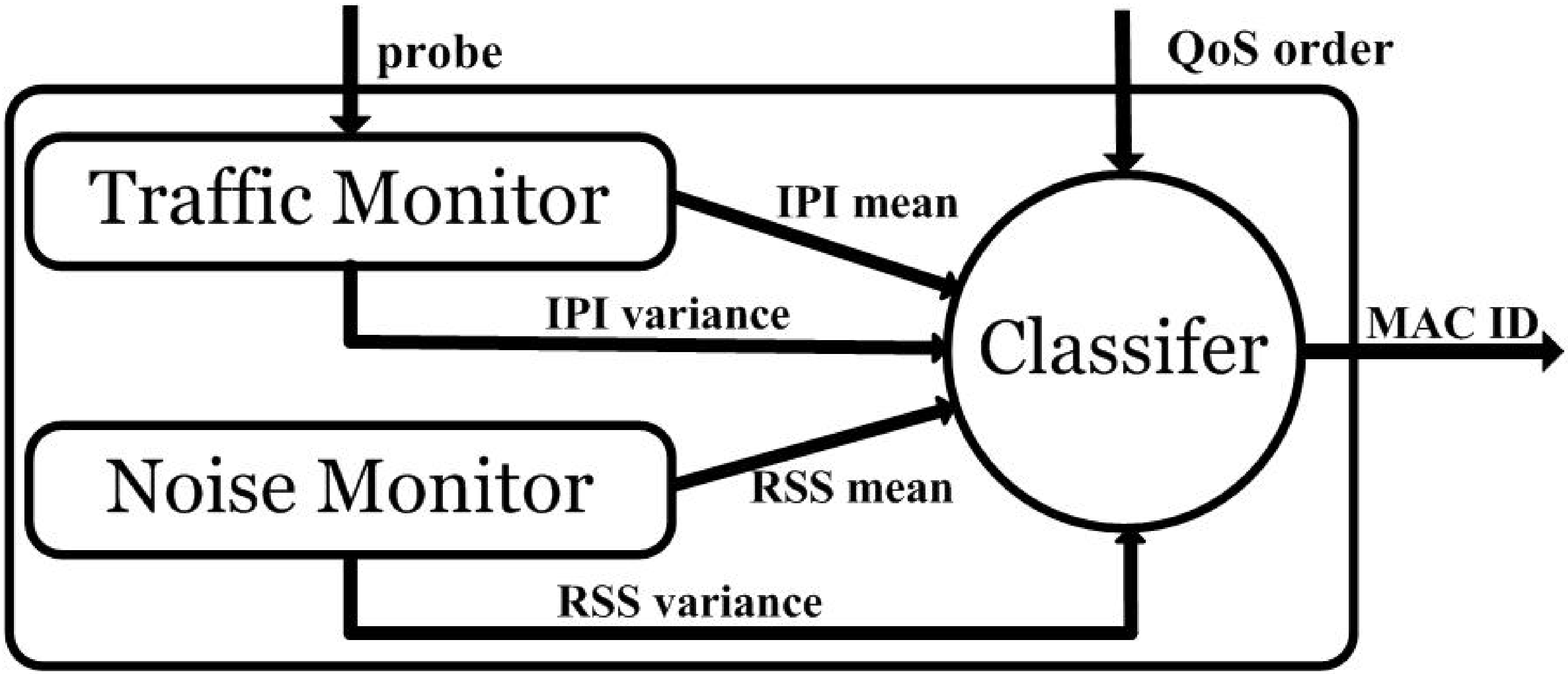


MAC Control Engine

CPSL
Cyber-Physical Systems Laboratory

- Protocol Control
  - Responsible for synchronizing active MAC ID in all wrappers/switches within a node
  - Update active MAC IDs during switch atomically
- Network Control
  - Manage nodes join/leave network
  - Hub: periodically broadcast current active MAC ID
  - New node: run baseline-MAC to snoop broadcasted ID
- Switching Control
  - Reliably notify all network devices when switching MAC

MAC Selection Engine



- **Traffic Monitor** keeps track of mean/variance of Inter-Packet Interval (IPI)
- **Noise Monitor** measures the external interference level in the environment
- **Classifier** determines the best MAC according to the current REL order

Decision Tree Classifier

- Why decision tree classifier?
  - Limited, discrete choices to make
  - Fast at run-time
  - Consumes small memory footprint

- Decision tree training
  - Run offline experiments that vary the features while recording the operating characteristics and the MAC protocol in use.
  - Characteristics: reliability, energy consumption, and latency
  - Features: QoS order, mean and variance of Inter-packet Interval, mean and variance of the RSSI

Implementations

- Implemented in TinyOS 2.1.1 on TelosB
- Select five MACs as examples
  - BoX-MAC, pure TDMA, RI-MAC, adaptive TDMA, ZigBee-like
- Build three prototypes of RMA
  - CSMA/TDMA: BoX-MAC + pure TDMA
  - SI/RIMAC: BoX-MAC + RI-MAC
  - 5-MAC prototype
- Decision tree classifier
  - Use Weka with C4.5 algorithm
  - Offline experiments for MAC comparisons: 4624 training examples

Memory Footprint

CPSL Cyber-Physical Systems Laboratory

RMA CSMA/TDMA adds 11% ROM and 4% RAM to a single MAC

	ROM (bytes)	RAM (bytes)
BoX-MAC	25308	1114
pure TDMA	25362	1202
RI-MAC	25132	1268
adaptive TDMA	25418	1126
ZigBee MAC	27168	1272
RMA CSMA/TDMA	28016	1254
RMA SI/RI-MAC	27752	1896
RMA 5-MAC	29990	1968

ROM and RAM usage for each single MAC and RMA prototype

Memory Footprint

CPSL
Cyber-Physical Systems Laboratory

RMA 5-MAC adds 10% ROM and 55% RAM to a single MAC

	ROM (bytes)	RAM (bytes)
BoX-MAC	25308	1114
pure TDMA	25362	1202
RI-MAC	25132	1268
adaptive TDMA	25418	1126
ZigBee MAC	27168	1272
RMA CSMA/TDMA	28016	1254
RMA SI/RI-MAC	27752	1896
RMA 5-MAC	29990	1968

ROM and RAM usage for each single MAC and RMA prototype

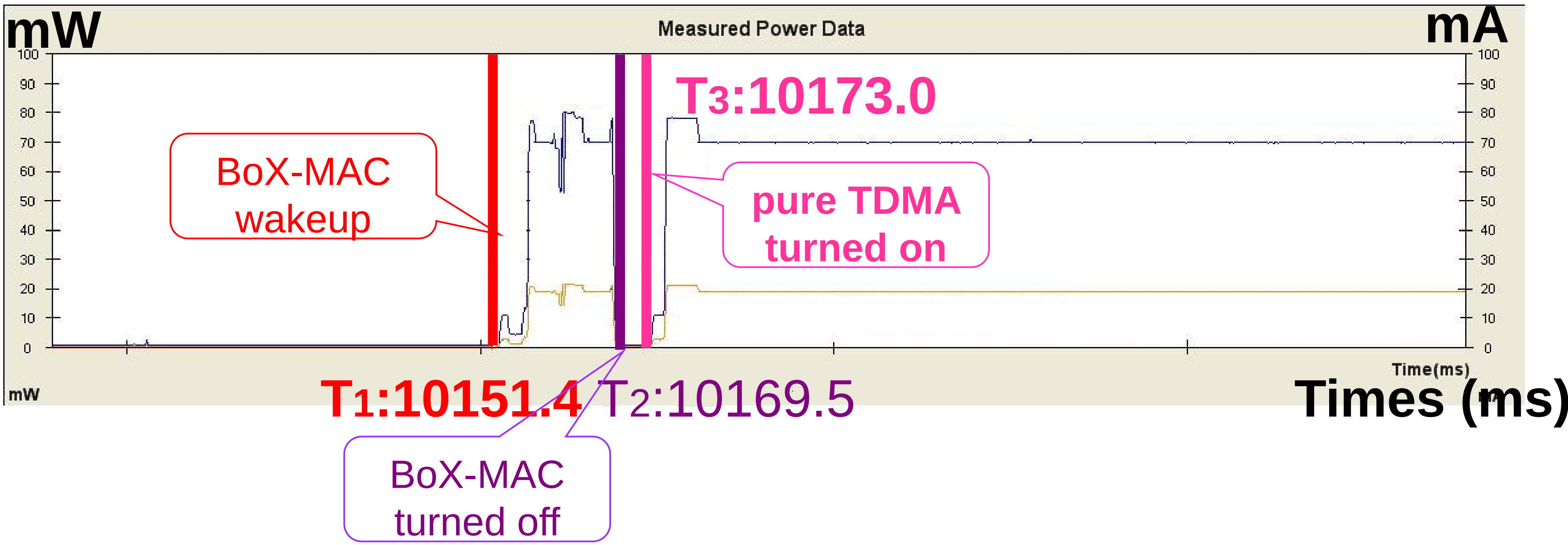
Benefit from MAC components reuse in SAML

Micro-benchmark Experiments

CPSL Cyber-Physical Systems Laboratory

The switching process

- takes **3.5** ms
- consumes **2.9** *u*J of energy



Current and time measured by a power meter from Monsoon Solutions during switch from BoX-MAC to pure TDMA.

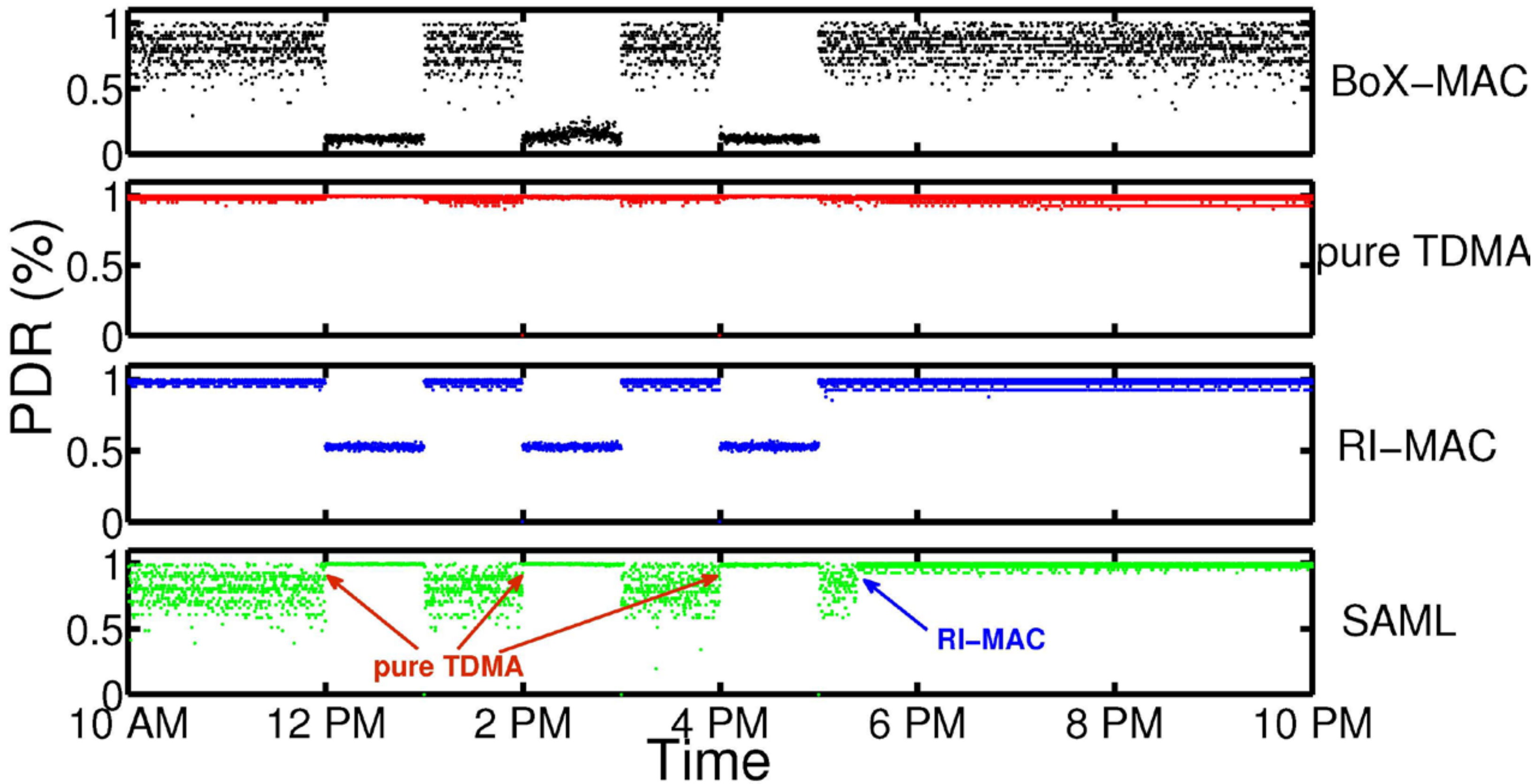
Case Study

- Emulating a wireless health application
  - One sensor on wrist, one sensor on chest, and one hub in pocket
  - Sensors generate packets at 2 pkt/s at regular and 20 pkt/s for an hour after detecting an abnormal event
  - QoS order (regular): Energy efficiency>Reliability>Latency
  - QoS order (abnormal): Reliability>Latency>Energy efficiency

- Experimental setup
  - 1st: BoX-MAC, 2nd: pure TDMA, 3rd:RI-MAC, 4th: SAML
  - 12-hour measurement per day (10am-10pm)
  - Volunteer follows the same schedule for daily activities
  - 3 abnormal event generated daily at 12pm, 2pm, and 4pm

Case Study

- SAML meets reliability requirement (>99%)
- and saves 32% of energy (1451.7 J)



PDR of BoX-MAC, pure TDMA, RI-MAC, and SAML during 12 hours

Related Works

- MLA [SenSys’07]
  - Library of reusable components for MAC implementation
- Distributing new TinyOS image or fragments of code
  - Deluge [SenSys’04], Task-Cruncher [IPSN’10]
  - High communication & runtime overhead
- Hybrid MACs
  - Z-MAC [SenSys’05], Funneling-MAC [SenSys’06], IDEA [MobiSys.10]
  - Monolithic implementation with fixed set of features
- Runtime parameter adaption
  - pTunes [IPSN’12]
- Adaptive MACs in IEEE 802.11 networks
  - MULTIMAC [JSAC’10]
  - Optimize for single dimension and high static overhead

Conclusion

- A fixed MAC protocol cannot meet varying requirements in dynamic environments
  - Challenge with the convergence of mobile phones and sensors
- SAML: Self-Adapting MAC Layer for WSNs
  - Reconfigurable MAC Architecture (RMA): switch MAC protocols on the fly
  - A learning-based MAC Selection Engine: selects protocol most suitable for the current condition and requirements
  - Implemented in TinyOS 2.1.1 on TelosB
- SAML effectively adapts MAC layer protocol to meet varying application requirements in dynamic environments