

## Programmable MAC in Body Area Networks, One Command at a Time

Costas Michaelides\* and Foteini-Niovi Pavlidou\*\*

School of Electrical and Computer Engineering, Aristotle University of Thessaloniki, Thessaloniki, 54124, Greece

\* Student Member, IEEE

\*\* Senior Member, IEEE

**Abstract**—The discussion concerning programmable MAC in Body Area Networks (BANs) is due to the demand for simple and low-power sensor nodes. Additionally, the diverse applications in BANs require low-level modifications to support adaptive services or custom functions. In this work, we propose a novel scheme for programmable MAC, which requires a minor modification to the beacon frame of IEEE 802.15.6-2012. Specifically, our main contribution is the attachment of a command to the beacon, which is broadcasted at the beginning of each superframe by the hub. The hub requests an action, typically a modification of a MAC capability field, by the nodes with a metric which satisfies a constraint. Thus, one command at a time, the proposed scheme is applied with a negligible overhead. Two adaptive use cases, based on signal strength, are implemented to demonstrate this scheme. Firstly, the hub requests the nodes with high signal strength to enable relay support and secondly, the hub requests the nodes with low signal strength to set a sleeping pattern. In the first case packet delivery increases significantly, while in the second case each node saves an amount of energy.

**Index Terms**—Body area networks, sensor networks, network protocols, data link controller, simulation.

### I. INTRODUCTION

A Body Area Network (BAN) is defined as a group of sensor nodes inside or on the surface of the human body, coordinated by a hub. IEEE 802.15.6-2012 [1] specifies the functions for the communication in one-hop star or two-hop extended star topology. A BAN as a small-scale, centralized sensor network is an appropriate field to experiment with remote commands. The hub has no obvious limitations, but the nodes are expected to be simple, low-power and interchangeable. Thus, in this letter, we propose a hybrid centralized-decentralized administration of their MAC capabilities.

The concept of a programmable MAC is presented thoroughly in [2]. An early related work in BANs is [3], where a scheduler is employed to control a virtual MAC on top of the real MAC. Programmability in BANs concerns mostly the adaptive modification of MAC capability fields. Several works in BANs indicate the interest for dynamic and adaptive MAC applications [4]–[8]. Moreover, sensors have a very specific operation: to sense and transmit. Thus, the conventional assumption that a node is available for communication at any time is questionable. This observation is also stated in a recent patent [9], filed by Toshiba Corporation, which describes a way to configure the hub's MAC parameters, according to the signal strength of the nodes. In our work, we focus on the modification of the nodes' parameters instead of the hub's. In the tradition of Wake-On-LAN [10], where a magic packet is transmitted to wake up multiple nodes, our main contribution is a small modification to IEEE 802.15.6-2012, to allow the broadcast of one command at the beginning of each superframe. Surprisingly, this restrictive approach enables a field of simple and effective adaptive applications.

Corresponding author: Costas Michaelides (e-mail: cmich@computer.org).

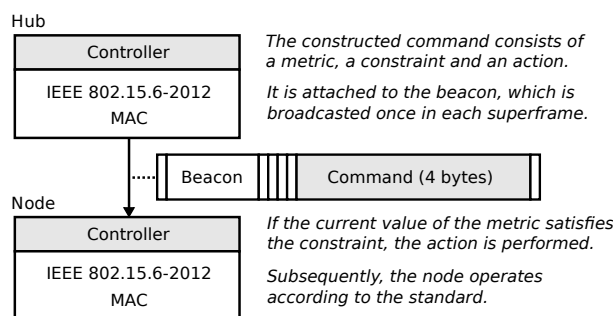


Fig. 1. The operation of the proposed scheme

In this work, we propose a hybrid centralized-decentralized scheme for programmable MAC in BANs by employing one command in each superframe, without any additional frame transmissions. At the hub, a data link controller processes a high-level request and constructs the required command based on the current values of system metrics. A command consists of a *metric*, a *constraint* and an *action*. It is attached to the beacon, the synchronization frame of the network, to be broadcasted at the beginning of the following superframe. Upon a beacon reception, the data link controller of each node checks if the current value of the metric satisfies the constraint. If the result is positive, the requested action is performed. The only modification of IEEE 802.15.6-2012, is the addition of a short command field to the beacon. The operation of the proposed scheme is depicted in Fig. 1. In order to showcase the feasibility of this scheme, we simulated two adaptive use cases which employ signal strength as a metric. Our system model was introduced in [11].

The rest of this paper is structured as follows. In Section II, there is a short note on the motivation. The proposed scheme is presented in more detail in Section III, followed by a simulation in Section IV. Section V concludes the paper.

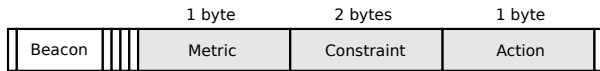


Fig. 2. Simplified structure of a beacon, including a command

## II. MOTIVATION

By design, the operation of a BAN is rather simple. The nodes connect to the hub and transmit their data packets. MAC is coordinated by the hub using predefined scheduled, improvised or random access schemes. However, the centralized topology of BANs suggests that the network operation can be modified dynamically at runtime by the hub, using remote commands. The commands may be used either for adaptive services or for occasional adjustments of the MAC fields of the nodes. Such a design, implies that the nodes are capable of receiving and executing remote commands.

A conventional design using remote commands, employs command frames. This is a well established method, applicable to any type of network and supported by IEEE 802.15.6-2012 as well. However, it is assumed that the nodes are able to receive any frames at any time. In BANs and generally in sensor networks, the nodes are intended to be primarily in Tx mode, transmitting their packets, or in sleep mode to save energy. Thus, in order to employ command frames, a special access phase has to be specified, just for this purpose. In practice, this is a huge disadvantage: the nodes are required to stay in Rx mode periodically, in order to be available to receive a possible command. Bearing in mind that the primary purpose of the nodes is to continuously sense and transmit, the specification of yet another access phase is an important design flaw which results to an additional amount of latency.

In this work, we present a scheme for programmable MAC in BANs, by employing a periodic transmission of a command attached to the beacon. Thus, no additional transmissions are required. The beacon is the synchronization frame of the network and has a relatively large size, so the additional command field is a negligible overhead, concerning bit errors. Broadcasting of commands is appropriate in a BAN because the nodes belong to a certain body and have common goals, typically to extend the lifetime of the network. Moreover, broadcasting provides new opportunities for intent-based applications. The hub suggests an action and the nodes may accept or reject it, depending on the current values of their metrics. In practice, the limitation of one command in each superframe provides a simple and flexible programming environment.

## III. PROPOSED SCHEME

The proposed scheme provides a hybrid centralized-decentralized programming environment in BANs. In this section, we describe the construction and the operation of the proposed scheme. The structure of a command is presented in III-A. The implementation of the controller is discussed in III-B. Lastly, in III-C we present the communication method.

### A. Commands

A command, as depicted in Fig. 2 consists of a metric, a constraint and an action. The metrics and actions of Table 1 are specifically useful in BANs and can be used in order to provide adaptive services to the nodes. Each metric is subject to a constraint and each action sets

TABLE 1. A few metrics and a core set of actions

Metrics	Actions
Transmitted packets	Enable (or disable) relay support
Packet delay	Enable (or disable) random access
Buffer state	Enable (or disable) scheduled access
Type of data	Set a sleeping pattern
Signal strength	Increase (or decrease) priority level
Remaining energy	Increase (or decrease) security level

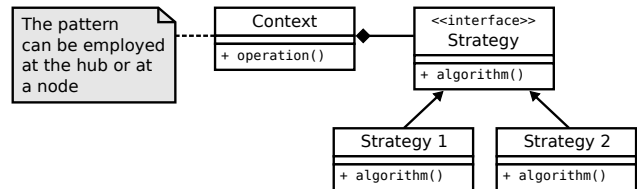


Fig. 3. A controller implements the strategy pattern

a value to a MAC capability field. Preferably, actions do not disrupt the low-level operation of MAC. A command can be expressed as: the node with a *metric* which satisfies a *constraint* may perform an *action*. For example, a simple command is to set a sleeping pattern (action) if the number of the transmitted packets (metric) has reached a threshold (constraint).

The size of each metric or action field is set to 1 byte, which is adequate for the specification of a large number of metrics and actions. Regarding the constraint, an integer value of 2 bytes is assumed, since a double precision value seems an unjustified overhead.

### B. Controller

The controller is included in data link layer but is decoupled from MAC operations. At the hub, it constructs a command according to a high-level request and the values of the system metrics. At each node, it decides whether the requested command is applicable, depending on the value of its metric and the constraint. Each controller can be implemented using the strategy pattern [12], as in Fig. 3, in order to perform operations at runtime. Strategy is a behavioral pattern, suitable for multiple implementations.

### C. Communication

The nodes communicate using beacon mode with superframes, in compliance with IEEE 802.15.6-2012. A command is attached to the beacon and is transmitted at the beginning of each superframe by the hub, as depicted in Fig. 4. The transmission of a beacon is repeated periodically for synchronization purposes. Upon reception, a received command is processed at the controller of a node. Subsequently, each node participates in the current superframe according to its updated MAC capabilities.

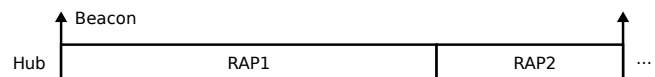


Fig. 4. Superframe, repeated over time

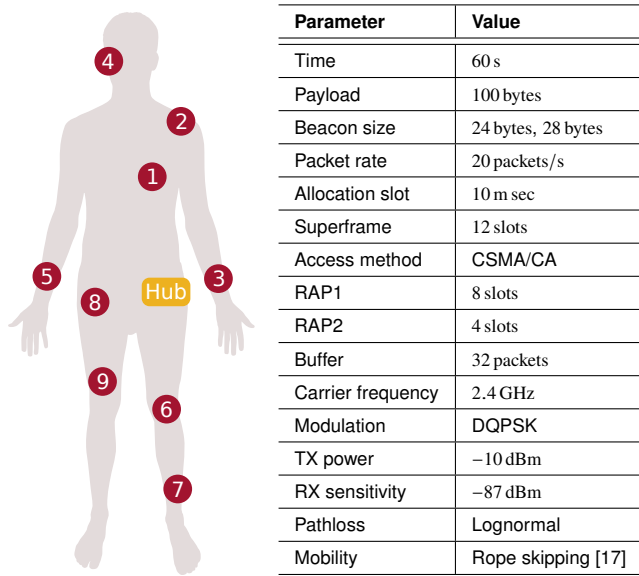


Fig. 5. Node placement and parameters

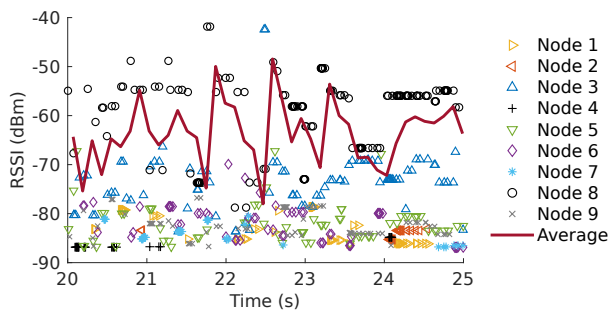


Fig. 6. Signal strength of each node, during a few seconds

#### IV. SIMULATION

It is important to investigate if the concept which has been descriptively put as "one command at a time" actually works. The following simulations are performed in a harsh environment with huge pathloss and high mobility. The system model of Fig. 5 is elaborated in detail in [13]. It is implemented with Castalia [14], a framework of OMNeT++ [15]. Nine nodes are placed on body surface and transmit data packets to the hub during the random access phase RAP1, as depicted in Fig. 4. Here, RAP2 is exclusive for relayed nodes. The mobility traces are extracted from [16].

To evaluate the performance of the proposed scheme we observe the performed actions, the received data packets, the consumed energy and application level latency of two use cases. In the first use case, the request is to set the nodes with signal strength above the average as relays, to forward the packets of relayed nodes to the hub. In the second use case, the request is to set the nodes with signal strength below the average to sleep. The two use cases are compared to a baseline implementation, in compliance with IEEE 802.15.6-2012. The presented results are the average values of 100 realizations and the error bars denote the standard deviation. In the rest of this section, to facilitate the discussion of the results, the first use case is denoted as *relaying* and the second one as *sleeping*.

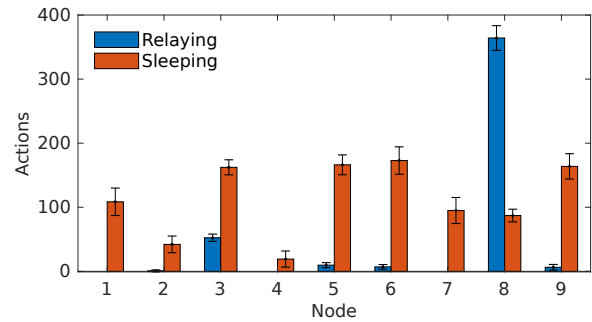


Fig. 7. Performed actions by each node

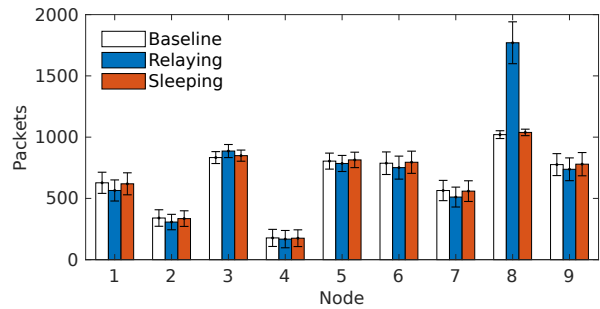


Fig. 8. Received data packets at the hub

The hub has access to several metrics, since each data packet contains a set of statistics in the payload. Thus, it keeps track of the received signal strength indication (RSSI) values. Fig. 6, shows the RSSI values of the nodes during a few seconds of one realization. The hub calculates the average RSSI value of the current superframe and constructs a command. Depending on the use case, the requested action is to enable relay support or to set a node to sleep. The average RSSI value serves as a constraint in both cases. Upon a beacon reception, a node performs the requested action if its current RSSI value satisfies the constraint. An action is valid only during the current superframe in order to guarantee that, in case of disconnection, MAC capability fields are not modified permanently.

In the case of *relaying*, node 8 performs the most actions (Fig. 7). This is also hinted by Fig. 6, where its signal strength is most of the time above the average. Packet reception is increased significantly due to relayed packets (Fig. 8). Inevitably, node 8 consumes an additional amount of energy (Fig. 9). In terms of latency, the results are comparable to the baseline implementation (Fig. 10).

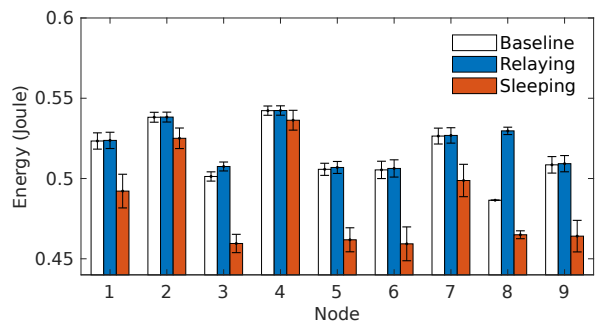


Fig. 9. Consumed energy by each node

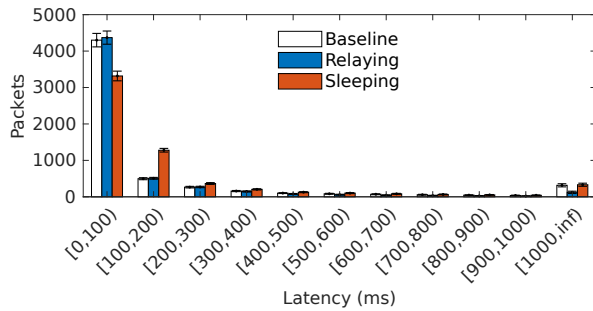


Fig. 10. Application level latency

In the case of *sleeping*, the requested action is performed by several nodes (Fig. 7). It appears that the signal strength of many nodes is below the average (Fig. 6). *Sleeping* reduces the energy consumption of each node (Fig. 9) and maintains similar results compared to the baseline implementation regarding packet delivery (Fig. 8). However, it has a slight negative impact on latency (Fig. 10).

It may be argued that node 4 deserves more sleep time to save energy (Fig. 9). Unfortunately, due to channel conditions, it receives very few commands (Fig. 7). The treatment of such extreme cases depends on the application. On the one hand, we may set a node to sleep permanently, on the other hand we may wait patiently for better future conditions.

Moreover, it appears that the overhead of the command is negligible, since the additional 4 bytes are too few compared to the total size of a beacon. Some issues are possible at very low signal strengths, where bit error probability is high. In our simulations, no error correction is performed; thus, a packet is discarded even for a single bit error. Considering that a node participates in a superframe only when a beacon is received, any issues concerning the reception of the beacon would have been obvious.

To sum up, some additional commentary is provided concerning the simulated use cases. In the first case, *relaying*, a node is placed close to the hub to boost packet delivery and is successfully detected as the best candidate for relaying. In the second case, *sleeping*, the hub implements a simple energy saving function by setting a sleeping pattern to the detected nodes with low signal strength.

## V. CONCLUSION

In this letter, we introduced the concept of low-level, intent-based application development in BANs. Specifically, we presented a novel scheme for programmable MAC in BANs, by employing one command in each superframe. A command consists of a metric, a constraint and an action. It is attached to the beacon, which is broadcasted by the hub periodically, at the beginning of each superframe. A node performs an action if the current value of its metric satisfies the constraint. The only modification of IEEE 802.15.6-2012 is the addition of a command field to the beacon. However, the incorporation of a set of actions in the standard, which modify the most common MAC capability fields, will allow interoperability and will encourage the use of commands instead of hardcoded implementations. We demonstrated the proposed scheme with two RSSI based simulations. Programmability is feasible and has a positive impact on the results. The proposed scheme offers a simple and effective way to control multiple nodes.

## REFERENCES

- [1] *IEEE Standard for Local and metropolitan area networks - Part 15.6: Wireless Body Area Networks*, IEEE Std. 802.15.6, Feb. 2012.
- [2] G. Bianchi and A. T. Campbell, "A programmable mac framework for utility-based adaptive quality of service support," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 2, pp. 244–255, Feb. 2000.
- [3] G. Zhou, J. Lu, C.-Y. Wan, M. D. Yarvis, and J. A. Stankovic, "Bodyqos: Adaptive and radio-agnostic qos for body sensor networks," in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, Phoenix, AZ, USA, Apr. 2008, pp. 565–573.
- [4] M. M. Alam, E. Ben Hamida, O. Berder, D. Menard, and O. Sentieys, "A heuristic self-adaptive medium access control for resource-constrained wban systems," *IEEE Access*, vol. 4, pp. 1287–1300, Apr. 2016.
- [5] C. Lin, K. C. Lin, and W. Chen, "Channel-aware polling-based mac protocol for body area networks: Design and analysis," *IEEE Sensors Journal*, vol. 17, no. 9, pp. 2936–2948, May 2017.
- [6] M. M. Alam, O. Berder, D. Menard, and O. Sentieys, "Tad-mac: Traffic-aware dynamic mac protocol for wireless body area sensor networks," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 1, pp. 109–119, Mar. 2012.
- [7] S. Moulik, S. Misra, and D. Das, "At-mac: Adaptive mac-frame payload tuning for reliable communication in wireless body area networks," *IEEE Transactions on Mobile Computing*, vol. 16, no. 6, pp. 1516–1529, Jun. 2017.
- [8] D. Chen and W. Chiu, "Collaborative link-aware protocols for energy-efficient and qos wireless body area networks using integrated sensors," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 132–149, Feb. 2018.
- [9] S. K. Subramani, W. H. Chin, and M. Sooriyabandara, "Configuring mac parameters in body area networks," U.S. Patent, Jun., 2018.
- [10] H. M. Garrett, F. E. Noel Jr, and L. A. Tomek, "Wake multiple over lan," U.S. Patent, Sep., 1997.
- [11] C. Michaelides, M. Iloridou, and F. Pavlidou, "In search of the lost nodes in bans," in *2018 IEEE 26th International Conference on Network Protocols (ICNP)*, Cambridge, UK, Sep. 2018, pp. 249–250.
- [12] E. Gamma, J. Vlissides, R. Helm, and R. Johnson, *Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley Professional, Oct. 1994.
- [13] C. Michaelides, M. Iloridou, and F. Pavlidou, "An improved mobility aware relaying scheme for body area networks," *IEEE Sensors Journal*, pp. 1–8, Apr. 2019.
- [14] Castalia. [Online]. Available: <https://github.com/boulis/Castalia/>
- [15] Omnet++. [Online]. Available: <https://www.omnetpp.org/>
- [16] M. Müller, T. Röder, M. Clausen, B. Eberhardt, B. Krüger, and A. Weber, "Documentation mocap database hdm05," Universität Bonn, Tech. Rep. CG-2007-2, Jun. 2007.
- [17] Rope skipping. [Online]. Available: <https://youtu.be/yDXF1JRYPSo>