IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 30, 2016, accepted February 21, 2016, date of current version April 12, 2016.

Digital Object Identifier 10.1109/ACCESS.2016.2539299

A Heuristic Self-Adaptive Medium Access Control for Resource-Constrained WBAN Systems

MUHAMMAD MAHTAB ALAM[1], ELYES BEN HAMIDA[1], OLIVIER BERDER[2], DANIEL MENARD[3], AND OLIVIER SENTIEYS[2]
[1]Qatar Mobility Innovations Center, Qatar Science and Technology Park, Doha 210531, Qatar
[2]Institut de Recherche en Informatique et Systèmes Aléatoires, University of Rennes 1, Lannion 22300, France
[3]Institut National des Sciences Appliquées de Rennes, Rennes 35708, France

Corresponding author: M. M. Alam (mahtaba@qmic.com)

This work was supported by the Qatar National Research Fund (a member of Qatar Foundation) within the National Priorities Research Program under Grant 6-1508-2-616.

ABSTRACT This paper presents a heuristic-based approach to self-adapt and reconfigures the wake-up schedule of the nodes in wireless body area networks (WBANs). A latency-energy-optimized traffic-aware dynamic medium access control protocol is presented. The protocol is based on an adaptive algorithm that allows the sensor nodes to adapt their wake-up and sleep patterns efficiently in static and dynamic traffic variations. The heuristic approach helps to characterize the algorithmic parameters with an objective to investigate the behavior of the convergence patterns of the WBAN nodes in a non-linear system. An open-loop form is developed by keeping the wake-up interval (I_{wu}) fixed followed by the closed-loop adaptive system which updates I_{wu} on every wake-up instant. An exhaustive search is conducted for different initial wake-up interval values which show that (on average) the algorithm parameters behave monotonically in open-loop systems, whereas a decaying function in the closed-loop form. Various performance metrics, such as energy consumption, packet delay, packet delivery ratio, and convergence speed (for reaching a steady state), are evaluated. It is observed that the convergence time varies from 8 to 72 s under fixed packet transmission rate, whereas the algorithm re-converges (within 8 s) whenever the transmission rate changes.

INDEX TERMS Wireless body area networks, dynamic traffic-aware MAC protocol, optimized low duty cycle MAC protocol, WSNet simulator, energy consumption, packet latency, packet delivery ratio, convergence time.

I. INTRODUCTION

Self-organization is extremely vital in resource constrained systems such as wireless body area networks (WBAN). Generally, natural phenomenon and design strategy are two approaches for the self-organization [1]. Natural phenomenon is like a distributed system, where components interact on the small-scale and organize themselves in a way that leads to large-scale organization. Whereas in design strategy, system components have to follow certain local rules intended to reach a desired global behavior.

WBAN is a self-organized network at human body scale. It consists of smart devices which are low-power, miniaturized, hardware constrained, attached to or implant inside a body [2]. Typically, WBAN nodes have limited energy capacity, constraints on the miniature size, low signal processing capabilities and low storage capacity [3]. However, energy efficiency remains one of the most important issue in WBAN. In this work we present an approach towards self-organization with a focus on local energy optimization which can lead towards network level global energy optimization.

In a typical WBAN platform [9], the radio transceiver consumes most of the power. Radio activity is controlled by the medium access control (MAC) layer, therefore it is important to design an ultra low power and energy efficient MAC protocol suitable for WBAN. Whilst, low duty-cycle MAC protocols such as preamble sampling are efficient as they reduce the unnecessary energy waste [10], but their performance (in terms of energy efficiency) is questionable under variable traffic particularly for WBAN nodes in which different sensors have different data rates requirements. Number of different static low-duty cycle MAC protocols exist in the literature but to best of our knowledge there is a lack of dynamic optimized duty cycle protocols specific to real-time traffic variations of WBAN.

In our previous work [11], we introduced a traffic-aware dynamic MAC protocol (TAD-MAC). It relies on an adaptive technique in which every node adapts its wake-up interval (I_{wu}) dynamically with due account of the amount of traffic (i.e. data packets) it receives and consequently optimizes the energy consumption. The algorithm keeps on updating the wake-up interval until it reaches a steady state in which the idle listening is extremely minimized. In this initial study, we proposed the design principle of TAD-MAC algorithm and presented some simulation results under fixed and varying traffic. However, the convergence pattern and specific priorities are not classified and further the energy consumption analysis does not explicitly analyze the convergence behavior and the corresponding cost of reaching towards a steady state. This work is necessary to consolidate and complete our previous study, in particular, to address the important questions; i) whether TAD-MAC protocol can always converge or not?, ii) what is the deterministic cost (i.e., performance loss in terms of energy, packet loss and delay) for reaching to a steady state? Finally, by adding the convergence cost, is TAD-MAC still an energy efficient protocol compare to classical preamble sampling MAC protocols or not?

To address aforementioned questions, this paper provides multiple contributions. First, a simple yet effective traffic estimation technique is proposed with detailed discussions on the convergence patterns of the sensor nodes (at the coordinating node). Second, the adaptive algorithm (proposed in the previous work) is heuristically modeled as a non-linear control system to understand the convergence behavior of the parameters of the algorithm. In this regard, open loop and closed-loop forms are developed and analyzed. Further, a detailed insight of the convergence patterns is presented. Third, a state machine is developed for the MAC protocol and the complete system is implemented in a packet-oriented event driven network simulator (i.e., WSNet [12]) for the performance analysis. The performance metrics, i.e. energy consumption, packet latency, packet delivery ratio, and convergence time, are calculated (for reaching a steady state) for different traffic rates through a small-scale single-hop WBAN network. The results show that I_{wu} converges to a steady state value after a certain number of wake-ups efficiently while configuring the algorithm with suitable parameters. Further, best, worst, and average convergence times are analyzed. Finally, traffic characteristics are exploited from the network simulator to evaluate the energy consumption for both transmit and receive nodes.

The rest of this paper is organized as follows. In section II, the related work is presented followed by TAD-MAC protocol description in section III. The heuristic approach and the system characterization of the TAD-MAC protocol is presented in section IV. Further, the implementation and performance evaluation are explored and discussed in section V. Finally, the paper ends with a conclusion.

II. STATE OF THE ART

Wireless body area networks (WBAN) are different than classical wireless sensor networks by many ways including radio propagation and channel characteristics [7], medium access and communication constraints [6], ultra low power dissipation and consumption [8], nodes density and network topology [4]. Generally WBAN consist of few nodes including coordinator sensors and/or actuator devices as shown in fig. 1. Typically the on-body sensor nodes sense the physiological signals or parameters from the body such as vital signs, ECG, EEG etc., and send it periodically to the on-body coordinating node. The coordinating node is often more powerful (in terms of processing, memory storage, throughput, range etc.), and communicates with the external networks infrastructure.

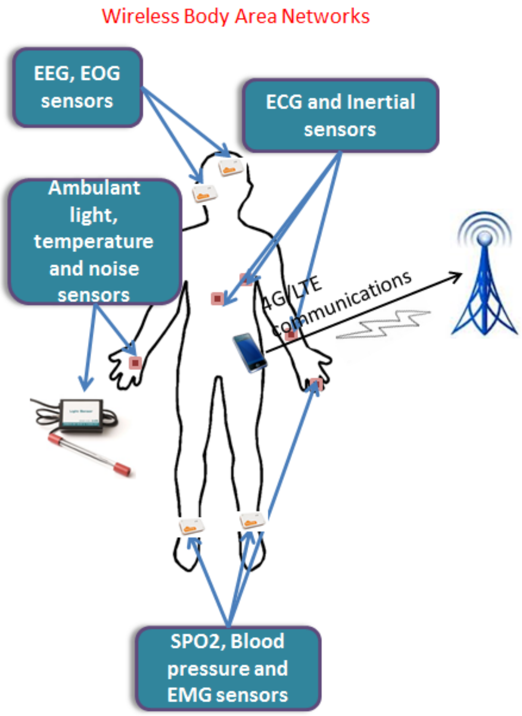

FIGURE 1. A typical wireless body area networks architecture.

Over the last decade, many applications have emerged for wearable sensor networks. While fitness and health-care remain the most dominant, other applications include fashion and entertainment, augmented reality [5], rescue and emergency management are emerging as well [21], [36]. In this regard, contex-aware WBAN are reviewed for medical and non-medical applications in [3]. It highlights the importance of being adaptive to users environment, especially their mental states and biophysical signals to understand the complete situation. To cover diverse range of WBAN applications, initially IEEE 802.15.4 standard was mostly used. However, due to lack of specific WBAN channel characteristics, MAC and PHY features, it was not widely accepted. In 2012, IEEE formally released a WBAN specific standard [15], which is being adapted in many recent research and development.

The IEEE 802.15.6 standard provided a great flexibility to adapt the medium access according to the specific users requirements [15]. The MAC layer includes *Aloha, Slotted Aloha, CSMA/CA, scheduled* and *unscheduled access* as well as *polling* and *posting* channel access mechanisms. In classical health-care WBAN systems, *time division multiple access* (TDMA) based medium access control is most often considered. Every sensor node has a dedicated slot to transfer its data to the other sensors or coordinator. Moreover, works such as [16], [17], and [29] can further help to optimize the slot scheduling based on the traffic load. Historically, limited attention has been given to carrier sense multiple access *CSMA/CA*, however, very-low duty cycle *CSMA/CA* based protocols such as [11] and [28] seem very attractive for ultra low power applications.

In CSMA/CA priorities are assigned based on the traffic, for example emergency traffic has higher priority in comparison to normal WBAN traffic. Further, based on these priorities, contention window sizes and back-off mechanisms are proposed [15]. In scheduled access, TDMA slots can be allocated by the coordinator to specific nodes based on their requirements. To compare these two schemes, scheduled access has very high performance in terms of packets delivery ratio but at an expense of synchronization overheads and the cost of keeping the nodes always in synchronization which is very critical for the complete network performance. Moreover, in time-critical applications (such as [36] and due to dynamic nature of WBAN traffic, scheduled based approaches can be degraded significantly. As per CSMA/CA-based MAC approach is concerned, it consumes more energy but it can be improved provided the packets collisions are minimized and the packet delivery is maximized [14].

To take advantage of both CSMA/CA and TDMA based techniques, in this paper, we use controlled dedicated beacon enabled CSMA/CA approach to minimize the energy consumption and collisions and to improve the packet delivery ratio. This approach minimizes the energy consumption by synchronizing the wake-up schedule and achieve the advantage of TDMA-based mechanisms in a distributed fashion. In this context we present a self-organized preamble sampling MAC protocol.

Various low-power MAC protocols exist in the literature for example, WiseMAC [13] can be considered as the best MAC protocol in the class of preamble sampling, but its performance degrades significantly under variable traffic load. It is important to highlight the exact difference between TAD-MAC and WiseMAC. WiseMAC is used for the downlink of infrastructure network (from access point to a sensor node), which is a low traffic as it only contains the configuration and querying requests [13]. However, in the case of variable and high traffic the performance of WiseMAC degrades rapidly [20]. Whereas, the focus of TAD-MAC is to remain generic and applicable to variable traffic loads to meet the demands of various WBAN applications [21]. Basically, both protocols reduce idle listening by synchronizing the wake-up schedule. Another fundamental difference is that, WiseMAC is based on a fixed wake-up interval which is learned (from the receive node to the transmit node) after every communication between the two nodes but this wake-up interval remains fixed irrespective of variation in the traffic between different sensor nodes. Whereas, TAD-MAC is based on an evolutionary process of evaluating the wake-up schedule of the sensor nodes through an estimation of traffic load and the process is adaptable to variable traffic load. Further, [18] presents an extended WiseMAC to improve the throughput but it does not address the issue of fixed wake-up interval of the nodes which is the main problem under variable traffic load.

Few other MAC protocols such as [19], [20], and [22] are low-energy MAC protocols that optimize the energy waste due to idle listening, collisions and interference. [20] proposed a predictive wake-up (PW-MAC) protocol in which each node wake-up time is calculated through a random generator. All the nodes have knowledge of the initial *seed* value of their neighbor nodes. It is an efficient technique for static networks but it requires certain enhancements in particular to re-configure as soon as the traffic changes to broadcast the new *seed* value to the neighbors. Moreover, the technique has fundamental limitations due to the fact that the random generator is not a precise measure of packet generation. For example, a source node may have a data packet to transmit, but it will delay its transmission until the wake-up time of the receive node, which results in a transmission delay. Furthermore, the protocol is completely distributed and in asynchronous sensor networks there is always certain probabilities of wake-up and data collisions [23], [24] which are not discussed and explored.

Dynamic WBAN MAC protocols is an active area of research especially for low power devices to not only increase their lifetime but also to improve the packet reception performance [30]. For example, works such as [31] and [32], [31] adapt time division scheduled MAC based on the body shadowing due to various postures and corresponding channel variations. In addition, MedMAC [31], propose an energy efficient synchronization mechanism for multi-channel MAC in comparison to IEEE 802.15.4 standard. Similarly in [11] and [33], authors proposed an adaptive MAC protocols based on the variable WBAN traffic load using IEEE 802.15.4 standard. Most of the above mentioned studies focus on the improvements of the energy consumption without considering the applications constraints such as achievable throughput and delay requirements.

To sum up, TAD-MAC adds an important WBAN dynamic protocol in the class of preamble sampling MAC protocols. In this paper the behavior of the TAD-MAC protocol is modeled, characterized and validated in terms of its convergence towards a steady state through rigorous experiments. Further, the exact cost of reaching to the steady state is evaluated.

III. TRAFFIC-AWARE DYNAMIC MAC PROTOCOL

Wake-up interval and wake-up time are the most important design parameters in the preamble sampling class of MAC protocols. For periodic sensing and variable sensing applications, the wake-up interval is usually kept fixed, which results in degrading the performance as well as the energy efficiency when traffic and data rates vary. In this section a dynamic MAC protocol (called traffic-aware dynamic MAC, i.e., TAD-MAC [11]) is presented which adapts the wake-up schedule based on the estimation of the traffic load. The proposed approach can further be applied to other low power asynchronous MAC protocols such as B-MAC [26], X-MAC [27], WiseMAC [13], [13], CyclED Receiver [25], RI-MAC [19] with minor modifications suggested in [11].

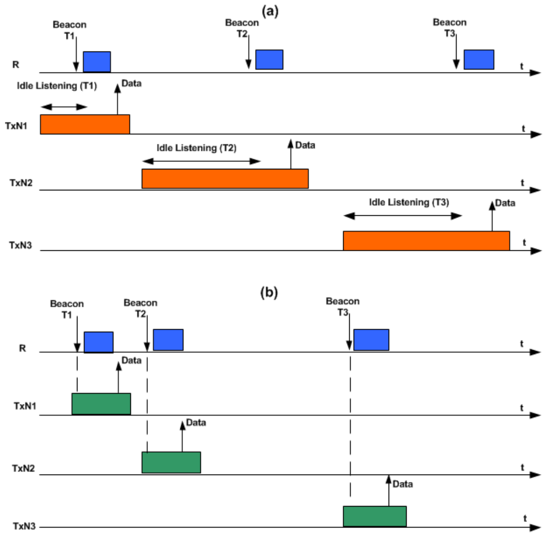

FIGURE 2. Traffic-aware dynamic MAC protocol is divided into two phases; the evolution phase before convergence (a) and the steady state phase after convergence (b). It is important to note that in the second phase the receive node (coordinator) has adapted its I_{wu} schedule as such that the idle listening is reduced [11].

A. PROTOCOL DESCRIPTION

Figure 2 describes the basic principle of the TAD-MAC protocol (which is initiated by the receive node). It shows an example of a small network which includes one receive node and three transmit nodes. The figure is divided into two phases (i.e., (a) before convergence and (b) after convergence), and presents the communication of the three transmit nodes (*TxN1* to *TxN3*) attempting to transmit data to a coordinating node (*R*). During the first phase (which we termed as an 'evolution phase'), before reaching a steady state (Fig. 2-a), each transmit node (*TxNi*) waits for the beacon signal from the coordinator before sending its data. The beacon packet is solicited to a specific transmit node containing its unique node ID (identifier). Whereas, other intending transmit nodes continue to wait for their respective beacons. After several wake-ups, the coordinating node adapts its I_{wu} based on the traffic it receives from each transmit node.

In the second phase (i.e., after reaching the convergence as shown in Fig. 2-b), the coordinator has adapted its I_{wu} in such a way that idle listening is minimized. In order to cater for the clock drift and hardware latencies, the receive node sends the wake-up beacon (WUB) slightly after its scheduled time to ensure that the intending transmit node is already awake.

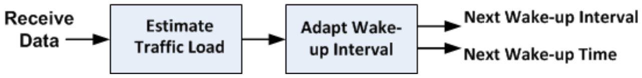

FIGURE 3. Traffic-aware dynamic MAC protocol consists of two components (i.e., traffic estimation and adaptive wake-up interval algorithm).

The main novelty of the TAD-MAC protocol exist in dynamic adaptation of the wake-up interval under variable traffic. There are two fundamental parameters (i.e. wake-up interval and wake-up time) that are required to achieve synchronization between transmit and receive nodes. Fig. 3 shows the basic building blocks of the TAD-MAC protocol. Since TAD-MAC is based on receiver-initiated mechanism, the receive node estimates the traffic load and based on this estimation it controls/adapts the wake-up interval in order to synchronize with the data transmission time of the transmit node. Consequently the next wake-up time and interval are calculated which is used by the coordinating node.

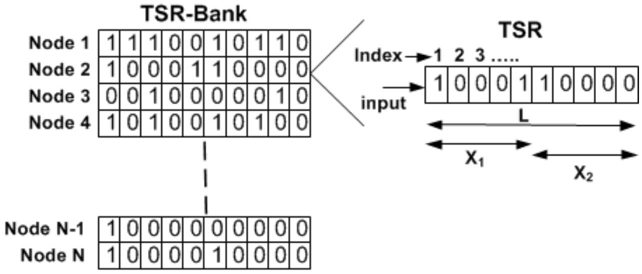

FIGURE 4. Traffic status register bank: it contains *N* registers for *N* neighbor nodes. If the receive node receives data, the register (of the corresponding transmit node) is filled with '1' and if it does not receive data, the TSR is filled with '0' [11].

B. TRAFFIC ESTIMATION

The traffic estimation technique has to be effective as well as low in complexity such that it can be executed fast enough under constrained processing unit. In this regard, we introduce a traffic status register (TSR) that contains a status information of the traffic at each node. As the receive node wakes-up, it sends a wake-up beacon, if it receives a data in response, the TSR corresponding to the transmitted node is filled with a value '1' and if it does not receive data, the TSR is filled with a value '0'. The new status (either 1 or 0) is filled in the 1st index of the register as shown in Fig. 4. The register contents are shifted one bit right before inserting a new status. This is a blind way of keeping the track of traffic load without having any information about the data rate of the transmitter. However, this blind estimation is not a quantitative evaluation of the data rate but rather a relative measure of either the traffic is increasing (which means multiple consecutive ones in TSR) or decreasing (which means multiple zeros in TSR).

In this context, for the adaptation of I_{wu}, each node contains a traffic status register bank (as shown in Fig. 4), to keep the traffic status of all the neighbor nodes. Each register corresponds to an individual neighbor node and is updated (as either '1' or '0') based on the either received data packet or not. The receive node wakes up and sends its beacon (solicited) to the node which has the nearest wake-up time T_{wu}. The wake-up interval for an individual node is estimated based on the contents of the traffic status register. The register is divided into two halves in order to consider a dominant impact from the most recent traffic (which resides in *most significant* part) in comparison to the relatively old traffic (which resides in the *least significant* part).

The size of the register is a parameter that can be tuned to achieve fast convergence speed as detailed in [11]. The TSR can have different patterns depending upon the variations in the traffic and it is important to mention the selection criteria for the desired TSR pattern or a sequence of convergence which is targeted for TAD-MAC protocol. The algorithm (presented in Sec. III-C) is designed in such a way that as it converges to a steady state value of the wake-up interval, the TSR contains a sequence of [10101010 . . .] pattern. This typical sequence seems the best trade-off between the optimal wake-up interval [11111111 . . .] (i.e. each wake-up of the node is followed with a successful data reception), and too frequent wake-up [100010001 . . .] due to several reasons:

- Sequence of [11111111 . . .] is an ideal sequence which means that each time the receive node wakes up it receives a data packet. However, in addition to transmission losses especially due to WBAN's shadowing (i.e. space and time varying radio links [34]), it always has the probability of missing data packet as soon as the traffic rate increases (at the transmit side). The wake-up interval of the receive node will keep converged at lower traffic rate (irrespective of the fact that the traffic rate has increased at the transmit side) because of [11111111 . . .] sequence but it will lost number of packets. Moreover, the packets keep on aggregating at the transmitter side which will degrade the performance in terms of energy consumption, packet delay and packet delivery ratio.
- Whereas, if the traffic rate reduces then on the next wake-up, [11111111 . . .] sequence will become [01111111 . . .] and consequently the wake-up interval will no more remained converge and therefore the convergence process will start again. So in the case of reducing the traffic rate there is no problem while selecting [11111111 . . .] sequence because it can learn that the traffic has reduced and the wake-up interval should be reduced. But it is not prone to accommodate the lost packets.
- With regards to the sequence [10001000 . . .], the receive node wakes up too often and results in unnecessary beacon transmission and energy wastage.
- [11101110 . . .] sequence is another alternate which increases the transmission capacity and can be more energy efficient. However, its response to an increase traffic is slow because in worst-case it will take $(\frac{N}{2} - 1)$ more wake-ups to be able to know that the traffic has increased. This is especially critical for the burst traffic when the response has to be extremely quick.
- [10101010 . . .] sequence seems to be the best trade-off between [11111111 . . .], [11101110 . . .] and [10001000 . . .] sequences under the above considerations and is selected as a steady state in this work.

C. ADAPTIVE ALGORITHM

The adaptive algorithm computes the wake-up interval (I_{wu}) based on the contents of the TSR for a specific node as shown in Fig. 4. The wake-up interval is updated for the time instant (t_{i+1}) based on the previous time instant (t_i), and update factors as,

$$I_{wu}(t_{i+1}) = I_{wu}(t_i) + [\mu(t_i) + e(t_i)] \cdot t_{ref} \cdot \tag{1}$$

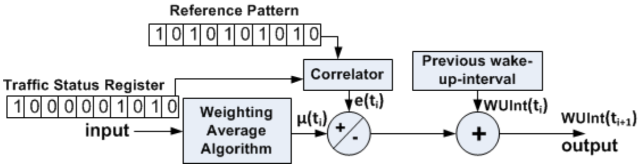

FIGURE 5. Adaptive wake-up interval system [11].

Where, μ is the output of the weighting average algorithm and e is the correlation error as shown in Fig. 5. In order to have a unique reference of time among I_{wu}, μ, and e, the t_{ref} is multiplied with μ and e, and it is defined with reference to the system/simulator clock. The update factor $\mu(t_i)$ is calculated based on the computation of two weighted values (i.e., most significant $X_1(t_i)$ and least significant $X_2(t_i)$) as shown in Fig. 4) as

$$\mu(t_i) = \alpha \cdot X_1(t_i) + (1 - \alpha) \cdot X_2(t_i), \tag{2}$$

where α is a constant weighting factor whose variations and impact can be seen in [11]. $X_1(t_i)$ and $X_2(t_i)$ are defined as

$$X_1(t_i) = \frac{N_{0,1}}{L/2} \cdot Nc_{0,1} - \frac{N_{1,1}}{L/2} \cdot Nc_{1,1}, \tag{3}$$

$$X_2(t_i) = \frac{N_{0,2}}{L/2} \cdot Nc_{0,2} - \frac{N_{1,2}}{L/2} \cdot Nc_{1,2}, \tag{4}$$

where $N_{0,1}$, $N_{1,1}$, $Nc_{0,1}$, and $Nc_{1,1}$ are the number of zeros, number of ones, number of pairs of consecutive zeros and number of pairs of consecutive ones in X_1 respectively, whereas, $N_{0,2}$, $N_{1,2}$, $Nc_{0,2}$, and $Nc_{1,2}$ are the number of zeros, number of ones, occurrence of consecutive zeros and the occurrence of consecutive ones in X_2 respectively. For example, a sequence [11100100] has $Nc_{1,1}$ equals to 2 and $Nc_{0,1}$ equals to 0, whereas, $Nc_{1,2}$ is equal to 0 and $Nc_{0,2}$ equal to 1. The optimization parameter L is the length of the TSR, which can be tuned to achieve fast convergence in the context of WBAN as explained in [11].

The adaptive wake-up interval system is illustrated in Fig. 5. After the weighting average algorithm computation of the wake-up interval $\mu(t_i)$ through (2) to (4), a cross correlator is used to smooth the output of the weighting average algorithm. The correlator is fed with the current contents of TSR and it computes a correlation after a comparison with the reference patterns that is [10101010...]. The correlator provides the output error either positive (which means that the TSR sequence contains more zeros than ones, hence, the correlator output should contribute to increase the wake-up interval) or negative (which means that the TSR sequence contains more ones than zeros, hence, the correlator output should contribute to decrease the wake-up interval). Consequently it guides the adaptation of wake-up interval towards the desired sequence. Finally, the previous wake-up interval value is added with $\mu(t_i)$ and $e(t_i)$ to compute the next wake-up interval at the output according to Eq. 1.

In the proposed algorithm the most influential parameter is Nc (number of consecutive pairs of zeros or ones). Even though there are multiple ones or zeros in the TSR, they will not make an impact on the update value of wake-up interval until there are consecutive pairs of zeros or ones. Multiple consecutive zeros employ that the next wake-up interval should be increased in comparison to the previous value, whereas multiple consecutive ones imply that the next wake-up interval should be decreased.

IV. HEURISTIC-BASED APPROACH FOR SYSTEM CHARACTERIZATION

To effectively exploit TAD-MAC protocol, it is necessary that the adaptive algorithm should always converge towards a steady state. To validate that, in particular, the parameters of Eq. 1, (i.e., I_{wu}, μ, and e) are investigated with the objective to understand their behavior. To observe and monitor the behavior of the system, it is characterized as a control system (i.e., open-loop and closed-loop forms). The motivation of this characterization comes from the fact that, WBAN is a non-linear system with various on-body sensors consistently vary over space and time, so channel link between the sensors and on-body coordinator keeps changing [34]. In addition, both hardware and software delays are non-linear with reference to the time. Therefore, at first, it is important to model an open-loop behavior in order to understand the non-linearity that exists in the system.

The system Eq. 1 contains two fixed parameters (initial wake-up interval $I_{wu}(t_0)$ and reference time t_{ref}) and three variable parameters (wake-up interval I_{wu}, update factor μ and error factor e). First in the open-loop form, two variables μ and e are investigated against number of different (fixed) I_{wu} values (that is same as $I_{wu}(t_0)$ value) and a fixed t_{ref} value (without adaptation of the wake-up interval). Second, all three variables (including I_{wu}) are explored in a closed-loop where I_{wu} is updated at each wake-up and the experiments are conducted for number of different I_{iwu} values and fixed t_{ref}.

Let us first analyze the formal description of the adaptive algorithm to understand the complexity of the convergence problem. Let us note w_i, the vector of length L representing the content of the TSR register at time index i such as

$$w_{i,j} = [w_{0,i}, w_{1,i}, \cdot \cdot \cdot \cdot w_{L-1,i}] \tag{5}$$

with

$w_i \in [0, 1]$ and $\forall j \in [1, \cdot \cdot \cdot n_r]$, where n_r is the number of neighbor nodes. The number of ones and zeros and consecutive ones and zeros in the $w_{i,j}$ for a given time index i are calculated through $N_{1,b}$, $N_{0,b}$ $Nc_{1,b}$ $Nc_{0,b}$.

$$N_{1,b} = \sum_{j=(b-1)L/2}^{b((L/2)-1)} w_{i,j} \tag{6}$$

$$N_{0,b} = \sum_{j=(b-1)L/2}^{b((L/2)-1)} (1 - w_{i,j}) \tag{7}$$

$$Nc_{1,b} = \sum_{j=(b-1)L/2}^{b(L/2-1)} w_j \cdot w_{j+1} \tag{8}$$

$$Nc_{0,b} = \sum_{j=(b-1)L/2}^{b(L/2-1)} (1 - w_j) \cdot (1 - w_{j+1}) \tag{9}$$

where b $\in$ $\{1, 2\}$. The algorithm converges to a steady state, i.e. $I_{wu}(t_{i+1}) = I_{wu}(t_i)$ as soon as the following two conditions are met:

$$Nc_{1,b}(t_i) = Nc_{0,b}(t_i) = 0 \tag{10}$$

$$Nc_{1,b}(t_{i+1}) = Nc_{0,b}(t_{i+1}) = 0. \tag{11}$$

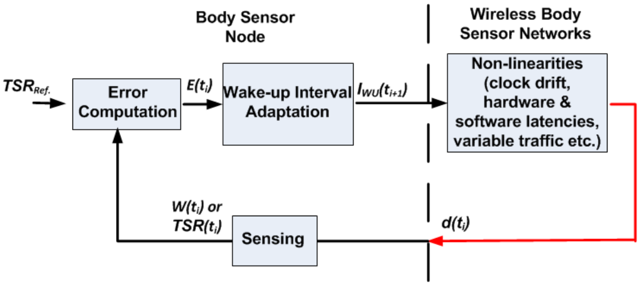

FIGURE 6. System model used to characterize the adaptive wake-up interval algorithm.

Figure 6 shows the model of the complete system from a sensor node perspective. The model is initialized through an error computation block which provides an input to the wake-up interval adaptation block to update the wake-up interval. Further, this value is used by a node to wake-up and sense the channel if there is any data (represented as $d(t_i)$) for it. It is important to note that WBAN has non-linear behavior (due to variable traffic, clock drift, hardware/software latencies, channel variations, mobility etc.), and it is quite complex to develop analytic models to justify above conditions of convergence under these variations. Therefore, an experimental search is conducted (as explained later in this section) to show the behavior of all the variables and parameters of the adaptive algorithm.

The simulation setup considers different traffic rates, packet length of 16 bytes, free space propagation and binary phase shift keying (BPSK) modulation scheme. 0dBm of transmit power is used and receiver sensitivity of −92dBm is considered. The application model is basically controlled by the transmission rate of $1,\ 2\ and\ 5\ packets/s$, for open-loop and closed-loop experiments. The network is simplified up to three nodes (two receive nodes and one transmit node) as the purpose is to understand the behavior of the variables and parameters of the protocol therefore a small scale WBAN is sufficient. At the MAC layer, maximum waiting time at the transmit node is set as $500ms$ and the acknowledgment waiting time is $5ms$. Whereas, maximum waiting time for receiving the data (at the receive node) after it transmits a beacon is $5ms$. A state machine is implemented at the MAC layer that includes an adaptive algorithm and all the necessary communication states of the TAD-MAC protocol. More details about the execution of state machine can be found in Sec. V.

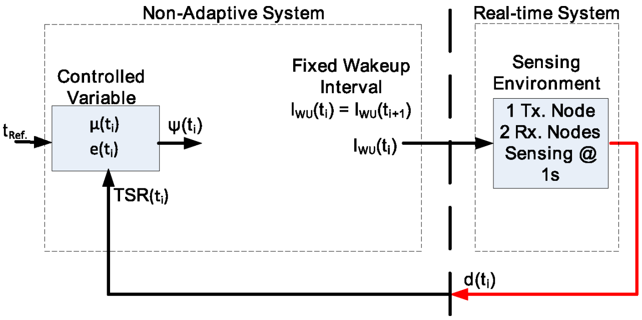

FIGURE 7. **The open loop form of the system consists of two parts. The first one is a non-adaptive system that computes ψ (for the purpose of analyzing controlled variables) but does not use it to update the wake-up interval whereas, the second part is real-time system which is the wireless sensor network.**

A. OPEN LOOP SYSTEM

The purpose of the open loop form of the system is to evaluate the behavior of controlled variables under static condition (i.e., fixed wake-up interval). The open loop form of the system, (shown in Fig. 7), is divided into non-adaptive and real-time systems. In an open-loop form, the wake-up interval is not updated and remains fixed therefore, it is termed as non-adaptive system. The wake-up interval is used by a node to sense the environment and is initialized by a fixed value. The purpose of the open loop form of the system is to evaluate the behavior of controlled variables under static conditions (fixed wake-up interval) which remains constant during the whole observation time.

A sensor node wakes up and receives a value that can be either '1' or '0' which is used to update the contents of the TSR register. The contents of the TSR-register at time t_i are used to compute the controlled variables $\mu(t_i)$, and $e(t_i)$ and their behaviors are individually explained below, whereas, the symbol $\psi(t_i)$ is the output of the product of the t_{ref} and the controlled variables as shown in Fig. 7. The controlled variables can be combined or separately selected to constitute ψ (depending upon the trade-off between performance and complexity). In the open-loop form even though $\psi(t_i)$ is computed, though as the wake-up interval is kept fixed, it does not have any influence for sampling the next value.

For the transmission of $1\ packet/s$ which means that the coordinating (receive) node will have $500ms$ as the optimal wake-up interval to match the selected reference pattern (i.e., [10101010...]). Now the number of different wake-up interval values are chosen starting from $100ms$ until $1000ms$ with an increment of $50ms$ and three variables are observed. It is to note that average values of all the controlled variables are calculated for a given time instant over the length of the TSR-register, i.e. L (which is kept to a fixed value and is equal to 8).

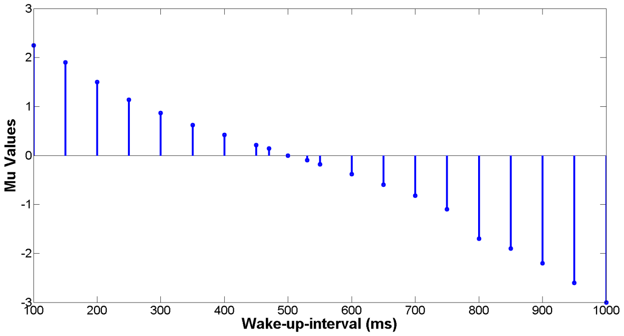

FIGURE 8. **Variations of mu variable against different (fixed) wake-up interval values.**

Figure 8 shows the behavior of μ (which is computed through Eq. 2) versus different wake-up interval values. It can be noticed that when the wake-up interval reaches $500ms$, then the μ value becomes 0, whereas on the two other extremes (such as 100ms and 1s) it has maximum opposite value. This is due to the fact that with $500ms$ value of wake-up interval the contents of TSR-register perfectly match with the desired pattern, which means that the number of zeros and ones are equal without consecutive pair of zeros or ones in TSR and consequently μ is zero. Each fixed wake-up interval value results in fixed periodic sequence (TSR-Register contents) and an average value is calculated from the periodic sequence which is shown in Fig. 8. It can be seen that, the variation of μ is monotonic and almost linear, hence it is predicted that μ will always helps to converge the algorithm towards a steady state as the second order derivative of this monotonic function is zero.

Figure 9 shows the behavior of variable e computed through correlation. The correlation is performed between a reference pattern and the contents of TSR. Mean and absolute values of the correlation are calculated over the length of TSR (i.e. for one current and seven previous values of correlation). It can be seen that at a $500ms$ value of wake-up interval, both sequences (i.e. reference and TSR) perfectly match therefore, the mean value of correlation is 0 whereas,

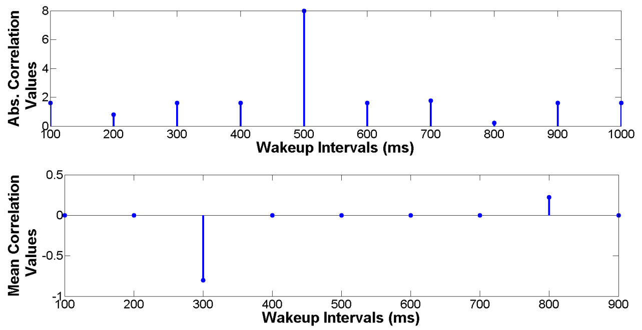

FIGURE 9. Behavior of *e* variable against different (fixed) wake-up interval values calculated through correlation technique. First is the mean of the absolute value of correlation and second is only the mean of correlation.

the absolute of mean value is 8 (maximum). It can be observed that correlation values in two different forms have predictable behavior (at 500ms) as can be seen in Fig. 9. However, it has non-monotonic behavior for most of the wake-up interval values.

To conclude the open-loop system, it is observed that the controlled variable μ is a suitable parameter and can be considered in a closed-loop form of the system. Whereas, variable e (computed through correlation technique) has non-linear behavior and can not be considered for the adaptation of the wake-up interval in the context of real network.

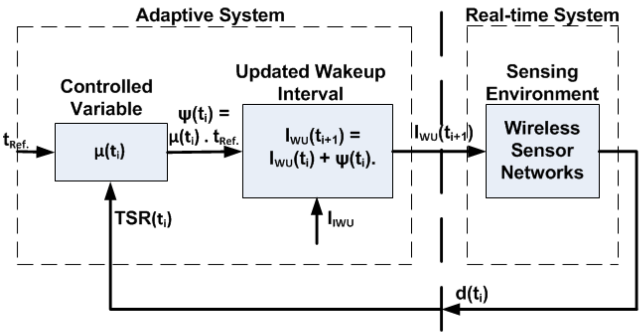

FIGURE 10. Closed-loop form of the system considers the influence of feedback into the computation of the update of the wake-up interval in the adaptive system. It is to note that only μ is considered as controlled variables in the adaptive system.

B. *Closed-Loop System*

The closed-loop form of the system, shown in Fig. 10, is divided into adaptive and real-time systems. The main difference between open-loop and closed-loop forms is that ψ is used to update the next wake-up interval that is used by a sensor node to wake-up and sense the environment in closed-loop. At the start, the wake-up interval is initialized and afterwards it is updated at each time index (wake-up instant). A sensor node samples the environment with this updated time interval and receives a value $d(t_i)$ which updates the contents of the TSR register.

The content of the TSR-register is used to compute the controlled variable μ. In the closed-loop form, the experiments are conducted with fixed data transmission time of 500ms between two packets (packet rate) and receive node adapts its wake-up interval according to this rate. In this regard, different initial wake-up interval values are used ranging from 100ms up to 1000ms. It is to mention here that, as the receive node reaches a steady state, it has a value of wake-up interval exactly half to that of the packet rate (which is the target rate of convergence), because of the desired sequence [10101010...].

In order to achieve the best convergence time which helps to reduce energy consumption and delay with least processing power (i.e. by utilizing less parameters), following experiments compute ψ by using only μ.

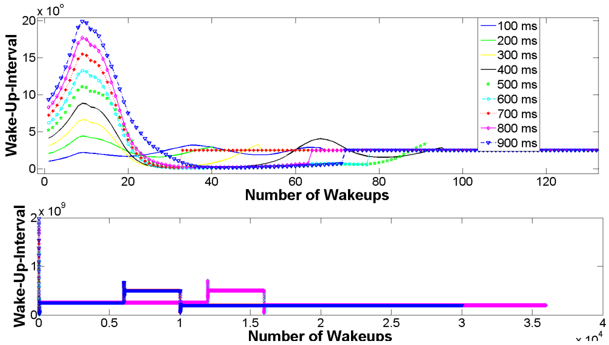

FIGURE 11. Wake-up interval adaptation for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one, also showing the behavior of convergence under variable traffic.

Figure 11 shows the behavior of the wake-up interval adaptation for number of different initial wake-up starting values. It can be seen that by using μ as a controlled variable for the update of wake-up interval helps to improve the convergence speed. Moreover, the adaptation resembles to typical adaptive or control systems with monotonically decaying towards a steady state without any continuous oscillations. Fig. 11 is divided into zoomed version and full version in which traffic rate also changes from 500ms to 1000ms and further 400ms and it is observed that under variable traffic receive node adapts very fast to new steady state value.

Figure 12 shows the variations of μ in closed-loop form with adaptation of wake-up interval. It can be seen that μ variations in all different initial wake-up interval values is a decaying function and reaches back efficiently towards a steady state as soon as the traffic rate changes.

The closed-loop form using μ performs much better both in terms of convergence speed and with minimum processing power along with an adaptive system without oscillations [35]. Number of different sensing rates are verified and few of them are presented with traffic statistics and their results in next section. It is important to mention here that once a node converged towards a steady state it takes much less time to converge back after the traffic rate changes, which shows that TAD-MAC performs much better

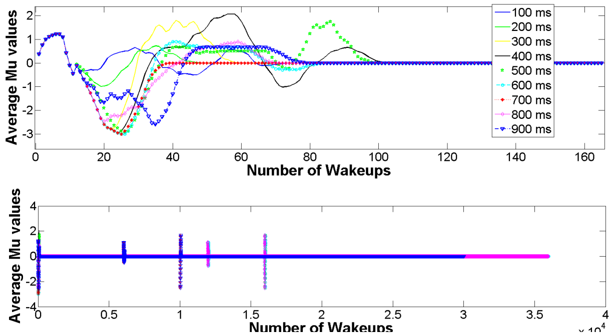

FIGURE 12. Behavior of μ for number of different initial wake-up interval values. Top figure is a zoomed version of the bottom one.

in comparison with other protocols under variable traffic as reported in [11].

V. IMPLEMENTATION AND PERFORMANCE EVALUATION

In this section the implementation details of TAD-MAC protocol in a network simulator are explained. Mainly the performance metrics such as energy consumption, packet latency, packet delivery ratio and nodes convergence time are presented.

During the implementation, it is observed that by applying an adaptive algorithm using μ to reach a steady state is not sufficient because μ can be zero (which means no update in the wake-up interval) through 10 different sequences whereas only one of them is a desired sequence of convergence. To resolve this issue two different processes are implemented, i.e. tracking and locking. During the tracking phase the algorithm allows the node to converge to a steady state (that can be any of the 10 sequences) which can result in μ equals to zero. However, the value of the wake-up interval achieved with this sequence may not be the best value. Therefore, to ensure that the node converges only to [10101010] sequence, wake-up interval is locked only when a node has converged to it and not when it has converged to other sequences. Locking process is divided into two steps, in the first step the difference of the wake-up time between the transmit and receive node is calculated and added in the current wake-up interval (at the receive node) to compute next wake-up interval. In the second step the receive node adjusts its wake-up time according to transmit node by knowing the waiting time of the transmit node (that is transmitted inside the last received data packets MAC frame header).

Figure 13 shows the flow of the state machine used to simulate TAD-MAC protocols in the WSNet simulator. The flow depends for example, if a node has a data packet to transmit it follows the transmit mode (*Tx Flow*) and if it has no data it follows the receive mode (*Rx Flow*). The scheduler moves to the next state based on the occurrence of an event, otherwise it keeps on waiting until the timer expires and then it proceeds backward. For example, a node wants to transmit a data packet and it waits for the wake-up beacon from the receive node; if it does not receive a beacon in the *maximum waiting time for beacon* then it goes to 'sleep state'. On the other hand if a transmit node receives a beacon from the receive node it moves forward to next state, i.e. the clear channel assessment (CCA) state.

The TSR contents are updated in the *RX Flow* during the 'wait for data state'. If a node receives a data, then, '1' is inserted in TSR otherwise '0' is inserted. Whereas the adaptive algorithm is implemented in the 'adaptive algorithm state' just before going to sleep. This state computes the variable μ, the *next wake-up interval*, and the *next wake-up time* of the receive node. Executing 'adaptive algorithm state' just before going to sleep has two advantages, first it avoids any delay during the communication between two nodes and second radio transceivers can be switched off after the completion of transmission and reception, which avoids extra cost (energy, delay) due to processing of an algorithm as only the microprocessor remains active.

The 'adaptive algorithm state' also evaluates which specific neighbor (transmit node) to be served (on the next wake-up) with a wake-up beacon. For this purpose every (receive) node contains a bank of TSR as shown previously in Fig. 4, where each TSR (that is specific to each neighbor) evaluates the next wake-up time for that neighbor. During rigorous simulations in WSNet it is found that a conflict (due to overlapping) of serving a wake-up beacon among multiple nodes can occur before all the nodes reach a steady state. In order to deal with such conflicts, certain priorities are defined at the receive (coordinating) node.

- **Strict Priority**: If a (transmit) node 'A' is already synchronized, whereas the (transmit) node 'B' is not yet synchronized with regards to receive node 'R', then node 'B' has a higher priority over node 'A', when a wake-up beacon conflict occurs between them. The node 'B' has higher priority because of two reasons. First, node 'A' which is already converged to a steady state with respect to 'R' demands only minor update at 'R' in its corresponding TSR. The receive node will insert dummy information '1' if it did not received data packet when it last time served a wake-up beacon to node 'A', whereas, '0' if it had received data last time to keep the convergence of node 'A' intact. Second, giving higher priority to the node 'B' which is not yet synchronized allows its evolution/convergence process to not deviate as it may be close to converge. It is important to mention here that, if the data rate changes at the time of conflict, then multiple ones or zeros will appear in TSR (within 2 or 3 wake-up) and as a result the process of re-convergence to another traffic rate will be started.
- **Lenient Priority**: Either if node 'A' and node 'B' are both already synchronized with regards to 'R', or both are not yet synchronized, then any one of them can be selected to be served with a wake-up beacon. Though, it is important to note that in case of both 'A' and 'B' are synchronized and suppose node 'A' is selected, then the TSR contents of node 'B' must be updated depending upon if it has to receive '1' or '0', as explained above.

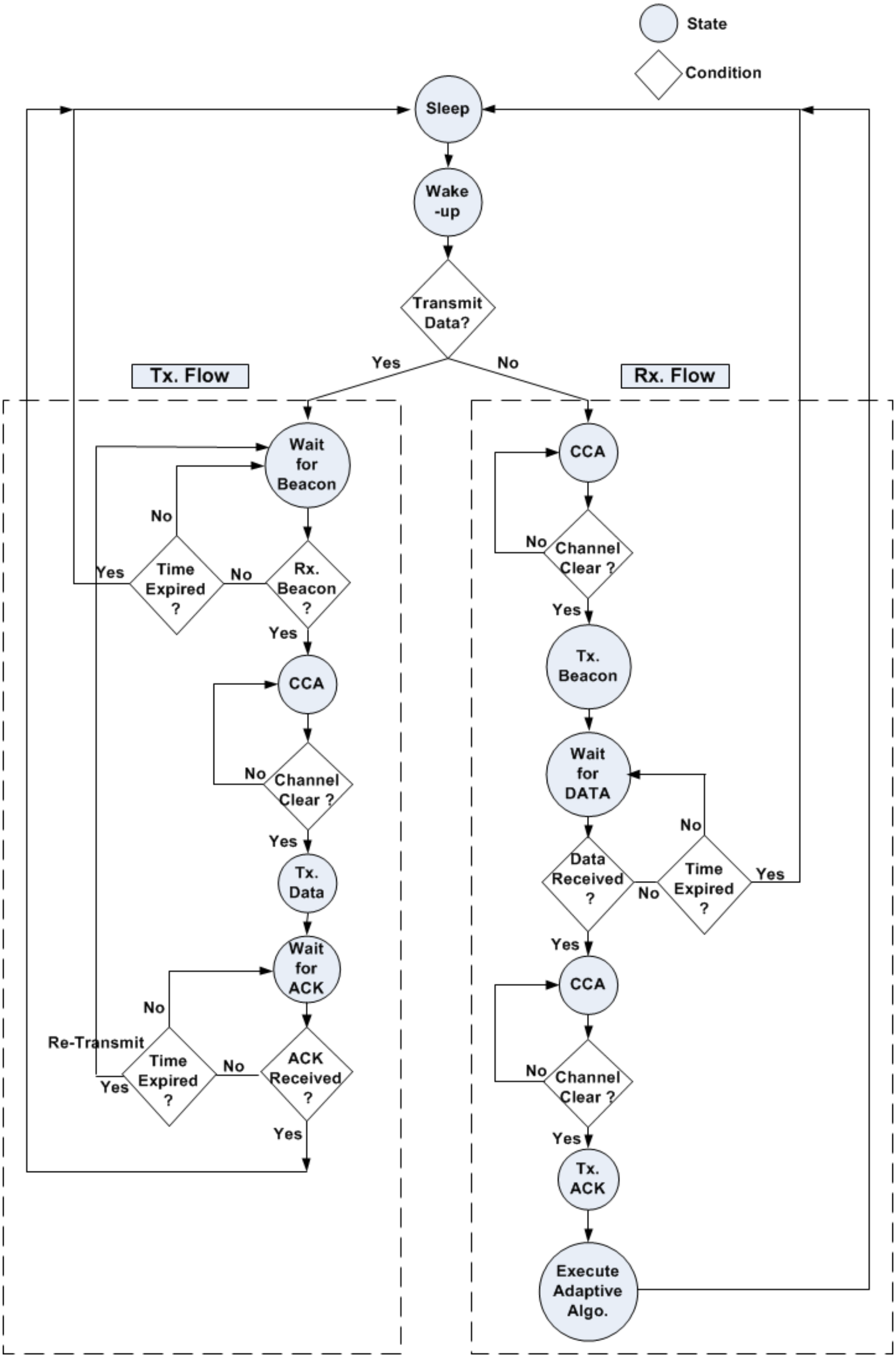

FIGURE 13. State machine being used to simulate TAD-MAC protocols in WSNet simulator. The scheduler moves to the next state based on the occurrence of an event, otherwise it keeps on waiting until the timer expires and then it proceeds backward.

The focus of this section is to evaluate the cost of the evolution phase for reaching a steady state because as the nodes are converged then all the performance metrics are optimized and those results were presented previously in [11]. In this regard, number of data and control packets transmitted and received for reaching a steady state are exploited from the network simulator whereas our precise real-time hybrid energy model presented in [23] is used for the evaluation of accurate energy consumption.

In this regard many experiments are conducted with varying traffic rates and two of them are discussed as an examples. The experimental setting is the same as explained in Sec. IV.

TABLE 1. **The number of beacons and acknowledgments transmitted and the number of data packets received by the WBAN coordinating node before reaching the convergence for sensing rate of 1s and 500ms respectively. Three cases (average, best and worst) are presented based on the performance of three (average, best and worst) initial wake-up interval values.**

Sensing rate = 1s	**Best Case**	**Average Case**	**Worst Case**
Number of Beacons Transmitted	22	72.9	150
Number of Data Packets Received	8	37.6	73
Number of Ack. Transmitted	8	37.6	73
Sensing rate = 500ms	**Best Case**	**Average Case**	**Worst Case**
Number of Beacons Transmitted	33	53	83
Number of Data Packets Received	16	31	42
Number of Ack. Transmitted	16	31	42

The convergence cost in terms of the traffic statistics of the WBAN coordinator are shown in Table 1. It includes the number of beacons and acknowledgments transmitted by the receive node and the number of data packets received by the receive node before reaching a steady state. The results presented are for the sensing rate of 1s and 500ms. Generally the traffic loads differ from one node to another depending upon the specific physiological monitoring signals, nodes position in the networks. For example, the sink node and the ones close to it have much more load in comparison to other nodes. The purpose of Table 1 is to understand how many data and control packets are necessary before the transmit and receive nodes is become synchronized.

Table 1 contains three different cases; *best case* is the fastest convergence case, *worst case* represents the longest time to converge and then the *average case* provides the average value. These three cases correspond to different initial wake-up interval values (ranging from 100ms to 1s) that are used to extract the convergence characteristics. It is observed that by using 100ms as an initial wake-up interval value nodes always converge faster for both sensing rates. Moreover it can be noticed that there exists a great variation between Table 1 and then, from best to worst cases of two traffic rates. Whereas, the average case is much closer in both traffic rates.

Table 2 shows the performance metrics results for three different cases, i.e. *best case*, *average case* and the *worst case* that corresponds to different initial wake-up interval values for two traffic rates. These performance metrics are number of packets being lost, latency (i.e. idle listening) and convergence time before reaching a steady state and are explained below.

- **Packet Losts**: It is the total number of data packets being lost before reaching towards a steady state.

TABLE 2. **Performance metrics for sensing rates of 1s and 500*ms* before reaching a steady state.**

Sensing rate = 1s	**Best Case**	**Average Case**	**Worst Case**
Number of Packet Lost	2	12.5	30
Latency (sec.)	1.52	6.57	15.33
Convergence Time (sec.)	8.43	45.5	72.59
Sensing rate = 500ms	**Best Case**	**Average Case**	**Worst Case**
Number of Packets Lost	0	8	12
Latency (s.)	2.18	4.34	5.7
Convergence Time (s.)	9	21.8	38

  Since we did not consider any first in first out (FIFO) in our implementation, a packet is considered to be lost if the waiting time is finished to receive a beacon. It can be seen from Table 2 that, on one extreme number of packets being lost can be 0 which means 100% delivery ratio, whereas, on the other extreme number of packets being lost can reduce the packet delivery ratio until 70% and hence increase the time of convergence and as a result impact on the energy consumption as well.
- **Latency**: It is defined as the time a transmit node has to wait before it transmits a data packet. In TAD-MAC protocol we termed this time as *idle listening*, further the total time (which is an addition of all the *idle listening* until a node reaches a steady state) is shown in Table 2. Since it is the transmit node who has to wait for the beacon before being able to transmit therefore all the contribution and impact of latency is from transmit node. For the best case it is 1 or 2 seconds, whereas in the wort case it can reach up to 15s.
- **Convergence Time**: It is the total time before a transmit node converged with respect to the receive node. The fast convergence speed is the best in terms of energy, latency and packet delivery ratio. The best and worst convergence times are 9s and 73s respectively whereas on average it is around half a minute.

TABLE 3. **Total energy consumed by a receive node before reaching towards a steady state for sensing rate of 1*s* and 500*ms*.**

Rx Energy Consumption		**Best Case**	**Average Case**	**Worst Case**
Beacon (mJ)	Table I	1.3	4.4	9.1
Data (mJ)	Table I	1.2	5.5	10.8
ACK (mJ)	Table I	0.04	1.9	3.7
Beacon (mJ)	Table I	2.0	3.2	5.0
Data (mJ)	Table I	2.4	4.6	6.2
ACK (mJ)	Table I	0.08	1.6	2.1

The results for the energy consumption of the receive and transmit nodes are presented in Table 3 and Table 4 respectively. The detail computations are explained below.

- **Energy Consumption (RX)**: It is the total energy consumed by a receive node before reaching a steady state.

TABLE 4. Total energy consumed by a transmit node before reaching towards a steady state for transmission rate of 1 $packet/s$ and 2 $packet/s$.

Tx Energy Consumption		Best Case	Average Case	Worst Case
Idle Listening (mJ)	Table II	100	450	1005
Data Transmission (mJ)	Table I	0.5	2.6	5.0
Idle Listening (mJ)	Table II	150	297	390
Data Transmission (mJ)	Table I	1.1	2.1	2.9

Table 3 shows the energy consumption at receive node by calculating the total number of beacons being transmitted, total number of data packets being received and total number of acknowledgments being transmitted. The best, worst and average values corresponds to different initial wake-up interval values are presented. Table 1 is used and the energy consumption of the receive node before reaching a steady state is calculated through accurate energy model [23]. It provides the energy consumption of the wake-up beacon, acknowledgment, and data transmission/reception, that are used to calculate the energy consumption presented in Table 3.

- **Energy Consumption (TX)**: It is the total energy consumed by a transmit node before reaching a steady state. Table 4 shows the energy consumption of the transmit node due do transmission of data packets as well as due to waiting time (i.e. *idle listening*) to receive wake-up beacon and are presented similarly for two different data rates (i.e. 1s and 500ms). The total number of data packets transmitted by the transmit node is the sum of the number of packets received (shown in Table 1) and the number of packets being lost (shown in Table 2). Whereas, the *idle listening* is the latency that is presented in Table 2 for the two data rates.

To conclude, performance evaluation of TAD-MAC protocol has shown that the protocol is optimized in terms of energy consumption as well as latency. On average it takes between 30s to 40s to converge towards a steady state. Moreover, the adaptive algorithm converges efficiently towards various traffic rates and is validated through experiments.

VI. CONCLUSION

The idle energy consumption is the dominant energy waste in WBAN and it is multiple times more energy consuming than the actual transmission. In this paper, we proposed a novel MAC protocol (TAD-MAC) that allows the sensor nodes to adapt themselves dynamically according to the traffic. The dynamic adaptation of TAD-MAC results in ultra low energy consumption from idle listening, overhearing, collisions and unnecessary wake-up beacon transmission. Each node has a traffic status register bank which contains the traffic status according to the data received from all the neighbor nodes. Further, each sensor node adapts its wake-up interval through TSR in such a way that idle listening is minimized and the convergence to a steady state is achieved through continuous update of wake-up interval. In order to make sure that TAD-MAC always converges towards a steady state value an exhaustive search method is used to characterize the system variable μ that is used in the computation of proposed adaptive algorithm. Number of different initial wake-up interval values are used to observe the behavior of μ in open loop and closed-loop forms that are used to characterize the system. It is shown that μ has a monotonic behavior in open loop form. Whereas, in closed-loop form it varies like a decaying function that reaches zero within short time and consequently it shows that the algorithm reaches a steady state. TAD-MAC protocol is implemented in a network simulator (WSNet) and various performance metrics such as energy consumption, latency, convergence time and delivery ratio are defined and evaluated through rigorous simulations. In this regard, traffic statistics are utilized from simulator to estimate performance metrics. Best, worst and average convergence time results are presented that are $8s$, $72s$ and $35s$ respectively, whereas algorithm also converges very fast (within $8s$) whenever the data rate changes due to the traffic variations. It is also shown that if no FIFO is used to buffer data packets the delivery ratio can be reduced up to 70% before a node reaches a convergence state.

ACKNOWLEDGMENT

The statements made herein are solely the responsibility of the authors.

REFERENCES

[1] K. L. Mills, "A brief survey of self-organization in wireless sensor networks," *Wireless Commun. Mobile Comput.*, vol. 7, no. 7, pp. 823–834, Sep. 2007.

[2] M. I. Qaisar and M. H. Rehmani, "Role of wireless sensor network in emerging communication technologies: A review," in *Emerging Communication Technologies based on Wireless Sensor Networks: Current Research and Future Applications*. Boca Raton, FL, USA: CRC Press, 2016.

[3] D. P. Tobón, T. H. Falk, and M. Maier, "Context awareness in WBANs: A survey on medical and non-medical applications," *IEEE Wireless Commun.*, vol. 20, no. 4, pp. 30–37, Aug. 2013.

[4] P. Honeine, F. Mourad, M. Kallas, H. Snoussi, H. Amoud, and C. Francis, "Wireless sensor networks in biomedical: Body area networks," in *Proc. 7th Int. Workshop Syst., Signal Process. Appl. (WOSSPA)*, May 2011, pp. 388–391.

[5] S. Mann, "Phenomenal augmented reality: Advancing technology for the future of humanity," *IEEE Consum. Electron. Mag.*, vol. 4, no. 4, pp. 92–97, Oct. 2015.

[6] A. Pyattaev, K. Johnsson, S. Andreev, and Y. Koucheryavy, "Communication challenges in high-density deployments of wearable wireless devices," *IEEE Wireless Commun.*, vol. 22, no. 1, pp. 12–18, Feb. 2015.

[7] S. Movassaghi, M. Abolhasan, J. Lipman, D. Smith, and A. Jamalipour, "Wireless body area networks: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1658–1686, Jan. 2014.

[8] R. Cavallari, F. Martelli, R. Rosini, C. Buratti, and R. Verdone, "A survey on wireless body area networks: Technologies and design challenges," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1635–1657, Aug. 2014.

[9] Accessed on Mar. 21, 2016. [Online]. Available: http://www.shimmersensing.com/images/uploads/docs/ExG_Revision_1_3.pdf

[10] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, "MAC essentials for wireless sensor networks," *IEEE Commun. Surveys Tuts.*, vol. 12, no. 2, pp. 222–248, 2010.

[11] M. M. Alam, O. Berder, D. Menard, and O. Sentieys, "TAD-MAC: Traffic-aware dynamic MAC protocol for wireless body area sensor networks," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 2, no. 1, pp. 109–119, Mar. 2012.

[12] INRIA Research Center. *WSNet Simulator*, accessed on Mar. 21, 2016. [Online]. Available: http://wsnet.gforge.inria.fr/download.html

[13] A. El-Hoiydi and J. D. Decotignie, "WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks," in *Proc. 9th Int. Symp. Comput. Commun.*, vol. 2. USA, 2004, pp. 244–251.

[14] M. M. Alam and E. B. Hamida, "Strategies for optimal MAC parameters tuning in IEEE 802.15.6 wearable wireless sensor networks," *J. Med. Syst.*, vol. 39, no. 9, pp. 1–16, 2015.

[15] Accessed on Mar. 21, 2016. [Online]. Available: http://standards.ieee.org/getieee802/download/802.15.6-2012.pdf/

[16] S. J. Marinkovic, E. M. Popovici, C. Spagnol, S. Faul, and W. P. Marnane, "Energy-efficient low duty cycle MAC protocol for wireless body area networks," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 915–925, Nov. 2009.

[17] O. Omeni, A. C. W. Wong, A. J. Burdett, and C. Toumazou, "Energy efficient medium access protocol for wireless medical body area sensor networks," *IEEE Trans. Biomed. Circuits Syst.*, vol. 2, no. 4, pp. 251–259, Dec. 2008.

[18] P. Hurni, T. Braun, and M. Anwander, "Evaluation of WiseMAC and extensions on wireless sensor nodes," *Telecommun. Syst.*, vol. 43, nos. 1–2, pp. 49–58, 2010.

[19] Y. Sun, O. Gurewitz, and D. B. Johnson, "RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks," in *Proc. 6th ACM Conf. Embedded Netw. Sensor Syst.*, USA, 2008, pp. 1–14.

[20] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "PW-MAC: An energy-efficient predictive-wakeup MAC protocol for wireless sensor networks," in *Proc. IEEE 31st Int. Conf. Comput. Commun.*, Apr. 2011, pp. 1305–1313.

[21] M. M. Alam and E. B. Hamida, "Surveying wearable human assistive technology for life and safety critical applications: Standards, challenges and opportunities," *Sensors*, vol. 14, no. 5, pp. 9153–9209, Mar. 2014.

[22] L. Tang, Y. Sun, O. Gurewitz, and D. B. Johnson, "EM-MAC: A dynamic multichannel energy-efficient MAC protocol for wireless sensor networks," in *Proc. 12th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, France, 2011, pp. 23:1–23:11.

[23] M. M. Alam, O. Berder, D. Menard, T. Anger, and O. Sentieys, "A hybrid model for accurate energy analysis of WSN nodes," *EURASIP J. Embedded Syst.*, vol. 2011, pp. 4.1–4.16, Jan. 2011.

[24] Accessed on Mar. 21, 2016. [Online]. Available: http://powwow.gforge.inria.fr/

[25] E. Y. A. Lin, J. M. Rabaey, S. Wiethoelter, and A. Wolisz, "Receiver initiated rendezvous schemes for sensor networks," in *Proc. IEEE Global Telecommun. Conf.*, vol. 5. Dec. 2005, pp. 3117–3122.

[26] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *Proc. ACM 2nd Int. Conf. Embedded Netw. Sensor Syst.*, USA, 2004, pp. 95–107.

[27] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks," in *Proc. ACM 4th Int. Conf. Embedded Netw. Sensor Syst.*, USA, 2006, pp. 307–320.

[28] P. Hurni and T. Braun, "MaxMAC: A maximally traffic-adaptive MAC protocol for wireless sensor networks," in *Wireless Sensor Networks* (Lecture Notes in Computer Science). 2010, pp. 289–305.

[29] M. Anwander, G. Wagenknecht, T. Braun, and K. Dolfus, "BEAM: A burst-aware energy-efficient adaptive MAC protocol for wireless sensor networks," in *Proc. 7th Int. Conf. Netw. Sens. Syst. (INSS)*, Jun. 2010, pp. 195–202.

[30] M. M. Alam, D. Ben-Arbia, and E. Ben-Hamida, "Joint throughput and channel aware (TCA) dynamic scheduling algorithm for emerging wearable applications," in *Proc. IEEE Wireless Commun. Netw. Conf.*, 2016.

[31] N. F. Timmons and W. G. Scanlon, "An adaptive energy efficient MAC protocol for the medical body area network," in *Proc. 1st Int. Conf. Wireless Commun., Veh. Technol., Inf. Theory Aerosp. Electron. Syst. Technol.*, May 2009, pp. 587–593.

[32] M. Maman and L. Ouvry, "BATMAC: An adaptive TDMA MAC for body area networks performed with a space-time dependent channel model," in *Proc. 5th Int. Symp. Med. Inf. Commun. Technol. (ISMICT)*, Mar. 2011, pp. 1–5.

[33] S. Ullah and K. S. Kwak, "An ultra low-power and traffic-adaptive medium access control protocol for wireless body area network," *J. Med. Syst.*, vol. 36, no. 3, pp. 1021–1030, 2012.

[34] M. M. Alam and E. B. Hamida, "Performance evaluation of IEEE 802.15.6 MAC for wearable body sensor networks using a space-time dependent radio link model," in *Proc. 11th AICCSA Conf.*, Nov. 2014, pp. 441–448.

[35] M. M. Alam, "Power-aware adaptive techniques for wireless sensor networks," Ph.D. dissertation, Univ. Rennes 1, Rennes, France, 2013.

[36] E. Ben-Hamida, M. M. Alam, M. Maman, B. Denis, and R. D'Errico, "Wearable body-to-body networks for critical and rescue operations—The CROW^2 project," in *Proc. IEEE PIMRC, Workshop Converg. Wireless Technol. Pers. Healthcare*, USA, 2014, pp. 2145–2149.

MUHAMMAD MAHTAB ALAM received the Ph.D. degree from the INRIA Research Center and the IRISA Laboratory, France, in 2012. In 2013, he was an Assistant Professor with the Swedish College of Engineering and Technology, Pakistan. From 2007 to 2009, he was a Research Engineer with the Center for Software Defined Radio, AAU, Denmark. Over the past eight years, he has contributed in number of European projects, such as GEODES, GRECO, and MS-SDR. He has been a Research Scientist with the Qatar Mobility Innovations Center, Qatar, since 2014. He is with the National Priority Research Program of Qatar Foundation in wearable WSN for rescue and critical applications. His research interests are in the fields of self-organized and self-adaptive wireless sensor and body area networks, in particular toward energy-efficient communication protocols and algorithms, radio-link and mobility modeling, digital signal processing, algorithm-architecture optimizations, and software defined radios. He is a member of IET-UK.

ELYES BEN HAMIDA received the Dipl.-Ing. degree from INSAT, Tunisia, in 2005, and the M.S. and Ph.D. degrees from INSA Lyon, France, in 2006 and 2009, respectively, all in computer networks and telecommunications. He is currently an M2M/IoT Product Manager and an R&D Expert with the Qatar Mobility Innovations Center, Doha, and has more than nine years of R&D experience in the fields of wireless networking, telecommunications, and security. He has been involved in several international and industrial R&D projects, and he had the chance to lead the design and specification of several products currently on the market. He holds four patents and has authored over 40 publications. He is currently leading two funded QNRF projects on Wearable Wireless Sensors Networks (NPRP 6-1508-2-616), and Security in Intelligent Transport Systems (NPRP 7-1113-1-199).

OLIVIER BERDER received the B.S., M.S., and Ph.D. degrees from the University of Bretagne Occidentale, Brest, in 1998, 1999, and 2002, respectively, all in electrical engineering. From 2002 to 2004, he was with the Laboratory for Electronics and Telecommunication Systems (LEST-UMR CNRS 6165), Brest. From 2004 to 2005, he was with the Speech and Sound Technologies and Processes Laboratory, FT R&D, Lannion, France. In 2005, he joined ENSSAT Lannion, University of Rennes 1, where he is currently a Professor, and the Laboratory IRISA, Rennes. His research interests focus on multiantenna systems and cooperative techniques for mobile communications and wireless sensor networks.

DANIEL MENARD received the Engineering and M.Sc. degrees in electronics and signal processing engineering from the University of Nantes Polytechnic School, in 1996, the Ph.D. degree in signal processing and telecommunications from the University of Rennes 1, in 2002, and the Habilitation à Diriger des Recherches degree from the University of Rennes 1, in 2011. From 1996 to 2000, he was a Research Engineer with the University of Rennes 1. He is currently an Associate Professor of Electrical Engineering with the University of Rennes 1 and a member of the Computing ArchItectures Embedding Reconfigurable resources for eNergy-efficient systems-on-chip research team at the IRISA/INRIA Laboratory. His research interests include implementation of signal processing and mobile communication applications in embedded systems, fixed-point conversion, embedded software, low-power architectures, and arithmetic operator design.

OLIVIER SENTIEYS received the M.Sc. and Ph.D degrees in electrical engineering (signal processing) from the University of Rennes 1, Brittany, France, in 1990 and 1993, respectively. After completing his Habilitation thesis in 1999, he joined the University of Rennes 1 (at ENSSAT, a Graduate School in Electronics Engineering), as a Full Professor in 2002. He is leading the CAIRN Research Team common to INRIA Institute (National Institute for Research in Computer Science and Control) and the IRISA Laboratory. (Research Institute in Computer Science and Random Systems). His research activities are in the two complementary fields of embedded systems and signal processing with research interests, including finite arithmetic effects, low-power and reconfigurable system-on-chip, design of wireless communication systems, and cooperation in mobile systems. He has authored or co-authored more than 150 journal publications or peer-reviewed conference papers and holds five patents.

• • •