Wireless Medium Access Control under Mobility and Bursty Traffic Assumptions in WSNs

Georgios Papadopoulos, Vasileios Kotsiou, Antoine Gallais, Thomas Thomas Noël, Thomas Noël

Mobile Netw Appl
DOI 10.1007/s11036-015-0608-1

Wireless Medium Access Control under Mobility and Bursty Traffic Assumptions in WSNs

Georgios Z. Papadopoulos[1] · Vasileios Kotsiou[2] · Antoine Gallais[1] · Periklis Chatzimisios[2] · Thomas Noël[1]

Abstract In Wireless Sensor Networks (WSNs) the nodes can be either static or mobile depending on the requirements of each application. During the design of Medium Access Control (MAC) protocols, mobility may pose many communication challenges. These difficulties require first a link establishment between mobile and static nodes, and then an energy efficient and low delay burst handling mechanism. In this study, we investigate preamble-sampling solutions that allow asynchronous operation. We first introduce anycast transmission to ContikiMAC where a mobile node emits an anycast data packet whose first acknowledging node will serve as responsible to forward it towards the sink. Once this link is established, burst transmission can start, according to the respective burst handling mechanism of ContikiMAC. Although it is considered as negligible in the literature, such an anycast-based on-the-fly operation actually results in high packet duplication at the sink. Hence, we demonstrate that even a basic anycast-based M-ContikiMAC would fail to handle bursty traffic from mobile nodes mainly due to increased unnecessary traffic and channel occupancy. We then propose Mobility-Enhanced ContikiMAC (ME-ContikiMAC), a protocol that reduces packet duplications in the network by more than 90 % comparing to M-ContikiMAC. Moreover, our results in a 48-node scenario show that ME-ContikiMAC outperforms a number of state-of-the-art solutions (including MoX-MAC and MOBINET), by terms of reducing both delay and energy consumption.

Keywords Wireless Sensor Networks · Medium Access Control · Mobility · Neighbor discovery · Bursty traffic · Energy efficiency · ContikiMAC

✉ Georgios Z. Papadopoulos
gpapadopoulos@unistra.fr

Vasileios Kotsiou
vkotsiou@sch.gr

Antoine Gallais
gallais@unistra.fr

Periklis Chatzimisios
peris@it.teithe.gr

Thomas Noël
noel@unistra.fr

[1] ICube Laboratory, University of Strasbourg, Strasbourg, France

[2] Hellenic Open University and ATEITHE, Patras, Greece

1 Introduction

Over the previous years, a new era of Wireless Sensor Networks (WSNs) has emerged, with deployments ranging from monitoring of patients to animals. In such applications, requirements such as mobility and bursty traffic are appear to be extremely important. In contrary to the traditional a priori known time-driven traffic patterns, event-driven networks face occasional, bursty and unanticipated multi-hop data transmissions. In wildlife monitoring for instance, the nodes (usually with limited-memory devices) operate under limited internet access for the majority of the time. When a network connection is detected, a surge of traffic should be handled. More specifically, the mobile nodes should immediately upload their stored data (bursts) at a more powerful device before losing again the connection (i.e. sink) [1]. Such sudden dynamic and bursty traffic cause certain

Published online: 17 May 2015

anomalies in the network and fuel the research community to find appropriate solutions.

The Medium Access Control (MAC) layer is responsible for the communications among the nodes. In this study, we consider contention-based MAC protocols, mainly due to scalability (e.g. local decisions) and dynamic nature of the network (e.g. mobility). By employing these protocols, nodes in the network asynchronously sample the wireless medium for incoming packets at regular intervals. In between, they turn OFF their radio to reduce energy consumption (i.e. duty cycling). Many MAC protocols that have been proposed in the literature (i.e. [2]) deal with mobility to some extent. However, to the best of our knowledge very few of them address the needs implied by the presence of variable and bursty traffic in the network. As a result, no successful WSN deployment with mobility handling has been proposed so far.

In [3], we introduced the basic M-ContikiMAC mechanism, which is compliant with preamble-sampling MAC protocols. We illustrated this by embedding our proposed solutions as anycast-based extensions of ContikiMAC [4], which presents some anomalies in the network due to its anycast-based transmissions. In this article, we further discuss how to allow mobile nodes to co-exist and communicate with static nodes in the network. We also further detail our Mobility-Enhanced ContikiMAC (ME-ContikiMAC) protocol that handles mobility even under very dense WSN scenarios. This allows us to evaluate ME-ContikiMAC in a large-scale environment where we verify ME-ContikiMAC's efficiency in dense network scenarios. In addition to the original description of M-ContikiMAC and ME-ContikiMAC [3], the contributions presented in this paper are threefold:

- We present some optimizations of our packet duplication control mechanism in order to mitigate the duplications, which in turn reduces the channel occupancy, as well as energy consumption when compared to basic M-ContikiMAC. We especially emphasize the 1-hop delay related increased performances that can be achieved.
- We discuss the remaining limitations of M-ContikiMAC and propose to overcome them by introducing some optimizations of ME-ContikiMAC. Hence, we show to what extent it allows reduced energy consumption, along with channel occupancy and delay reduction.
- Finally, we enhance our initial performance evaluation in order to compare ME-ContikiMAC proposal against our previous work M-ContikiMAC as well as against other state-of-the-art solutions (such as MoX-MAC [5] and MOBINET [6]).

The remainder of our paper is organized as follows. We review some of the most pertinent related works in Section 2. We provide the necessary background information and we present the foundations of the problem that we intend to address in Section 3. In Section 4, we present detailed description of both M-ContikiMAC and its enhanced version ME-ContikiMAC. We implement ME-ContikiMAC on top of the Contiki OS [7] and present the performance evaluation of our protocol in Section 5. Finally, Section 6 provides the concluding remarks of our work and certain suggestions for future work.

2 Related work

During the previous decade, many MAC layer solutions were proposed to handle mobility in WSNs. These protocols can be classified into various categories such as synchronous, asynchronous and hybrid [2], [8]. As previously reported, in this paper we consider contention-based MAC protocols and in particular preamble-sampling solutions. Hereafter, we present the most recent and relevant approaches related to our investigation.

In [9], the authors present MA-MAC (Mobility-Aware Medium Access Control), an extension of the X-MAC [10] protocol. MA-MAC defines two thresholds to handle mobility in WSNs. The first threshold is triggered in order the mobile node to initiate a handover, while the second one sets an upper limit (i.e. distance) that a mobile node should move before it establishes a new temporary parent. The mobile node constantly evaluates the Received Signal Strength Indicator (RSSI) values of incoming ACK packets from static nodes. Thus, if the mobile node perceives that the distance between the current receiver and itself exceeds the first threshold, it initiates a neighborhood discovery procedure by transmitting broadcast data packets in which handover requests are embedded. MA-MAC introduces a new header in the payload part of the packet, depends on nodes scheduling and network density, and relying on the two reported thresholds is fairly critical.

In [11], Dargie et al. present the MX-MAC, which allows a mobile node to transmit in burst once it gains access to the wireless medium. The protocol utilizes a Least Mean Square (LMS) filter that continuously evaluates the RSSI values of received ACKs from its temporary parent, and thus, if it detects a persisting deterioration in the link quality it initiates a handover procedure. Moreover, MX-MAC introduces three different types of MAC addresses: multicast, neighbor discovery, and unicast address. The MX-MAC protocol is particularly suitable for environments where the mobile nodes are few and its efficiency strongly depends on the network density. Indeed, if the number of neighboring static nodes is small, then the probability of discovering a new static relay node reduces significantly, (similarly to MA-MAC).

Kuntz et al. [12] propose X-Machiavel, a X-MAC-based solution. X-Machiavel has been designed to reduce the delay of accessing the wireless medium for the mobile nodes both under high and low contention scenarios. The authors consider that the data packets that originate from mobile nodes have higher value compared with the static nodes. Thus, X-Machiavel provides the mobile nodes with the privilege to "steal" the wireless medium from a static node that has gained it earlier. More specifically, a mobile node overhears the medium to detect a preamble's strobe from a transmission between two static nodes. Once it receives a strobe packet, it transmits its own data packet back to the transmitter. As a results, static nodes are forced to postpone their own data transmissions, which eventually leads to increase both 1-hop and end-to-end delays. X-Machiavel, is thus considered as a "non fair" contention based protocol for the nodes in a WSN.

In [5], the authors present the MoX-MAC protocol. Under MoX-MAC the mobile nodes do not transmit strobe packets during the preamble period in contrary to static nodes. In fact, when a mobile node expects to transmit a data packet, it samples the medium hoping to detect an ACK packet transmission, that originally is sent to a static node. Thus, if a mobile node receives the ACK packet, it waits until the end of the scheduled transmission, and afterwards, it transmits its data packet to the transmitter static node. Otherwise, if no ACK packet is detected, it follows the default procedure of the X-MAC. The efficiency of this approach strongly depends on the communication frequency between the static nodes. Moreover, if no mobile node transmits data packet, the transmitter static nodes unnecessarily consume energy by keeping their radio ON to potentially receive data packets from mobile nodes.

MOBINET [6] allows the mobile nodes to detect the surrounding static nodes (if there are any) in a passive listening mode. A mobile node when enters a static network, it builds a neighborhood table with destination addresses of the static nodes of its transmission range, by overhear the medium for transmitted packets from its temporary neighbors. Later, when the mobile node desires to transmit, it sends a data packet in unicast to one of the destination addresses listed in its neighborhood table. MOBINET comes with two methods, the random and selective method respectively. In the first approach, the next-hop selection is randomly selected among the ones available in the neighborhood, while on the other method the mobile node transmits to the "best" sensor located in its neighborhood in terms of number of hops for instance.

Most of the previously presented protocols may not satisfy our objectives of addressing bursty traffic in mobile environments, in a highly proactive manner and by attaining low 1-hop and end-to-end delay values under mobile environments.

Hence, as summarized in Table 1, MA-MAC and MX-MAC approaches are highly reactive solutions. Since MX-MAC requires a significant number of packet transmissions in order to estimate quality of the link before its establishment, its application induces potential delays or losses in the network. Furthermore, these solutions strongly depend on the network density and they are designed for small-scale networks. On the other hand, X-Machiavel even though being a traffic independent protocol, it strongly depends on features of the X-MAC protocol, such as strobe packets in the preamble, and moreover, it suffers under scenarios where we have more mobile over static nodes.

Finally, MoX-MAC and MOBINET being proactive protocols, appear as the most relevant to our targeted context. Moreover, the previously mentioned solution are independent from the underlying MAC protocol. Indeed, they can be implemented both on top of strobe-based (e.g. X-MAC) and data-based (e.g. ContikiMAC) MAC protocols. We therefore have selected these contributions as best candidates for further comparison during our evaluation campaign.

Table 1 Summary of state-of-the-art preamble-sampling based MAC layer contributions addressing mobility in WSNs

MAC protocol	Advantages	Drawbacks
MA-MAC [9],	traffic independent	reactive protocol
MX-MAC [11]	efficient handover mechanism	network density dependency
		designed for very small networks
X-Machiavel [12]	traffic independent	underlying protocol dependency
	hybrid protocol (reactive and proactive)	proportion of mobile to static nodes dependency
	overhead minimization (preamble-less)	non-fair contention-based protocol
MoX-MAC [5]	proactive protocol	traffic dependent (passive protocol)
	overhead minimization (preamble-less)	unnecessarily consume energy (for static nodes)
MOBINET [6]	proactive protocol	traffic dependent (passive protocol)
	optimal next-hop selection	increase of idle listening (energy consumption)

3 Overview

In this Section, we first provide the necessary background on ContikiMAC protocol and its drawbacks. We then perform a high-level description of our proposed approach, ME-ContikiMAC.

3.1 ContikiMAC protocol

We have chosen to develop our mechanism over the ContikiMAC protocol but the mechanism is general enough to be used in any preamble-sampling oriented protocols. ContikiMAC, the default MAC layer protocol in Contiki OS, embeds most of innovative features of existing preamble-sampling protocols. In particular, periodic wake-ups have been suggested by X-MAC, the phase-lock optimization has been presented by WiseMAC [13] and the use of data packet copies as a wake-up strobe has been previously introduced by the BoX-MAX [14], the default low-power MAC protocol in TinyOS. Furthermore, ContikiMAC comes with a bursty transmission handling mechanism [15]. In [16], after a thorough performance evaluation, the results illustrate that ContikiMAC achieves a better delay performance and significantly lower energy consumption compared to X-MAC. Therefore, we consider ContikiMAC as the leading protocol in the preamble-sampling family of MAC protocols above which we demonstrate that our proposal can operate efficiently.

ContikiMAC originally provides two types of transmissions, the so-called unicast and broadcast. Under unicast mode, the sender repeatedly transmits its data packet that contains the payload and the destination address until it receives a link layer acknowledgment from the receiver. On the other side, the intended receiver periodically wakes-up to sample the medium for packet transmissions from its neighbors. Once a transmission is detected during a wake-up, the receiver keeps the radio ON to receive the packet that will follow. When a data packet transmission is successful, the receiver replies with an ACK packet. Under broadcast mode, the potential receivers do not acknowledge the received data packet. The sender actually repeatedly transmits the data packet during the entire preamble period to ensure that all its neighbors have received it. The concept of unicast and broadcast transmissions according to the ContikiMAC is illustrated in Fig. 1.

Furthermore, ContikiMAC provides a burst handling mechanism to anticipate high traffic periods in the network [15]. In particular, under burst mode a transmitter expects to transmit multiple packets in a row. To do so, the sender modifies the header for each data packet of the queue (except the last). In fact, it sets a flag that notifies the receiver that another packet follows. On the other side, the node receiving the flagged packet, appropriately adapts its radio duty cycle. Indeed, the receiver keeps its radio ON to receive the following packets. As a result, the receiver node switches into Carrier Sense Multiple Access (CSMA) mode until it receives a data packet (conventionally the last packet of the queue) that is not labeled with the burst flag notification. Finally, the transmitter first waits for the acknowledgement of the ongoing transmission before transmitting the next data packet (see Fig. 1).

3.2 Challenge

Even though the ContikiMAC protocol is well designed for static networks, it does not perform effectively under environments where static and mobile nodes co-exist. In fact, since the mobile nodes do not participate in the construction of the routing tree, they are not aware of the next-hop address and this actually prevents them from utilizing unicast transmissions. Considering the default unicast and broadcast functions of ContikiMAC, a mobile

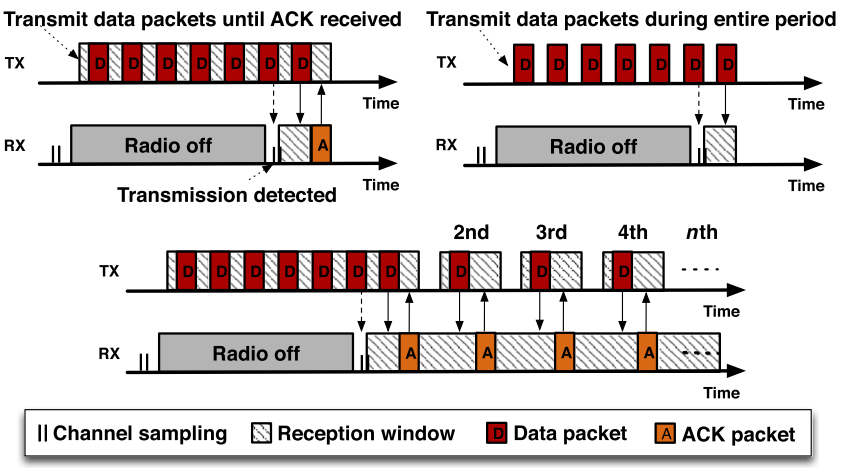


Fig. 1 Transmission modes of ContikiMAC protocol, namely Unicast, Broadcast and Burst respectively

node should transmit its packets by employing broadcast. Since broadcasting is a costly alternative, mobile nodes fail to access the medium to communicate with static nodes in an efficient manner (resulting in low energy consumption and delay performance). In the case of a burst transmission, a mobile node may either transmit all n packets in broadcast, or it transmits the first data packet in broadcast to discover a temporary parent and then switches to unicast mode to dequeue its buffer. As a result, the default transmission modes of ContikiMAC induce certain network deficiencies when it comes to mobile nodes.

3.3 ME-ContikiMAC in a nutshell

Since the mobile nodes do not utilize any routing scheme, there is a need for an efficient parent discovery mechanism. We here, introduce the Mobile-ContikiMAC mechanism (M-ContikiMAC), an extension of the ContikiMAC protocol, and its enhanced version ME-ContikiMAC. In this study, we depart from the unicast design paradigm. Instead, we propose an additional transmission mode, which allows to any given node that is located in the transmission range of a mobile transmitter to be its receiver, (acting as a temporary parent). To implement this approach, we introduce an anycast transmission, where a packet is transmitted opportunistically to the first potential forwarder acknowledging the corresponding packet. Hence, a mobile node chooses the next-hop, static node that wakes-up the soonest. Note that in anycast mode, the potential receivers are all identified by the same destination address.

4 Design of M-ContikiMAC & ME-ContikiMAC

This Section presents the core mechanisms of our proposed protocols. More specifically, we provide a detailed description about how M-ContikiMAC interacts with the underlying base MAC protocol, and the way that the mobile nodes co-exist with the static nodes in the network by utilizing M-ContikiMAC. We then propose two enhancements over the basic M-ContikiMAC: packet duplication control and delay optimization, and we further explain how these extensions improve network performance by also combining the minimization of energy consumption.

4.1 Basic M-ContikiMAC

In this study, we consider scenarios where the mobile nodes tend to transmit packet bursts (the burst notification flag is activated) under dense network scenarios. Let us assume a mobile node (TX), that expects to send n packets in a burst. As previously stated, TX is not aware about the next-hop as well as about the surrounding nodes within its transmission range. Thus, TX sets an anycast destination address for its first data packet of the queue that allows its temporary neighbors to receive it. Furthermore, on the first data packet we set an additional one byte flag (besides the burst flag), called ReqHop (Request for a next-Hop) that gets the value zero when the transmitter searches for a next-hop. Hence, TX repeatedly transmits its first data packet in anycast until it receives a link layer acknowledgment from a potential forwarder (similar to the original version of the ContikiMAC) which in turn will be its new next-hop. Later, TX sends the remaining $n-1$ packets on its queue in unicast to its temporary parent, while ReqHop is switched to one. Note that according to the current M-ContikiMAC parent discovery configuration, only static nodes are allowed to respond to anycast transmissions from a mobile node. The detailed process of parent discovery is illustrated in Fig. 2.

On the other side, a static node (for instance RX1) that wakes-up to sample the medium for an upcoming packet performs the following procedure: *i)* it checks if the destination address of the data packet contains its own address *ii)* if not, it then checks whether is a broadcast address *iii)* finally, if neither of these two transmission types correspond to the destination address it then checks if it is an anycast address. If so, RX1 checks the value of ReqHop whether it is *zero* that practically allows RX1 to accept the packet (otherwise it rejects it). Once RX1 receives the first data packet, it responds with an ACK including its own address, while keeping the radio ON to receive the remaining packets, as originally ContikiMAC was designed. Later, once TX receives the ACK, it sets ReqHop to one and transmits the rest of the packets to its new temporary parent. In case, another static node (e.g. RX2) wakes-up during the burst transmission, it will realize that the packets are not intended to itself. Since the destination address of the packet is neither anycast nor its own, and moreover ReqHop is equal to 1, thus, RX2 will turn its radio OFF after the sampling procedure. The detailed function of M-ContikiMAC is presented in Algorithm 1.

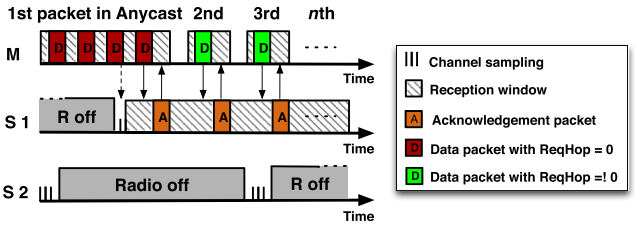


Fig. 2 M-ContikiMAC in basic mode where the transmitter transmits first in anycast following by unicast

Algorithm 1: Functionality of M-ContikiMAC

```
begin
    when Receiver (R) receives a packet (P) then
        if P.address is R.address (Unicast) then
            Accept the packet;
            Sent ACK;
            if Burst flag ON then
                Keep radio ON for the next packet;
            else
                Turn radio OFF
            end
        else if P.address is Broadcast then
            Accept the packet; Turn radio OFF
        else if P.address is Anycast && ReqHop == ∅ then
            Accept the packet;
            Sent ACK including Receiver's address;
            if Burst flag ON then
                Keep radio ON for the next packet;
            else
                Turn radio OFF
            end
        else if P.address is Anycast && ReqHop == 1 then
            Accept the packet;
            Sent ACK;
            if Burst flag ON then
                Keep radio ON for the next packet;
            else
                Turn radio OFF
            end
        else
            Reject the packet;
        end
    end
end
```

4.2 Reconnection procedure of M-ContikiMAC

When mobile nodes are present in WSNs, link disconnection between a mobile and a static node is a very frequent phenomenon, mainly due to bad link quality or mobility (mobile node moves away from the range of its temporary parent). During the burst transmission period, if a mobile node does not receive the expected ACK for its ongoing transmission, it assumes that it is disconnected from its temporary parent. In order to anticipate this situation, it enables a reconnection mechanism to discover a new forwarder. Indeed, it sets the ReqHop flag back to zero and retransmits the same data packet in anycast mode to find a new parent to continue transmitting the remaining packets of the queue as shown in Fig. 3.

4.3 Drawbacks of M-ContikiMAC

The previously presented mechanism induces certain anomalies due to the nature of anycast transmission mode. In fact, if two or more nodes simultaneously sample the medium, they may receive the same transmitted packet, while their ACKs may collide. Indeed, the probability of two or more nodes simultaneously sampling the medium for incoming packet is strongly correlated to the total number of the forwarders. Thus, the probability of having duplicated packets at the sink node increases in dense networks, which in turn induces unnecessary traffic in the network (i.e. including both originally transmitted and forwarded packets). Hence, the traffic in the network, as well as the congestion, the channel occupancy and the competition for medium access increase, which in turn enlarges the probability of packet retransmissions due to potential collisions. As a result, network performance significantly degrades while energy consumption for the whole network attains higher values as shown in [17].

Let us assume that TX transmits repeatedly the first of the total n packets of its queue by employing the anycast mode with ReqHop equal to zero. Thus, any static node in transmission range of TX is privileged to receive it. Thus, two (e.g. RX1 and RX2) or more static nodes having the same or almost the same sampling frequency phase may wake up and sample the medium simultaneously, as a result they will receive and consequently will acknowledge the data packet. Note that, there is high probability that the ACK packets may collide. Thus, according to basic M-ContikiMAC, if TX does not receive its expected ACK, it will postpone the burst transmission for the next preamble cycle and will retransmit in anycast the collided data packet (see Fig. 3). However, since RX1 and RX2 are not aware about the collision of their acknowledgements, they will forward the previously received packet further to the sink

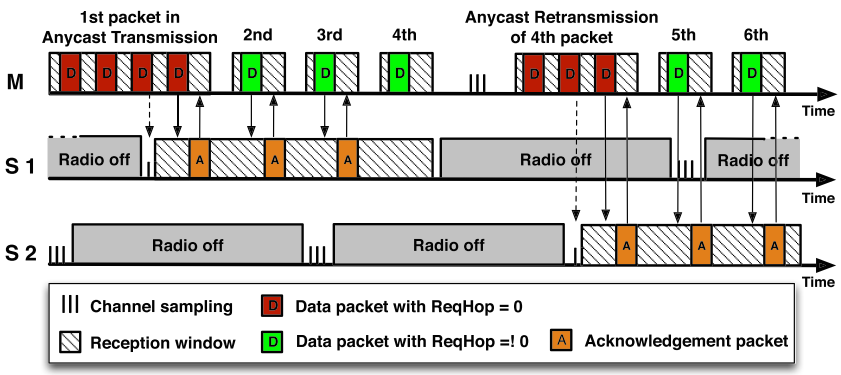
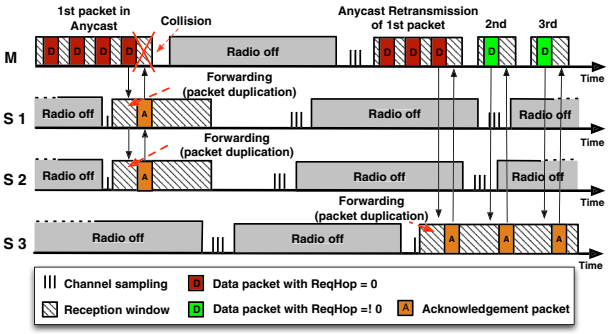
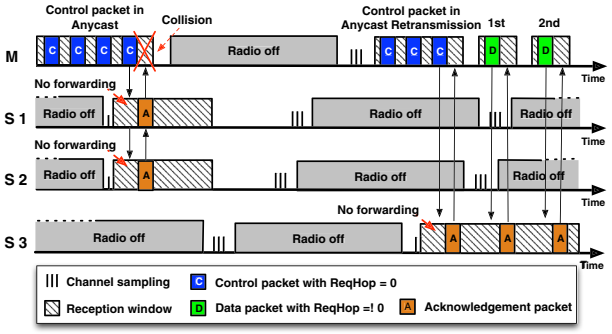


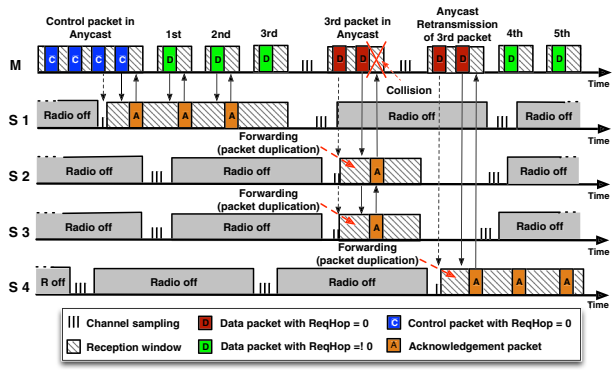
Fig. 3 Connection recovering with M-ContikiMAC



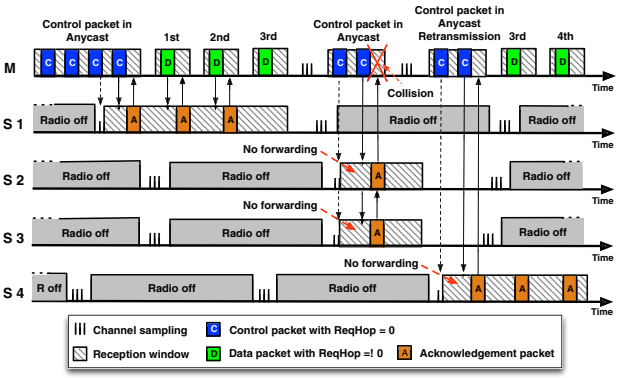
(a) When two or more nodes have the same wake-up slot, then a packet may be received by two or more nodes, that will in turn forward it. This situation leads to packet duplication at the sink.



(b) Packet duplication control mechanism of ME-ContikiMAC: By introducing control packets before the burst transmission we achieve the reduction of the duplications in the network.



(c) To recover a connection by transmitting a data packet, we may end up with duplications if two or more nodes will sample the medium simultaneously.



(d) ME-ContikiMAC in enhanced connection recovering depiction: Upon a link disconnection the mobile node transmits a control packet upfront to avoid duplications.

Fig. 4 Scenarios where packet duplication issues arise (*left*) and ME-ContikiMAC with packet duplication control mechanisms (*right*)

and will keep their radio turned ON, since they consider that more packets will arrive due to the burst flag (Fig. 4a). As a result, this situation generates duplicate packets in the network that lead to collisions as well as high interference energy consumption values.

4.4 Toward ME-ContikiMAC

Hereafter, we present ME-ContikiMAC, the enhanced version of M-ContikiMAC, to handle mobility under dense networks by avoiding duplicates and resulting in a significant delay degradation.

4.4.1 Packet duplication control mechanisms

TX expecting to transmit n data packets in burst, will transmit one additional control packet upfront, $n + 1$ packets. In particular, it will repeatedly send a control packet in anycast which will be labeled not to be forwarded, while ReqHop equals to zero. If there will be an ACK collision for the control packet, the sender will retransmit it, while the receivers will not forward it further to the sink. Once the transmitter receives the corresponding ACK for the control packet, it will initiate the burst procedure (with data packets) to its newly temporary parent, as it is depicted in the Fig. 4b). As a result, according to our simulation evaluation (see later in Section 5), we significantly reduce the unnecessary traffic in the network.

More specifically, two static nodes, RX1 and RX2, that sample the medium simultaneously, both will receive the control packet (recall that RX1 and RX2 will not forward the received control packet) and consequently will respond with an ACK which eventually will collide. Hence, TX will not be acknowledged for its control packet, thus, it will postpone the burst transmission of n data packets to the following preamble cycle. Once TX wakes-up, it will proceed again to the parent discovery mechanism by retransmitting the control packet until it receives an ACK from a static node (e.g. RX3). Afterwards, TX will initiate the burst process to the new discovered parent (Fig. 4b).

We now further optimize the packet duplication control mechanism. Indeed, we observed that when a mobile

node intends to establish a new connection, after a link disconnection, by utilizing the previously presented recovering mechanism of M-ContikiMAC (Fig. 3), multiple packet reception issue arises again (see Figure 4c). To overcome this phenomenon, we configure the TX during next-hop discovering procedure, instead of transmitting (by employing anycast) the data packet with ReqHop set to *zero*, to retransmit upfront a control packet (with ReqHop equals to 0) which will be labeled not to be forwarded. Thus, RX1 or RX2 will forward a data packet only when a link with TX is established. As a result, network performance is improved, congestion and collisions in the network are avoided while better energy efficiency is achieved. The detailed procedure of network reconnection of ME-ContikiMAC is illustrated in Fig. 4d).

4.4.2 Delay enhancement

As can be observed from Figs. 4a and b, according to the original version of ContikiMAC once the transmitted packet collides, the transmitter stops the ongoing preamble cycle and cancels the packet transmission. TX then retransmits either data or control packet in the following preamble cycle. Thus, TX waits for a complete preamble period (e.g. 125 ms, 500 ms) before retransmitting the packet. Apparently, this default operation of ContikiMAC induces high delays, especially when the nodes in the network are configured in long preamble-sampling frequencies such as 500 ms or 1 s. In order to overcome this barrier, we consider that the mobile nodes should have the priority to access the medium and, thus, we introduce aggressive nodes in the network. In particular, we configure the mobile nodes to behave more actively compared to the static, in order to receive privilege over the static nodes and gain the wireless medium. In order to do so, we appropriately modify the ME-ContikiMAC protocol to allow the mobile nodes to continue transmitting their packets during the preamble cycle even if their ACKs collide (see Fig. 5). By doing so, we achieve to significantly reduce the attained delay values.

5 Performance evaluation

In the previous Section, we have presented the design of both M-ContikiMAC and ME-ContikiMAC, and discussed to which extent they may improve network performance when compared to major contributions in the related literature. In order to evaluate the efficiency of ME-ContikiMAC, we have run a set of simulations over COOJA [18] with Sky motes. Moreover, we utilized BonnMotion [19], a tool to generate mobility in the network. For comparison purposes, we also implemented and compared ME-ContikiMAC both against our previous work M-ContikiMAC and state-of-the-art protocols such as MoX-MAC and MOBINET, both selective (i.e. MOBINET-S) and random (i.e. MOBINET-R) mode. Note that we deactivate the phase-lock optimization function from the default configuration of ContikiMAC, in order to provide fair analysis and thorough comparative study.

Our simulation scenario involves 40 fixed nodes (including the sink) that are uniformly (i.e. grid) or randomly distributed in an area of $50 \times 40\, m$, with network degree 13.6 in average, similarly to dense wireless lighting control networks [20]. Moreover, there are 8 mobile nodes that move by employing a random waypoint mobility model, with two different velocities. More specifically, the low speed (i.e. from 0.5 m/s *to* 2 m/s) that represents a human walk, and high speed (i.e. from 2 m/s *to* 8 m/s) that represents a typical jogging speed. In this study, we present application-dependent (i.e. time-driven) results where the mobile nodes transmit bursts of 16 packets every 90 sec while the static nodes transmit by utilizing a Constant Bit Rate (CBR) of 1 pkt per 30 sec, having as a result more than 8000 $pkts$ transmissions in total. As far it concerns the MAC layer, we

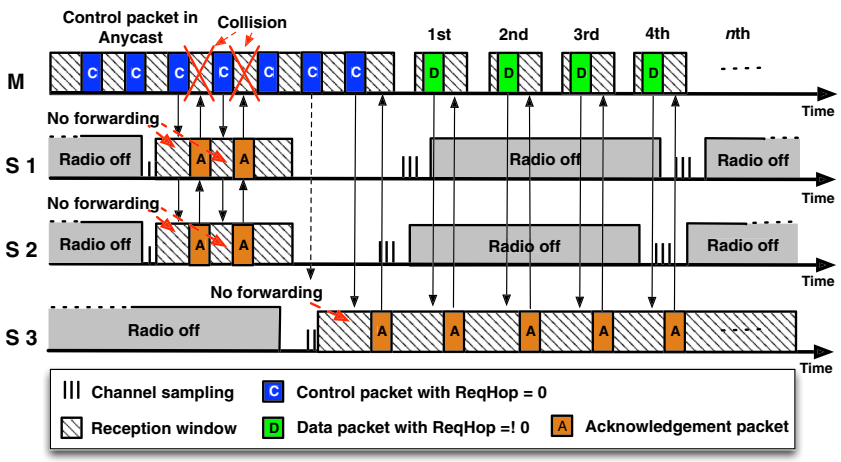


Fig. 5 ME-ContikiMAC in delay-enhanced illustration: Mobile nodes in agressive state

have set a maximum of three retransmissions and the sampling frequency to 125 ms. We choose the packet size to be equal to 33 $bytes$ that corresponds to all necessary information for MAC, routing and application operations (e.g. node ID, packet sequence, burst and ReqHop flags, sensed values). Furthermore, we used a radio model based on disks for the sake of clarity, where each node emits at $-10\ dBm$ transmission power, imposing thus, multi-hop communications among the mobile nodes and the sink (up to five hops). At the routing layer, we rely on a broadly used scalable and under realistic conditions gradient protocol [21] that generates a routing tree rooted at the sink (i.e. by employing as a metric a number of hops towards to the sink) having low overhead. Finally, each simulation lasted 54 min. The details of the simulation setup are exposed in Table 2. The results hereinafter show the performance gain of our proposal in terms of delay (i.e. both 1-hop and end-to-end) and energy consumption. In fact, we demonstrate that two different MAC configurations (i.e. statically oriented and mobile oriented) can cooperate with each other, so that the mobile nodes can smoothly coexist within a static network, without causing inefficiencies in the network.

Table 2 Simulation setup

Topology parameters	Value
Topology	Regular grid & random (50×40)
Number of nodes	40 fixed & 8 mobile sensors
Number of sources	47
Node spacing	$x = 6\ m$ / $y = 8\ m$
Network degree	13.6
Mobility parameters	**Value**
Mobility model	Random waypoint
Velocity	Low speed: from 0.5 m/s *to* 2 m/s
	High speed: from 2 m/s *to* 8 m/s
Simulation parameters	**Value**
Duration	54 $minutes$
Application model	Mobile nodes: Burst: 16 $pkts/90\ s$
	Static nodes: CBR: 1 $pkt/30\ s$
Number of events	Mobile nodes: 4096 $pkts$
	Static nodes: 3990 $pkts$
Payload size	33 $Bytes$
Routing model	Static network: Gradient [21]
	Mobile nodes: Opportunistic
Number of hops	Multihop (5 hops maximum)
MAC model	Mobile nodes: MoX-MAC, MOBINET-S, MOBINET-R, M-ContikiMAC & ME-ContikiMAC
	Static nodes: ContikiMAC
Sampling frequency	125 ms
Maximum retries	3
Hardware parameters	**Value**
Antenna model	Omnidirectional CC2420
Radio propagation	2.4 GHz
Modulation model	O-QPSK
Transmission power	$-10\ dBm$

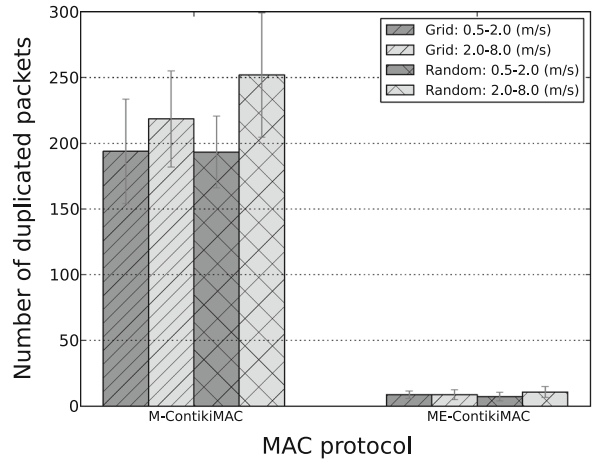
5.1 Packet duplication

Figure 6a illustrates the total number of packet duplications at the sink node both for M-ContikiMAC and ME-ContikiMAC. As can be observed, uncontrolled anycast transmissions may cause multiple packet receptions in the network. Indeed, the more dense is the network higher the probability of having packet duplications. As a result, network traffic congestion, channel occupancy and medium access competition increase, and, the probability of packet retransmission gets higher values due to the potential collisions in the network. The proposed ME-ContikiMAC succeeds in significantly reducing the multiple reception of a single packet at the sink node by employing the proposed packet duplication control mechanisms. More specifically, we succeed to reduce duplications by more than 90 % comparing to our primary work, M-ContikiMAC. Consequently, ME-ContikiMAC significantly decreases the number of unnecessary packets in the network and, thus, potential collisions.

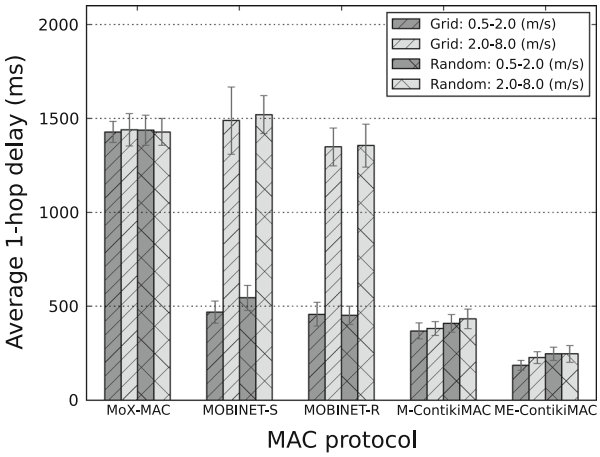
5.2 Delay performance

Figure 6b and c illustrate the average 1-hop (from mobile to any static node) and end-to-end (from mobile to sink node) delay per packet transmission. Both 1-hop and end-to-end delay include the channel sampling period, initial back-off, potential congestion back-off, potential retransmission delay and the transmission time of the preamble. Overall, the protocols within high velocity scenarios perform worse than in the low ones, mainly due to the difficulties of a link establishment between the mobile and static node (i.e. more frequent connections/disconnections). Furthermore, in the end-to-end delay, all protocols perform worse in random topologies. This phenomenon takes place due to the potential bottleneck links that are more prone to appear in random topologies, having as a result nodes to handle heavy traffic.

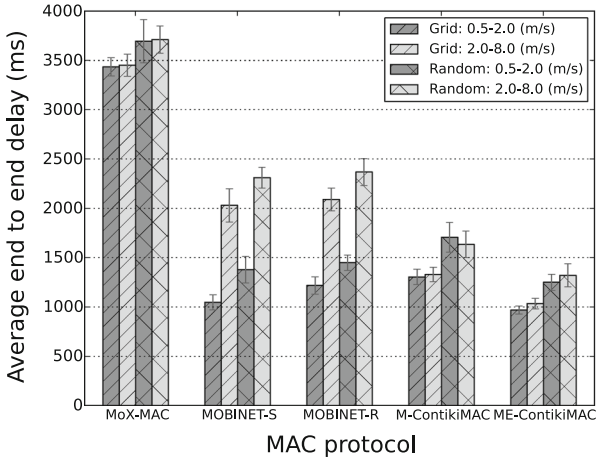
Furthermore, our simulation results show that both in 1-hop and end-to-end delay, MoX-MAC attains the worst performance. This could be explained by the phenomenon that in MoX-MAC a mobile node first overhears the whole transmission between the static nodes, and later it transmits its data packet to the detected sender. On the other side, MOBINET shows promising results in end-to-end



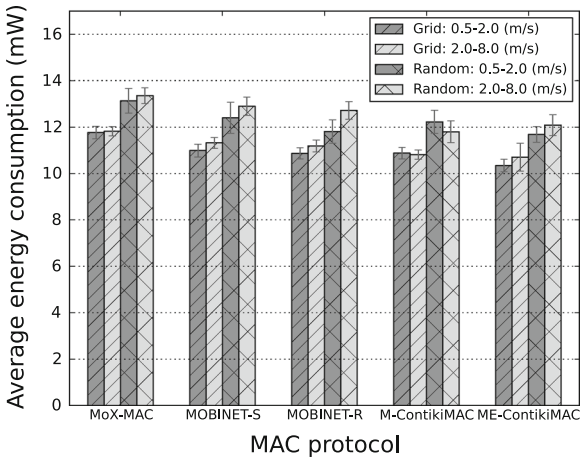
(a) Total number of packet duplications at the sink.



(b) Average one hop delay (from mobile to static node).



(c) Average end to end delay (from mobile to sink).



(d) Average energy consumption.

Fig. 6 A thorough performance evaluation of M-ContikiMAC and ME-ContikiMAC in terms of packet duplication, energy consumption and delay, (both in regular grid and random topologies)

delay for low speed scenarios, and more specifically, for the selective version of MOBINET, since it selects the best parent among the list of the potential next hop nodes (see Table 3). As a result, it picks up the node closest to the sink in terms of hop. However, MOBINET presents poor performance in high speed scenarios, it may due to the insufficient time for a mobile node to overhear the transmissions from neighborhood nodes, before to transmit its data packets.

Finally, ME-ContikiMAC significantly improves both 1-hop and end-to-end delay for all the considered scenarios. Indeed, it reduces up to 60 % the performance in high speed scenarios. These results are mainly due to the delay enhancement that we have presented in the previous section by introducing aggression to the mobile nodes. In addition, the reduction of the unnecessary transmissions in the network decreases potential collisions, and consequently retransmissions that have a major impact on the

Table 3 Average (along with confidence interval) number of hops, from mobile to sink

Scenario	MoX-MAC	MOBINET-S	MOBINET-R	M-ContikiMAC	ME-ContikiMAC
Grid: $0.5-2.0$ (m/s)	3.34 (0.04)	2.53 (0.05)	2.99 (0.04)	3.46 (0.06)	3.26 (0.07)
Grid: $2.0-8.0$ (m/s)	3.31 (0.05)	2.47 (0.05)	2.95 (0.06)	3.48 (0.09)	3.28 (0.04)
Random: $0.5-2.0$ (m/s)	3.64 (0.09)	3.09 (0.13)	3.45 (0.09)	4.02 (0.29)	3.70 (0.22)
Random: $2.0-8.0$ (m/s)	3.65 (0.09)	2.95 (0.15)	3.45 (0.09)	3.97 (0.22)	3.71 (0.10)

delay performance. Furthermore, as can be observed from the Table 3, all solutions have more or less the same amount of hops (the lower being for MOBINET because of its next-hop handling at MAC layer), thus meaning that the end-to-end delay reduction is independent to this metric. As a result, with ME-ContikiMAC we achieve a significant improved communication in mobile WSNs.

5.3 Energy consumption

Energy consumption: All energy consumption results were retrieved by utilizing the Energest module of Contiki [22]. This energy estimation module maintains a table with entries for all components, the Central Processing Unit (CPU), and the radio transceiver. Each table entry contains the total time that the corresponding component has been turned on, more specifically, it monitors in real-time the radio and CPU usage by saving the duration spent in each state (e.g. transmitting, receiving data, awaken, sleeping). This information is then combined with the energy values that are detailed in the component datasheet for each state in order to provide an accurate calculation of energy consumption per node. In Fig. 6d the average energy consumption per second for the whole network is presented for both grid and random topologies. The results show that the overhearing procedure has a straightforward impact on energy dissipation. As can observed, ME-ContikiMAC consumes less energy (i.e. 1 mW in average) network-wide when compared to MoX-MAC, and both selective and random-based MOBINET protocols.

6 Conclusion and future work

In this paper, we have introduced ME-ContikiMAC, an enhanced version of M-ContikiMAC protocol from our previous work, for tackling mobility issues and provide reliable, low delay and energy efficient communication between mobile and static nodes for WSNs. Our investigation demonstrated that two different configurations (i.e. statical oriented and mobile oriented) of the same MAC protocol can be combined, so that the mobile nodes can smoothly co-exist within a static network, without causing performance degradation for the static nodes that reside in the network. Note that, the proposed mechanism in this study can be applied to various preamble-sampling protocols (e.g. X-MAC). We performed a thorough simulation performance evaluation over two topologies (uniform and random nodes distribution) on top of COOJA simulator that demonstrates promising results. In fact, according to our results ME-ContikiMAC significantly enhances the overall network performance by reducing packet duplications (up to 90 %), channel occupancy and delay while keeping at low level the energy consumption, when compared to M-ContikiMAC and other state-of-the-art protocols.

Our ongoing and future work consists of further investigating this lead in mobile sensors. More specifically, we will continue our study of the energy consumption and handover delay of mobile sensor nodes and will try to reduce it with optimized algorithms. Moreover, our vision is to further explore ME-ContikiMAC by performing a set of experimental studies over FIT IoT-LAB,[1] a very large scale WSN testbed [23]. Thus, we plan to evaluate our mechanism under real-world scenarios and improve it by learning from the challenges that may arise from the experimental procedure. In the long term, we would like to also investigate solutions that detect arrivals of mobile nodes in a new wireless sensor network, in order to anticipate the interaction of mobile and static sensor nodes is a promising approach that will allow real-world deployments, such as advanced surveillance systems [24].

References

1. Zhang P, Sadler CM (2004). In: Lyon SA, Martonosi M (eds) Hardware design experiences in ZebraNet. In: Proceedings of the 2nd ACM conference on embedded networked sensor systems (Sensys), pp 227–238
2. Dong Q, Dargie W (2013) A survey on mobility and mobility-aware MAC protocols in wireless sensor networks. IEEE Commun Surveys Tutorials 15(1):88–100
3. Papadopoulos GZ, Gallais A, Noel T, Kotsiou V, Chatzimisios P (2014) Enhancing ContikiMAC for Bursty Traffic in Mobile Sensor Networks. In: Proceedings of the IEEE sensors
4. Dunkels A (2011) The ContikiMAC radio duty cycling protocol. *Swedish Institute of Computer Science, Technical Report*
5. Ba PD, Niang I, Gueye B (2014) An optimized and power savings protocol for mobility energy-aware in wireless sensor networks. Telecommun Syst 55(2):271–280
6. Roth D, Montavont J, Noel T (2011) MOBINET: mobility management across different wireless sensor networks. In: IEEE wireless communications and networking conference (WCNC)
7. Dunkels A., Gronvall B, Voigt T (2004) Contiki - a Lightweight and Flexible Operating System for Tiny Networked Sensors. In: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN), pp 455–462
8. Loo J, Mauri JL, Ortiz JH (2011) Mobile Ad hoc networks: current status and future trends. CRC Press, Taylor and Francis
9. Zhiyong T, Dargie W (2010) A mobility-aware medium access control protocol for wireless sensor networks. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM) Workshops, pp 109–114
10. Buettner M, Yee GV, Anderson E, Han R (2006) X-MAC: a short preamble MAC protocol for duty-cycled wireless sensor networks. In: Proceedings of the 4th ACM conference on embedded networked sensor systems (Sensys), pp 307–320
11. Dargie W, Wen J (2014) A seamless handover for WSN Using LMS filter. In Proceedings of the 39th IEEE conference on

[1] FIT IoT-LAB - https://www.iot-lab.info/

local computer networks (LCN), pp. 287–288. [Online]. Available: http://www.rn.inf.tu-dresden.de/uploads/Publikationen/LCN2014V2.pdf
12. Kuntz R, Montavont J, Noël T (2013) Improving the medium access in highly mobile wireless sensor networks. Telecommun Syst 52(4):2437–2458
13. El-Hoiydi A, Decotignie J-D, Enz C, Le Roux E (2003) WiseMAC, an ultra low power MAC protocol for the wiseNET wireless sensor networks. In: Proceedings of the 1st ACM conference on embedded networked sensor systems (Sensys), pp 302–303
14. Moss D, Levis P (2008) BoX-MACs: exploiting physical and link layer boundaries in low-power networking, Technical Report SING-08-00, Stanford University
15. Duquennoy S, Österlind F, Dunkels A (2011) Lossy links, low power, high throughput. In: Proceedings of the 9th ACM conference on embedded networked sensor systems (Sensys), pp 12–25
16. Michel M (2014) ContikiMAC vs X-MAC performance analysis. University of Mons, Technical Report
17. Papadopoulos GZ, Beaudaux J, Gallais A, Chatzimisios P, Noel T (2014) Toward a packet duplication control for opportunistic routing in WSNs. In: Proceedings of the IEEE global communications conference (GLOBECOM), pp. 94–99
18. Osterlind F, Dunkels A, Eriksson J, Finne N, Voigt T (2006) Cross-level sensor network simulation with COOJA. In: Proceedings of the 31st Annual IEEE International Conference on Local Computer Networks (LCN)
19. Aschenbruck N, Ernst R, Gerhards-Padilla E, Schwamborn M (2010) BonnMotion: a mobility scenario generation and analysis tool. In: Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques (SIMUTools)
20. Dandelski C, Wenning B-L, Perez D, Pesch D, Linnartz J-P (2015) Scalability of dense wireless lighting control networks. IEEE Commun Mag 53(1):157–165
21. Watteyne T, Kris Pister DB, Dohler M, Auge-Blum I (2009) Implementation of gradient routing in wireless sensor networks. In: Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM), pp 1–6
22. Dunkels A, Osterlind F, Tsiftes N, He Z (2007) Software-based on-line energy estimation for sensor nodes. In: Proceedings of the 4th ACN Workshop on Embedded Networked Sensors (EmNets), pp 28–32
23. Papadopoulos GZ, Beaudaux J, Gallais A, Noel T, Schreiner G (2013) Adding value to WSN simulation using the IoT-LAB experimental platform. In: Proceedings of the 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp 485–490
24. de Freitas EP, Heimfarth T, Vinel A, Wagner FR, Pereira CE, Larsson T (2013) Cooperation among wirelessly connected static and mobile sensor nodes for surveillance applications. Sensors 13(10):12903–12928