

Delay-efficient MAC protocol with traffic differentiation and run-time parameter adaptation for energy-constrained wireless sensor networks

Messaoud Doudou¹  · Djamel Djenouri¹ · Jose M. Barcelo-Ordinas² · Nadjib Badache³

Published online: 2 June 2015
© Springer Science+Business Media New York 2015

Abstract This paper presents an asynchronous cascading wake-up MAC protocol for heterogeneous traffic gathering in low-power wireless sensor networks. It jointly considers energy/delay optimization and switches between two modes, according to the traffic type and delay requirements. The first mode is high duty cycle, where energy is traded-off for a reduced latency in presence of realtime traffic (RT). The second mode is low duty cycle, which is used for non-realtime traffic and gives more priority to energy saving. The proposed protocol, DuoMAC, has many features. First, it quietly adjusts the wake-up of a node according to (1) its parent's wake-up time and, (2) its estimated load. Second, it incorporates a service differentiation through an improved contention window adaptation to meet delay requirements. A comprehensive analysis is provided in the paper to investigate the effectiveness of the proposed protocol in comparison with some state-of-the-art energy-delay efficient duty-cycled MAC protocols, namely DMAC, LL-MAC, and Diff-MAC. The network lifetime

and the maximum end-to-end packet latency are adequately modeled, and numerically analyzed. The results show that LL-MAC has the best performance in terms of energy saving, while DuoMAC outperforms all the protocols in terms of delay reduction. To balance the delay/energy objectives, a runtime parameter adaptation mechanism has been integrated to DuoMAC. The mechanism relies on a constrained optimization problem with energy minimization in the objective function, constrained by the delay required for RT. The proposed protocol has been implemented on real motes using MicaZ and TinyOS. Experimental results show that the protocol clearly outperforms LL-MAC in terms of latency reduction, and more importantly, that the runtime parameter adaptation provides additional reduction of the latency while further decreasing the energy cost.

Keywords Wireless sensor networks · MAC protocols · Energy-efficiency · Delay · Modeling · Optimization

1 Introduction

1.1 Problem statement and contribution

Nowadays wireless sensor networks (WSN) applications, ranging from health-care, to industrial monitoring, transportation, and automation [15], are complex and have several requirements. In addition to the energy constraint that was almost the only metric for consideration in earlier applications, new constraints appear with emerging monitoring system applications, such as latency, security, and reliability. Given that the MAC layer manages the medium access and controls the radio interface, it plays a key role in determining the system performance in terms of e2e packet latency, and power-consumption (network

✉ Messaoud Doudou
doudou@mail.cerist.dz

Djamel Djenouri
ddjenouri@acm.org

Jose M. Barcelo-Ordinas
joseb@ac.upc.edu

Nadjib Badache
badache@cerist.dz

¹ DTISI, CERIST Research Center, Algiers, Algeria

² Universitat Politcnica de Catalunya-BarcelonaTECH (UPC), Barcelona, Spain

³ University of Sciences and Technology Houari-Boumediene (USTHB), Algiers, Algeria

lifetime). Energy saving is achieved at the MAC layer by duty-cycling the radio and switching between active/sleep modes. That is, repeatedly switching the radio off/on. In active mode, a node can receive and transmit packets, while in the sleep mode, it completely turns off its radio to save energy. Hence, a node needs to be aware of its neighbors' wake-up time to tune its transmission accordingly. This has a direct impact on the forwarding delay, also known as sleep delay, which is the delay spent at every hop by the transmitter waiting for the receiver to wake up.

The forwarding delay problem is targeted herein, where a novel dual-mode asynchronous cascading wake-up scheduled MAC protocol, called DuoMAC, is presented. In absence of realtime (RT) traffic, DuoMAC runs in a low duty-cycling (LDC) mode and behaves as an energy-efficient MAC following an EEF (Energy Efficiency First) strategy. However, when an RT event is detected, a node enters in a high duty-cycling (HDC) mode and behaves as a delay-efficient MAC to forward RT traffic following a delay efficiency first (DEF) strategy. In addition to operating in two modes, DuoMAC precipitates packet forwarding by adopting a dynamic cascading of active periods and adjusting the wake-up time of a node according to, (1) its parent's wake-up time and, (2) its estimated load. To meet the dynamic traffic requirements of each class of traffic, DuoMAC processes every packet according to its degree of importance. This is achieved by applying an effective service differentiation based on a hybrid prioritization scheme, where the contention window size is adapted by implementing an effective link quality estimator. The proposed solution also integrates a runtime parameter adaptation mechanism and formulates a convex optimization problem with energy consumption as the objective function constrained by the delay required for realtime traffic.

1.2 Illustrative system scenario

For illustration and without loss of generality, a typical scenario of an urban traffic monitoring system is depicted in Fig. 1. The proposed protocol can be used in any monitoring system with a joint time/energy constrained application, where sensed data must be gathered and forwarded to a central station within a specific time proportional to the application requirements. In the given example, nodes are uniformly distributed to perform realtime monitoring of traffic, measurement of road conditions such as glaze, weather condition, etc., which helps optimizing traffic management and avoiding urban traffic jams. The monitoring system is typically connected to an intelligent traffic system (ITS) that provides collision avoidance

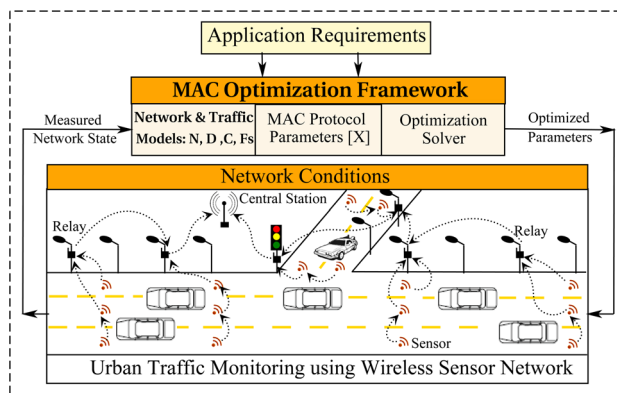


Fig. 1 Wireless sensor network for urban traffic monitoring system with runtime parameters adaptation

services and might display the crash warning messages to the drivers as soon as crash-prone conditions are detected. In this scenario, sensor samples should be communicated to the control center for being analyzed. Some data such as traffic flows must reach the central station within a short delay, no later than a predefined deadline L_{max} . This is known as realtime traffic, *RT*, while the traffic that is less delay-prone is named non-realtime traffic, *NRT*. The system should be able to satisfy the delay constraints on *RT* traffic, while considering minimizing energy consumption, E , subject to achieve maximum network lifetime.

In this paper, a comprehensive design approach that embraces the factors mentioned above is presented. DuoMAC protocol is presented along with a comprehensive mathematical analysis that shows the effectiveness of the proposed protocol from the energy-delay perspective in comparison with some state-of-the-art energy-delay efficient duty-cycled MAC protocols: (1) DMAC [20], (2) LL-MAC [34], and (3) Diff-MAC [33]. Further, the proposed protocol has been enhanced with a runtime parameter adaptation based on solving an optimization problem. This allows to dynamically find the optimal MAC tunable parameters to achieve better performance for the concurrent objectives. The proposed protocol enhanced with the runtime parameter adaptation has been implemented in MicaZ sensor motes with TinyOS and evaluated both by simulation, and in a testbed experimentation. The remainder of the paper is organized as follows. Section 2 presents the related work. Section 3 provides a detailed description of DuoMAC. Section 4 presents a mathematical analysis through generalized network and traffic models, and some numerical comparisons. Section 5 illustrates the DuoMAC parameter optimization, while Sect. 6 validates the optimization approach through extensive simulation and experimentations with real sensor motes. Finally, Sect. 7 draws the conclusions.

2 Related work

Existing WSN MAC protocols are basically grouped based on the access management to the shared medium [2, 6, 9, 16, 29], or according to some targeted performance metrics such as energy, delay, and reliability [28]. One possible classification is to split the protocols into three classes contention-based, scheduled, and hybrid, based on the mechanism employed to avoid collision. The authors in [17] propose to categorize MAC protocols based on how nodes are organized to access the shared channel into three classes random-based, slotted-based, and frame-based. Depending on how contention-based MAC protocols schedule the node's duty-cycle (turning the radio on/off), authors in [10] and [11] have grouped duty-cycled contention-based MAC into two schemes; Synchronous vs. Asynchronous based on how the senders join their intended receivers. Following this principle, synchronous protocols like SMAC [32] specify and coordinate active/ sleep periods by exchanging SYNC packets for synchronization. Many recent protocols are based on SMAC, such as Diff-MAC [33], which integrates traffic differentiation to reduce the delay for realtime traffic. Asynchronous duty-cycling schemes have the advantage over synchronous ones to eliminate the need of clock synchronization. Furthermore, their contention-based feature makes these protocols conceptually distributed and more dynamic compared to scheduled protocols, notably TDMA-based ones. However, the communicating nodes are prone to spend more time waiting for the active period of each other, which inevitably influences the cumulative e2e delay. In [10], different categories of asynchronous protocols like preamble-based protocols (e.g., X-MAC [3] and WiseMAC [12]) and beacon-based protocols (e.g., receiver-initiated protocols such as RI-MAC [27]) have been reported and discussed from the delay-efficiency perspective. The analysis also reveals that collaborative scheduled wake-up protocols, e.g. LL-MAC [34] and DMAC [20], provide better e2e delay reduction among other classes of protocols. The principle is that neighbor nodes collaborate to set their wake-up offset in a cascading way. However, none of these protocols consider traffic load of intermediate nodes when targeting the delay. CyMAC [24] is yet another asynchronous receiver-initiated protocol that targets to reduce the idle-listening of nodes by employing a rendezvous mechanism. Each node learns about the upcoming traffic from its neighbors and schedules the wakeup beacons to the appropriate time. Generally, the delay requirement is expressed in terms of absolute e2e delay in time critical applications [28] and the packet has to arrive as fast as possible and thus is not a question of relative delay, but a question of spend more energy for arriving faster. Furthermore, the delay requirement may not be the

same for each kind of traffic which has not been taken into account in most of these protocols.

The most compelling purpose for performance analysis through protocol modeling techniques is the capability to design optimized protocol parameters for a given application. While most of the energy-efficient MAC protocols for WSNs had taken pure experimental approaches, few works have attempted to model and analyze MAC protocols. First, the analytical model of well known preamble-sampling protocols, B-MAC and X-MAC, have been derived by Polastre et al. [25], and Buettner et al. [3], respectively. Langendoen and Meier [18] consider traffic and network models for very low data rate applications, analyzing energy consumption and average latency of well known MAC protocols. Markov models have been developed to evaluate the energy consumption of some MAC protocols, such as SMAC [30], and DMAC [35]. Protocol optimization have been investigated by Ye et al. [31], but this work is limited to optimize parameters for energy minimization in SCP-MAC protocol. Ergen et al. [13] propose a protocol engine that selects the adequate protocol with optimized parameters satisfying the application requirements for a fixed network topology. Those approaches are inefficient; yet their results are used at compile time, ignoring thus the high dynamic change of the network state at runtime.

Lately, numerous efforts to support MAC optimization at run-time have been published. First, Breath, by Park et al. [23], is a randomized protocol for control systems where parameters are controllable by the central coordinator. Meier et al. [21] propose the provision for dynamic calibration of MAC parameters at runtime, taking into account of topology variability. Zimmerling [36] presents an automatic centralized parameter adaptation framework in which MAC protocol parameters are abstracted in a protocol-independent way to meet a given application requirements in term of energy, delay, etc. This abstraction is chosen to match any protocol-dependent parameters. In this paper, a new asynchronous cascading wakeup MAC is proposed, and the modeling methodology given in [18] is extended and applied to the proposed protocol, which permits to set the parameters at runtime in order to improve the performance by considering the heterogeneous traffic support; RT and NRT. The performance is achieved by maximizing the network lifetime while satisfying the application requirement in terms of e2e delay constraints on RT traffic.

3 DuoMAC protocol description

DuoMAC is a contention-based MAC protocol inspired from a control loop system. It adapts its wake-up schedule to the network traffic conditions and switches between two

operating modes, (1) a LDC state whose aim is energy-efficiency for non-real-time (NRT) traffic and, (2) a HDC state to accelerate real-time (RT) packet forwarding. To gather the sensed data, the sink node first runs the tree construction. When a node is powered-on, it runs the connection phase and tries to join the network. The node monitors the channel and receives beacons from its neighboring nodes, choosing the parent with minimum hop-count, best link quality, and storing its wake-up scheduling information. Once connected, the node is ready to send and receive beacons and data packets. For simplicity, we assume the use of a routing protocol with an established tree topology rooted at the sink, where the nodes are classified according their distance to the sink in terms of the number of hops. Consequently, the sink is at level $d = 0$, nodes at one hop are at level $d = 1$, and the furthest nodes are at level $d = D$. The choice of the routing protocol does not impact the MAC protocol performance since nodes at level d assume that they will receive packets from children nodes at level $d + 1$, that they will forward packets to parent nodes at level $d - 1$, and that there will be background nodes that interfere at level d . In the following, the main features of DuoMAC are discussed.

3.1 LDC and HDC operation modes

Every cycle, t_m , contains an active period T_{active} , a sleep period, T_{sleep} , and N_{cp} channel polling periods of size T_{cp} . The later checks for possible incoming RT traffic in the middle of the sleep period. The number of quick channel polling periods must trade-off energy with delay. By default, Fig. 2(a), nodes run in LDC mode, where T_{recv} and T_{send} respectively limit the reception and transmission of NRT data packets.

A node wakes up exactly at its scheduled time, updated at each cycle, and sends a B_w beacon indicating the

beginning of T_{recv} period. The receive period is terminated when a node receives a B_w beacon of its parent. At the end of the T_{recv} period, a node calculates its next wake-up time schedule according to the node’s NRT time traffic, which is then broadcasted in a B_{nw} beacon. After sending this beacon, a node then proceeds to forward its queued packets.

As depicted in Fig. 2(a), the B_w beacon signals a current wake-up time of the transmitting node. The B_{nw} beacon’s role is twofold. First, it indicates the end of the T_{recv} period and the beginning of the T_{send} period. Second, it contains information about the next wake-up time $W_{n_i}(t_m)$, as well as the distance d_{n_i} of a node n_i from the sink. Nodes contend to send NRT packets using CSMA and each transmission includes a data packet and its acknowledgment (ACK).

In order to detect RT traffic, nodes wake up in every T_{cp} period for T_{on} seconds to poll the channel, then return back to sleep if there is no activity. A node can detect a RT event and then trigger its own RT packets, or it can receive RT packets from other nodes during its periodic channel polling. In both cases, a node switches to the HDC mode, Fig. 2(b). To send a RT packet, a node transmits a sequence of RTS strobos of size T_{rts} as a wake-up preamble, containing the identifier of the receiver, to reach its forwarder node. Strobos continue for a period sufficient to make at least one strobe overlap with a receiver wake-up. When this happens, the receiver replies with a CTS packet of size T_{cts} as an acknowledgment, and it keeps the radio on awaiting the transmission of the data packet. The sender transmits the data packet upon receiving the CTS and waits for the ACK from the receiver. Note that T_{on} must be at least equal to the transmission time of RTS, T_{rts} , plus the interval time, T_{cl} , between two RTS strobos where the sender waits to a CTS. This ensures the wake-up packet to

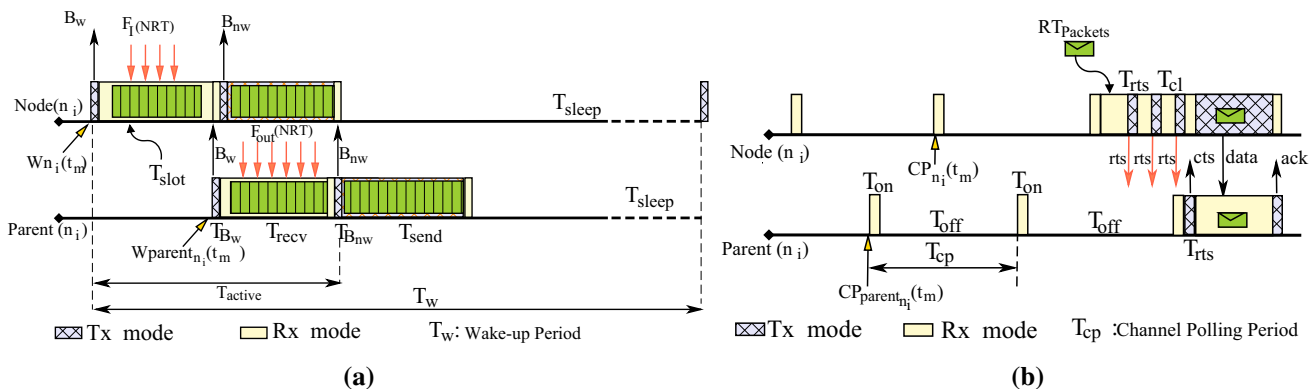


Fig. 2 a LDC operation mode. b HDC operation mode

be heard, Fig. 2(b). Once the flag *More_bit* contained in the data frame is set to 0, then, each node receiving this packet will be aware that it is the last RT packet, and it consequently switches back to LDC mode.

3.2 Cascading wake-up schedule

To reduce the delay wasted when a transmitter remains waiting for the receiver to wake-up (in idle listening), the active time of both nodes must be proportional to the propagation time value plus the time required to receive a packet, which represents the idle listening interval when perfect synchronization is assured. DuoMAC is based on adaptive self-adjusting cascading wake-up scheduling. The dynamically updated cascading wake-up feature of DuoMAC permits to wake up nodes asynchronously and independently from their neighbors (Fig. 2(a)). If $parent_{n_i}$ is the selected next forwarder (parent) of node n_i , then, the wake-up time, W_{n_i} , must be shifted σ_{n_i} (amount of time) before $W_{parent_{n_i}}$. Consequently, the closer the node is to the sink, the more its corresponding σ_{n_i} time will increase. The wake-up time W_{n_i} of a node is not static, but it depends on the traffic load to be handled at the current cycle t_m and the wake-up time of its parent, which makes the wake-up schedule adaptive to the traffic to be handled at each node. As a result, $W_{n_i}(t_{m+1})$ can be calculated as a function of σ_{n_i} and $W_{parent_{n_i}}$, i.e., $W_{n_i}(t_{m+1}) = f(\sigma_{n_i}(t_m), W_{parent_{n_i}}(t_m))$. For an operation cycle t_m , $\sigma_{n_i}(t_m)$ is the required time to receive the estimated load (NRT traffic load), $L(t_m)$, by node n_i , from its children during the next cycle, which may vary in time. For instance, a node needs to estimate the upcoming traffic load for the next operation cycle so that it can calculate its next wake-up time.

3.3 Estimating the traffic load

Each node in the network has to estimate the load it expects to receive in the coming period in order to calculate the wake up schedule. To determine the best traffic load estimator, we rely on the fact that the average input traffic depends linearly on the data traffic rate, Sect. 4.1, Eq. (10), which justifies the choice of a linear estimator for traffic. To validate this assumption, we have performed extensive experimentations using a 20 MicaZ testbed, where the traffic load flowed by the nodes has been tracked. In the testbed, each node generates traffic with a frequency of 2 pkt/sec. The estimated and received average loads in some nodes are plotted in Fig. 3. The selected nodes are labeled and depicted in Fig. 4. Figure 3(a) shows that the traffic load has a linear form. At a given operation cycle t_m , the node gathers information about the received traffic, $L_{n_i}(t_m)$ at node n_i . In the following, n_i is omitted for clarity. To calculate the estimated traffic for the next cycle t_{m+1} , the current load value of the nodes are predicted according to the m previous load values, using a linear prediction scheme. Assume that the load values $L(t_k)$ at times t_k ($k = 1, \dots, m$) are known, and that the load value at time t_{m+1} , $L(t_{m+1})$, has to be predicted. This load is calculated using the least mean square (LMS) method: $L(t_{m+1}) = at_{m+1} + b$, where coefficients a and b are obtained from solving the system equations of Eq. (1):

$$\begin{cases} \sum_{k=1}^m t_k L(t_k) &= a \sum_{k=1}^m t_k^2 + b \sum_{k=1}^m t_k \\ \sum_{k=1}^m L(t_k) &= a \sum_{k=1}^m t_k + bm \end{cases} \quad (1)$$

Resolving the above system yields the following solution for a and b :

$$a = \frac{\sum_{k=1}^m L(t_k) \sum_{k=1}^m t_k - m \sum_{k=1}^m t_k L(t_k)}{(\sum_{k=1}^m t_k)^2 - m \sum_{k=1}^m t_k^2}, \quad (2)$$

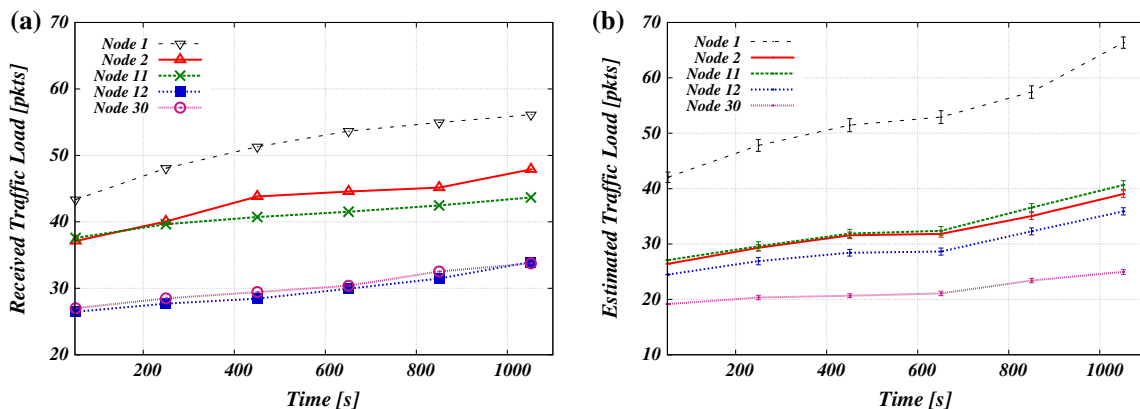
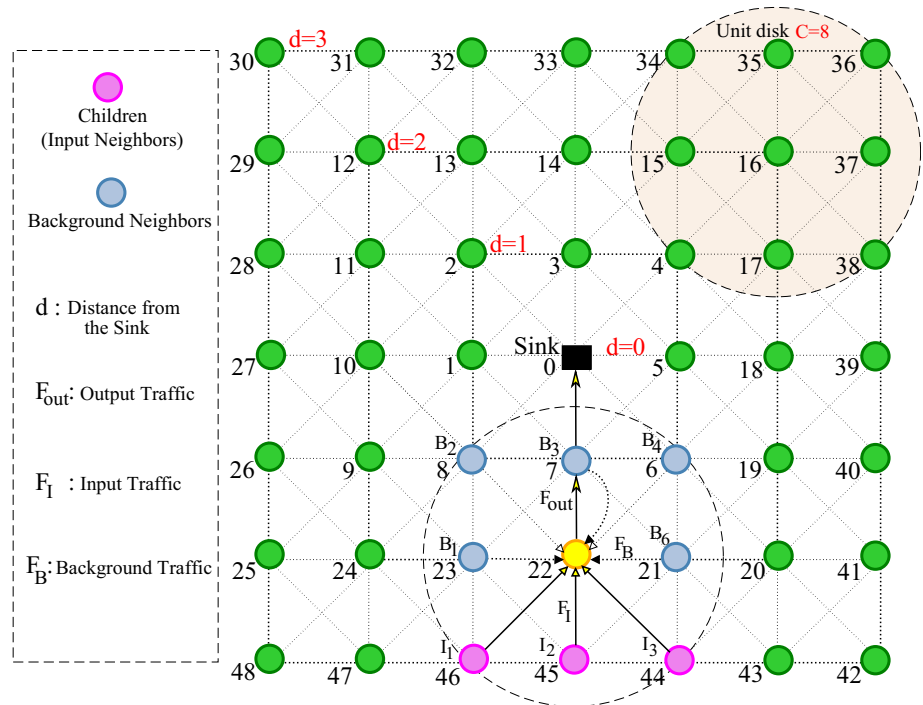


Fig. 3 **a** Received load and **b** estimated load at intermediate nodes (1, 2, 11, 12, 30) with traffic rate = 2 pkts/sec and parameters $a/b = 0.9710/0.3735$

Fig. 4 Network topology and local node traffic flowing



$$b = \frac{\sum_{k=1}^m t_k L(t_k)}{\sum_{k=1}^m t_k} - \left(\frac{\sum_{k=1}^m t_k^2}{\sum_{k=1}^m t_k} \right) \left(\frac{\sum_{k=1}^m L(t_k) \sum_{k=1}^m t_k - m \sum_{k=1}^m t_k L(t_k)}{(\sum_{k=1}^m t_k)^2 - m \sum_{k=1}^m t_k^2} \right) \quad (3)$$

In the experiment, m has been fixed to 100, a and b have been accordingly calculated. To investigate how the linear estimator performs, the estimated load, Fig. 3(b), has been recorded in the same intermediate nodes as in the experiment i.e. nodes 1,2,11,12, and 30, and under the same traffic rate of 2 pkt/sec. The results show a smooth increase in time, an approximative approach to the real values, Fig. 3(a), for all nodes. These results confirm the effectiveness of a linear estimator. In general, other topologies such as random deployment, [18] also present linear dependency with respect to the data rate. Further, topologies that produce other traffic profiles and require other estimators, e.g. non-linear, can be easily integrated in the proposed protocol.

3.4 Adjusting the wake-up time

As discussed earlier, the wake-up time W_{n_i} of node n_i can be calculated as a function of two parameters;

$W_{n_i}(t_{m+1}) = f(\sigma_{n_i}(t_m), W_{parent_{n_i}}(t_m))$, where $\sigma_{n_i}(t_m)$ is as defined in Sect. 3.2. Suppose TX_i is the time required to transmit one packet from an arbitrary node, n_i , including contention and acknowledgement times: $TX_i = T_{cw} + T_{data}$. T_{cw} is the contention window time and it is calculated as the contention window size multiplied by the time-slot, and $T_{data} = T_{hdr} + P/R + T_{ack}$. Definitions and values for T_{cw} , T_{hdr} , P , R and T_{ack} depends on the protocol, they are given in Table 2. Then $\sigma_{n_i}(t_m)$ can be calculated as follows,

$$\sigma_{n_i}(t_m) = T_{tx}^{B_w} + TX_i * L(t_m) + T_{rx}^{B_w} \quad (4)$$

where $T_{tx}^{B_w}$ and $T_{rx}^{B_w}$ are the required times to send and receive the wake-up beacon, B_w , including contention, $L(T_m)$ is the number of packets that the node estimates it will receive in the coming cycle, and TX is the time that a children node needs to transmit every packet. Given the value of $\sigma_{n_i}(t_m)$, the next wake-up time, $W_{n_i}(t_{m+1})$, can be calculated as:

$$W_{n_i}(t_{m+1}) = W_{parent_{n_i}}(t_m) - \sigma_{n_i}(t_m) \quad (5)$$

3.5 Service differentiation and CW adaptation

In order to provide hybrid prioritization levels between the traffic type and the traversed hop-count, an effective

Algorithm 1 Contention Window Adaptation

Input:

cw_{min}, cw_{max} : lower and upper bound of the contention window.

PRR : packet reception rate of the link in the current cycle.

b_{min}, b_{max} : link burstiness metrics in the current cycle.

Output:

cw_{cur} : current contention window.

- 1: $cw_{cur} = cw_{min}$;
 - 2: $P_{LQ}(t_i) = PRR * \frac{b_{min}}{b_{max}}$;
 - 3: **if** $P_{LQ}(t_i) < P_{LQ}(t_{i-1})$ **then**
 - 4: $\Delta cw = \alpha_{down} * (cw_{min} - cw_{cur})$;
 - 5: **else**
 - 6: $\Delta cw = \alpha_{up} * (cw_{max} - cw_{cur})$;
 - 7: **end if**
 - 8: $cw_{cur} = cw_{cur} + \Delta cw$;
-

service differentiation scheme that relies on contention window (CW) size adaptation mechanism is proposed. Saxena et al. [26] propose a CW adaptation algorithm based on loss probabilities by defining for each traffic class a targeted contention window, CW_{target} , to be reached after several steps. Yigitel et al. [33] propose a CW adaptation approach that initiates the CW of each traffic class from the medium value, i.e., $\frac{cw_{min} + cw_{max}}{2}$, then it moves it up and down according to PRR (Packet Reception Ratio) measurements. We propose a CW adaptation algorithm that differs from those in [26] [33] in two main points. First, the PRR is combined with a link quality metric called *link burst length*, which uses the concepts of b_{min} and b_{max} as defined in [22]. With this metric, a good link is characterized with a short b_{max} , which defines the maximum consecutive packet transmission errors in a burst, and with a long b_{min} , which defines the minimum consecutive successful packet transmissions between two bursts of packets. For example, for a burst of 10 transmitted packets, in the sequence 0110010011, a 0 at position i indicates that the i th packet

transmission has failed, where 1 indicates that the packet transmission has been successful. In this example, $b_{max} = 2$ and $b_{min} = 1$ for packet bursts of window size equal to 3. Authors in [22] show that these two parameters outperform PRR and capture better the link quality. Second, our CW adaptation scheme is initiated to cw_{min} (to reduce e2e delay), and it is then moved up and down based on the estimated link quality $P_{LQ}(t_i)$ given by Eq. (6).

$$P_{LQ}(t_i) = PRR * \frac{b_{min}}{b_{max}} \quad (6)$$

To summarize, DuoMAC runs a CW adaptation scheme, as described in *Algorithm 1*, for each N traffic samples that constitutes a burst of length N . For every burst t_i , the algorithm computes $P_{LQ}(t_i)$, and accordingly moves the CW . Finally, DuoMAC implements the same principle as introduced in [33] for setting non-overlapping CW sizes, where $CW^{RT} < CW^{NRT}$. The adaptation coefficient are chosen to fulfil the following conditions: $\alpha_{up}^{RT} < \alpha_{up}^{NRT}$ and $\alpha_{down}^{RT} > \alpha_{down}^{NRT}$.

3.6 Dealing with topology changes

Due to the use of wireless communication channels, WSNs are prone to frequent link breakdown, which causes routes changes. To deal with this problem in DuoMAC, every node stores information about its potential forwarders during its connection phase, and connects to one parent (the one with best link quality). In case a node has not received beacons from its parent for a number of consecutive cycles (set in our implementation to 3), DuoMAC considers this as an indication of that the link was broken down and the parent becomes unreachable. In this case, the node must chose another parent from the forwarder list and follows its schedule. It is obvious that frequent link breakdown deteriorates the performance of MAC protocols, especially those based on scheduling. Nonetheless, DuoMAC is relatively less affected by this problem, since (1) it is not based on schedule establishment, and (2) the wakeup time to receive NRT traffic is updated at every cycle, which permits a fast re-connection after link breakdown.

4 Comparative analysis

A comprehensive analysis of the proposed MAC protocol is given in the following, in comparison of some competitors from the literature. The chosen protocols are DMAC [20], LL-MAC [34], and Diff-MAC [33], which are all energy-delay efficient duty-cycled MAC protocols, and thus the most relevant for comparison. We first define the network and traffic models, which permit to determine the topology information and the traffic load at each node. Then, the protocols are analyzed through their operating modes (idle, transmission, receiving, and sleep modes). For every protocol, the energy consumption function, E , and the maximum e2e (end to end) packet delay, L , are derived.

4.1 Network and traffic model

An unsaturated network with low traffic is considered, as a typical scenario of WSN applications. A grid topology is adopted for the sake of simplification, and which is also suitable in scenarios similar to that presented in Fig. 1. However, the analysis can be easily adapted to other irregular deployments, as well as regular topologies by reformulating the input, output, and background traffic handled by each node in the network (e.g., the network and traffic models derived in [18] for ring topologies). A spanning tree is constructed, where nodes are static and maintain a unique path to the sink. The shortest paths to the sink are used, which are generally time-varying depending on the link quality. The maximum path length is denoted by D , i.e., the depth of the tree. A uniform node density on

the plane domain and a unit disk graph communication model are used, with $C + 1$ nodes supposed to be on the unit disk. That is, all nodes are in communication range with C neighboring nodes,¹ except those in the boundary of the grid as depicted in Fig. 4. The nodes are layered into levels according to their distance to the sink, in terms of minimal hop count, d .

In the following, the traffic model derived in [18] is extended for our network model. Let us consider periodic traffic generation, where every source node generates traffic with frequency, $F_s = F_{RT} + F_{NRT}$ for protocols that consider heterogeneous traffic, e.g., DuoMAC. Then, for every node, the input F_I^d , output F_{out}^d , background F_B^d traffic, and average number of input links I^d are derived accordingly. The background traffic is defined as the traffic overheard by a node that is submitted by neighboring nodes not selected as children. Background nodes may be peers at the same level of the tree, as well as nodes at adjacent (below or above) levels. The latter group includes (among others) the parent node forwarding a node’s outgoing traffic further up in the tree. Figure 4 shows an example, in which node 22 is the parent of nodes 44, 45 and 46. The parent of node 22 is node 7. Neighboring nodes 6, 7, 8, 21 and 23 are background nodes. The different symbols introduced in the analysis related to network and traffic model are summarized in Table 1 with typical values. The neighboring nodes can be classified as the set of children (input) nodes, I , and the set of overheard (background) nodes, B , such as, $C = |I| + |B|$. Since the first level contains C nodes (refer Fig. 4), the number of nodes N_d at level d can be expressed by Cd , except for $d = 0$ where it is 1 (reserved to the sink). This permits to compute the average number of input links, I_d , of a node at level, d , as:

$$I_d = \begin{cases} 0, & \text{if } d = D, \\ C, & \text{if } d = 0, \\ \frac{N_{d+1}}{N_d} = \frac{d+1}{d}, & \text{if } 0 < d < D. \end{cases} \tag{7}$$

Every input link among the I_d links of a node at level d transfers both RT and NRT traffics. The average output traffic frequency of a node at level d , F_{out}^d , is thus given by,

$$F_{out}^d = F_I^d + F_s = I_d \cdot F_{out}^{d+1} + F_s \tag{8}$$

By iterating Eq. (8) at each level starting from leave nodes to the sink, the following generalized traffic equation is obtained,

$$F_{out}^d = \begin{cases} F_s & \text{if } d = D, \\ 0 & \text{if } d = 0, \\ \frac{D^2 - d^2 + D - d}{2d} F_s & \text{if } 0 < d < D. \end{cases} \tag{9}$$

¹ For a communication range of unit 1, two neighboring nodes must be separated by a distance of $1/\sqrt{2}$.

Table 1 CC2420 Radio constants [4], network and traffic model with typical parameter values

Parameter description		Values
CC2420 Radio		
R	Rate (kbyte/s)	31.25
θ	Frequency tolerance (ppm)	30
T_{cs}	Time (ms) to turn the radio on and probe the channel (carrier sense)	2.60
T_{up}	Time (ms) to turn the radio on into RX or TX	2.40
L_{pbl}	Packet preamble length (byte)	4
Traffic		
P	Data payload [byte]	32
F_{RT}/F_{NRT}	Generation frequency of RT/NRT traffic (pkt/node/min)	[0.005, 20]
F_s	Sampling rate (pkt/node/min)	$F_{RT} + F_{NRT}$
F_I^d	Node’s input traffic frequency at level d	$F_s \frac{D^2 - d^2 + 2d - 1}{2d - 1}$
F_B^d	Background node’s traffic frequency at level d	$F_{out}^d - F_s$
F_{out}^d	Node’s output traffic frequency at level d	$ B^d F_{out}^d$
Network		
N	Network size (number of nodes) (#nodes)	[9, 512]
D	Network depth (#levels)	[1, 10]
C	Network density (Connectivity) (#neighbors)	8
Topology	Grid	150 * 150 m^2
Routing	Min hop count	–

From Eqs. (8), and (9):

$$F_I^d = \begin{cases} 0 & \text{if } d = D, \\ C \frac{D^2 + D}{2} F_s & \text{if } d = 0, \\ \frac{D^2 - d^2 + D - 3d}{2d} F_s & \text{if } 0 < d < D. \end{cases} \quad (10)$$

For the background traffic, B nodes are generating the same load F_{out}^d as the node itself [18]. Therefore, the average background traffic frequency of a node at level d , say F_B^d , is given by,

$$F_B^d = (C - |I_d|) F_{out}^d = \begin{cases} C * F_s & \text{if } d = D, \\ 0 & \text{if } d = 0, \\ \frac{(Cd - d - 1)(D^2 - d^2 + D - d)}{2d^2} F_s & \text{otherwise.} \end{cases} \quad (11)$$

The traffic model is valid for low data rate applications in which the sampling rate $F_s < 0.1$ Hz. The radio model uses realistic hardware values, Table 1, based on the CC2420 chipset. Since the objective is to compare the relative performance of the DuoMAC protocol with respect to other MAC protocols, we do not compute absolute energy consumption values and only effective duty cycles. That means that energy consumption is computed as a fraction of the time the radio is switched on. Thus, the radio can be modeled using the time needed to power up the radio, the

radio baud rate, and the time needed to do a carrier sense. We refer to paper [18] for a discussion on the radio model and traffic model. The same traffic model may be adapted to other topologies, including irregular ones. The traffic model assumes that there are no losses, i.e., no retransmissions, and thus the free-space propagation model is used. The traffic model has been validated in [18] through intensive simulations. The authors show that the analytical model provides results with $< 5\%$ of delivery ratio and energy consumption. This represents a powerful tool for comparing MAC protocols for low data rate applications.

4.2 Energy and delay model

Let **Network Energy Consumption** be defined as the amount of energy consumed by the radio duty of each node in the network according to its position and the amount of traffic it handles. Thus, the node’s energy consumption is the sum of energy consumed in each operating mode, which depends on the exchanged traffic load and the MAC intrinsic parameters. For example, let E_{idle}^n , E_{tx}^n , and E_{rx}^n be the energy consumed fractions in idle listening,² transmitting and receiving modes, the node’s energy

² The term idle listening is not explicitly mentioned in equations. It is modeled under different names that differs from a protocol to another. For example, terms T_{cs} and T_{up} in Table 1 that are common for all MAC protocols, terms T_0 for Diff-MAC, T_{cl} and T_{on} for DuoMAC in Table 2 are all fractions of time where the node is in idle listening and the radio is consuming energy.

consumption can be calculated as $E^n = E_{idle}^n + E_{tx}^n + E_{rx}^n$. The normalized energy consumption (in Joules) can be calculated by multiplying the obtained expressions in each mode by the current draws of the radio defined in the datasheet for each mode (eg. I_{idle} , I_{tx} , and I_{rx}). Given the energy consumption, E^n , of node, n , the Network Energy Consumption E for all nodes in the network can be expressed as,

$$E = \sum_{n \in N} (E^n) \tag{12}$$

The end-to-end (e2e) packet delay (latency), L^n , is defined as the expected time between the first transmission of a packet at node $n \in N$, and its reception at the sink. It is then a per-topology parameter, in the sense that it depends on the position of the node that generates the data. L^n denotes the sum of per-hop latencies of the shortest path \mathcal{P}^n from node n to the sink, where L_l^n is one-hop latency on each link $l \in \mathcal{P}^n$. The maximum end-to-end latency, L , is defined as the maximum latency from any node to the sink:

$$L = \max_{n \in N} (L^n) = \max_{n \in N} \left(\sum_{l \in \mathcal{P}^n} L_l^n \right) \tag{13}$$

The network energy consumption as well as the maximum e2e packet delay are MAC dependent. Expressions for Eqs. (12) and (13) for some canonical MAC protocols are developed in the following subsections and details are given in “Appendix: MAC energy-delay models” of Appendix.

4.3 DMAC protocol

Data-gathering MAC (DMAC) [20] is a synchronous protocol that addresses the delay of convergecast based data delivery, by scheduling the receiving and sending slots in a wave-like chain according to the nodes level in the gathering tree. Referring to Fig. 5, DMAC’s frame has a size of T_w ; the wake-up period. It is composed by a receive slot, a

transmission slot, and a number of sleep slots (1). The receive slot of each node coincides with the transmission slot of down-level nodes (children) (2), whereas the transmission (3) slot coincides with the receive slot of the parent. Nodes must contend in each slot T_{slot} using CSMA with acknowledgments (4), and they compensate for clock drift with a guard time T_{guard} , $T_{guard} = 4\theta T_{sync}$ (5). Synchronization messages are exchanged every T_{sync} interval using regular sending and receiving slots (6). Additional slots are added in the sleep period when there is more data to be sent using a data-prediction scheme (7). In DMAC, energy is consumed in transmission, E_{tx} , reception, E_{rx} , and in data prediction, E_{dp} , modes [18]. The per-node energy consumption based on the protocol operation modes and the e2e packet delay are provided in “DMAC Protocol” of Appendix. From Eqs. (28) and (29), we define the following energy-delay functions, where T_w , and T_{sync} are the key DMAC’s parameters:

- (a) The network energy consumption function:

$$E^{DMAC} = \frac{\alpha_1}{T_w} + \alpha_2 \frac{T_{sync}}{T_w} + \alpha_3 T_{sync} + \frac{\alpha_4}{T_{sync}} + \alpha_5 \tag{14}$$

where $\alpha_1 = \sum_{n=1}^N (T_{up} + T_{cw} + T_{data})$, $\alpha_2 = \sum_{n=1}^N 2\theta$, $\alpha_3 = \sum_{n=1}^N 2\theta F_l^n$, $\alpha_4 = \sum_{n=1}^N (2\theta I^n + (T_{cs} + T_{data}) F_{out}^n + (T_{up} + T_{cw} + T_{data}) F_l^n)$, and $\alpha_5 = \sum_{n=1}^N (T_{cs} + T_{hdr} + (T_{up} + T_{cw} + T_{data}) I^n)$.

- (b) The e2e packet delay function:

$$L^{DMAC} = \max_{n \in N} (\beta_1 T_w + \beta_2 T_{sync} + \beta_3) \tag{15}$$

where $\beta_1 = 1/2$, $\beta_2 = \sum_{i=1}^{d^n} 2\theta$ and $\beta_3 = \sum_{i=1}^{d^n} (T_{cw}/2 + T_{data})$. Parameters used in the formulas for the D-MAC protocol are defined in Table 2.

Fig. 5 DMAC’s transmission, receiving, and data prediction modes

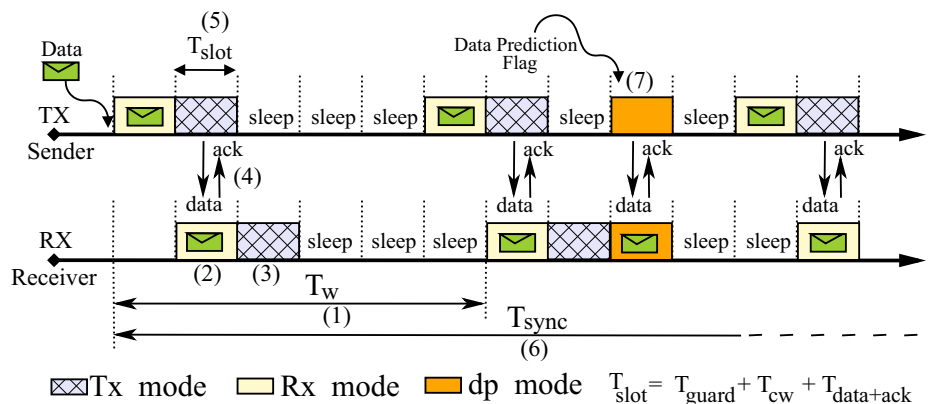


Table 2 DMAC, LL-MAC, Diff-MAC, and DuoMAC symbols used in energy & Delay equations with typical values [18, 33, 34]

MAC	Parameter and description	Values	
DMAC	T_w DMAC frame duration (ms)	[1000, 10,000]	
	T_{sync} DMAC synchronization period (ms)	60,000	
	T_{cw} Contention window size (ms)	15×0.62	
	T_{hdr}, T_{ack} pkt header & Ack duration (ms)	$\frac{10+L_{pkt}}{R}$	
LL-MAC	T_w LL-MAC wake-up period (ms)	[1000, 10,000]	
	T_{cw} Contention window size (ms)	15×0.62	
	T_{hdr}, T_{ack} pkt header & Ack duration (ms)	$\frac{9+L_{pkt}}{R}$	
Diff-MAC	T_w Diff-MAC slot duration (ms)	[1000, 10,000]	
	T_{sync} Diff-MAC synchronization period (ms)	$T_{guard} + T_{cw} + T_{hdr}$	
	T_{active} Diff-MAC active period (ms)	$(0.05, 0.5) * T_w$	
	T_{sleep} Diff-MAC sleep period (ms)	$T_w - (T_{active} + T_{sync})$	
	T_0 Diff-MAC discovery interval (s)	360	
	T_{cw}^{RT} RT Contention window size (ms)	12×0.62	
	T_{cw}^{NRT} NRT Contention window size (ms)	24×0.62	
	T_{hdr} Control packets (RTS/CTS/Ack) duration (ms)	$\frac{8+L_{pkt}}{R}$	
	DuoMAC	T_{cp} DuoMAC channel polling period (ms)	$T_{cp}^* \in [100, 5000]$
		T_w X-MAC wake-up period (ms)	$T_w^* [1000, 10, 000]$
T_{on} Time (ms) to turn radio on and probe channel for RT		$T_{cs} + T_{rts} + T_{cl}$	
T_{cl} CTS listen period (ms)		0.95	
T_{rts} RTS Strobe preamble duration (ms)		$\frac{5+L_{pkt}}{R}$	
T_{cts} CTS (or early acknowledgement) packet duration (ms)		$\frac{5+L_{pkt}}{R}$	
T_{cw}^{RT} RT Contention window size (ms)		12×0.62	
T_{cw}^{NRT} NRT Contention window size (ms)		24×0.62	
T_{hdr}, T_{ack} pkt header and Ack duration (ms)		$\frac{9+L_{pkt}}{R}$	
$T_{rx}^{B_w}, T_{rx}^{B_{nw}}, T_{tx}^{B_w}, T_{tx}^{B_{nw}}$ B_w and B_{nw} beacons TX and RX durations (ms)		$T_{hdr} + P/R$	
\mathcal{P}_{lq}^n Link quality at node n		$PRR * \frac{b_{min}}{b_{max}} \in]0, 1[$	
a/b Linear estimator coefficients		–	

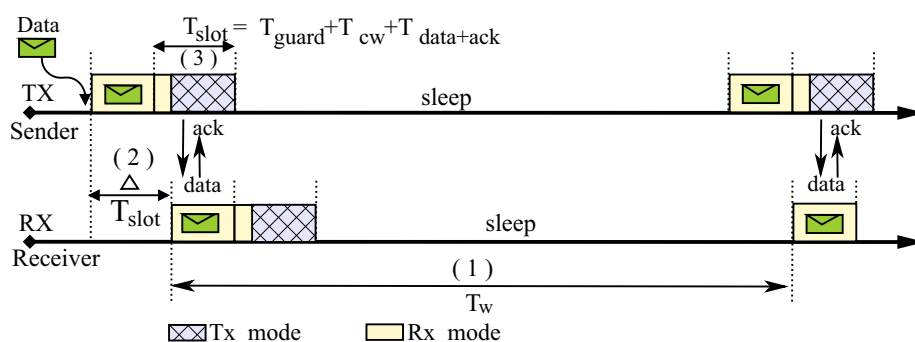


Fig. 6 LL-MAC’s transmission, receiving, and sleep modes

4.4 LL-MAC protocol

LL-MAC (Low Latency MAC) [34] is an asynchronous protocol that uses the same staggered active times of nodes– similarly to DMAC– to reduce sleep delay in a

multi-hop transmissions. Referring to Fig. 6, nodes wake-up every T_w to receive and send packets (1). If Δ is the necessary time to transmit one packet (slot duration), then the node’s reception time will be shifted Δ time before its parent’s wake-up time (2). Due to the asynchronous

scheme of LL-MAC, nodes must compensate with a guard time T_{guard} ($T_{guard} = 40T_w$) for clock drift in every operation cycle T_w (3). The transmission slot includes contention, guard time, and data transmission with acknowledgment. Energy is spent in transmitting and receiving modes, respectively E_{tx} and E_{rx} . The LL-MAC per-node energy consumption based on the protocol operation modes and the e2e packet delay are provided in “LL-MAC Protocol” of Appendix . From Eqs. (30) and (31), the following energy-delay functions are derived, where T_w is the key LL-MAC’s parameter:

- (a) The network energy consumption function:

$$E^{LLMAC} = \frac{\gamma_1}{T_w} + \gamma_2 \tag{16}$$

where $\gamma_1 = \sum_{n=1}^N (T_{up} + T_{cw} + T_{data})$ and $\gamma_2 = \sum_{n=1}^N (2\theta + (T_{cs} + T_{data})F_{out}^n)$.

- (b) The e2e packet delay function:

$$L^{LLMAC} = \max_{n \in N} (\delta_1 T_w + \delta_2) \tag{17}$$

where $\delta_1 = 1/2 + \sum_{i=1}^{d^n} 2\theta$ and $\delta_2 = \sum_{i=1}^{d^n} (T_{cw}/2 + T_{data})$. Parameters used in the formulas for the LL-MAC protocol are defined in Table 2.

4.5 Diff-MAC protocol

Diff-MAC [33] is QoS-aware MAC protocol that adopts a service differentiation through inter and intra queue prioritization. It also uses contention window (CW) adaptation for different traffic: multimedia traffic (RT), non realtime traffic (NRT), and best effort (BE) traffic. This is to provide energy-delay application efficiency as described in Sect. 3.5. In this paper, the analysis is limited to only RT and NRT traffics. Diff-MAC is implemented on the top of a well-known slotted duty-cycled sensor MAC protocol

(SMAC) [32], and it adopts all its features [33]. As depicted in Fig. 7, nodes are synchronized to a common slot structure of a fixed length T_w (1). The slots are divided into an active period T_{active} (2), a sync phase T_{sync} (3), and a sleep phase T_{sleep} (4). Nodes broadcast SYNC beacons to keep the network synchronized (5). The nodes are therefore synchronized every $(C + 1)$ slots and require a clock-drift compensation of $T_{guard} = 2\theta T_w (C + 1)$. The sync phase has a length of $T_{sync} = T_{guard} + T_{cw} + T_{hdr}$. In the active phase, the nodes contend for the channel using RTS/CTS handshake with acknowledgment to send RT and NRT data packets. A different contention window is used for each traffic category used, respectively CW_{RT} and CW_{NRT} (6). Nodes switch off their radios for the duration indicated in RTS/CTS packets to avoid overhearing. In Diff-MAC, energy is spent in the synchronization and in active phases, except when background traffic is exchanged [18]. Diff-MAC per-node energy and RT/NRT e2e packet delay are provided in “Diff-MAC Protocol” of Appendix. From Eqs. (32) and (33), the following energy-delay functions are defined, where T_w , T_{active} , and T_{sleep} are the key Diff-MAC’s parameters:

- (a) The network energy consumption function:

$$E^{Diff-MAC} = \frac{\lambda_1}{T_w} + \frac{T_{active}}{T_w} + \frac{\lambda_2}{T_{sleep}} + \lambda_3 \tag{18}$$

where $\lambda_1 = \sum_{n=1}^N (T_{cw}^{NRT} + T_{hdr})$, $\lambda_2 = \sum_{n=1}^N \frac{1}{T_0}$, and $\lambda_3 = (2\theta - (T_{data} - T_{hdr} - T_{up})F_B^n)$.

- (b) The e2e packet delay function:

$$L^{Diff-MAC}(RT) = \max_{n \in N} (\mu_1 T_w + \mu_2 T_{sleep} + \mu_3 \frac{T_w}{T_{active}} + \frac{\mu_4}{T_{active}} + \mu_5) \tag{19}$$

where $\mu_1 = \theta$, $\mu_2 = 1/2$, $\mu_3 = \sum_{i=1}^{d^n} (\frac{T_{cw}^{RT}}{2} + T_{data})$,

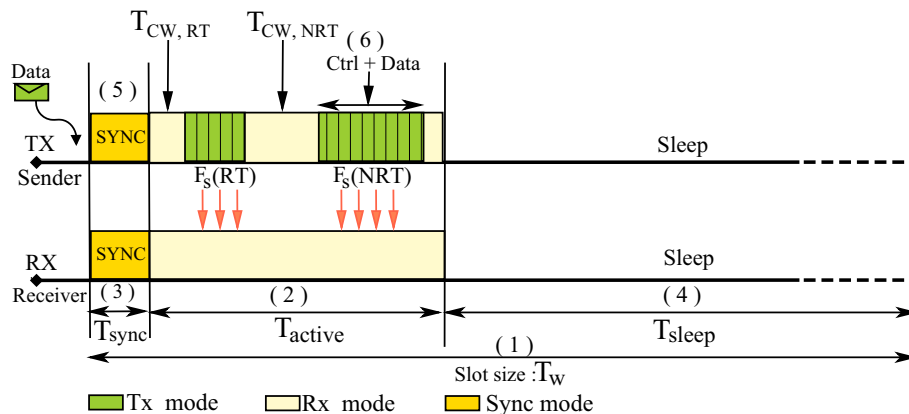


Fig. 7 Diff-MAC’s synchronization, active, and sleep modes

$$\mu_4 = \sum_{i=1}^{d^n} \left(\frac{T_{cw}^{RT}}{2} + T_{data} \right)^2, \text{ and } \mu_5 = \frac{T_{cw}^{NRT} + T_{hdr}}{2}.$$

$$L^{\text{Diff-MAC}}(NRT) = \max_{n \in N} (\varepsilon_1 T_w + \varepsilon_2 T_{sleep} + \varepsilon_3 \frac{T_w}{T_{active}} + \frac{\varepsilon_4}{T_{active}} + \varepsilon_5) \tag{20}$$

where $\varepsilon_1 = \theta$, $\varepsilon_2 = 1/2$, $\varepsilon_3 = \sum_{i=1}^{d^n} \left(\frac{T_{cw}^{NRT}}{2} + T_{data} \right)$, $\varepsilon_4 = \sum_{i=1}^{d^n} \left(\frac{T_{cw}^{NRT}}{2} + T_{data} \right)^2$, and $\varepsilon_5 = \frac{T_{cw}^{NRT} + T_{hdr}}{2}$. Parameters used in the formulas for the Diff-MAC protocol are defined in Table 2.

4.6 DuoMAC protocol

DuoMAC protocol handles two kinds of traffic, namely RT and NRT. Consequently, it uses two operation modes, LDC and HDC, to forward data packets. In this analysis, both LDC and HDC modes are considered to derive the delay and the energy equations. Details concerning the per-node energy and e2e packet delay for RT (HDC mode) and NRT (LDC mode) traffic of DuoMAC are given in “DuoMAC Protocol” of Appendix (description of every term used in the formulas can be found in Table 2). From Eqs. (34–37), we define the following energy-delay functions, where T_{cp} and T_w are the key DuoMAC’s parameters to be considered.

(a) The network energy consumption function:

$$E^{\text{DuoMAC}} = \zeta_1 T_{cp} + \frac{\zeta_2}{T_{cp}} + \zeta_3 T_w + \frac{\zeta_4}{T_w} + \zeta_5 \tag{21}$$

where

$$\zeta_1 = \sum_{n=1}^N \frac{F_{out}^{RT}}{2},$$

$$\zeta_2 = \sum_{n=1}^N \left(T_{on} + \frac{T_{rts} + T_{cl} + 2T_{cts} + 2T_{data}}{4} 3T_{rts} \cdot F_B^{RT} \right),$$

$$\zeta_3 = \sum_{n=1}^N 4\theta F_I^{NRT},$$

$$\zeta_4 = \sum_{n=1}^N (T_{rx}^{B_w} + T_{rx}^{B_{nw}} + T_{tx}^{B_w} + T_{tx}^{B_{nw}}), \quad \zeta_5 = \sum_{n=1}^N \left((T_{on} + \frac{T_{rts} + T_{cl}}{2} + T_{cts} + T_{data}) F_{out}^{RT} + E_{rx}^d + \frac{3}{4} T_{rts} F_B^{RT} + (T_{cw} + T_{data}) F_I^{NRT} + (T_{cs} + T_{data}) F_{out}^{NRT} \right).$$

(b) The e2e packet delay function:

$$L^{\text{DuoMAC}}(RT) = \max_{n \in N} (v_1 T_{cp} + v_2) \tag{22}$$

where $v_1 = \sum_{i=1}^{d^n} 1/2$, $v_2 = \sum_{i=1}^{d^n} \left(\frac{T_{cw}^{RT}}{2} + T_{data} \mathcal{P}_{lq} \right)$.

$$L^{\text{DuoMAC}}(NRT) = \max_{n \in N} (\eta_1 T_w + \eta_2) \tag{23}$$

where $\eta_1 = 1/2$, $\eta_2 = \sum_{i=1}^{d^n} \left(T_{rx}^{B_w} + T_{tx}^{B_{nw}} + \left(\frac{T_{cw}^{NRT}}{2} + T_{data} \mathcal{P}_{lq} \right) \frac{F_{I,NRT}^i}{F_{NRT}} \right)$.

4.7 Analytical results

The equations derived thus far for each protocol are used in the following to compare the protocols under different network, traffic and MAC parameters configurations.

4.7.1 The wake-up period

The network energy, E , and the maximum e2e delay L are measured when varying the wake-up period, T_w . The network connectivity, C , and depth, D , are set to, 8 and, 10, respectively. The traffic generation rate of both RT and NRT is fixed to 0.5 pkt/min. The synchronization period of DMAC is $T_{sync} = 60s$, and the active period of Diff-MAC is set to $0.4 * T_w$, while the channel polling period of DuoMAC is set to $0.1 * T_w$. The wake-up period T_w is varied from 1 to 10 s, and the results are depicted in Fig. 8(a, b). The results show that LL-MAC outperforms all MACs from the energy perspective. This is because LL-MAC does not rely on synchronization and handles only a single packet per cycle. DuoMAC consumes lower energy compared to Diff-MAC when increasing the wake-up

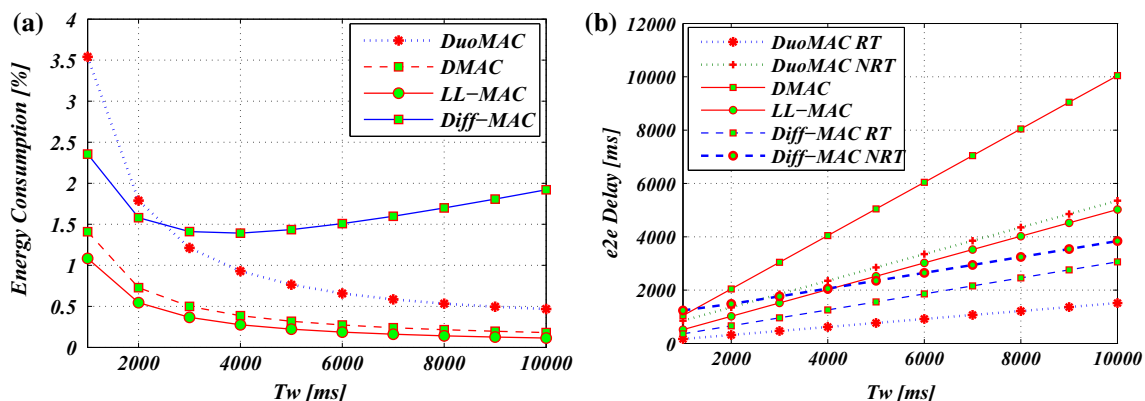


Fig. 8 a Energy E obtained by varying $T_w \in [1000, 10,000]$ ms. b e2e Latency L obtained by varying $T_w \in [1000, 10,000]$ ms

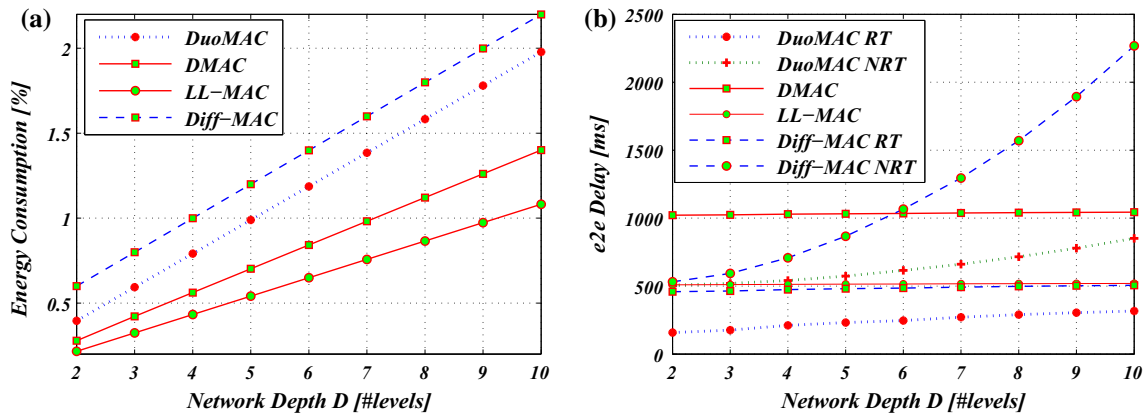


Fig. 9 a Energy E obtained by varying $D \in [1, 10]$ levels. b e2e Latency L obtained by varying $D \in [1, 10]$ levels

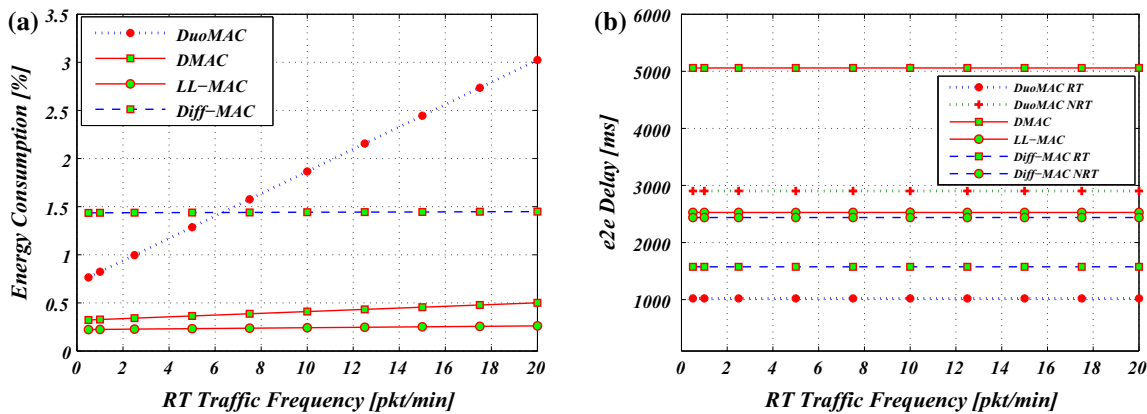


Fig. 10 a Energy E obtained by varying $F_{RT} \in [0.5, 20]$ pkt/min. b e2e Latency L obtained by varying $F_{RT} \in [0.5, 20]$ pkt/min

period, due to the use of the cascading scheme. With respect to latency, DuoMAC outperforms the other protocols for RT traffic and it performs better than DMAC and closer to LL-MAC for NRT traffic [Fig. 8(b)]. From both figures, the conclusion is that although LL-MAC offers better performance with respect to energy consumption it does not have the possibility to differentiate between NRT and RT traffic. On the other hand, DuoMAC, for NRT offers similar latencies than LL-MAC and trade-offs energy consumption to offer lower latencies for RT traffic. Diff-MAC also offers similar trade-offs than DuoMAC; it offers better energy consumption but worst RT latency than DuoMAC. The main conclusion of this set of plots is that in order to offer better RT e2e delay, it is necessary to sacrifice energy. Section 5 is devoted to optimize DuoMAC parameters in order to improve energy consumption without penalizing too much the RT e2e delay.

4.7.2 The network depth

Here the network depth D is varied from 1 to 10 levels to measure the performance metrics that are depicted in Fig.

9(a, b). The network connectivity, C , is set to 8. The traffic generation of RT and NRT is fixed to 0.005 pkt/min. The wake-up period of all protocols is set to 1000 ms, the synchronization period of DMAC to 60 s, the active period of Diff-MAC to 200 ms, and the channel polling period of DuoMAC to 200 ms. The results show that the energy consumption of DuoMAC is higher than LL-MAC and DMAC and lower than Diff-MAC in all the network depth configurations. The reason again is that LL-MAC and DMAC does not differentiate between RT and NRT traffic. However, these protocols offer worst latencies than the DuoMAC for RT traffic. On the other hand, Diff-MAC consumes more energy and produces higher latencies than DuoMAC.

4.7.3 RT traffic frequency

In this part, the RT traffic generation frequency, F_{RT} , is varied from 0.5 to 20 pkt/min. The network connectivity and depth are set to 8 and 10, respectively. The wake-up period of all protocols is set to 5000 ms, while the traffic generation of NRT is fixed to 1 pkt/min. The results are

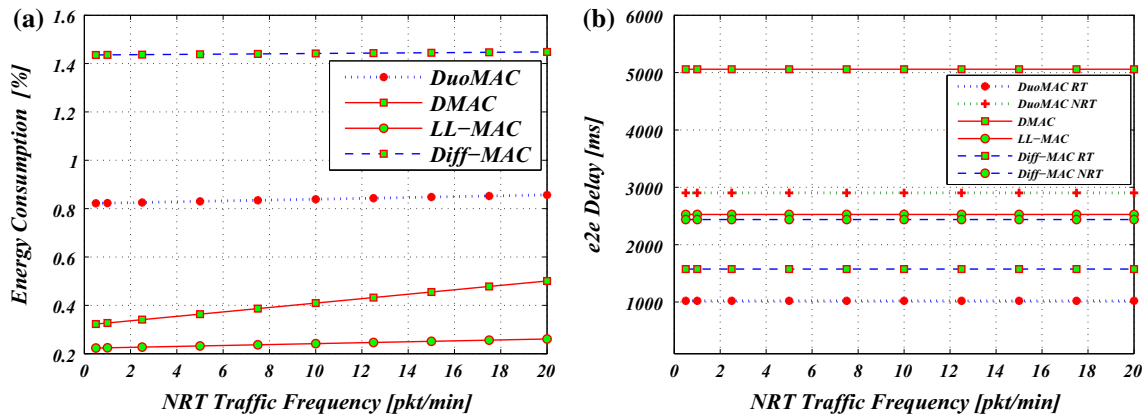


Fig. 11 a Energy E obtained by varying $F_{NRT} \in [0.5, 20]$ pkt/min. b e2e Latency L obtained by varying $F_{NRT} \in [0.5, 20]$ pkt/min

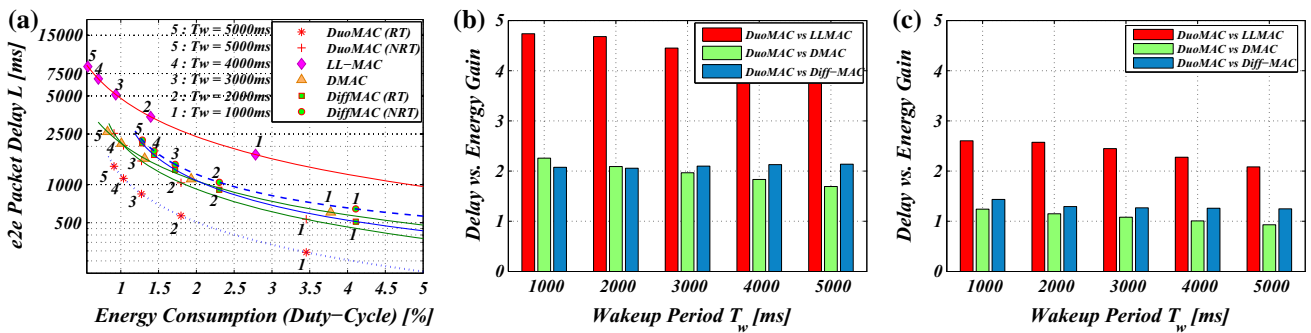


Fig. 12 a Delay versus energy for different values of wakeup periods T_w and b the gain under RT traffic for different values of wakeup periods T_w and c the gain under NRT traffic for different values of wakeup periods T_w

depicted in Fig. 10(a, b). The energy consumption of DuoMAC is normally affected by increasing the RT traffic rate, while those of the other protocols increase slightly. This is because DuoMAC switches more frequently to HDC mode as the RT rate rises to quickly forward realtime packets. The benefit from this is the lower latency compared to all the other protocols, Fig. 10(b). Note that due to the low data rate, delays related to retransmissions and buffering are not considered in the analysis, which justifies independency of the e2e delay with traffic variation.

4.7.4 NRT traffic frequency

Here, the RT traffic generation is set to 1 pkt/min, and the NRT traffic F_{NRT} is varied from 0.5 to 20 pkt/min. The results are depicted in Fig. 11(a, b). In this case, DuoMAC’s energy is not affected by NRT traffic variability, as it becomes less delay sensitive. DuoMAC performs better than Diff-MAC, while LL-MAC achieves the best results in energy consumption. The same results as when varying RT traffic can be observed for the e2e delay. The results show that DuoMAC outperforms all the other protocols by

assuring the lowest latency for RT, but at the price of an increase in power consumption. The results also show that there is an important gap in power consumption between RT and NRT modes of DuoMAC, which justifies the switching strategy between the two modes adopted by DuoMAC. The results also reveal that some intrinsic parameter such as, T_w , have significant effect. So there is ample room for improvement by optimizing such parameters, which is the aim of Sect. 5. Finally, the results show that LL-MAC has the best performance with respect to power consumption, so it will be used as a reference for comparison in the testbed/simulation protocol evaluation.

4.7.5 Delay versus energy gain

We have shown the comparison between MAC protocols for different network, traffic and MAC parameters configurations. The results reveal that DuoMAC protocol can reduce much better the e2e delay over other MAC protocols especially for RT traffic at the cost of some energy consumption. Given that the energy and the delay are the conflicting performance metrics, an efficient MAC protocol

must be tuned with the optimal parameters that permit to achieve the trade-off between them. Figure 12(a) depicts the energy and the e2e delay resulted for different wake-up periods T_w which represents the key MAC parameter that affects both performance metrics (energy and delay). It can be observed that for values of $T_w \in [1000, 1500]$ ms, DuoMAC can achieve best tradeoff between the two metrics. Quantitatively speaking, we propose a metric, G , unifying the energy and the delay performance metrics that expresses the gain of our protocol over other MAC protocols. This metric represents the ratio between what we win in terms of e2e delay reduction ratio, and what we pay in terms of additional energy consumption ratio. This metric relating DuoMAC, to some other protocol, say MAC_x , is calculated as follows:

$$G = \frac{R_L}{R_E} \tag{24}$$

where R_L represents the ratio of the delay reduction (L^{MAC_x} / L^{DuoMAC}) and R_E represents the ratio of the cost (E^{DuoMAC} / E^{MAC_x}). Figure 12(b, c) depict the gain of DuoMAC over other MAC for RT and NRT traffic respectively. It can be observed that the gain is much better for RT than NRT traffic. The gain for instance is more important when comparing with LL-MAC than DMAC and Diff-MAC. It can be observed also that this gain decreases generally with the increase of T_w values where the reduction in delay and the additional energy spent become less important.

5 Parameters optimization

5.1 Optimization problem

Given the application requirements in terms of initial energy budget and the tolerated e2e packet delay, the choice of MAC parameters is of great importance; yet their choice is currently done by system designers based on repeated real experiences [5], which may be optimal for one objective that is not necessarily optimal for others and can yield a performance far off the desired results. DuoMAC protocol is optimized dynamically as a constrained optimization problem. Key performance metrics of the system scenario are considered, namely energy consumption, E , and the e2e latency of RT traffic, L . The MAC parameter optimization problem involves optimizing the network energy E as objective function subject to e2e latency L as constraint. In long-term traffic monitoring systems, the major concern is typically the system lifetime expressed by the energy consumption, but at the same time with some bound guarantee on delay in delivering measurements of RT traffic $L^{DuoMAC}(RT) \leq L_{RT}^{max}$. The link quality represents

the only network state information used as input for our protocol optimization. It is expressed by \mathcal{P}_{lq}^n , the probability of successful transmission over a link at node n , and it is calculated using the effective link quality estimation metric defined in Sect. 3.5.

There are two specific and adjustable parameters in DuoMAC implementation that affect the protocol performance, (1) the channel polling period to check for possible incoming RT traffic, T_{cp} and (2) the wake-up period to send beacons and NRT traffic, T_w . Both variables are expressed as a vector $X = [T_{cp}, T_w]$. Based on this approach, minimizing the energy consumption subject to a minimum e2e latency on RT traffic is specified as,

$$\begin{aligned} \text{(P1)} \quad & \text{Minimize} \quad E(X) \\ & \text{S. t.} \quad L_{RT}(X) \leq L_{RT}^{max} \\ & \text{Var.} \quad X, \end{aligned} \tag{25}$$

From the network energy E^{DuoMAC} and the e2e latency functions on realtime traffic $L^{DuoMAC}(RT)$ defined by Eqs. (21,22), the optimization problem (25) becomes:

$$\begin{aligned} \text{(P2)} \quad & \text{Minimize} \quad \left(\zeta_1 T_{cp} + \frac{\zeta_2}{T_{cp}} + \zeta_3 T_w + \frac{\zeta_4}{T_w} + \zeta_5 \right), \\ & \text{S. t.} \quad T_{cp} \leq \frac{L_{RT}^{max} - v_2}{v_1}, \\ & \text{Var.} \quad T_{cp}, T_w \end{aligned} \tag{26}$$

The solution of the optimization problem (P2), (T_{cp}^*, T_w^*) , is derived in the following. The objective function of (P2) is a function of two independent variables, T_{cp} and T_w , where ζ_i and v_i are positive constants. Let us denote the function by $f(x, y)$, which is in the form of: $\zeta_1 x + \frac{\zeta_2}{x} + \zeta_3 y + \frac{\zeta_4}{y} + \zeta_5$. This two-variable function has a global minimum at point $(x_c, y_c) = \left(\sqrt{\frac{\zeta_2}{\zeta_1}}, \sqrt{\frac{\zeta_4}{\zeta_3}} \right)$, which represents the solution of the system. Therefore, the global optimal parameters (T_{cp}^*, T_w^*) of DuoMAC are given by,

$$\begin{aligned} T_{cp}^* &= \begin{cases} x_c = \sqrt{\frac{\zeta_2}{\zeta_1}} & \text{if } x_c \leq \frac{L_{RT}^{max} - v_2}{v_1}, \\ \frac{L_{RT}^{max} - v_2}{v_1} & \text{otherwise} \end{cases} \\ T_w^* &= y_c = \sqrt{\frac{\zeta_4}{\zeta_3}} \end{aligned} \tag{27}$$

5.2 Collection and dissemination

In DuoMAC, collection of network state information is achieved using data packet headers and the dissemination

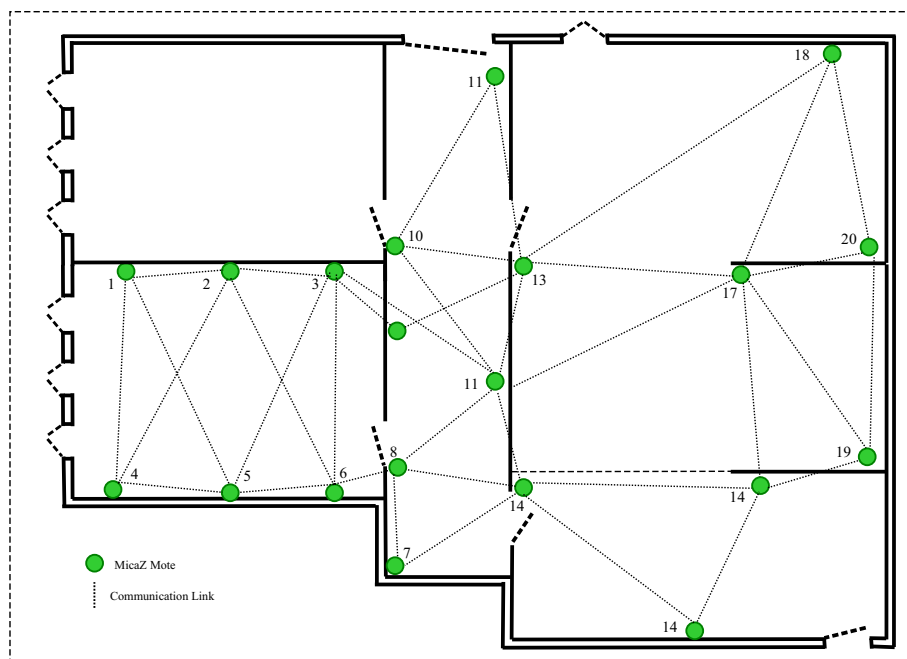


Fig. 13 TestBed platform with MicaZ motes placement and communication links

of new MAC parameters T_w^* and T_{cp}^* is realized using regular B_w beacons exchange. First, the optimization is triggered periodically by the sink. It includes a start command in the B_w beacon, which is propagated by each node to a connected children. Every node runs in one of the protocol operation modes (LDC or HDC) and forwards data packets from level d_i to level d_{i-1} . When a node n receives a B_w beacon with the start command, it piggybacks its network state information \mathcal{P}_{lq}^n in the data packet header. By receiving this information from all nodes, the sink is responsible of calculating and disseminating the new optimal MAC parameters (T_w^* and T_{cp}^*) to all the network using B_w beacons. The time of a complete optimization round depends on the necessary time to propagate the start command and receive the input information to and from all the nodes. The propagation of new parameters from the sink to all nodes necessitate $D \times T$, where T is the cycle duration and D is the network depth. We believe that despite the existence of many collection and dissemination mechanism like CTP [14] and Dip [19] that can be integrated, the choice of using the built-in packet exchange of DuoAMC is reasonable and will achieve better performance since it does not require any additional overhead, so no more energy is consumed.

6 Experimental performance evaluation

The protocol optimization is evaluated through an extensive set of experiments under the considered traffic and network model, using MicaZ platform. The optimized

DuoMAC, termed in what follows DuoMAC*, has been compared with a non-optimized version DuoMAC (without parameter optimization module), and LL-MAC [34], which is shown to be the best candidate among other MAC protocols in the analytical comparison (Sect. 4) with respect to the energy consumption. The evaluation is carried out with both simulations using Avrora [1], which accurately emulates MicaZ, i.e. the AVR cycle execution and the RF CC2420 physical layer [4], and with experimentations on MicaZ real motes using a Testbed. The testbed contains 20 MicaZ motes placed on top of walls in several offices in our building as depicted in Fig. 13. The figure also shows the communication links between the deployed nodes. We point out, from the experiments performed, that the communication links have relatively high loss rates (up to 20 %). Every node acts as source and generates RT and NRT packets periodically, respectively with frequency F_{RT} and F_{NRT} . Very low data rates are considered (0.32–20 packet/min) which is common in low data rate applications [18]. In the simulation, a grid topology as depicted in Fig. 4 is used, with maximum range of 15 m [1], and the unit disk with $C = 8$. The delay requirement of RT traffic was chosen according to the representative monitoring system, urban traffic monitoring, and set to $L_{RT}^{max} = 1s$. Tables 1 and 2 sketch all the setup configuration of the considered network and traffic model as well as the typical parameter values for the CC2420 radio from [18]. In both simulation and experimentation, a phase of neighbors discovery and tree construction with a minimum hop-count routes, according to the network model of Sect. 4.1, is executed before

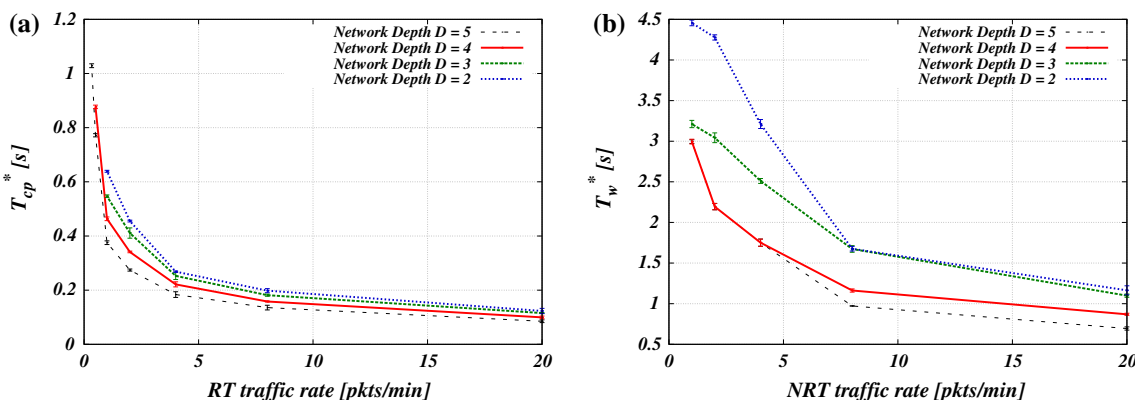


Fig. 14 **a** Optimized channel polling period T_{cp}^* and **b** optimized wakeup period T_w^* of DuoMAC for different network depths D : 2, 3, 4, and 5 and varying the RT traffic rate in $[0.32\text{--}20]$ $\text{pkts}/\text{node}/\text{min}$

starting experiments. Each experiment is repeated 30 times and each point in the following curves comes from the average result of all experiments, where bars are represented with 95 % of confidence interval.

6.1 Optimal parameters

First, the simulation results of the protocol optimization with parameter adaptation are presented. The RT and NRT traffic generation frequency has been varied from 0.32 to 20 $\text{pkts}/\text{node}/\text{min}$. The resulted optimal parameters are traced for different network sizes and diameters ($D = 2, 3, 4, \text{ and } 5$). Figure 14(a, b) depict the optimal channel polling T_{cp}^* and the optimal wake-up period T_w^* parameters, respectively.

T_{cp}^* and T_w^* progressively decrease with the increase of traffic rate and have lower values for large networks ($D = 4, D = 5$) compared to small network diameters ($D = 2, D = 3$). This can be explained by the fact that the increase in NRT and RT traffic causes the increase of the node active period, which requires to imperatively lower the duration of T_{cp} and T_w . On the other hand, it is obvious that the farther the node is from the sink ($D = 4, D = 5$), the more the wakeup period will be lowered to meet the delay requirement.

6.2 The e2e delay

Figure 15(a) shows the average e2e delay of DuoMAC*, DuoMAC, and LL-MAC for RT packets. These results are obtained using Avrora and by fixing the network size to 55 nodes, and increasing the traffic rate from 0.32 to 20 $\text{packet}/\text{node}/\text{min}$. The resulted average e2e delay of DuoMAC*, DuoMAC, and LL-MAC for NRT packets is depicted in Fig. 15(b). The results show that the average delay decreases with the increase of the traffic rate, and that

DuoMAC* outperforms DuoMAC with regard to RT traffic (about 79ms lower for RT and 1.06 s lower for NRT on average), and the difference is much more important when compared with LL-MAC for both RT and NRT traffic (2.4s in LL-MAC vs. 0.3 in DuoMAC, and 0.22 s in DuoMAC*, as average RT latency). This is due to the automatic decrease of T_{cp} and T_w periods. The same scenario was investigated in real motes using the TestBed, where the results are depicted in Fig. 16(a) for RT traffic, and in Fig. 16(b) for NRT traffic, respectively. Results confirm superiority of DuoMAC* regarding the delay reduction.

In a second scenario, traffic rate has been fixed to 20 pkts/min , and the average e2e delay is measured using Avrora, for different network depths ($D = 2, 3, 4, 5$). Note that this scenario cannot be tested in our testbed due to limitation in number of motes, which prevents construction of 4 and 5 levels networks. The resulted average delay for RT and NRT traffic is depicted in Fig. 17(a, b), respectively. It can be noted that the average delay increases with the network depth, and DuoMAC* delay is much lower than DuoMAC for RT traffic. The delay of DuoMAC* and DuoMAC gets close to each other for NRT traffic, for which there is no delay constraint. Both protocols clearly outperform LL-MAC. This confirms the results obtained in Sect. 6.1 and shows that DuoMAC* tunes adequately its parameters T_{cp} , and T_w , to meet delay requirements.

6.3 The duty cycle

The duty cycle of a node is defined as the ratio of the active time to the sleep time. It is obvious that the lower is the duty-cycle, the better is the performance of the protocol in terms of energy consumption. So in the following, energy consumption is expressed in terms of duty cycle. Protocols are compared using Avrora under a network configuration with 55 nodes and a maximum depth $D = 5$. We vary the

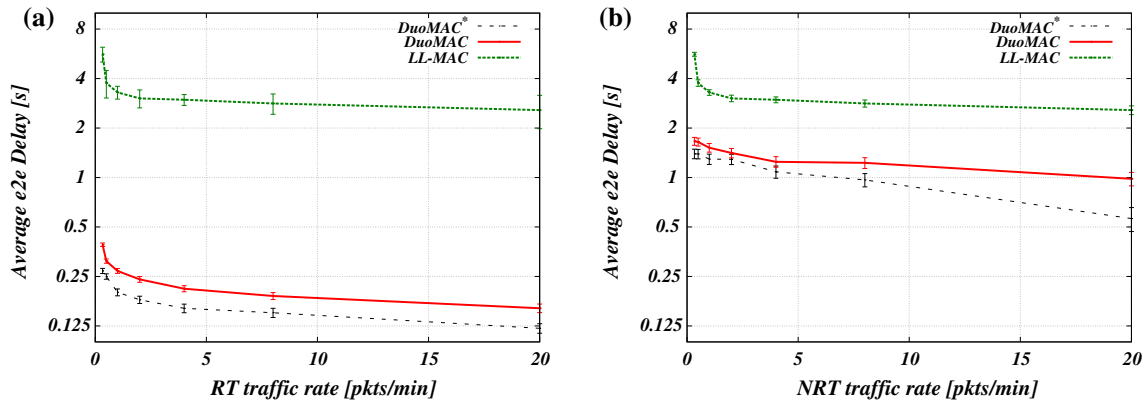


Fig. 15 The average e2e delay of **a** RT traffic and **b** NRT traffic obtained by DuoMAC*, DuoMAC, and LL-MAC when increasing the traffic rate using Avrora

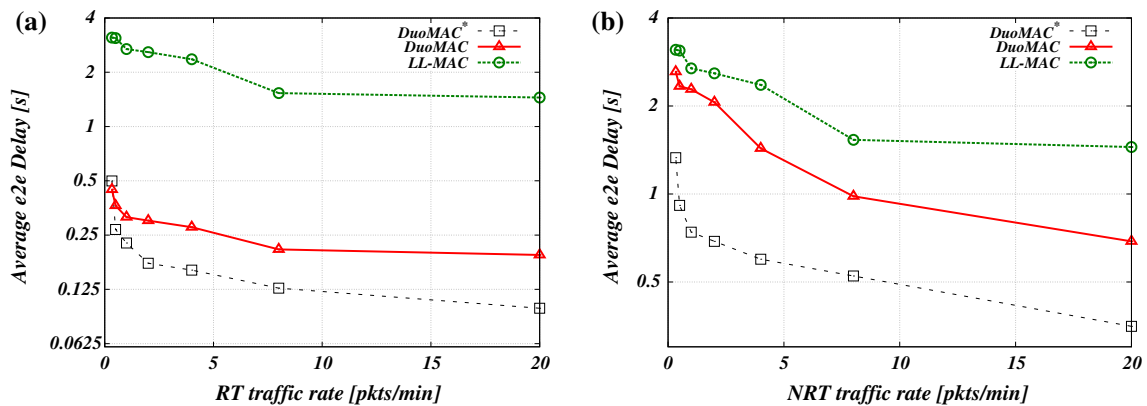


Fig. 16 The average e2e delay of **a** RT traffic and **b** NRT traffic obtained by DuoMAC*, DuoMAC, and LL-MAC when increasing the traffic rate using TestBed

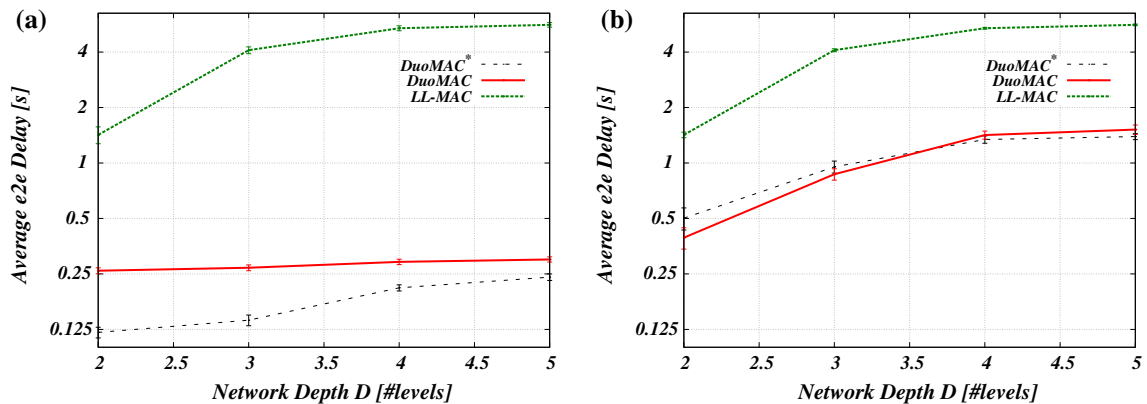


Fig. 17 The average e2e Delay of **a** RT traffic and **b** NRT traffic obtained by DuoMAC*, DuoMAC, and LL-MAC when varying the network depth ($D = 2, 3, 4,$ and 5) with Traffic rate = 20 pkts/min

traffic generation rate of RT and NRT from 0.32 to 20 pkts/node/min. The average duty-cycle of the network has been measured and the results are depicted in Fig. 18(a). Results

demonstrate that the average duty-cycle of the network increases for DuoMAC* and DuoMAC compared to LL-MAC. This is because the later handles one packet per

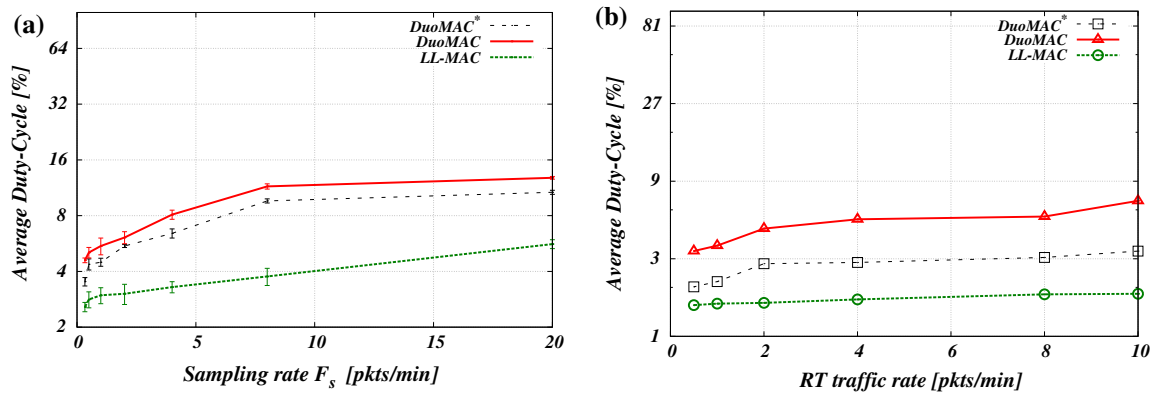


Fig. 18 The average Duty-cycle obtained by DuoMAC*, DuoMAC, and LL-MAC when **a** increasing the sampling rate from 0.32 to 20 pkts/min using Avrora and **b** fixing NRT traffic rate to 10 pkts/min and decreasing RT Traffic from 10 to 0.16 pkts/min using TestBed

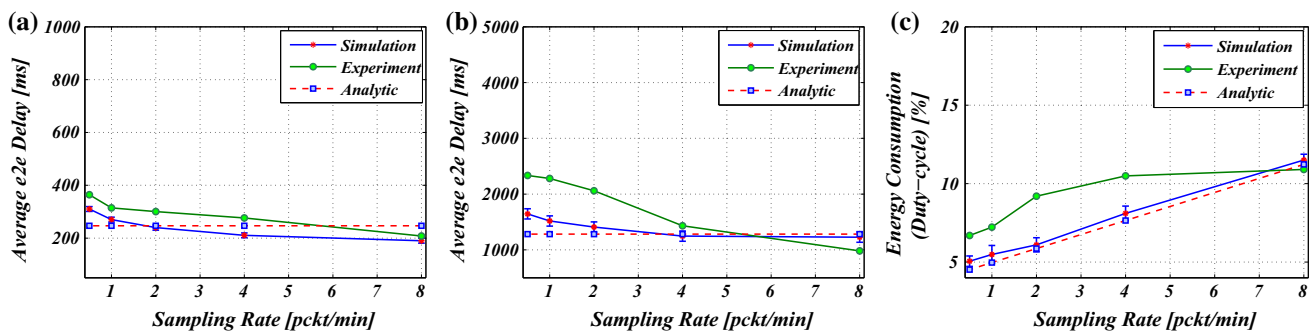


Fig. 19 The results obtained from DuoMAC Protocol analysis, simulation, and experimentation for **a** the e2e delay of RT traffic, **b** the e2e delay of NRT traffic, and **c** the energy consumption when varying the sampling rate from 0.16 to 10

cycle, and does not make any adaptation for RT traffic, where the other protocols balance energy optimization and delay optimization to accelerate RT packets delivery. On the other hand, DuoMAC* achieves better energy reduction, about 2 % compared to DuoMAC. This is due to the accurate built-in parameter adaptation of DuoMAC*, where T_w and T_{cp} are set optimally. Figure 18(b) shows the average duty-cycle of experimental results using the Test-Bed. The aim of this experiment is to measure the energy consumption of the network when the RT frequency gets very low. The generation rate of NRT traffic has been fixed to 10 pkts/node/min and the RT rate has been varied from 10 pkts/node/min (equal to NRT rate) down to 0.16 pkts/node/min. It has been observed that the power consumption of DuoMAC and DuoMAC* is higher than that of LL-MAC, and it increases with the rise of RT rate. This is because these protocols run most of time in HDC mode to forward RT traffic and thus consume more energy. However, it is noted that DuoMAC* considerably reduces the energy consumption compared to DuoMAC. Figure 19 shows the analytical, the simulation, and the experiments results of DuoMAC obtained for (a) the e2e delay of RT, (b) the e2e delay of NRT, and (c) the energy when varying

the sampling rates $\in (0.16, 10)$ pkts/min. The plots show how the analytical model fits quite well the simulations and are near the experimental results, confirming the conclusions drawn from the protocol modeling.

7 Conclusion

DuoMAC has been presented in this paper, a new MAC protocol with runtime parameter adaptation that targets applications with heterogeneous traffic. The proposed protocol assures energy efficiency and delay constrained data delivery by balancing energy-minimization with delay-minimization, and adaptively switching between two states according to the dominating traffic in the network. Such protocol can be used in monitoring system applications where sensed data of the monitored area must be delivered to the control center by satisfying the application requirements in terms of e2e delay and network lifetime. A comprehensive analysis has been performed to compare DuoMAC protocol against some state-of-the-art energy-delay efficient duty-cycled MAC protocols, respectively DMAC, LL-MAC, and Diff-MAC. The results rise that

LL-MAC is the best candidate for energy consumption in all cases while DuoMAC uses to provide the lowest latency to realtime traffic (RT). The results of this analysis have driven the parameter optimization approach that has been integrated to the proposed protocol, where an energy/delay constrained optimization problem has been formulated, and runtime resolution has been proposed to accordingly derive the optimal parameters that nodes must tune-up with in order to achieve optimized performance, and satisfy the application requirements. The proposed optimization has been evaluated through simulations and experimentations on MicaZ notes. Results show that optimized version of the proposed protocols (DuoMAC*) is tunable and meets delay requirements, and it shows clear reduction of the delay over LL-MAC and the ordinary DuoMAC (with static parameters). The improvement has been for both RT and NRT traffic types. Both versions clearly outperform LL-MAC with respect to e2e latency. However, this gain in delay minimization comes at the cost of a moderate additional energy consumption when compared to LL-MAC due to the increase of the channel polling period needed to satisfy delay constraints on RT traffic, which inevitably rises the duty-cycling. In addition to delay improvement, DuoMAC* reduces the energy cost compared to DuoMAC and exhibits a good distribution of the radio duty-cycle, which enables to improve the network lifetime.

Acknowledgments This works is partly supported by Algerian Ministry of Higher Education through the DGRSDT, and the National Spanish funding TIN2013-47272-C2-2-R.

Appendix: MAC energy-delay models

In this section, the per node energy and e2e delay models is provided for DMAC, LL-MAC, Diff-MAC and DuoMAC protocols following the operation modes of each protocol. The selected MAC protocols are chosen as representative of families of synchronous and asynchronous protocols as mentioned in the related work. Terms appearing in equations are described in Table 2.

DMAC protocol

The DMAC [20] per-node energy and e2e packet delay are given by:

(a) The Energy of node n :

$$E^n = E_{tx}^n + E_{rx}^n + E_{dp}^n \tag{28}$$

where $E_{tx}^n = (T_{cs} + T_{data})F_{out}^n$, $E_{rx}^n = \frac{T_{up} + 4\theta T_{sync}/2 + T_{cw} + T_{data}}{T_w} + \frac{T_{cs} + T_{hdr}}{T_{sync}}$, and $E_{dp}^n = (F_l^n + |I_d| \frac{1}{T_{sync}})(T_{up} + 2\theta T_{sync} + T_{cw} + T_{data})$.

(b) The delay of node n at level d^n :

$$L^n = \frac{T_w}{2} + \sum_{i=1}^{d^n} \left(\frac{4\theta T_{sync}}{2} + T_{cw}/2 + T_{data} \right) \tag{29}$$

where $T_{data} = T_{hdr} + P/R + T_{ack}$.

LL-MAC protocol

The LL-MAC [34] per-node energy and e2e packet delay are given by,

(a) The Energy of node n :

$$E^n = E_{tx}^n + E_{rx}^n \tag{30}$$

where $E_{tx}^n = (T_{cs} + T_{data})F_{out}^n$ and $E_{rx}^n = \frac{T_{up} + 4\theta T_w/2 + T_{cw} + T_{data}}{T_w}$.

(b) The delay of node n at level d^n :

$$L^n = \frac{T_w}{2} + \sum_{i=1}^{d^n} \left(\frac{4\theta T_w}{2} + T_{cw}/2 + T_{data} \right) \tag{31}$$

where $T_{data} = T_{hdr} + P/R + T_{ack}$.

Diff-MAC protocol

The Diff-MAC [33] per-node energy and RT/NRT e2e packet delay are given by,

(a) The Energy of node n :

$$E^n = \frac{T_{cw}^{NRT} + T_{hdr}}{T_w} + \frac{T_{active}}{T_w} + \frac{T_{sleep}}{T_0} + 2\theta - (T_{data} - T_{hdr} - T_{up}) \tag{32}$$

(b) The RT and NRT delays of node n at level d^n :

$$L^n(RT) = \frac{T_{sleep}}{2} + \theta T_w + \frac{T_{cw}^{NRT} + T_{hdr}}{2} + \sum_{i=1}^{d^n} \left(\left(\frac{T_{cw}^{RT}}{2} + T_{data} \right) \frac{T_w}{T_{active}} + \frac{(T_{cw}^{RT}/2 + T_{data})^2}{T_{active}} \right) \tag{33}$$

$$L^n(NRT) = \frac{T_{sleep}}{2} + \theta T_w + \frac{T_{cw}^{RT} + T_{hdr}}{2} + \sum_{i=1}^{d^n} \left(T_{msg} \frac{T_w}{T_{active}} + \frac{T_{msg}^2}{T_{active}} \right)$$

$$T_{msg} = \left(\frac{T_{cw}^{RT}}{2} + T_{data} \right) F_l^n + \left(\frac{T_{cw}^{NRT}}{2} + T_{data} \right)$$

where $T_{data} = 4T_{hdr} + P/R$.

DuoMAC protocol

(a) HDC mode (RT traffic):

In HDC mode [illustrated in Fig. 2(b)], the energy is divided into the energy spent (1) for periodic channel polling, E_{cp} , (2) for sending RT packets E_{tx} , (3)

receiving RT packets, E_{rx} , and (4) overhearing background node’s RT packet transmissions, E_{ovr} . Before start sending strobos, a node checks for channel activity for T_{on} . The maximum number of strobos is $\lceil \frac{T_{cp}}{T_{rts}+T_{cl}} \rceil$, and the strobos’ transmission time is $\lceil \frac{T_{cp}}{T_{rts}+T_{cl}} \rceil \frac{T_{rts}+T_{cl}}{2}$ on average. At reception, a node receives half of the packet strobe before the first strobe packet, and a node turns off its radio for T_{cl} between the strobos, to save energy. For the overhearing, the average number of channel polling during one packet transmission is $\frac{T_{rx}}{T_{cp}}$. A node, n , spends energy overhearing one strobe and a half to decode the identifier for all background node’s transmissions, F_B . This yields,

$$E_{HDC}^n = E_{cp}^n + E_{tx}^n + E_{rx}^n + E_{ovr}^n \tag{34}$$

where $E_{cp}^n = \frac{T_{cp}}{T_{cp}}$, $T_{tx} = \lceil \frac{T_{cp}}{T_{rts}+T_{cl}} \rceil \frac{T_{rts}+T_{cl}}{2} + T_{cts} + T_{data}$, $E_{tx}^n = (T_{on} + T_{tx}) \cdot F_{out}^{RT}$, $E_{rx}^n = (\frac{3}{2} T_{rts} + T_{cts} + T_{data}) F_I^{RT}$, $E_{ovr}^n = (\frac{T_{rx}}{T_{cp}} \frac{3}{2} T_{rts}) F_B^{RT}$. For the e2e delay, the RT packet is delayed on average by half the channel polling period, T_{cp} . When accounting for the contention and the transmission time, an RT packet in DuoMAC has a maximum e2e latency of,

$$L^n(RT) = \sum_{i=1}^d \left(\frac{T_{cp}}{2} + \frac{T_{cw}^{RT}}{2} + T_{data} \mathcal{P}_{lq} \right) \tag{35}$$

(b) LDC mode (NRT traffic):

As illustrated in Fig. 2(a), a node spends energy in this mode receiving or sending beacons and NRT packets, respectively E_{rx} and E_{tx} , during T_{recv} and T_{send} , respectively. This includes on one hand, the reception and the transmission of B_w and B_{nw} beacons, the reception of NRT input traffic, F_I , and on the other hand the transmission of NRT output traffic, F_{out} . Note that to receive a packet, a node remains in the receiving state for a complete time-slot T_{slot} . Due to the absence of explicit synchronization in DuoMAC, nodes are required to guard for a minimal beacon exchange, which corresponds to the wake-up time, T_w . The duration of the time-slot must cover the potential clock drift, which is proportional to the time since the last beacon. Consequently, the energy of node, n , is expressed by,

$$E_{LDC}^n = E_{rx}^n + E_{tx}^n = \frac{T_{rx}^{B_w} + T_{rx}^{B_{nw}}}{T_w} + T_{slot} F_I^{NRT} + \frac{T_{tx}^{B_w} + T_{tx}^{B_{nw}}}{T_w} + (T_{data}) F_{out}^{NRT} \tag{36}$$

where $T_{slot} = T_{guard} + T_{data}$, $T_{guard} = 4\theta T_w$, and $T_{data} = T_{hdr} + P/R + T_{ack}$. For the e2e delay, the NRT

packet, on average, is delayed by half the wake-up period T_w , added to the parent’s receive period that includes the time to receive and send beacons, respectively B_w and B_{nw} , as well as the time to receive NRT packets from the children nodes. A NRT packet in DuoMAC, when accounting for the contention and the transmission time of each data packet, has a maximum e2e latency of,

$$L^n(NRT) = \frac{T_w}{2} + \sum_{i=1}^{d^n} T_{recv}^{i-1} \tag{37}$$

where $T_{recv}^i = T_{rx}^{B_w} + T_{tx}^{B_{nw}} + (\frac{T_{cw}^{NRT}}{2} + T_{data} \mathcal{P}_{lq}) \frac{F_I^{NRT}}{F_{NRT}^{RT}}$. The E_{rx} term in Eq. (34) does not depend on T_{cp} . Using the property $\frac{T_{cp}}{T_{rts}+T_{cl}} \leq \lceil \frac{T_{cp}}{T_{rts}+T_{cl}} \rceil \leq \frac{T_{cp}}{T_{rts}+T_{cl}} + 1$, and by taking the upper bound, Eq. (34) can be rewritten as,

$$E_{HDC}^n = \frac{T_{on}}{T_{cp}} + \left(T_{on} + \frac{T_{cp} + T_{rts} + T_{cl}}{2} + T_{cts} + T_{data} \right) F_{out}^{RT} + E_{rx}^n + \left(\frac{T_{cp} + T_{rts} + T_{cl}}{2T_{cp}} + \frac{T_{cts} + T_{data}}{T_{cp}} \right) \frac{3}{2} T_{rts} F_B^{RT} \tag{38}$$

Eq. (36) of LDC becomes,

$$E_{LDC}^n = \frac{T_{rx}^{B_w} + T_{rx}^{B_{nw}} + T_{tx}^{B_w} + T_{tx}^{B_{nw}}}{T_w} + (4\theta T_w + T_{cw}^{NRT} + T_{data}) F_I^{NRT} + (T_{cs} + T_{data}) F_{out}^{NRT} \tag{39}$$

References

1. Avrora, The AVR simulation and analysis framework. <http://compilers.cs.ucla.edu/avrora/>
2. Bachir, A., Dohler, M., Watteyne, T., & Leung, K. K. (2010). MAC essentials for wireless sensor networks. *IEEE Communications Surveys & Tutorials*, 2(12), 222–248.
3. Buettner, M., Yee, G. V., Anderson, E., & Han, R. (2006). X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *ACM SenSys* (pp. 307–320).
4. CC2420 Single-Chip 2.4 GHz RF Transceiver, Texas Instruments. (2010). <http://focus.ti.com/lit/ds/symlink/cc2420>
5. Ceriotti, M., Corrà, M., D’Orazio, L., Doriguzzi, R., Facchin, D., Guna, et al. (2011). Is there light at the ends of the tunnel? Wireless sensor networks for adaptive lighting in road tunnels. In *IPSN* (pp. 187–198).
6. Demirkol, I., Ersoy, C., & Alagoz, F. (2006). MAC protocols for wireless sensor networks: A survey. *IEEE Communications Magazine*, 06, 115–121.
7. Dong, Q., Dargie, W., & Schill, A. (2010). The energy cost of control packets in hybrid MAC Protocols. In *HPCC* (pp. 609–615).
8. Dong, Q., & Dargie, W. (2012). Analysis of the cost of handover in a mobile wireless sensor network. In *WWIC* (pp. 315–322).
9. Dong, Q., & Dargie, W. (2013). A survey on mobility and mobility-aware MAC protocols in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 15(1), 88–100.
10. Doudou, M., Djenouri, D., & Badache, N. (2013). Survey on latency issues of asynchronous MAC protocols in delay-sensitive

- wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 15(2), 528–550.
11. Doudou, M., Djenouri, D., Badache, N., & Bouabdallah, A. (2014). Synchronous contention-based MAC protocols for delay-sensitive wireless sensor networks: A review and taxonomy. *Journal of Network and Computer Applications*, 38, 172–184.
 12. El-Hoiydi, A., & Decotignie, J. D. (2005). Low power downlink MAC protocols for infrastructure wireless sensor networks. *Mobile Networks and Applications MONET*, 10(6), 675–690.
 13. Ergen, S. C., Marco, P. D., & Fischione, C. (2009). MAC protocol engine for sensor networks. In *IEEE GLOBECOM* (pp. 1–8).
 14. Gnawali, O., Fonseca, R., Jamieson, K., Moss, D., & Levis, P. (2009). Collection tree protocol. In *ACM SenSys, Berkeley, CA, USA* (pp. 1–14).
 15. Gungor, V. C., & Hancke, G. P. (2009). Industrial wireless sensor networks: Challenges, design principles, and technical approaches. *IEEE Transactions on Industrial Electronics*, 56(10), 4258–4265.
 16. Kredo, K. I., & Mohapatra, P. (2007). Medium access control in wireless sensor networks. *Computer Networks*, 51(51), 961–994.
 17. Langendoen, K. (2007). Medium access control. In H. Wu & Y. Pan (Eds.), *Wireless sensor networks* (Chap. 20, pp. 535–560). Netherlands: Nova Science Publishers.
 18. Langendoen, K., & Meier, A. (2010). Analyzing MAC protocols for low data-rate applications. *ACM Transactions on Sensor Networks TOSN*, 7(2), 10:1–10:34.
 19. Lin, K., & Levis, P. (2008). Data discovery and dissemination with DIP. In *ACM IPSN* (pp. 433–444).
 20. Lu, G., Krishnamachari, B., & Raghavendra, C. S. (2007). An adaptive energy-efficient and low-latency MAC for tree-based data gathering in sensor networks. *Wireless Communications and Mobile Computing*, 7(7), 863–875.
 21. Meier, A., Woehle, M., Zimmerling, M., & Thiele, L. (2010). Zerocal: Automatic MAC protocol calibration. In *IEEE DCOSS* (pp. 31–44).
 22. Munir, S., Lin, S., Hoque, E., Nirjon, S. M. S., Stankovic, J.A., & Whitehouse, K. (2010). Addressing burstiness for reliable communication and latency bound generation in wireless sensor networks. In *IEEE IPSN* (pp. 303–314).
 23. Park, P. G., Fischione, C., Bonivento, A., Johansson, K. H., & Sangiovanni-Vincentelli, A. L. (2011). Breath: An adaptive protocol for industrial control applications using wireless sensor networks. *IEEE Transactions Mobile Computing*, 10(6), 821–838.
 24. Peng, Y., Li, Z., Qiao, D., & Zhang, W. (2011). Delay-bounded MAC with minimal idle listening for sensor networks. In *IEEE INFOCOM* (pp. 1314–1322).
 25. Polastre, J., Hill, J. L., & Culler, D. E. (2004). Versatile low power media access for wireless sensor networks. In *ACM SenSys* (pp. 95–107).
 26. Saxena, N., Roy, A., & Shin, J. (2008). Dynamic duty cycle and adaptive contention window based QoS-MAC protocol for wireless multimedia sensor networks. *Computer Networks*, 52(13), 2532–2542.
 27. Sun, Y., Gurewitz, O., & Johnson, D. B. (2008). RI-MAC: A receiver-initiated asynchronous duty cycle MAC protocol for dynamic traffic loads in wireless sensor networks. In *ACM SenSys, New York, NY, USA* (pp. 1–14).
 28. Suriyachai, P., Roedig, U., & Scott, A. (2011). A survey of MAC protocols for mission-critical applications in wireless sensor networks. *IEEE Communications Surveys & Tutorials, Issue*, 99, 1–25.
 29. Yahya, B., & Ben-Othman, J. (2009). Towards a classification of energy aware MAC protocols for wireless sensor networks. *Wireless Communications & Mobile Computing*, 12(9), 1572–1607.
 30. Yang, O., & Heinzelman, W. R. (2009). Modeling and throughput analysis for S-MAC with a finite queue capacity. In *ISSNIP* (pp. 409–414).
 31. Ye, W., Silva, F., & Heidemann, J. (2006). Ultra-low duty cycle MAC with scheduled channel polling. In *SenSys* (pp. 321–334).
 32. Ye, W., Heidemann, J., & Estrin, D. (2002). An energy-efficient MAC protocol for wireless sensor networks. *INFOCOM*, 3, 1567–1576.
 33. Yigitel, M. A., Incel, Ö. D., & Ersoy, C. (2011). Design and implementation of a QoS-Aware MAC protocol for wireless multimedia sensor networks. *Computer Communications*, 34(16), 1991–2001.
 34. Yu, Q., Tan, C., & Zhou, H. (2007). A low-latency MAC protocol for wireless sensor networks. In *WiCom* (pp. 2816–2820).
 35. Zheng, T., Radhakrishnan, S., & Sarangan, V. (2011). Modeling and performance analysis of DMAC for wireless sensor networks. In *MSWiM* (pp. 119–128).
 36. Zimmerling, M., Ferrari, F., Mottola, L., Voigt, T., & Thiele, L. (2012). Ptunes: Runtime parameter adaptation for low-power MAC protocols. In *IEEE IPSN* (pp. 173–184).



Mr. Messaoud Doudou He obtained his Engineer Degree in computing science from National Institute of Computer science (INI-Algiers) in 2005. After 2 years, he obtained his master degree from the University of Science and Technology USTHB Algiers in 2008, where he worked mainly on Wireless networks and especially on wireless mesh networks. Actually, he is a Ph.D student at the USTHB University. He is also a permanent Full-time Researcher at the Center of Research on Scientific and Technical Information (CERIST) in Algiers since 2009. He is a member of SensLab research group at CERIST. His research interest includes medium access control, wireless sensor networks, and QoS in wireless networks.



Dr. Djamel Djenouri He obtained the Ph.D in Computer Science from the University of Science and Technology USTHB Algiers, Algeria, in 2007. From 2008 to 2009, he was granted a post-doctoral fellowship from the European Research Consortium on Informatics and Mathematics (ERCIM), and he worked at the Norwegian University of Science and Technology (NTNU), in Trondheim, Norway, where he participated in the MELODY project supported by the Norwegian Research Council. Currently, Dr Djenouri is a permanent full-time senior researcher at the CERIST research center in Algiers, where he is leading the WSN group, and adjunct professor at the university of Blida. He is working on topics related to wireless and sensor networking, with focus on quality of service, power management and energy harvesting, routing protocols, MAC protocols, time synchronization, fault tolerance, sensor and actuator networks, vehicular applications, smart building and green applications, Internet of things. He has been conducting several

research projects with international collaborations on these topics. Dr Djamel Djenouri participated in many international conferences worldwide, and he has been granted short-term research visitor internships to many renown universities including SICS in Stockholm, university of Cape town, UPC Barcelona, and the university of Oxford. He published more than 70 papers in international peer-reviewed journals and conference proceedings, and two books. He is a professional member of the ACM, IEEE, and he chaired workshops held in conjunction with IEEE DCOSS and IEEE GlobeCom. He also served as TPC member of many international conferences, guest editor of a special issue with *Int. J. Communication Networks and Distributed Systems*, and reviewer for many international Journals.



Dr. Jose M. Barcelo-Ordinas

He is an Associate Professor at Universitat Politècnica de Catalunya (UPC). He received his Ph.D and M.Sc at UPC (1998 and 1991). He joined the CompNet research group at UPC in 1993. He has participated in several European projects such as EXPLOIT, BAF, EXPERT, NETPERF, MOEBIUS, WIDENS projects, and EuroNGI, EuroNFI and EuroNF (VII FP) Networks of Excellence (NoE). His currently

research areas are Delay Tolerant Networks, VANETs, Wireless Sensor Networks, mobile networks and citizen science.



Pr. Nadjib Badache He was with CISTTT (Centre d'Information Scientifique et Technique et de Transfert Technologique) now CERIST (Centre de Recherche sur l'Information Scientifique et Technique) at Algiers, from 1978 to 1983, working on applied research projects in information retrieval, operating systems and library management. In 1983, he joined USTHB University of Algiers, as Assistant Professor and then Professor, where he teaches op-

erating systems design, distributed systems and networking with research mainly in distributed algorithms and mobile systems. From 2000 to 2008, he was head of LSI (Laboratoire des Systèmes Informatiques of USTHB University) where he conducted many projects on routing protocols, energy efficiency and security in Mobile Ad-hoc Networks and Wireless Sensor Networks. Since March 2008, he is CEO of CERIST and Professor at USTHB University.