

A Synchronous Engine for Wireless Sensor Networks

Fernando Royo, Teresa Olivares and Luis Orozco-Barbosa

Albacete Research Institute of Informatics

University of Castilla La Mancha

02071 Albacete, SPAIN

[froyo, teresa, lorozco]@dsi.uclm.es

Abstract. The development and deployment of robust and highly populated Wireless Sensor Networks require addressing a wide variety of challenges. Among them, the synchronization and energy management of the nodes composing the network have been identified as two challenges to overcome. In this paper, we follow a cross-layer approach by defining a synchronous engine built across the radio, MAC and routing layers. The design of our proposal has been based on the results of our previous research efforts dealing with various experimental platforms and experimental trials.

1 Introduction

Traditionally, an energy efficient wireless MAC protocol is a protocol that minimizes idle listening and overhearing [1]. In addition, as any other MAC protocol, it should keep to a minimum the number of collisions and the protocol overhead. Idle listening is a dominant factor on energy waste in most sensor network applications. The central approach to reducing energy lost to idle listening is to lower the radio duty cycle by turning the radio off part of the time. Duty cycle is the ratio between listen time and a full listen/sleep interval.

Radio duty cycling, where the radio is off by default but wakes up periodically to participate in potential network communication, has received significant attention in the literature. However, the duty-cycling benefits achieved in theory and simulations have often not translated to practice. This can be attributed mainly to the problem of time uncertainty between sensor nodes. If the sleep/wakeup schedules of nodes do not intersect, the communication can not take place. Note that each sensor node may have its own notion of time governed by its local clock. The approaches used by MAC protocols to address this problem of *time uncertainty* determine their energy consumption [8]. The lack of techniques to accurately estimate time uncertainty also impacts the ability to deploy long-lived sensor network applications.

Recent studies on MAC protocols for sensor networks observe that there is not clear trend indicating that medium access for sensor networks is converging towards a unique best solution [13]. Besides, many of the protocols being introduced in the literature have only been evaluated in simulation. In order to impact the market, a

Please use the following format when citing this chapter:

Royo, F., Olivares, T., Orozco-Barbosa, L., 2007, in IFIP International Federation for Information Processing, Volume 248, Wireless Sensor and Actor Networks, eds. L. Orozco-Barbosa, Olivares, T., Casado, R., Bermudez, A., (Boston: Springer), pp. 107-118.

protocol not only must perform well in simulation; it must also integrate well within the implementations of wireless sensor network protocol architectures

This paper presents SA-MAC (Synchronous after Awake MAC), a new MAC protocol for wireless sensor networks. Our objective is to address some of the major issues currently present in this type of networks. We are particular interested on those issues related to the lower layers, such as energy management, overhearing, packet overhead or idle listening. Furthermore, the synchronization utilized in the protocol denotes a cross layer view, fusing some characteristics from the network layer over the MAC layer.

This article presents this new and robust protocol, as a component of a complete network architecture and coexisting with other important protocols. The remainder of this article is structured as follows. Section 2 reviews the state-of-the art on MAC protocols for sensor networks. Section 3 shows our previous work with sensor networks, implementations and results. Section 4 describes our proposal. Section 5 concludes the paper and outlines our future work.

2 Background

Medium access for sensor networks is a very active research area. The B-MAC protocol has been an important input to the MAC protocols world for wireless sensor network. We show in the following table the most important protocols for wireless sensor networks in the last five years, and then we briefly explain the most important differences between them. Existing works mainly focus on two directions: TDMA and contention-based protocols. Two classes of contention-based protocols are those that add schedules and those that employ channel sampling. All protocols control the radio duty cycle to avoid energy waste in idle listening; some also take approaches to avoid overhearing or add other optimizations [10]. The major advantage of scheduling is that a sender knows a receiver's wakeup time and thus transmits efficiently. However, the cost of listening for an entire contention interval is about ten times the cost of channel sampling, thus the overhead in lightly used networks is higher than Low Power Listening (LPL) based approaches [4].

S-MAC, Sensor Media Access Control [1, 2], is a contention based MAC protocol that adds into the MAC layer power management, link-level retransmission, duplicate packet suppression, hidden terminal avoidance using RTS/CTS, and link-quality estimation. S-MAC periodically sleeps, wakes up, listens to the channel, and then returns to sleep. S-MAC is designed to operate like a "black box"—it is optimized for a representative set of workloads. S-MAC does not have any hooks to change its duty cycle or reconfigure its parameters; instead, it combines link, routing, organization, synchronization, and fragmentation services into a single protocol. All of these services may be used by network protocols. In the power management scheme in S-MAC, each active period is of fixed size, 115 ms, with a variable sleep period. The length of the sleep period dictates the duty cycle of S-MAC. At the beginning of each active period, nodes within a cell exchange synchronization information. S-MAC uses Adaptive Listen, which allows S-MAC to achieve lower latency when relaying multihop traffic. By changing the duty cycle, S-MAC can trade energy for latency [3].

Table 1. Mac Protocols

NAME	YEAR	UNIVERSITY	CHARACTERISTICS
S-MAC	2002	USC, USA	Scheduled Fixed duty cycle
T-MAC	2003	Delft, Holland	Scheduled Adaptive duty cycle
WISE-MAC	2004	CSEM, Switzerland	Synchronized preamble sampling
B-MAC	2004	UCB, USA	Adaptive preamble sampling (CCA & LPL)
Z-MAC	2005	NCSU, USA	hybrid protocol (TDMA/CSMA)
UB-MAC	2005	UM, USA	Uncertainty-driven time synchronization
B-MAC+	2006	Pisa, Italy	B-MAC wake-up preamble division
SCP-MAC	2006	USC, USA	LPL + Scheduling
Crankshaft	2007	Delft, Holland	SCP-MAC for Dense WSN
SA-MAC	2007	UCLM, Spain	Synchronization based packet and take advantage it for routing

To handle load variations in time and location T-MAC, Timeout MAC [5], introduces an adaptive duty cycle in a novel way: by dynamically ending the active part of it. This reduces the amount of energy wasted on idle listening, in which nodes wait for potentially incoming messages, while still maintaining a reasonable throughput. T-MAC in variable workload uses one fifth the power of S-MAC. In homogeneous workloads, T-MAC and S-MAC perform equally well. T-MAC suffers from the same complexity and scaling problems of S-MAC [3].

WiseMAC (Wireless Sensor MAC) [5] is based on the preamble sampling technique. This technique consists in regularly sampling the medium to check for activity. By sampling the medium, we mean listening to the radio channel for a short duration, e.g. the duration of a modulation symbol. All sensor nodes in a network sample the medium with the same constant period. Their relative sampling schedule offsets are independent and constant. This technique provides a very low power consumption when the channel is idle. The disadvantages of this technique are that the (long) wake-up preambles cause a throughput limitation and large power consumption overhead in reception. The novel idea introduced by WiseMAC consists in letting the access point learn the sampling schedule of all sensor nodes. Knowing the sampling schedule of the destination, the access point starts the transmission just at the right time with a wake-up preamble of minimized duration. WiseMAC differs from previous research on ad-hoc sensor networks, mainly because it focus on an infrastructure topology, and investigate how the unconstrained energy supply of the access point can be exploited.

B-MAC, Berkeley Media Access Control [4], a carrier sense media access protocol for wireless sensor networks that provides a flexible interface to obtain ultra low power operation, effective collision avoidance, and high channel utilization, is motivated by the needs of monitoring applications. B-MAC supports on-the-fly reconfigu-

ration and provides bidirectional interfaces for system services to optimize performance, whether it be for throughput, latency, or power conservation [3]. To achieve low power goals and workload requirements, S-MAC is not only a link protocol, but also a network and configuration protocol. Applications and services must rely on policies internal to S-MAC to adjust operation as node and network conditions change; such changes are opaque to the application. In contrast, the B-MAC protocol contains a small core of media access functionality. B-MAC uses clear channel assessment (CCA) and packet back offs for channel arbitration, link layer acknowledgments for reliability and low power listening (LPL) for low power communication. To achieve low power operation, B-MAC employs an adaptive preamble sampling scheme to reduce duty cycle and minimize idle listening. WiseMAC meets many of B-MAC goals except that it has no mechanism to reconfigure based on changing demands from services using the protocol [4].

Z-MAC [6] is a hybrid protocol for wireless sensor network that combines the strengths of TDMA and CSMA while offsetting their weaknesses. Like CSMA, Z-MAC achieves high channel utilization and low-latency under low contention and like TDMA, achieves high channel utilization under high contention and reduces collision among two-hop neighbours at a low cost. Z-MAC has the setup phase in which it runs the following operations in sequence: neighbour discovery, slot assignment, local frame exchange and global time synchronization. These operations run only once during the setup phase and does not run until a significant change in the network topology occurs. Z-MAC uses the CCA and LPL features of B-MAC. Thus, its energy efficiency is no better than B-MAC's under low data applications.

UB-MAC, uncertainty B-MAC [7], integrate an uncertainty-driven time synchronization scheme with B-MAC, and empirically demonstrate one or two orders of magnitude reduction in the transmit energy consumption at a node with negligible impact on the packet loss rate. B-MAC uses an asynchronous technique that involves no time synchronization or clock estimation to deal with the time uncertainty. Other techniques such as S-MAC and T-MAC use synchronized techniques where explicit time synchronization beacons are transmitted periodically between neighbouring nodes. This enables the transmitter to turn on the radio at the right moment, but the inability to deal effectively with time varying changes in clock drift force these techniques to re-synchronize frequently. In [7] the authors, first, experimentally obtained long time-scale data sets both in indoor and outdoor settings for Berkeley mica2 motes. They then performed a detailed characterization period on accuracy and energy requirements. Second, they used the results of the empirical analysis to design an adaptive time synchronization protocol and, lastly, they developed a prototype implementation on mica2 motes for sense-response applications (only with three nodes).

B-MAC+ [9] is an enhancement of B-MAC. The basic idea consists in replacing the pattern of the wake-up preamble of B-MAC, with a new pattern that contains information about the size of the remaining part of the preamble not yet transmitted. This information can be used by receivers to avoid wait states during significant portions of the preamble transmission time, i.e. going to sleep and waking up when the data payload is actually transmitted. The wake-up preamble of B-MAC+ is obtained by dividing into slots the wake-up preamble of B-MAC. The experiments were made only with two Tmote nodes, one sender node and one receiver node connected to the

USB port of a PC running a modified version of TOS_Base. The results show statistics about the time the radio must be on to receive a packet, and relates the collected data with the analytical mean values of both B-MAC+ and B-MAC. B-MAC+ outperforms B-MAC, as the countdown packets technique reduces the waste of time in receiving long wake-up preambles.

SCP-MAC, Scheduled Channel Polling [10], eliminates long preambles in LPL for all transmissions, and is able to operate ultra-low duty cycles when traffic is light by synchronizing the channel polling times. It has been designed with two main goals: first, to push the duty cycle an order of magnitude lower than in practical with current MAC protocols and second, to adapt to variable traffic loads common in many sensor networks applications. To schedule coordinated transmission and listen periods is the approach of S-MAC and T-MAC. The schedule determines when a node should listen and when it should sleep. In S-MAC and T-MAC nodes adopt common schedules, synchronizing with periodic control messages. A receiver only listen to brief contention nodes participating in data transfer remain awake after contention periods, while others can then sleep. Overhead is due to schedule maintenance and listening during contention intervals if there is nothing to send. Another technique is LPL presented in WiseMAC and B-MAC. In LPL, nodes wake up very briefly to check channel activity without actually receiving data. The authors of SCP-MAC call this action channel polling (polling refers only to each node sampling the channel to check for activity). Unfortunately, current LPL-based protocols have some problems: the duty cycle is limited to 1-2% because the polling frequency needs to balance the cost on sending preambles and polling the channel; this balance between sender and receiver costs makes LPL-based protocols very sensitive to tuning for an expected neighbourhood size and traffic rate; finally, it is challenging to adapt LPL directly to newer radios like 802.15.4, since the specification limits the preamble size.

SCP-MAC adopts channel polling from LPL approaches. However, unlike LPL, SCP-MAC synchronizes the polling times of all neighbouring nodes. SCP-MAC distributes schedules such as developed by S-MAC: each node broadcast its schedule in a SYNC packet to its neighbours every synchronization period. The key is to discover the optimal synchronization period and wakeup tone length that minimizes the overall energy consumption. Also SCP-MAC eliminates long preambles, so its energy performance is not sensitive to varying traffics loads. The authors of SCP-MAC have implemented the protocols in TinyOS over the mica2 motes with the CC1000 radio, and to provide a clean comparison of LPL and scheduling, they implement SCP as a layer over basic LPL. They also describe a preliminary port to micaZ motes with the CC2420 radio supporting IEEE 802.15.4. The relative performance of SCP improves on never, faster radios like the CC2420, while that of LPL degrades.

Crankshaft [11] is a MAC protocol specifically targeted at dense wireless sensor networks. It employs node synchronization and offset wake-up schedules to combat the main cause of inefficiency in dense networks: overhearing by neighbouring nodes. Further energy savings are gained by using efficient channel polling and contention resolution techniques. Crankshaft employs a mechanism of channel polling very similar to the SCP-MAC.

3 Protocol Engineering

Protocol engineering becomes a key area of research enabling the development and deployment of power efficient networks. Cross-layer protocol design and power management have become two main approaches towards the development of such networks.

In order to explore the capabilities of state-of-the-art technology, we have conducted a set of experimental trials [14]. The hardware equipment used throughout our experiments consisted of ten micaZ nodes (MOTE-KIT2400)[6] and a Stargate node used as base station. This latter node gathers the data collected from the ten sensor nodes. The nodes were located throughout different locations in a building characterized by different environmental conditions: sun light conditions, number of people normally working at a given lab, among others.

Our first experiment consisted in deploying them operating under the control of the application provided by the manufacturer. This application collects and sends the data to the sink every eight seconds to the base station (high load scenario). We observe that the battery life span was 82 hours.

Due to the limited life span of the system, we did focus our research on the analysis of the power consumption, in particular, on the transmission system and associated protocols. It is well known that overhearing is the main source of power mismanagement. We choose two key elements to improve the lifetime of the network:

- To program a new application for data capture and data transmission to the base station
- To efficiently control the radio use. In a previous work, we have shown that the radio device consumes as much as 65% of the overall energy [12]

Regarding the second, we focus on the MAC layer and the different MAC protocols implemented for wireless sensor network: S-MAC, B-MAC or Wise-MAC [12]. All these protocols are different to the standard IEEE 802.15.4 [16]. They have important characteristics addressing the power consumption at the MAC layer of wireless sensor network. However, they have been developed for the mica2-based platforms (CC1000 radio chip) [17]. We have started by adapting the B-MAC to the micaZ (CC2420 radio chip) [15].

B-MAC is able to reduce the *idle-listening*, i.e., the time that the node spends listening the channel. B-MAC requires that each node should wake up periodically to verify the channel activity. In case of detecting activity, the node keeps sensing the channel. On the contrary, no activity is detected, it falls asleep. The time between two consecutive wake-up periods is fixed by the check interval. B-MAC defines eight check intervals, each one corresponding to a different listening mode.

In order to ensure that all the packets are properly received, the packets are sent with a preamble whose length is longer than the check interval. B-MAC defines 8 sizes of preamble, each one related to a different way of transmission, which the protocol denominates transmit mode.

Another advantage of B-MAC, with respect to S-MAC, is its modularity and flexibility. B-MAC provides accessible interfaces to the upper layers, allowing the upper layers to transmit or listen in various modes.

B-MAC has been originally designed over the CC1000 radio chip (mica2). Therefore, the first task has been to analyze the main differences between the two radio chip systems. These are listed in Table 2.

Table 2. Summary of main sensor radio chip characteristics.

Chipcon CC1000	Chipcon CC2420
<ul style="list-style-type: none"> • Chip UHF RF transceiver 	<ul style="list-style-type: none"> • 2.4 Ghz IEEE 802.15.4 systems
<ul style="list-style-type: none"> • Frequency Range 300-1000 Mhz 	<ul style="list-style-type: none"> • 250 Kbps
<ul style="list-style-type: none"> • Integrated bit synchroniser 	<ul style="list-style-type: none"> • Suitable for both RFD and FFD
<ul style="list-style-type: none"> • Consumption: RX 9.6 mA TX 26.7 mA 	<ul style="list-style-type: none"> • Consumption: Rx 18.8 mA TX 17.4 mA
	<ul style="list-style-type: none"> • Independent data buffering for rx and tx
	<ul style="list-style-type: none"> • Encrypt (AES 128)

As seen from the table, the use of the radio CC2420 provides some advantages to micaZ nodes, characteristics as throughput or the possibility to encrypt, but the fact that these new chip radio comply standards like IEEE 802.15.4, limits them in certain aspects. The preamble length is fixed by the standard. This is the main obstacle to the implementation of B-MAC on the CC2420 radio chip. B-MAC contains three important interfaces *MacControl*, *MacBackoff* and *LowPowerListening*, although the two early interfaces have already been implemented, the interface *LowPowerListening* (LPL) has been until today an insurmountable step.

Our proposal, according to the recommendations of the protocol designer, it is break with the idea of, if sender is not capable of prolonging the preamble for the total reception of the packet on the part of the receiver, that be the receiver that before the packet arrival event maintains the radio on for its total reception (see Figure 1). This is the main idea before the preamble problem, but will also be necessary to implement the rest of the interface that permit to level of application, and of transparent way, to define the listening and transmit mode desired.

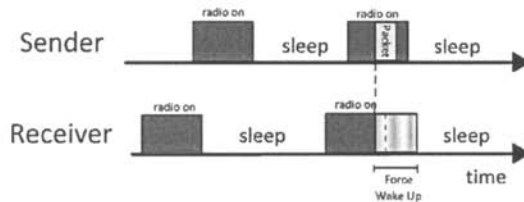


Fig. 1. On-off Timing.

The mechanism forcing the radio to keep awake should be incorporated into the LPL interface, i.e., made available to the upper levels. Our objective is testing the improvement using B-MAC and our own application, *IntellBuildApp*. We design different tests to evaluate the lifetime of network and to collect data allowing us to monitor the environmental conditions of the building. Four sets of trials were defined:

- **Test 1:** Use of the application as defined by the manufacturers based on the MTS400 and micaZ board. It sends a packet every 8 seconds, with all the data from sensor board. It makes use of the IEEE 802.15.4 MAC protocol.
- **Test 2:** Our own protocol. A packet is sent every 8 seconds, only temperature and humidity data are sent. It uses a multihop routing protocol and the standard 802.15.4 MAC protocol.
- **Test 3:** Same configuration as Test 2, but it uses the new interface implemented over B-MAC, using listening mode 3 and transmit mode 4 (1200 ms radio on, 1100 ms radio sleep).
- **Test 4:** Same configuration as Test 2, but it uses the new interface implemented over B-MAC, using listening mode 5 and transmit mode 3(2500 ms radio on, 900 ms radio sleep).

We evaluated the performance of the different set-ups in terms of the network lifetime and packet loss count. In this way, we can evaluate the energy-efficiency of B-MAC for wireless sensor networks. In terms of the network lifetime, the best results were obtained when the listening mode was longer than the transmit mode. Figure 2 graphically shows the results obtained. However, Figure 3 shows that the packet loss count is extremely high. This is due to the lack of synchronisation among the nodes. While some nodes may attempt to transmit, the potential receivers may be sleeping. It is clear that there is a need of a synchronisation scheme among the nodes.

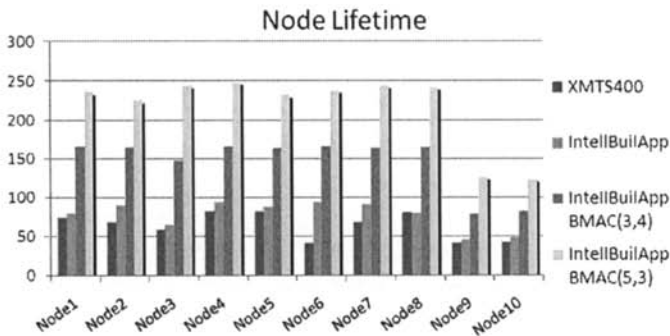


Fig. 2. Lifetime of different nodes

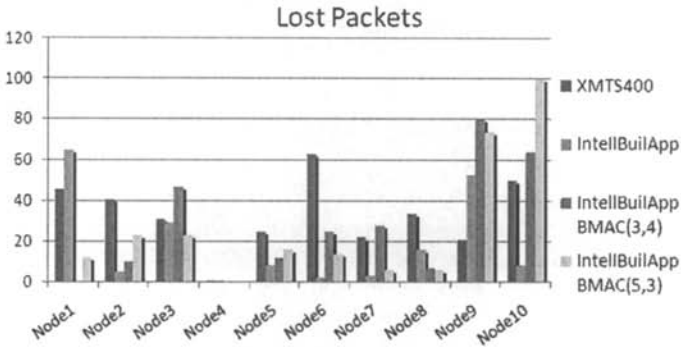


Fig. 3 Packet loss by node

4 SA-MAC: The Synchronous after Awake MAC engine

Bearing in mind the problem of packet loss and the need of developing an energy efficient protocol, we have undertaken the design of SA-MAC. We consider the design of a node synchronization engine. This synchronization algorithm would manage to save energy and establish routes, solving two issues in the area of wireless sensor networks. To achieve the design goal, we developed SA-MAC that consists of two phases: neighbour discovery phase and synchronization phase.

The main motivation towards the design of a synchronization scheme among neighbouring nodes is to limit the idle listen. Since the wake-up procedure implies a transient period, the synchronization should be designed by limiting the number of wake up steps. Towards this end, all the children nodes to a parent should send one after another before going back to sleep. In this way, the sink node only has to turn on its radio for a shortest period of time.

In our proposal, the nodes exchange their schedules by broadcasting them to all their immediate neighbours. It is also important to take into account the clock drift between different nodes since this is a critical point in synchronization

4.1 Neighbour Discovery Phase

In this first phase, each node sends periodic neighbour discovery packets by broadcasting them to all its immediate neighbours. In these packets, the node includes the time, it is important for establishing the listening period for the parent node. In our implementation, the discovery packets are sent every 20 seconds. If a node does not receive any discovery packet from a node for two minutes, it is locally considered inactive, i.e., no longer present.

When a node receives a neighbour discovery packet, it takes out the node id, its send time and cost to send through it and save it in its neighbour table, see Figure 4. Moreover, it adds an entry in its Planning Table.

Neighbour Table			Planning Table			
id	send time	active	id	cost	stage	delay
...

Fig. 4. Neighbour and Planning Tables.

For its election, it compares between the costs of the possible candidates, in this first evaluation we work with hop number to the base station. Initially all entries set up cost, stage, delay and active to not valid values, except for packets receive from base station, this packet sets up its cost to 1, in such a way that this entry has priority when is found for the scheduling algorithm and the node establishes it like parent setting up stage to value *parent*. When a node has fixed its parent, it has to inform to it for to be considered in the parent listen period. The node can also include the new cost to arrive to the base station in its discovery packet. In this way the network topology is generated as a tree, with the base station being the root with children nodes in a first level (hop) and the remaining ones in successive levels.

4.2 Synchronization Phase

In this second phase, we describe how the node chosen as parent synchronizes with all its children: As already stated, the children should send their packets one after the other, remaining the father the minimum time in the listen state.

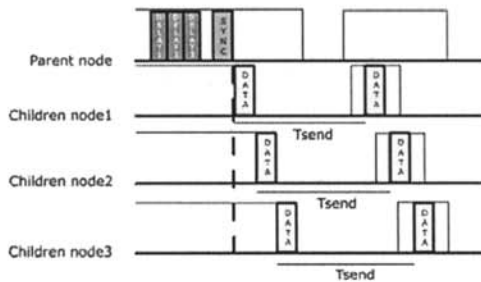


Fig. 5. Synchronization phase.

This task begins when a node receives a packet from another node. The parent sets up a record, in the Planning Table, for each child. In it, the parent indicates the time delay for the associated child. This time delay is fixed in accordance with the number of children associated to the father at that moment. We denote t_{ln} as the time that the parent node listens to each one of its children nodes and $N_{ch(i)}$ as the number of children of a parent at time i . This time is transmitted by the father to each child node.

$$delay_{node} = t_{ln} \times N_{ch(i)}$$

Once all children of a parent have received their delay, the parent sends a packet via broadcasting to synchronize all its children nodes. Figure 8 depicts the ideal situation of a parent node synchronized with three nodes.

It is a dynamic process in which new nodes can be easily integrated. A new node can be simply added to the end of the listening period. In this way SA-MAC is able to adapt the topology of the network due to the addition of a new node. In a similar way, nodes leaving the network can be simply discarded without requiring a long signalling process.

The following code shows the activity of node when a packet is received.

```

receive(packet)
{
    switch(packet_type)
    {
        case 'discovery':
            if (node_is_known)
                refresh_table();
            else insert_to_table();
            if(not_base_node)
                planning_algorithm();
        case 'information':
            if(node_is_known)
                insert_to_table();
            refresh_table();
            send_delay_to_children_node();
        case 'synchronization':
            save_delay();
        case 'start':
            send_event_to_application_layer_after_delay();
    }
}

```

5 Conclusions and Future works

This paper introduces the implementation of synchronized MAC for CC2420 radio chip. Energy efficiency is the primary goal in the protocol design. This is done by addressing the key features of the MAC layer, such as, overhearing and idle listening. By adopting a cross-layer design, we take advantage of the synchronization done at the physical level for developing a routing strategy. We are currently working on the development and testing of our proposal over an actual wireless sensor network.

References

- [1] W. Ye, J. Heidemann and D. Estrin, An Energy-efficient MAC protocol for wireless sensor networks, In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2002), June 2002

- [2] W. Ye, J. Heidemann and D. Estrin, Medium access control with coordinated, adaptative sleeping for wireless sensor networks, In IEEE Transactions on Networking, April 2004.
- [3] J. Polastre, A Unifying Link Abstraction for Wireless Sensor Networks, Doctoral Dissertation, University of California, Berkeley. October 25, 2005.
- [4] J. Polastre, J. Hill and D. Culler, Versatile Low Power Media Access for Wireless Sensor Networks, In proceedings of the ACM SenSys'2004, November 2004, Baltimore, Maryland, USA
- [5] T. van Dam and K. Langendoen, An adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks, In proceeding of the ACM Sensys'2003, November 2003, Los Angeles, California, USA
- [6] I. Rhee, A. Warrier, M. Ahia and J. Min, Z-MAC: a Hybrid MAC for Wireless Sensor Networks, In proceedings of the 3rd International Conference on Embedded Networked Sensor Systems, 2005, San Diego, California
- [7] S. Generiwal, D. Ganesan, H. Sim, V. Tsatsis and M. Srivasta, Estimating Clock Uncertainty for Efficient Duty-Cycling in Sensor Networks, In proceedings of the ACM conference on Embedded Networked Sensor Systems, 2005, San Diego, California
- [8] V. Raghunathan, S. Generiwal and M. Srivastava, Emerging Techniques for Long Lived Wireless Sensor Networks, IEEE Communications Magazine, April 2006
- [9] M. Avvenuti, P. Corsini, P. Masci and A. Vecchio, Increasing the efficiency of preamble sampling protocols for wireless sensor networks, In proceedings of the 1st IEEE International conference on Mobile computing and Wireless Communications, Amman (Jordan), September 2006
- [10] W. Ye, F. Silva and J. Heidemann, Ultra-Low Duty Cycle MAC with Scheduled Channel Polling, In proceedings of the ACM conference on Embedded Networked Sensor Systems, 2006, Boulder, Colorado, USA.
- [11] G. P. Halkes and K.G. Langendoen, Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks, In proceedings of the 4th European Conference EWSN 2007, Delft, The Netherlands
- [12] T. Olivares, P.J. Tirado, L. Orozco-Barbosa, V. López y P. Pedrón: Simulation of Power-aware Wireless Sensor Network Architectures, in ACM Internacional Workshop on Performance Monitoring, Measurement, & Evaluation of Heterogeneous Wireless and Wired Networks. Torremolinos, Málaga (España)
- [13] K. Langendoen and G. Halkes, Embedded Systems Handbook, CRC Press, Aug. 2005
- [14] T. Olivares, P.J. Tirado, F. Royo, J.C. Castillo and L. Orozco-Barbosa, Intell-Building: A Wireless Sensor Network for Intelligent Buildings, In proceedings of the 4th European Conference EWSN 2007, Delft, The Netherlands
- [15] Crossbow Technology, INC MicaZ datasheet. www.xbow.com
- [16] IEEE, Inc. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), October 2003.
- [17] Crossbow Technology, INC Mica2 datasheet. [website. www.crossbow.com](http://website.www.crossbow.com)