

# Performance Evaluation of Receiver Based MAC Using Configurable Framework in WSNs

Rodrigo Vieira Steiner\*, Mohammad Reza Akhavan†, Antônio A. Fröhlich\*, A. Hamid Aghvami†  
\*Software/Hardware Integration Lab, Federal University of Santa Catarina, Florianópolis, SC, Brazil  
{rodrigo, guto}@lisha.ufsc.br

†Institute of Telecommunications, King's College London, London, UK  
{reza.akhavan, hamid.aghvami}@kcl.ac.uk

**Abstract**—Receiver-based MAC (RB-MAC) is a preamble-sampling MAC protocol for WSNs in which a receiver node is dynamically elected, among potential neighbors of the sender node, based on current channel conditions. In this paper, we evaluate the performance of RB-MAC and compare it with a sender-based preamble-sampling MAC protocol by using analytical methods, and implementation in real sensor nodes. We have used Configurable MAC (C-MAC), which is a framework to develop different MAC protocols in WSNs. This framework is realized as a component architecture that can produce application-specific communication protocols. The experimental results presented in the paper corroborate with the analytical and numerical results showing how RB-MAC outperforms sender-based MAC protocols in terms of transmission delay, and energy consumption.

**Keywords**—wireless sensor networks; MAC; configurability; energy efficiency; delay.

## I. INTRODUCTION

Solutions for WSN challenges are highly application-dependent and scenario-based. In this work, we consider a scenario with poor channel conditions—with high data error rate—in which low-power and low-cost sensor nodes, with limited transmission range, generate low data traffic. These nodes are distributed over a vast area and report gathered information towards one or more sinks.

According to the IETF's ROLL working group [1], a lossy link is not just a link with a high Bit Error Rate (BER). In lossy WSNs, extremely high and frequent packet losses may occur due to different reasons like interference [2], path loss, shadowing, and multipath fading which cause the reception power of a signal to vary significantly over time and space. Wireless links may become unavailable due to network dynamicity. For example: sensors nodes may run out of energy; be removed from the network due to hardware failure; or new nodes may be injected into the network changing the connectivity graph. Sensor nodes may also move to new positions out of range of their previous neighbors, and new communication links can be created.

It is a challenge to provide both low latency and reliable data delivery in such networks, due to limited energy resources of nodes and dynamicity of network. Significant energy savings can be achieved by keeping sensor nodes in sleep mode for longer durations, as idle listening is the main source of energy dissipation in WSNs. However, this reduces the availability

of the nodes for communication. Therefore, a mechanism is required to wake up the nodes whenever a packet needs to be forwarded. Preamble sampling MAC protocols [3] have been designed to tackle idle listening and support sleep/wakeup modes without synchronization overheads.

To provide a satisfactory level of data delivery in the network, retransmissions have been used in many MAC protocols. However, retransmissions are not energy-efficient and cause delay in transmissions. In [4], Akhavan et al. proposed receiver-based MAC (RB-MAC), in which a receiver node is dynamically elected among potential neighbour nodes, based on current channel conditions and other predefined objective function (e.g. distance to the sink, residual energy, or any combinations of these parameters). It is resilient to lossy links due to multiple receivers, yet more energy efficient as it reduces the number of retransmissions.

In this paper, we investigate the performance of the RB-MAC protocol in terms of transmission delay, and energy-efficiency by using analytical, simulation and real implementation. To implement RB-MAC we have used the Configurable MAC (C-MAC) [5] framework, which allowed us to evaluate and compare the protocol performance executing in real sensor nodes.

The rest of the paper is organized as follow. Section II describes related work, and the RB-MAC approach. Section III presents the analytical performance evaluation of RB-MAC together with simulation results. The Configurable MAC framework is explained in Section IV. Experimental results using real sensor nodes are presented in Section V. Finally, Section VI concludes the paper.

## II. PREAMBLE SAMPLING MAC PROTOCOL (PS-MAC)

In Preamble Sampling approach [3], generally known as low asynchronous power listening (LPL), each node selects its sleep/wakeup schedules independently of other nodes. Sensor nodes spend most of their time in sleep mode and wake up for a short duration called clear-channel-assessment (CCA) every checking interval (CI) to check whether there is an ongoing transmission on the channel. To avoid deafness, each data packet is preceded by a long preamble, with the same length as CI, to make sure that all receivers detect the preamble and then get the data frame. By tuning CI and CCA, average duty-cycles

of below 1% can be achieved without any need for scheduling or synchronization. Some MAC protocols that utilized this idea are: B-MAC [6], wiseMAC [7], X-MAC [8], MFP [9], CSMA-MPS [10], STEM [11], 1-hopMAC [12], and RB-MAC [4]. In B-MAC receiver nodes wake up every CI period to check for channel activity. If a preamble is detected, the radio is kept ON until the reception of the data. In MFP, the authors have proposed to replace the continuous preamble by a series of small frames, called micro-frames, to solve the problem of receiving irrelevant frames. Each micro-frame contains the destination address and a sequence number which serves as a countdown to the actual data transmission, allowing a receiver to switch off its radio during micro-frame transmissions and switch it on again just before the data frame transmission. 1-hopMAC takes this idea further by allowing a node to dynamically discover its neighborhood during each transmission. A sender node listens between sending the preamble and the data, leaving time for its neighbors to announce themselves.

Based on the forwarding mechanisms utilized, we categorize MAC protocols into two groups:

#### A. Sender-based MAC

The majority of MAC protocols in WSNs are sender-based. In sender-based MAC (SB-MAC) protocols, a sensor node has a table containing the addresses of its neighbors. The sender node selects the best receiver among the neighbors' from this table and transmits the packet after including the destination address in the packet header. It selects the appropriate receiver based on some predefined criteria (e.g. distance to the sink, energy level of the receivers, link quality or any combination of these). Creating and maintaining an address table is not energy efficient in lossy networks. In addition, many retransmissions (up to seven times in IEEE 802.15.4 [13]) may occur due to dynamicity of the network, which consume considerable amount of energy and cause delay. In this paper, we will use B-MAC as a representative example of a sender-based MAC since it is a classic preamble sampling MAC and is highly used for comparison with other protocols in the literature.

#### B. Receiver-based MAC

In receiver-based MAC, a sender node transmit its data without defining a particular node as a receiver. In RB-MAC [4], a representative example of this group, when a transmitter node has a packet to send, it sends the preamble followed by the payload. Similar to MFP, the preamble consists of micro-frames that contain useful information such as: countdown to data transmission, sender distance to sink, and payload sequence number. All neighboring nodes within communication range of the sender sense the channel every CI interval and obtain a micro-frame. Upon obtaining the information extracted from micro-frames, all eligible receivers (nodes closer to sink) go back to sleep and wake up to receive the data. If a receiver gets the data with error, it just drops the packet. This eliminates the need for retransmissions. Nodes that correctly received the data start a back-off timer (e.g. timer set based on remaining energy, etc.). When the timer elapse

a node performs CCA. The node with the shortest back-off timer will sense no channel activity, and will then proceed to transmit the preamble. Other nodes will detect the winners preamble containing the same sequence number of the same data. Therefore; they will go back to sleep mode and drop the data. Figure 1 illustrates the timeline of RB-MAC.

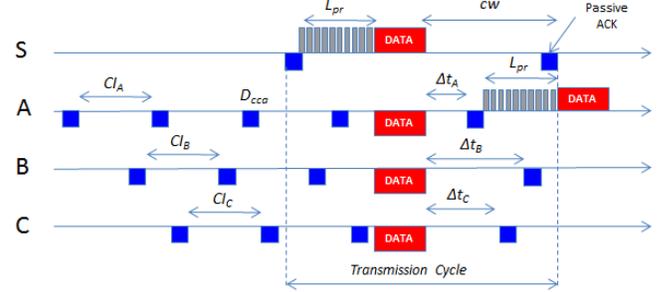


Figure 1: Timeline of RB-MAC.

### III. ANALYTICAL PERFORMANCE EVALUATION

In this section, we summarise the performance of RB-MAC in terms of total energy consumption, and delay as introduced in [4]. Table I presents used notations.

Table I: Notations and symbols.

$p$	probability of successfully receiving data
$n$	number of candidates in communication range of a node
$D_{DATA}$	transmission time of one data packet
$D_{cca}$	the carrier sense duration
$D_{CW}$	the contention window duration
$E_{REQ}$	energy drained in transmitting preamble
$P_{rx}$	power drained in receive mode
$P_{tx}$	power drained in transmit mode
$P_{sleep}$	power drained in sleep mode
$P_{listen}$	power drained in listening mode

#### A. Lower Energy Consumption

We use  $p$  as the probability that a receiver successfully receives the data from source without error. When using a sender-based MAC, the probability of retransmission is equal to  $1/p$ . Conversely, RB-MAC forwards the data to all of its neighbors, which drops the retransmission probability to [4]:

$$P_{Ret} = \frac{1}{1 - (1 - p)^n} \quad (1)$$

The total energy consumed in a successful transmission would be [4]:

$$E_{tot} = \frac{E_{basic}}{1 - (1 - p)^n} \quad (2)$$

where  $n$  is the number of receivers and  $E_{basic}$  is the energy cost for a single transmission. Energy drained in transmission, reception, and total energy can be calculated as follows [4]:

$$E_{Tx} = D_{cca}P_{listen} + E_{REQ} + D_{DATA}P_{Tx} + (D_{CW} - D_{cca})P_{sleep} + D_{cca}P_{Rx} \quad (3)$$

#### IV. C-MAC

$$E_{Rx} = (D_{REQ} + D_{CW} - 2D_{cca})P_{sleep} + (D_{DATA} + D_{cca})P_{Rx} + D_{cca}P_{listen} \quad (4)$$

$$E_{Total} = E_{Tx} + nE_{Rx} \quad (5)$$

Figure 2 presents the simulation results for energy consumption by a sender and its receivers for one transmitted packet, including energy drained in retransmissions. It is possible to notice that SB-MAC consumes more energy as channel conditions become poor. RB-MAC trigger fewer retransmissions, due to its cooperative nature, and its energy consumption is not significantly impacted by lower link probabilities.

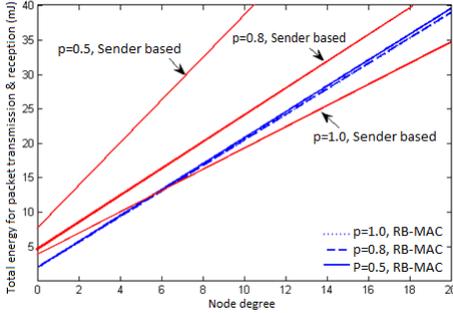


Figure 2: RB-MAC is more energy-efficient with lossy links in comparison to sender-based MAC [4].

#### B. Lower Delay

Similar to energy performance, total delay can be obtained as follows [4]:

$$D_{tot} = \frac{D_{basic}}{1 - (1 - p)^n} \quad (6)$$

where  $D_{basic}$  is the delay for one transmission. Figure 3 presents the simulation delay results of RB-MAC and SB-MAC.

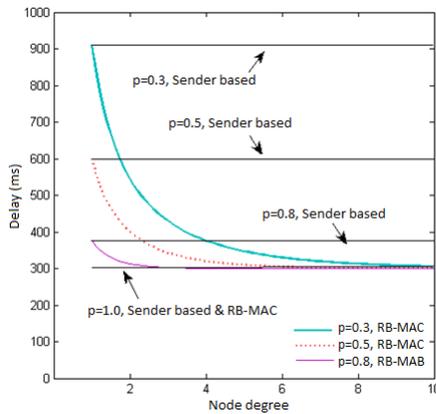


Figure 3: Delay performance in RB-MAC [4].

C-MAC is a framework of medium access control strategies that can be combined to produce application-specific protocols [5]. It enables application programmers to configure several communication parameters (e.g. synchronization, contention, error detection, acknowledgment, packing, etc) to adjust and create a protocol instance that fulfil the specific needs of their applications.

C-MAC is realized as a component architecture, illustrated in Figures 4, 5, and 6. Each activity in these diagrams is executed by a microcomponent which can have different implementations. These microcomponents alongside with the flow control can be combined to produce application-specific protocols. By using static metaprogramming techniques, microcomponents representing activities that do not make sense for a certain protocol can be completely removed. When an activity is removed, its inputs are forwarded to the activity targeted by its outputs, still maintaining the original flow semantics.

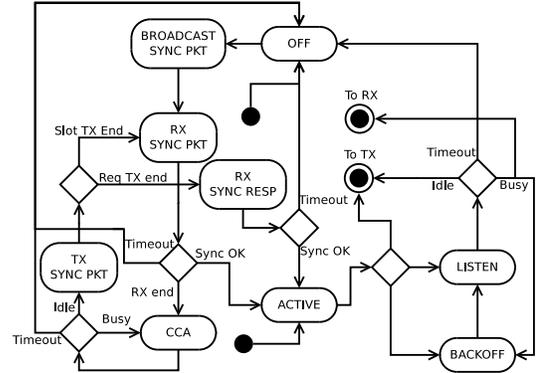


Figure 4: C-MAC Synchronization Activity Diagram [5].

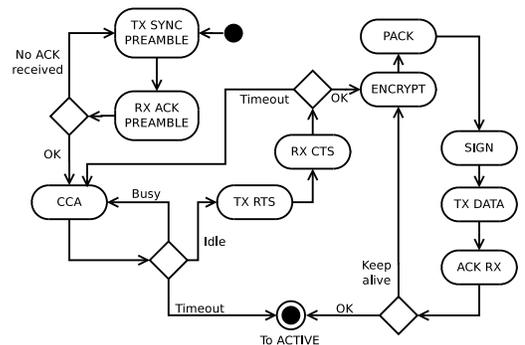


Figure 5: C-MAC Transmission Activity Diagram [5].

The use of static metaprogramming techniques (e.g. templates, inline functions, and inline assembly) ensures that configurability does not come at expense of performance or code size [5]. In this way, C-MAC's instances are fully

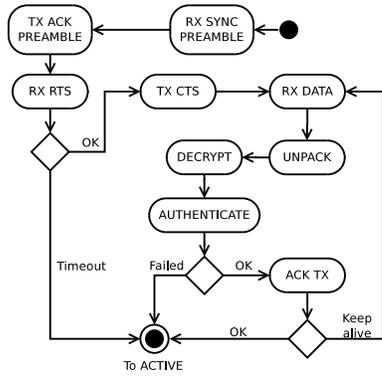


Figure 6: C-MAC Reception Activity Diagram [5].

customized at compile-time and yield extremely lean run-time MACs.

### A. Configuring C-MAC

C-MAC configuration is performed using the same concept of *traits* present in the C++ standard library [14]. Traits are parameterized classes whose static constant members describe the properties of a certain type [15]. Using the configuration file, shown in Figure 7, developers can choose which properties the protocol must possess. When a property is selected, its functionality is included in the protocol. This way it is possible to define the protocol behavior at compile time.

```

// ...
template <> struct Traits<CMAC<Radio_Wrapper>> : public Traits<void>
{
// ...
enum Synchronization {
    NO_SYNC = 0,
    PREAMBLE = 1,
    BEACON = 2,
};

static const int SYNC = PREAMBLE;

enum Preamble_Content {
    DUMMY = 0,
    TIME_TO_DATA_AND_RANK = 1,
    DESTINATION_ID = 2,
};

// ...
static const int PREAMBLE_CONTENT = TIME_TO_DATA_AND_RANK;
static const int PREAMBLE_LENGTH = 144;
static const int SLEEPING_PERIOD = 144;
// ...
static const bool forward_data_to_sink = true;
static const int RANK = 2;
// ...
};
// ...

```

Figure 7: Traits configuration file.

Figure 8 presents the C-MAC activity diagram for B-MAC and RB-MAC configurations. As can be seen, only states that are actually used are present. Therefore, there is no additional overhead to the generated code. Both B-MAC and RB-MAC use the same states. However, they use different microcomponents to handle preamble synchronization and to determine if a transmission was successful.

## V. EXPERIMENTAL RESULTS

In order to evaluate RB-MAC in real sensor nodes we chose to use the EPOSMote II, which is an easy to develop and low cost platform with an open hardware project [16]. Its hardware

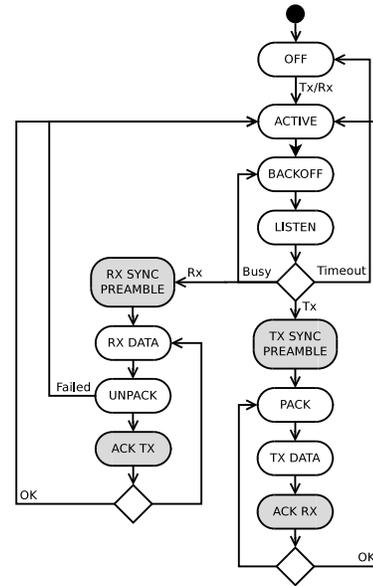


Figure 8: C-MAC Activity Diagram for B-MAC and RB-MAC. Highlighted states indicate which microcomponents have different implementation.

is designed as a layer architecture composed by a main module (responsible for processing, storage, and communication), a sensing module, and a power module. The model used in this research features a 32-bit ARM7 processor, 128kB of flash, 96kB of RAM, and an IEEE 802.15.4-compliant radio transceiver. Figure 9 shows the development kit which is slightly larger than a R\$1 coin. Table II shows a comparison between the EPOSMote II specification with other sensor nodes highly used in WSNs.



Figure 9: EPOSMote II SDK side-by-side with a R\$1 coin. On the left the sensing module. On the right the main module.

### A. Results

We have configured C-MAC to mimic both B-MAC and RB-MAC and the probability  $p$  of a node successfully receive a packet was ranged between 1, 0.8, 0.5, and 0.3. To simulate the channel conditions in this implementation we have used the pseudo random algorithm provided by the operating system. Each time a node receives a packet it generates a random number, inside an interval, and according with the result and the probability  $p$  under test the packet may be discarded.

Table II: Comparison of Mote Specifications [17], [18], [19], [20].

Mote	TelosB	Mica2	MicaZ	EPOSMote II
MCU	MSP430	ATmega128L	ATmega128L	ARM7TDMI-S
Processor Speed	8 MHz	8 MHz	8 MHz	26 MHz
Program Flash Memory	48 K	128 K	128 K	128 K
RAM	10 K	4 K	4 K	96 K
Processor Current Draw - active mode	1.8 mA	8 mA	8 mA	3.3 mA
Processor Current Draw - sleep mode	5.1 $\mu$ A	< 15 $\mu$ A	< 15 $\mu$ A	0.85 $\mu$ A
Radio	CC2420	CC1000	CC2420	Integrated transceiver
Radio frequency	2400 to 2483.5 MHz	315/433/915 MHz	2400 to 2483.5 MHz	2405 to 2480 MHz
Rx Current Draw	21.8 mA	10 mA	21.8 mA	22 mA
Tx Current Draw	19.5 mA	27 mA	19.5 mA	29 mA

Figure 10 illustrates the evaluation scenario. At first, one node is set as sender and another one as receiver. Both B-MAC and RB-MAC are evaluated for each probability  $p$ . When all probabilities are tested another receiver is added and this process repeats itself until the network possess a total of four receivers at the same time.

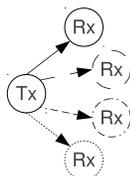


Figure 10: Evaluation scenario.

Both protocols were evaluated in terms of memory footprint, latency and energy consumption. For all experiments, we have used: the GNU GCC compiler for ARM, version 4.4.4; the ARM processor clock was set to 24 MHz; CCA duration of 1.43 ms; preamble duration of 143.8 ms; data packets size of 94 bytes; and 4.5 dBm of transmission power.

We have used the `arm-size` tool, from GNU Binutils version 2.20, to obtain the memory footprint of both protocols. Results are shown in Table III. As expected, RB-MAC presents a larger footprint since it uses more complex mechanisms. While B-MAC uses a dummy preamble content, RB-MAC uses helpful information allowing nodes to go back to sleep or even quit contention. Also, B-MAC uses simple acknowledgement packets as transmission confirmation, while RB-MAC uses a contention window mechanism to check if a transmitted packet is being properly forwarded.

Table III: Memory footprint (bytes).

Section	B-MAC	RB-MAC
.text	10780	11236
.data	38	38
.bss	5679	5679
TOTAL	16497	16953

In order to determine the necessary time and amount of energy required to transmit a packet for each protocol, we have used an oscilloscope as illustrated in Figure 11. A General

Purpose Input/Output (GPIO) pin corresponding to a LED output pin in EPOSMote II is connected to the oscilloscope and set to high before a transmission starts and reset to low right after the confirmation of reception.

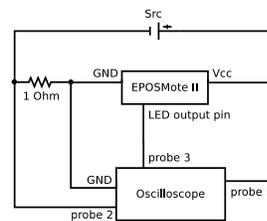


Figure 11: Schematic of the EPOSMote II connected to oscilloscope.

Figure 12 presents the necessary time to successfully transmit a packet, including retransmissions, varying the channel conditions for both protocols. It is possible to notice that B-MAC's performance deteriorates as the reception probability decreases and that it is not influenced by the amount of receiver nodes. This happens because, in B-MAC, the sender node sends the data to a specific receiver. Thus, the lower the probability of reception, the more retransmissions will be needed, thereby increasing the necessary time to successfully transmit a packet. As for the RB-MAC protocol, the sender does not have a specific receiver, and all neighboring nodes are potential receivers. The more neighbors, the greater the chances that a message will be successfully received, and therefore the number of retransmissions is reduced. When the probability of reception is equal to one, there is no need for retransmission, and both protocols have the same behavior.

The energy consumption to transmit a packet, including retransmissions, is illustrated in Figure 13. Since a transmission in B-MAC has a specific receiver, each other receiver in the network becomes synonymous of idle listening, in other words, wasted energy. Furthermore, the lower the probability of reception, the greater is the number of retransmissions, and the higher is the energy consumption. As for RB-MAC, more neighbors mean fewer retransmissions. Nevertheless, this does not always means less energy consumption. There is a relation between the probability of reception and the energy spent on unnecessary listening in which, after a certain point,

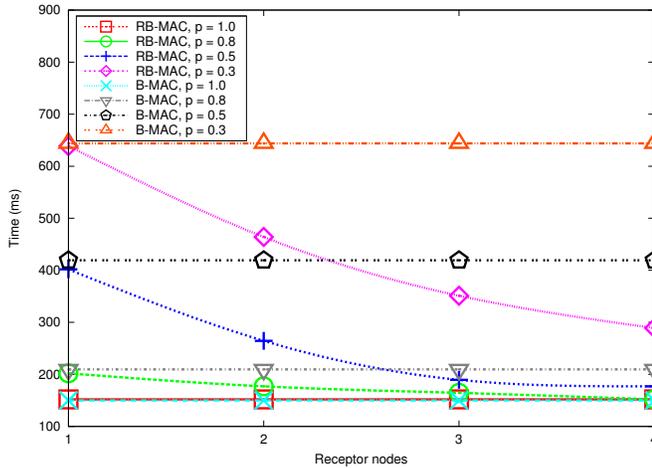


Figure 12: B-MAC and RB-MAC latency varying channel conditions.

adding a receptor does not reduce energy consumption. This happens because only one of the receivers has to get the data, so the energy spent by other receivers is wasted. Therefore, when the energy wasted by other receivers is higher than the energy saved by reducing retransmissions, then the total energy consumption increases. This relation can be observed in Figure 13. At the configuration RB-MAC with  $p = 0.5$ , the energy consumption reduces from one to two receivers, from two to three, but starts to increase with four receivers.

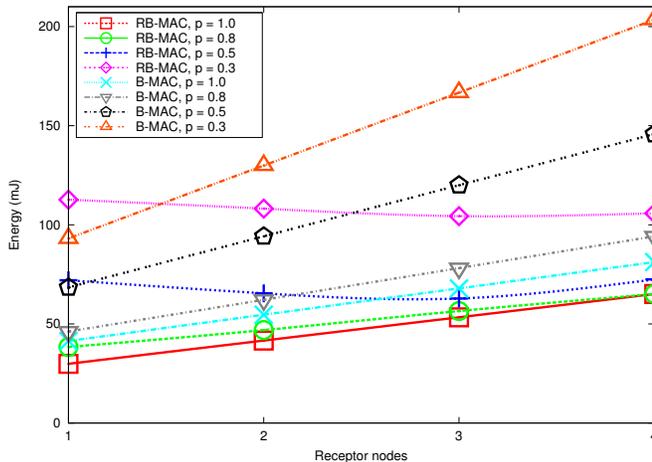


Figure 13: B-MAC and RB-MAC energy consumption varying channel conditions.

## VI. CONCLUSION

This paper presented the performance evaluation of RB-MAC under three different methods: analytical, simulation, and real implementation. Using the C-MAC framework, RB-MAC was compared with B-MAC executing in real sensor nodes. The results confirm that RB-MAC is significantly more resilient to lossy links when compared to sender-based MAC

protocols. As the number of network nodes increase, RB-MAC requires fewer retransmissions, consequently, reducing transmission delay and energy consumption. In our current work, we are studying the use of adaptive preamble lengths, adaptive CIs, and the behavior of the protocol in multi-hop transmissions.

## REFERENCES

- [1] T. Winter, P. Thubert, A. Brandt, J. Hui, R. Kelsey, P. Levis, K. Pister, R. Struik, J. Vasseur, and R. Alexander, "RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks," RFC 6550 (Proposed Standard), Mar. 2012. [Online]. Available: <http://www.ietf.org/rfc/rfc6550.txt>
- [2] J.-P. Vasseur and A. Dunkels, *Interconnecting Smart Objects with IP: The Next Internet*. San Francisco, USA: Morgan Kaufmann Publishers Inc., 2010.
- [3] A. El-Hoiydi, "Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks," in *IEEE ICC*, New York, USA, 2002, pp. 3418–3423.
- [4] M. Akhavan, T. Watteyne, and A. Aghvami, "Enhancing the performance of RPL using a Receiver-Based MAC protocol in lossy WSNs," in *IEEE ICT*, Ayia Napa, Cyprus, May 2011, pp. 191–194.
- [5] R. Steiner, T. Muck, and A. Frohlich, "C-MAC: a Configurable Medium Access Control Protocol for Sensor Networks," in *IEEE Sensors*, Hawaii, USA, Nov. 2010, pp. 845–848.
- [6] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *ACM SenSys*, New York, USA, 2004, pp. 95–107.
- [7] A. El-Hoiydi, J.-D. Decotignie, C. Enz, and E. Le Roux, "Poster abstract: wiseMAC, an ultra low power MAC protocol for the wiseNET wireless sensor network," in *ACM SenSys*, Los Angeles, USA, 2003, pp. 302–303.
- [8] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks," in *ACM SenSys*, Boulder, USA, 2006, pp. 307–320.
- [9] A. Bachir, D. Barthel, M. Heusse, and A. Duda, "Micro-Frame Preamble MAC for Multihop Wireless Sensor Networks," in *IEEE ICC*, Istanbul, Turkey, Jun. 2006, pp. 3365–3370.
- [10] S. Mahlke and M. Bock, "CSMA-MPS: a minimum preamble sampling MAC protocol for low power wireless sensor networks," in *IEEE International Workshop on Factory Communication Systems*, Vienna, Austria, Sep. 2004, pp. 73–80.
- [11] C. Schurgers, V. Tsiatsis, S. Ganerwal, and M. Srivastava, "Optimizing sensor networks in the energy-latency-density design space," *IEEE Trans. on Mobile Computing*, vol. 1, no. 1, pp. 70–80, Jan-Mar 2002.
- [12] T. Watteyne, A. Bachir, M. Dohler, D. Barthe, and I. Auge-Blum, "1-hopMAC: An Energy-Efficient MAC Protocol for Avoiding 1-hop Neighborhood Knowledge," in *IEEE SECON*, Sept. 2006, pp. 639–644.
- [13] "IEEE Standard for Information Technology - Telecommunications and Information Exchange Between Systems - Local and Metropolitan Area Networks Specific Requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs)," *IEEE Std 802.15.4*, pp. 1–670, 2003.
- [14] B. Stroustrup, *The C++ Programming Language*, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2000.
- [15] A. A. Frohlich, *Application-Oriented Operating Systems*. Sankt Augustin: GMD - Forschungszentrum Informationstechnik, 2001.
- [16] LISHA, "EPOS Project," 2008. [Online]. Available: <http://epos.lisha.ufsc.br>
- [17] Crossbow Technology, Inc., *TelosB Mote Platform*, Datasheet 6020-0094-01 Rev B. [Online]. Available: <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=252>
- [18] —, *Mica2 Wireless Measurement System*, ed., Datasheet 6020-0042-08 Rev A. [Online]. Available: <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=174>
- [19] —, *MicaZ Wireless Measurement System*, Datasheet 6020-0060-04 Rev A. [Online]. Available: <http://bullseye.xbow.com:81/Products/productdetails.aspx?sid=164>
- [20] Freescale Semiconductor, *MC1322x Advanced ZigBee - Compliant Platform-in-Package (PiP) for the 2.4 GHz IEEE 802.15.4 Standard*, Oct. 2010, Datasheet Rev. 1.3. [Online]. Available: [http://www.freescale.com/webapp/sps/site/prod\\_summary.jsp?code=MC13224V](http://www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC13224V)