

# CONFIGURABLE MEDIUM ACCESS CONTROL FOR WIRELESS SENSOR NETWORKS

Lucas F. Wanner, Augusto B. de Oliveira, and Antônio A. Fröhlich

*Laboratory for Software and Hardware Integration*

*Federal University of Santa Catarina*

*PO Box 476 – 88049-900 – Florianópolis, SC, Brazil*

*{lucas, agosto, guto}@lisha.ufsc.br*

*<http://www.lisha.ufsc.br/>*

## Abstract

This article presents C-MAC, a *Configurable Protocol* for medium access control in wireless sensor networks. C-MAC works as a framework of medium access control strategies, with a transparent configuration system. The protocol aggregates different services, each implemented in several different strategies. Applications may configure different communication parameters in compile-time and run-time. C-MAC's meta-programmed implementation yields smaller footprint and higher performance than equivalent protocols for wireless sensor networks.

## 1. Introduction

Narrow band radios typically support simple modulation schemes, leaving a high level of control to software, and thus paying a higher processing overhead. On the other hand, wide band radios support sophisticated modulation techniques, such as Direct-Sequence Spread Spectrum (DSSS), and Phase Shift Keying (PSK), which are more robust to noise and interference, but have small flexibility and high power consumption. Low power radios have thus been presented as a viable alternative for wireless communication for embedded systems such as sensor networks.

Given the simplicity of communication hardware for such systems, Medium Access Control protocols and other data link layer services must be implemented in software. Services such as data packet detection, error detection and treatment, addressing, packet filtering, and others traditionally implemented in hardware become one of the main parts of

a communication stack implemented by operating systems for wireless sensor networks.

In current hardware prototypes for sensor networks, transmitting a single byte of data through the radio consumes more power than executing hundreds of instructions in the main micro-controller (Langendoen and Halkes, 2005). Thus, a communication system for wireless sensor networks must use the communication resources in a conservative way, providing only the support needed for specific applications to implement their services.

On the other hand, the simplicity of communication hardware for wireless sensor networks is not a limiting factor, but a desirable characteristic, as it allows a wide range of configuration of the data communication channel. Previous studies (Langendoen and Halkes, 2005; Polastre et al., 2005) show medium access control implemented in software which are adequately designed and adapted to applications may be more efficient than standardized, hardware-implemented protocols.

Several protocols have been designed and implemented for medium access control in wireless sensor networks (El-Hoiydi, 2002; Hoesel and Havinga, 2004; Lu et al., 2004; Woo and Culler, 2001; Mahlknecht and Boeck, 2004; Ringwald and Roemer, 2005; Dam and Langendoen, 2003; Polastre et al., 2004; Ye et al., 2002). In the context of wireless sensor networks, where different applications may have very specific communication workloads, adaptability and configuration are key issues. However, several implementations for these protocols in real systems do not provide adequate mechanisms for the application to configure the communication channel. In fact, many protocols, such as S-MAC (Ye et al., 2002) are specifically designed to optimize predetermined workloads (such as multi-hop communications), and present little to nil opportunities for application-guided configuration.

This article presents the design and implementation of C-MAC, a *Configurable Protocol* for medium access control in wireless sensor networks. C-MAC works as a framework of medium access control strategies, with a transparent configuration system. The protocol aggregates different services (e.g. synchronization, data detection, acknowledgments, contention, sending and receiving), each implemented in several different strategies. Applications may configure different communication parameters in compile-time and run-time. C-MAC owes its fundamental concepts to different medium access control protocols, and presents a new project to well-known, efficient low-power wireless communication solutions, allowing applications to take advantages of strategies that fit their requirements. Section 2 discusses MAC protocols for wireless sensor networks. Section 3 presents the design and implementation of C-

MAC. Section 4 evaluates the presented solution, and Section 5 closes the paper.

## 2. MAC Protocols for Wireless Sensor Networks

A Medium Access Control (MAC) protocol decides when a network node may access the medium, and tries to ensure that one node doesn't interfere with the transmissions of a second node. The MAC protocol is also responsible of treating or signaling collisions to the upper layers in the protocol stack.

Most MAC protocols for wireless networks have been designed and optimized for satellite links and local wireless networks (*Wireless LANs*). Communication requirements for wireless sensors are considerably different from these scenarios, specially regarding the need of autonomous and energy efficient operation in a wireless sensor. In several application scenarios of wireless sensors, parameters such as low latency and high throughput have smaller importance (Ali et al., 2006).

In current hardware prototypes for sensor networks, transmitting a single byte of data through the radio consumes more power than executing hundreds of instructions in the main micro-controller. Thus, power efficient operation is the main design issue to be considered for MAC protocol design. Langendoen and Halkes, 2005 identify the main sources of power overhead in a MAC protocol for wireless sensors with small traffic:

**Idle listening:** If a node doesn't know when it will receive messages from one of its neighbors, it will have to keep the radio turned on in receive mode all the time. As the cost of reception is much greater than the cost of *standing-by*, this is perhaps the greatest source of overhead.

**Collisions:** If two nodes transmit at the same date, data are corrupted, both transmissions must be repeated, and the energy spent on the first tries is wasted.

**Overhearing:** As the radio channel is a shared medium, a receiver may hear packets which are not directed to itself.

**Traffic fluctuations:** When a phenomenon is detected by many neighboring nodes in a densely installed network (in the *local gossip* pattern), the nodes will compete for the radio channel, and will waste power while waiting for a transmission window.

Medium access control protocols for sensor networks compromise performance (latency, throughput) for cost (power consumption). Power

consumption is minimized mainly by shortening the period in which the radio listens to the channel when there are no communications (*idle listening*) (Langendoen and Halkes, 2005).

Contention-based protocols, such as B-MAC (Polastre et al., 2004) attain energy efficiency by increasing the message preamble, allowing the radio channel to be verified with lower periodicity. Slot-based protocols, such as S-MAC (Ye et al., 2002) and T-MAC (Dam and Langendoen, 2003), reduce power consumption by limiting communication to well-defined periods. Comparisons in different application scenarios show that there is no “optimal” protocol for wireless sensors (El-Hoiydi, 2002; Hoesel and Havinga, 2004; Lu et al., 2004; Woo and Culler, 2001; Mahlknecht and Boeck, 2004; Ringwald and Roemer, 2005; Dam and Langendoen, 2003; Polastre et al., 2004; Ye et al., 2002). The choice of an adequate MAC protocol for a wireless sensor network application depends on the level of compromise between power efficiency and communication flexibility. Characteristics such as complexity, special hardware requirements (e.g. synchronization hardware), and application data communication patterns must be taken into consideration when determining the ideal MAC for a given scenario. In what regards communication support in an operating system for wireless sensors applications, configuration flexibility may be considered the most desirable trait.

### 3. C-MAC

C-MAC is a *Configurable Protocol* for medium access control in wireless sensor networks equipped with low power radio transceivers. Its configurable characteristic allows the user to adjust several communication parameters (e.g. synchronization, data detection, acknowledgments, contention, sending and receiving), in order to adjust the protocol to the needs of different applications.

Santos and Fröhlich, 2005 have developed a static composition system of lightweight protocols for computer grids that has presented several advantages to monolithic layered based implementations, such as the Internet Protocol (TCP/IP). This system consists of a meta-programmed framework, which provides mechanisms to allow the selection, configuration, and combination of communication protocols according to application requirements; and a baseline architecture for communication on top of which the protocols are designed. This paradigm presents several advantages, including the ability to create new communication services on-demand, and to allow applications to experiment with different communication protocol configurations, by collecting metrics in order to identify the configuration best fitted to its requirements.

C-MAC uses these same premises to build a configurable communication kernel, over which other protocols may be composed. The list of configurable points in the C-MAC architecture was assembled by analyzing the commonalities and variabilities in the design of MAC protocols for wireless sensor networks, and aims at delivering the largest range of configurable points that, when combined, form a complete MAC protocol. Protocol configuration is performed at compile-time, and run-time configuration of protocol characteristics is not treated in the current C-MAC architecture. The overhead of maintaining several configuration possibilities programmed in the node, and the need of a second protocol for synchronization makes the use of a run-time configuration system impracticable for a protocol as widely configurable as C-MAC. The main C-MAC configuration points include:

**Basic Communication Characteristics:** These configurations are handled by the communication hardware, and include: transmission frequency and power (which may be altered in runtime); modulation type (e.g. Manchester, NRZ); transmission data rate.

**Duty cycle and organization:** The duty cycle determines the active period in which the radio may operate. In a simple CSMA-based configuration, the radio may transmit at any time it detects the channel is free. On the other hand, in a slot-based protocol, the duty cycle is limited to the active part of the protocol's time slot.

**Collision-avoidance mechanism:** The collision-avoidance mechanism in a wireless sensor networks MAC protocol may be comprised of a carrier sense algorithm, the exchange of contention packets (*Request to Send* (RTS) and *Clear to Send* (CTS)), or a combination of both. Furthermore, there must be the possibility to not use any collision-avoidance mechanism, for example, in a sparse network with little communication, in which eventually retransmitting corrupted packets is less costly than the mechanism itself.

**Collision-detection mechanism:** As hardware for communications in wireless sensor networks is mostly half-duplex, the most widely used mechanism for collision detection is the use of acknowledgment packets, sent from the receiving node to indicate that the data was correctly received. In situations where packet loss is not a problem (e.g. a densely installed network, where many information packets are redundant), the collision detection mechanism may be eliminated from the protocol configuration, thus increasing power efficiency.

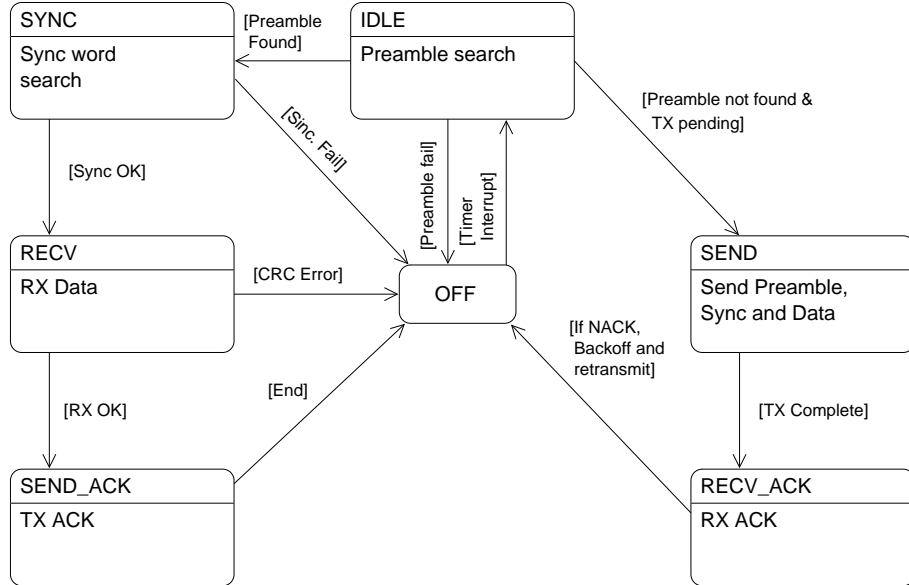


Figure 1. C-MAC State Machine Overview

**Collision handling mechanism:** When a collision is detected, the protocol may retransmit the packet, or simply increment a packet loss counter.

C-MAC is implemented through a state machine in which transitions are activated through interruptions of a dedicated counter (which controls duty cycle and backoffs), and through data interruptions of the communication hardware. Figure 1 presents a simplified overview of the protocol’s main state machine.

C-MAC’s configurable characteristics are selected by the programmer through *Configurable Traits* in EPOS (Polpetta and Fröhlich, 2004; Fröhlich, 2001), an application-oriented operating system. The EPOS system framework allows software components to be automatically adapted to fulfill the requirements of particular applications. Configurable Traits are parametrized classes whose static members describe the properties of a certain class. When a certain property is selected, the functionality it describes is included into the protocol. On the other hand, due to the use of function inlining and static meta-programming when a certain characteristic is *not* selected, no overhead associated with it is added to the final object code of the protocol. Furthermore, C-MAC’s modular design allows different radio transceivers to be used with no alterations in the protocol’s logic.

Table 1. Communication parameters

Parameter	Value
Duty cycle	100%
Transmission power	5 dBm
Modulation	19.2 kbps, Manchester encoding
Transmission backoff	0 ms
Collision detection and handling	None
Maximum theoretical data rate	16.4 kbps

Table 2. Memory footprint

System	Code (bytes)	Data (bytes)
EPOS	3888	108
TinyOS	8562	205

#### 4. Evaluation

In order to allow comparison between C-MAC in EPOS and other MAC protocols, C-MAC was configured in order to function like B-MAC in the TinyOS system. Table 1 summarizes the configuration used both in C-MAC and B-MAC in the tests presented in this section. Table 2 presents the system footprint for both TinyOS and EPOS. C-MAC's meta-programmed implementation, along with EPOS's component architecture delivers equivalent functionality, yielding a considerably smaller footprint than the B-MAC protocol in TinyOS.

Figure 2 presents packet loss in a network of two to four nodes, using both systems. In this test, a node was positioned in the center of an imaginary circle with a two-meter radius, and the other nodes were positioned in the border of this circle, equidistant from each other. Each node is in range, and may potentially interfere in the communications of every other node. Figure 3 shows the throughput of this same network. The differences between the two equally configured protocols may be attributed to a series of factors, including slight variations in the test environment and heterogeneity between nodes (e.g. antennas, slight variations in battery charge); design and programming problems; loss of interruptions; and the overhead of internal system procedures. It should be noted that repetitions in the tests diminish the impact of environmental variations and node heterogeneity.

C-MAC presented slightly superior performance than a protocol configured in an equivalent fashion, with smaller memory footprint. This

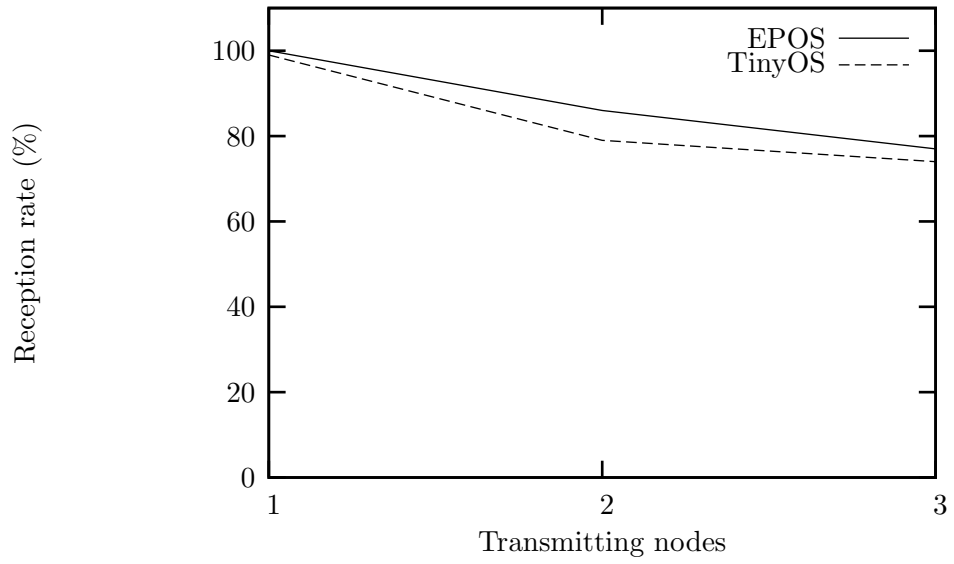


Figure 2. Reception rate (B-MAC and C-MAC)

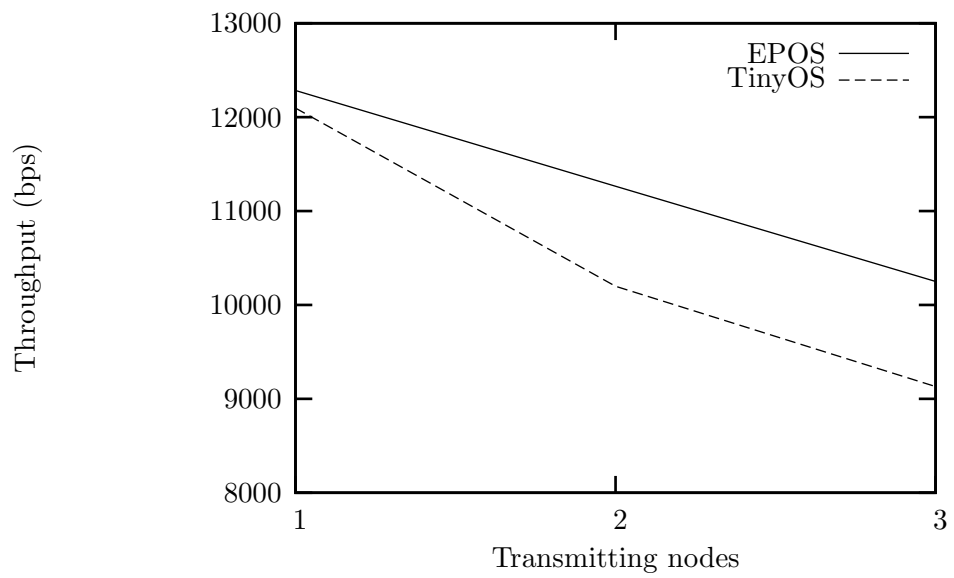


Figure 3. Throughput (B-MAC and C-MAC)



advantage is magnified by C-MAC's configuration system, which allows the creating of application-specific protocols, with only the necessary overhead.

## 5. Conclusion

The choice of an adequate MAC protocol for a wireless sensor network application depends on the level of compromise between power efficiency and communication flexibility. In what regards communication support in an operating system for wireless sensors applications, configuration flexibility may be considered the most desirable trait. The implementation of the B-MAC protocol for the TinyOS system allows dynamic and static configuration of basic communication parameters (e.g. frequency, power). Alterations in protocol logic demand direct interference of the user in the source code. In the same system, the S-MAC protocol implementation uses a monolithic design, with little space for configuration.

C-MAC's modular project and implementation allows applications to configure the data communication channel according to its needs. The protocol aggregates different services (e.g. synchronization, data detection, acknowledgments, contention, sending and receiving), each implemented in several different strategies. Applications may configure different communication parameters in compile-time and run-time. Tests using C-MAC in the EPOS operating system yielded better results than the B-MAC protocol in TinyOS, when both protocols were configured in an identical fashion.

Future developments in the C-MAC protocol shall allow a greater level of configuration including the possibility of global network synchronization, and adaptative duty cycle.

## References

- Ali, Muneeb, Saif, Umar, Dunkels, Adam, Voigt, Thimo, Romer, Kay, Langendoen, Koen, Polastre, Joseph, and Uzmi, Zartash Afzal (2006). Medium access control issues in sensor networks. *SIGCOMM Comput. Commun. Rev.*, 36(2):33–36.
- Dam, T. van and Langendoen, K. (2003). An adaptive energy-efficient MAC protocol for wireless sensor networks. pages 171–180, Los Angeles, CA.
- El-Hoiydi, A. (2002). Aloha with preamble sampling for sporadic traffic in ad hoc wireless sensor networks. In *IEEE International Conference on Communications (ICC)*, New York.
- Fröhlich, Antônio Augusto (2001). *Application-Oriented Operating Systems*. GMD - Forschungszentrum Informationstechnik, Sankt Augustin.
- Hoesel, L. van and Havinga, P. (2004). A lightweight medium access protocol (LMAC) for wireless sensor networks. Tokyo, Japan.
- Langendoen, K. and Halkes, G. (2005). *Embedded Systems Handbook*, chapter Energy-Efficient Medium Access Control. CRC press.

- Lu, G., Krishnamachari, B., and Raghavendra, C. (2004). An adaptive energy-efficient and low-latency MAC for data gathering in sensor networks. In *Int. Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, NM.
- Mahlknecht, S. and Boeck, M. (2004). CSMA-MPS: A minimum preamble sampling MAC protocol for low power wireless sensor networks. In *IEEE Int. Workshop on Factory Communication Systems*, pages 73–80, Vienna, Austria.
- Polastre, Joseph, Hill, Jason, and Culler, David (2004). Versatile low power media access for wireless sensor networks. In *SenSys '04: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107, New York, NY, USA. ACM Press.
- Polastre, Joseph, Szewczyk, Robert, and Culler, David (2005). Telos: Enabling ultra-low power wireless research. In *The Fourth International Conference on Information Processing in Sensor Networks: Special track on Platform Tools and Design Methods for Network Embedded Sensors (IPSN/SPOTS)*, Los Angeles, California.
- Polpeta, Fauze Valério and Fröhlich, Antônio Augusto (2004). Hardware Mediators: a Portability Artifact for Component-Based Systems. In *International Conference on Embedded and Ubiquitous Computing*, volume 3207 of *Lecture Notes in Computer Science*, pages 271–280, Aizu, Japan. Springer.
- Ringwald, M. and Roemer, Kay (2005). BitMAC: a deterministic, collision-free, and robust MAC protocol for sensor networks. In *Proc. IEEE European Workshop on Wireless Sensor Networks (EWSN) 2005*, pages 57–69, Istanbul, Turkey.
- Santos, Thiago Robert and Fröhlich, Antônio Augusto (2005). A Customizable Component for Low-Level Communication Software. In *19th Annual Symposium on High Performance Computing Systems and Applications*, pages 58–64, Guelph, Canada.
- Woo, Alec and Culler, David E. (2001). A transmission control scheme for media access in sensor networks. In *Proceedings of the 7th annual international conference on Mobile computing and networking*, pages 221–235, Rome, Italy.
- Ye, W., Heidemann, J., and Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. In *21st Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1567–1576.