

Efficient Cluster Mobility Support for TDMA-Based MAC Protocols in Wireless Sensor Networks

MAJID NABI and MARC GEILEN, Eindhoven University of Technology

TWAN BASTEN and MILOS BLAGOJEVIC, Eindhoven University of Technology and TNO-ESI

Node mobility is a key feature of using Wireless Sensor Networks (WSNs) in many sensory applications, such as healthcare. The Medium Access Control (MAC) protocol should properly support the mobility in the network. In particular, mobility is complicated for contention-free protocols like Time Division Multiple Access (TDMA). An efficient access to the shared medium is scheduled based on the node's local neighborhood. This neighborhood may vary over time due to node movement or other dynamics. In scenarios including body-area networking, for instance, some clusters of nodes move together, creating further challenges but also opportunities. This article presents a MAC protocol, MCMAC, that provides efficient support for cluster mobility in TDMA-based MAC protocols in WSNs. The proposed protocol exploits a hybrid contention-free and contention-based communication approach to support cluster mobility. This relieves the protocol from rescheduling demand due to frequent node movements. Moreover, we propose a listening scheduling mechanism to avoid idle listening to mobile nodes that leads to a considerable energy saving for sensor nodes. The protocol is validated by performing several experiments in a real-world large-scale deployment including several mobile clusters. The protocol is also evaluated by extensive simulation of networks with various scales and configurations.

Categories and Subject Descriptors: C.2.1 [Computer-Communication Networks]: Network Architecture and Design

General Terms: Design, Algorithms, Performance

Additional Key Words and Phrases: Wireless sensor networks, medium access control, cluster mobility, idle listening, energy efficiency, quality-of-service, TDMA

ACM Reference Format:

Majid Nabi, Marc Geilen, Twan Basten, and Milos Blagojevic. 2014. Efficient cluster mobility support for TDMA-based MAC protocols in wireless sensor networks. *ACM Trans. Sensor Netw.* 10, 4, Article 65 (June 2014), 32 pages.

DOI: <http://dx.doi.org/10.1145/2594793>

1. INTRODUCTION

Wireless sensor networks are going to be used as a key solution for many applications. Healthcare, traffic management, wild life monitoring, disaster management, and precision agriculture are some fields in which WSNs are useful. Wireless technology makes the sensor nodes easy to deploy and gives them the opportunity to be mobile. The ability of sensor nodes to move provides additional interesting applications for WSNs.

A preliminary version of this work was published in *Proceedings of the IEEE Communication Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON10)*.

This work was supported by the Dutch innovation program Point-One, through project ALwEN, grant PNE07007.

Authors' addresses: M. Nabi (corresponding author), M. Geilen, Electrical Engineering Department, Eindhoven University of Technology, Eindhoven, the Netherlands; email: m.nabi@tue.nl; T. Basten and M. Blagojevic, Electrical Engineering Department, Eindhoven University of Technology, Eindhoven, the Netherlands and TNO-ESI, Embedded Systems Innovation by TNO, Eindhoven, the Netherlands.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

2014 Copyright held by the Owner/Author. Publication rights licensed to ACM. 1550-4859/2014/06-ART65 \$15.00

DOI: <http://dx.doi.org/10.1145/2594793>

Network protocols should sufficiently support expected node mobility in the network. The MAC layer is primarily responsible for managing mobility as it should control the access to the shared medium. In schedule-based contention-free protocols in particular, accessing the shared medium is scheduled taking the neighborhood into account. Time slots that are dedicated to the nodes should be unique in their neighborhood; this provides an efficient use of the channel bandwidth by avoiding collisions. Because of this, these mechanisms are promising for communication in WSNs. However, with node movement, the neighborhood may change, so the slots need to be dynamically rescheduled. This process is time and energy consuming. Rescheduling mechanisms are well suited for network startup and for joining nodes. Static networks with very limited mobility (node relocation) can also be supported. However, applying such processes is really challenging for networks with high node mobility.

The mobility is even more challenging if several nodes in the network move together (cluster mobility). There are many applications in which a static sensor network is in place together with several groups of mobile nodes. Besides a sensing task, the static network then plays the role of a network backbone for data dissemination. Health monitoring, including wireless body area networks (WBANs), is a prominent example. In such scenarios, a WBAN consists of a group of nodes that move together with the human wearing them. To achieve a good performance, supporting these highly mobile clusters is of great importance.

Nodes in WSNs have generally very limited resources. Optimizing energy consumption is critical for such networks to provide a reasonable lifetime for the nodes. Low duty-cycling is an effective approach to reduce energy consumption by keeping the radio transceivers off and only turning them on periodically in specific time durations to perform radio activities. However, even in the short active durations, unnecessary radio activity should be avoided, because it wastes the energy resources. *Idle listening* to the wireless channel, in particular, is a source of major energy wastage and MAC protocols try to minimize it. Node mobility again plays a role here. When a node expects packets from a specific neighbor, it listens to the channel in the dedicated time slot. However, if the sender has moved to another neighborhood, the listening node does not receive anything. Thus mechanisms to avoid idle listening due to node movement will be of great benefit.

This article introduces a mechanism to support cluster mobility in networks with a static backbone using a contention-free MAC protocol. It exploits a contention-based medium access scheme to be used by mobile cluster nodes within the existing low-duty-cycle TDMA protocol. The mechanism is designed to keep the duty cycle of all nodes as short as possible and to be integrated with our listening scheduling mechanism, which is proposed to reduce idle listening to mobile clusters. The basic idea of using Carrier-Sense Multiple-Access (CSMA) within a TDMA schedule for this purpose was previously presented in Nabi et al. [2010] in which we used simulation to evaluate the protocol. The contributions of this article in addition to Nabi et al. [2010] are as follows. (1) A slotted ALOHA scheme is presented as an alternative contention-based medium access scheme to be used for mobile cluster nodes. Many wireless radio platforms either do not support a CSMA implementation or it is not efficient. (2) The behavior of both contention-based paradigms is analyzed and guidelines for choosing a contention resolution mechanism and protocol parameter settings are provided based on the hardware setup, mobile cluster density, and application requirements. (3) The idea is generalized to support crowded networks in which many mobile clusters exist in the network. The scheme of Nabi et al. [2010] may suffer many collisions in crowded networks. (4) An empirical exploration is done to study the trade-offs due to different parameters of the protocol. This is then used to select near-optimal configurations of the protocol. (5) The full protocol is implemented on real wireless nodes and several

large-scale experiments have been performed to validate the protocol and investigate its behavior in real-world networks. A static network of 60 nodes was deployed on two floors of an office building. Four volunteers were participating, each carrying four mobile nodes, while performing their daily mobility behavior. This makes a realistic network setup mimicking a health monitoring application to evaluate the performance of the MCMAC protocol.

The next section investigates different MAC mechanisms and the way they deal with clustering and mobility. The intended network and mobility behavior and a motivating application are given in Section 3. Section 4 describes the proposed protocol for supporting cluster mobility. The listening scheduling mechanism is explained in Section 5. The experimental setup and evaluation results are given in Section 6. Section 7 discusses an extension of the protocol for crowded networks. It also discusses possible effects of node mobility and the MCMAC mechanism on time synchronization of sensor nodes. Section 8 concludes.

2. RELATED WORK: ACCESSING SHARED MEDIUM AND MOBILITY

In this section we review prominent MAC paradigms in WSNs, showing their respective advantages and issues. In particular, their methods of dealing with clustering and mobility are studied.

2.1. MAC Paradigms for WSNs

There are several MAC protocols designed for WSNs, many of them targeting specific applications. The prominent MAC protocols for WSNs are explored in Langendoen and Meier [2010], in which their respective performance is analyzed. In general, MAC protocols for sharing the wireless channel in WSNs exploit one of the two major *contention-based* or *schedule-based* paradigms. Carrier-Sense Multiple-Access (CSMA) and ALOHA are the base mechanisms for designing many contention-based protocols in the literature of wireless ad hoc and sensor networks. Wise-MAC [El-Hoiydi and Decotignie 2004] (based on ALOHA) and B-MAC [Polastre et al. 2004] (based on CSMA) are known *asynchronous* protocols that rely on low-power listening (LPL) to reduce idle listening, while imposing the overhead of transmitting a long preamble on the sender side. X-MAC [Buettner et al. 2006] tries to alleviate the overhead by a shorter preamble using an acknowledge from the waked-up receiver. It also reduces the overhearing problem by embedding the target address in the preamble. S-MAC [Ye et al. 2002] and T-MAC [Dam and Langendoen 2003] synchronize the nodes to reduce idle listening through comprising a specific wake-up schedule within the frames. To further reduce energy consumption, SCP-MAC [Ye et al. 2006] shortens the length of the active part of the frames by synchronizing the channel probing (LPL) times of all neighboring nodes. Using scheduled channel probing, a very short preamble is required for senders to wake up the receivers in the range.

Contention-based protocols are considered well suited for WSNs due to their simplicity, flexibility, and little need for network infrastructure support. No assumption is made for network topology and node neighborhoods. Joining and leaving the network can be done without extra operations, so node mobility natively does not need specific support in this category of MAC protocols. However, these protocols suffer inefficient use of the bandwidth due to possible collisions. In particular, the mechanism is not well suited for dense networks. Although CSMA uses carrier sensing to avoid collisions, these can still happen due to the *hidden terminal* problem. Solutions such as RTS/CTS are very power costly for the rather short data packets in most WSN applications. The switching delay from carrier sensing to transmit mode is another native source of collisions in CSMA-based protocols.

Nodes using a schedule-based protocol, on the other hand, use TDMA to share the medium without collisions. These protocols are more efficient in terms of bandwidth usage. However, there are several issues that complicate these protocols. Among all, efficient slot scheduling plays an important role and is addressed in several publications. PEDAMACS [Ergen and Varaiya 2006] provides a centralized synchronization and slot scheduling mechanism using a high-powered access point. TRAMA [Rajendran et al. 2003], DRAND [Rhee et al. 2009], and LMAC [van Hoesel and Havinga 2005] are examples of efforts for efficient distributed slot scheduling in TDMA frames. The LMAC protocol, for instance, provides a distributed mechanism for occupying unique slots in a two-hop neighborhood. Every node propagates a bit-set detailing the occupied slots by itself and its one-hop neighbors. A node can detect the free slots in its two-hop neighborhood by OR-ing the received occupancy bit-sets and then selects a random slot among the unoccupied slots. The biggest challenge for such scheduling methods is network dynamics. In particular, node mobility can destroy the schedules and causes collisions. In this sense, TDMA-based protocols are said to be efficient protocols for static networks.

There are also some protocols in the literature that combine the two paradigms to collect their respective strengths. Z-MAC [Rhee et al. 2008], for instance, uses such a hybrid approach. The nodes in this protocol are synchronized. Under a low contention, nodes use CSMA in the TDMA slots and under high contention they try to use their own scheduled TDMA slots. Although Z-MAC is known as the first attempt of using a hybrid TDMA/CSMA approach for WSNs, the concept was previously exploited in Ephremides and Mowafi [1982] for a one-hop wireless LAN environment.

2.2. Cluster Communication

In general, node communication for data dissemination in WSNs is organized either by a hierarchical clustering architecture (using cluster heads) or a flat communication network. In the first approach, nodes are grouped into separate clusters with a certain hierarchy. Nodes within a cluster communicate with a cluster head that acts as an interface to the upper layer in the network hierarchy. The cluster head can be a preassigned node [Chen et al. 2004] or it can be selected periodically among the cluster nodes (e.g., in LEACH [Heinzelman et al. 2002]). This approach aims to provide an efficient and scalable network architecture. In a flat communication approach, every node communicates with its neighboring nodes to disseminate sensed data. Directed diffusion [Intanagonwiwat et al. 2003] and SPIN [Heinzelman and Balakrishnan 2002] are examples of the protocols that consider a flat network architecture. Depending on the context of the network and its application, one approach is selected.

In this work, we consider a flat architecture. We consider cases where clusters (groups) of nodes have application-level dependencies, such as in body-area networks. Nodes within a cluster are always together showing group mobility behavior. The mobile cluster nodes directly communicate with a (flat) backbone network and no cluster head is presumed. In specific applications, this provides better performance (overall latency and reliability) and makes the communication of the cluster nodes with the backbone network independent of other cluster nodes. As an example, in a health monitoring scenario in which there is a dense static network deployed in a building, our experiments show that some body nodes (in a WBAN) can better reach the static nodes in the neighborhood than other nodes in the WBAN (such as a cluster head). This may happen due to severe body shadowing. In this work, we provide mechanisms for efficient support of mobile clusters in applications in which all mobile cluster nodes directly send their packets to a static backbone network.

2.3. Mobility Support in TDMA-based MAC Protocols

Several *rescheduling* methods have been presented for TDMA protocols with the main objective of supporting joining and leaving nodes. A movement may be treated as leaving one neighborhood and joining another one. In LMAC [van Hoesel and Havinga 2005], for instance, a node may cause interference when it moves until the collisions are detected and slots are rescheduled in the neighborhood. Of course, a proper interference detection mechanism should be in place. In Jhumka and Kulkarni [2007], slot scheduling is done by cluster heads. Upon a node movement, the moving node may join another cluster and so it will be assigned a new slot by the head of the new cluster.

Such reactive rescheduling methods are effective for joining nodes and also for node relocation (very limited infrequent mobility). For instance, sensor nodes that are mounted on furniture can move and use this method to reschedule their transmit slot. Movement detection and new slot scheduling takes time. The mobile nodes interfere with other nodes for a while in between. It strongly affects the quality-of-service (QoS) of a running application if it should be continuously performed for highly mobile nodes.

M-LMAC [van Hoesel and Havinga 2008] supports mobility of singular nodes with event-based or very low sampling rate data generation. Static nodes use the LMAC slot scheduling mechanism. Mobile nodes, which are not synchronized to the static nodes, use carrier sensing whenever they have data to transmit. The time slots assigned to the static nodes are doubled in length to include a contention-based period. Every static node sends an Announcement Message (AM) at the beginning of its contention period and starts listening to the channel to receive possible responses from a mobile node. In the case that a mobile node in the neighborhood has data to transmit, it scans the channel to find an AM. If so, the mobile node tries to transmit its data by CSMA. If the data transmission succeeds, the static node sends an acknowledgement to the transmitting mobile node. This protocol is well suited for applications with sparse mobile node distributions having infrequent data transmissions, such as a fire fighting application (as mentioned and studied by the authors). However, it is not the best choice for applications like health monitoring, in which several clusters of mobile nodes exist with data transmission in every TDMA frame. In such applications, all nodes in a cluster (WBAN) are always in each other's interference range. Therefore, M-LMAC causes very many collisions that lead to a degraded performance. Moreover, a static node is only able to receive data from at most one mobile node in each frame, even if no collisions occur. This further limits the services for important body-related data in health applications. Moreover, it imposes high listening energy consumption overhead to the mobile nodes to scan the channel for AM. In a health monitoring application, in particular, body sensor nodes are very power-constrained nodes.

MMAC [Ali et al. 2005] provides a proactive rescheduling mechanism with the slot scheduling principles of TRAMA [Rajendran et al. 2003] as its core. A localization service is used to predict the mobility pattern of the mobile nodes. According to the predicted location of the nodes, the schedules are adapted. This way, the protocol tries to make itself ready for future movements. However, besides the complexity of the localization service and predictions, the performance of the protocol strongly depends on the accuracy of the predictions.

LWB [Ferrari et al. 2012] presents a cross-layer (MAC and routing) communication protocol for scenarios including mobile nodes. It uses a round-based centralized mechanism, performed by a host node, to schedule the access of wireless nodes to the medium. In each round, the nodes contend to inform the host of their traffic demands. The host then computes the schedule for the new round. Direct communication of all nodes with the host and the centralized scheduling mechanism put this work out of the scope of our intended scenarios.

Although the presented solutions for node mobility in TDMA-based MAC protocols improve performance of such protocols, none of them has specifically been designed for supporting group (cluster) mobility of nodes (like a WBAN). In this sense, MCMAC is the first work that addresses cluster mobility in WSNs. Several nodes may form a group and move together at once having a continuously high mobility. Moreover, the idle listening to the mobile nodes imposes a high power consumption overhead and methods to avoid it are worth considering. This is also not addressed properly in the current protocols. These issues are addressed in this article.

3. SYSTEM MODELS AND ASSUMPTIONS

3.1. A Motivating Application Scenario

There are several potential applications in which clusters of nodes are moving in presence of a static WSN. Healthcare applications are important examples. In particular, we consider healthcare monitoring applications such as elderly care or monitoring patients with chronic diseases like COPD (Chronic Obstructive Pulmonary Disease). In such a scenario, the patients are equipped with WBANs. The sensors are placed on several positions on the body to measure biological signals. Temperature, ECG, blood pressure, SpO2, and GSR are examples of biological sensors.

Additionally the building is equipped with sensors to measure required parameters. Sensors may be installed on the walls or on furniture such as chairs, beds, and electronic equipment to monitor the activity of the person. Sensor nodes are wireless for ease of installation and support of limited mobility, like refurbishing the house. An important task of this static network (besides sensing the relevant parameters) is to receive data from body sensor nodes and deliver it to the sink nodes through multihop communication. Sink nodes collect and process data from all nodes over the network. They can also send information to a medical center through a wired or wireless network, receive feedback, and inform the patient.

3.2. Network Architecture

Suppose that $S = \{s_k : 1 \leq k \leq N_s\}$ is the set of N_s static sensor nodes. These nodes are in principle static but they can be relocated (limited mobility). Moreover, a small subset of the set of static sensor nodes is considered as the sink nodes ($Sinks \subset S$). The number and location of the sink nodes are determined according to the circumstances in the real deployment considering the overall QoS requirements. Static nodes may be equipped with sensors and inject data into the network. However, the general view is that the static nodes have either infrequent event-based or low sampling rate data generation. Some static nodes without any sensing task can also be deployed in the network to maintain network connectivity.

In a deployment, we also assume that we have N_{mc} mobile clusters (MCs), each including a number of mobile sensor nodes. These nodes show a high mobility in the sense of both group mobility and individual mobility within the cluster. Assume that $C^i = \{c_1^i, c_2^i, \dots, c_{N_i}^i\}$, $1 \leq i \leq N_{mc}$ is the set of N_i sensor nodes of the i^{th} mobile cluster. The total number of sensor nodes in the network (network size N) is then $N = N_s + \sum_{i=1}^{N_{mc}} N_i$. In general, mobile cluster nodes are assumed to have tighter energy consumption constraints than the static nodes.

All wireless nodes in the network may generate data which is supposed to eventually reach the sink nodes. Beside sensing tasks, the static nodes are also responsible for dissemination of the sensed data from all mobile and static nodes toward the sink nodes using multihop communication. The exact data dissemination/routing algorithm is not relevant to this article.

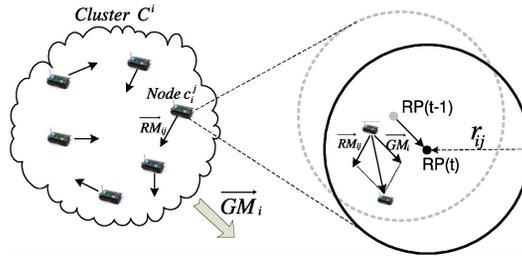


Fig. 1. RPGM model behavior, which illustrates the intended cluster mobility.

3.3. Cluster Mobility Definition and Modeling

We consider a mobile cluster C^i as a set of sensor nodes with a dependency in their physical location. This means that all nodes in a cluster are always in a bounded region with respect to each other. Of course, there might be an application-level dependency between the nodes in a cluster as well. Cluster nodes can have two kinds of mobility behavior. First, the cluster can move as a whole. Second, the nodes may have individual movement within the cluster bounds. The mobility only specifies the physical relationship between nodes within the cluster and does not presume any specific radio connectivity between them. We use the Reference Point Group Mobility (RPGM) [Hong et al. 1999] model to formally define such cluster mobility. This mobility model can properly illustrate the cluster mobility while retaining enough configurability for adapting the motion behavior according to the application scenario.

In the RPGM model, a Logical Center (LC) is set for every group of nodes (mobile clusters), the motion of which defines the overall cluster movement. Every group C^i has a group motion vector \vec{GM}_i that determines the motion of the group's LC. A reference point (RP_{ij}^t) is assigned individually to each node c_j^i in cluster C^i that moves with the group motion vector \vec{GM}_i . On the other hand, every node c_j^i may move within a predefined area (sphere with radius r_{ij}) around its reference point with a random motion vector \vec{RM}_{ij} . So the reference point allows independent motion of individual nodes in the group while the logical center provides the group movement. The location of the node c_j^i in each step is defined by the sum of the group motion and the random motion vectors ($\vec{GM}_i \mp \vec{RM}_{ij}$). Figure 1 illustrates this for an example cluster.

Besides using the RPGM model for defining the cluster mobility, we use this model for simulations that we perform to evaluate the performance of our protocol. By selecting a proper group motion vector (\vec{GM}_i) and the random motion vectors (\vec{RM}_{ij}), we model the movement patterns in the target application. As a common choice, we use the Random Waypoint Mobility Model (RWMM) [Zonoozi and Dassanayake 1977] for moving the logical center of the clusters. In each step, a random position is selected uniformly within the convex deployment area. A velocity value v_i is also selected from a given domain ($[v_{min} v_{max}]$) and the cluster moves towards the destination with the selected velocity. Once the logical center of the cluster reaches the destination point, a waiting time is uniformly randomly selected, considering a maximum waiting time w_{max} . So the cluster stays for a while in its location, selects the next destination and corresponding velocity, and starts to move again. For individual movement of sensor node c_j^i within cluster C^i , we set two parameters, namely the velocity (v_{ij}) and the radius (r_{ij}) of the sphere around its reference point. The decision is made based on the location of that node. All movement parameters are set according to the mobility in the target application.

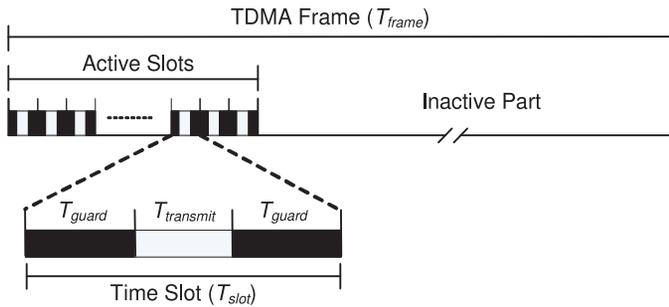


Fig. 2. The basic structure of the TDMA frames and time slots.

3.4. The Base TDMA MAC Mechanism

A TDMA-based mechanism [van der Wateren 2010] is considered as the base for communication of the wireless nodes. In the TDMA mechanism, the communication activities of the nodes are performed in a periodic manner using fixed-size TDMA frames. A *frame* consists of two parts, an *active* part and an *inactive* part. Figure 2 shows the general structure of a TDMA frame. The active part of the frame includes a fixed number of *time slots* that are used for communications. The channel is silent during the inactive part of the frame because none of the nodes transmits in that part. Therefore, all nodes go to the idle mode in the inactive part of the frame, leading to low average energy consumption for nodes. The slots in the active part of the frames are scheduled to different nodes in a neighborhood for their packet transmission. A unique slot assignment in the two-hop neighborhood, similar to the LMAC [van Hoesel and Havinga 2004] slot scheduling strategy, is used to assign slots to the nodes for transmission. A node transmits its data in its scheduled slot and listens in the other active slots to receive packets from its neighbors. This mechanism provides very low-duty-cycle TDMA frames, which leads to a very low energy consumption of wireless sensor nodes.

Synchronization is an important requirement for a schedule-based medium access scheme, as its performance and accuracy influence the performance of the network. In the base MAC layer, the wireless nodes are synchronized using a decentralized frame synchronization mechanism [Assegei 2008; van der Wateren 2010]. It achieves synchronization by adapting the phase of the node's clock to those of its neighbors without explicitly exchanging the sender's clock timestamps. The node computes the phase difference of its TDMA frame with the frames of its direct neighbors from the time of the packet reception from the neighbor (if any) relative to the expected time of the arrival of the packet. The node then adjusts its frame schedule taking the phase differences into account. Assegei [2008] investigates several methods for adjusting the TDMA frame. A median [Tjoa et al. 2004] algorithm, a least-square method, and a method using a discrete Kalman filter [Welch and Bishop 1995] are studied.

To avoid problems due to small phase errors that could lead to unintentionally overlapping transmissions, a guard time is inserted at the beginning and the end of every active slot (Figure 2). Thus one slot consists of two guard times and one transmission time ($T_{slot} = 2 \times T_{guard} + T_{transmit}$). In a transmit slot, relative to its own clock, the sender node waits for one guard time and then starts packet transmission. The receiver nodes start listening to the channel from the beginning of the slot, according to its own clock, until they receive the packet or reach the end of the slot. This way, phase errors between sender and receiver smaller than a guard time are tolerated.

In our work, we consider the TDMA-based MAC layer of van der Wateren [2010] as the base. This MAC takes care of synchronizing the nodes. The objective of MCMAC is to provide cluster mobility support for such a TDMA-based MAC protocol to efficiently

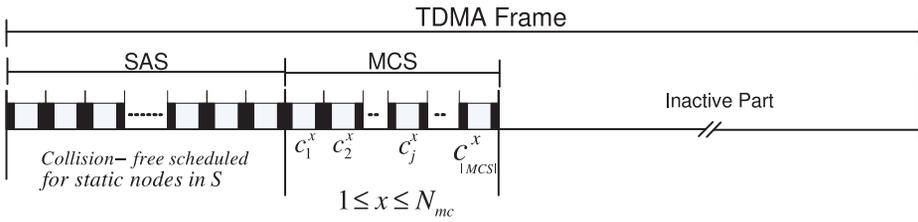


Fig. 3. A TDMA frame containing the SAS and MCS parts.

deliver the data generated by the mobile nodes to the static network and optimize energy consumption of nodes for performing this task. The mechanism is designed in a way that keeps the duty cycle of all nodes as short as possible.

4. SUPPORTING CLUSTER MOBILITY

The MCMAC protocol uses a hybrid schedule-based and contention-based communication mechanism to support cluster mobility. In this section, the MCMAC mechanism and its different variations are presented.

4.1. A Hybrid Communication Approach

In the base TDMA-based MAC layer explained in Section 3.4, slots in the active part of the frames are exclusively assigned to different sensor nodes in a neighborhood. In MCMAC, the slots that are used by static nodes and mobile cluster nodes are separated and different accessing paradigms are used for each part. The active part of the frames is split into two separate parts shown in Figure 3, a static active section (SAS) and a mobile cluster section (MCS). The SAS is used by the static nodes to transmit their data. A pure TDMA scheme with the distributed dynamic slot scheduling strategy of LMAC [van Hoesel and Havinga 2005] is exploited in this part. Using this approach we benefit from spatial reuse of RF channels, thus each static node occupies a slot from SAS that is unique in its two-hop neighborhood. However, other scheduling mechanisms can also be used here without impacting the other part of the protocol. In any case, these mechanisms try to minimize the number of SAS slots ($|SAS|$) through smart slot reuse. Static nodes exchange data with each other by transmitting packets in their scheduled slot and listening to other SAS slots.

A contention-based mechanism is exploited within the TDMA slots of the second part of the frame (MCS). Each node c_j^i in cluster C^i is assigned statically a unique slot in this part to transmit its data. The size of the MCS part ($|MCS|$) is then determined by the number of nodes in the mobile clusters. The maximum cluster size is taken as the MCS length. Nodes within a cluster can always access the channel without interfering with each other and without the need for rescheduling upon intra-cluster topology changes due to individual node movements. Given the fact that the nodes in a cluster are mostly within a one- or two-hop neighborhood of each other, statically assigning a unique slot to each node of a cluster is an efficient approach. Considering the high mobility of the nodes within the cluster, that leads to frequent cluster topology changes, adaptive slot reuse in this part is inefficient and costly. In Section 7.1, where we discuss supporting high cluster densities, the mobile clusters are classified taking their size into account in the case that the sizes of different clusters in the network vary considerably. Multiple MCS parts of different sizes are then used.

The sensor nodes from different mobile clusters share a single MCS part. For instance, node c_j^i from cluster C^i and node c_j^k from cluster C^k share the j^{th} MCS slot (for simplicity, we assume that the nodes in the cluster are indexed according to their assigned slots). Thus, when these two clusters are in each other's range, they compete

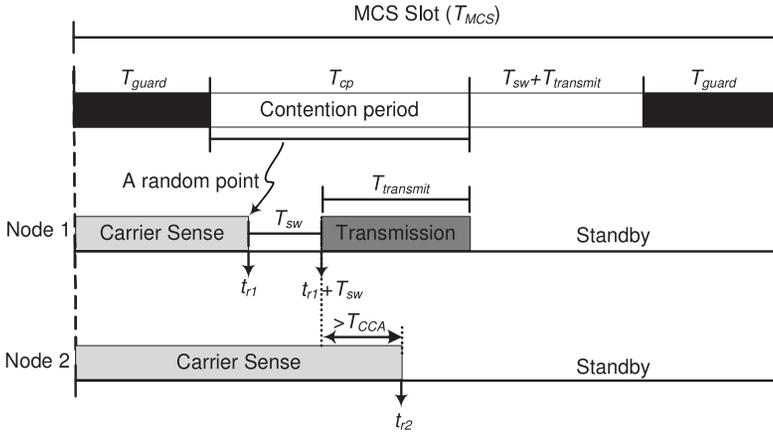


Fig. 4. The structure of a time slot prepared for performing CSMA in the MCS part of the frame.

to access the same slot and transmit their data. Figure 3 illustrates the structure of a frame and slot allocations in the MCMAC protocol. A contention-based mechanism is used for accessing such a TDMA slot. Two variations of the protocol are developed using CSMA or slotted ALOHA as the contention-based paradigms. The appropriate version is then selected according to the circumstances and hardware platform in the application. Both variations and guidelines for decision making are presented in detail in this section.

Static nodes listen in the MCS slots of the frame to receive data from cluster nodes and forward it to other static nodes to eventually be delivered to the sink node. As the position of the MCS is known in the frame, the static nodes do not need to listen in the whole inactive part of the frame to possibly receive packets from mobile clusters. This is the main reason for specifying a fixed MCS in the frames with a contention-based access by the mobile cluster nodes. Another important advantage is that, by developing a smart listening schedule, static nodes can avoid listening in MCS slots when no mobile cluster is around. We present such a mechanism later in this section. According to the application scenario, cluster nodes may also listen to the SAS to receive information from the static network. Feedback, comments, and alarms for the person wearing the WBAN are examples of such information.

4.2. Carrier-Sense Multiple-Access Paradigm

CSMA is used as the first contention-based mechanism for accessing the shared MCS slots. Every cluster node performs carrier sensing before transmitting in its dedicated slot. The duration of a time slot is extended by the duration $T_{cp} + T_{sw}$, where T_{cp} is the length of the *contention period* in which nodes compete to transmit their data. T_{sw} stands for the switching time of the radio transceiver from carrier sensing to transmit mode. Figure 4 shows the structure of one time slot prepared for performing CSMA. $T_{transmit}$ stands for the actual time required for transmitting one fixed-sized packet. Cluster nodes randomly pick a time point (t_r) in the contention period and sense the channel from the beginning of the slot until t_r . If the channel is found idle, the node switches to transmit mode to start sending its packet (the case for node1 in Figure 4). If the node detects a carrier, it waits for the next frames to try again (node2). Note that T_{CCA} is the clear channel assessment (CCA) time which is the minimum time that the radio chip requires to sense the wave to be able to state the existence of a carrier on the channel. For instance, the value of T_{CCA} in both Chipcon CC2420 [Texas

Instruments 2010] and Nordic nRF24L01 [Nordic Semiconductor 2007] transceivers is $128\mu s$ (8 symbol periods).

There are several aspects that should be considered for performing CSMA in MCS slots. The features of the carrier sensing facilities of the radio transceiver and the TDMA framing and slot length are important factors. To obtain a better understanding of how these factors influence the performance of CSMA, we analyse the probability of collision or successful transmission in an MCS slot. Assume that there are two nodes in each other's interference range. To have a successful transmission of one of them, the distance of the selected random points within the contention period should be at least $(T_{CCA} + T_{sw})$ apart. Suppose without loss of generality that in one round $t_{r1} < t_{r2}$. The real transmission of node1 starts at time $t_{r1} + T_{sw}$. Node2 should at least sense the signal of the first node for a duration of T_{CCA} to detect that. The probability of one transmission without collision is as follows.

$$P_{col}(Node1, Node2) = P(|t_{r1} - t_{r2}| > T_{sw} + T_{CCA}) \quad (1)$$

Points t_{r1} and t_{r2} are selections out of two independent uniform distributions from the contention period of length T_{cp} . Therefore, we can compute Eq. (2) to be the probability of a collision-free transmission of one of these two nodes. Note that the effect of the hidden terminal problem is not taken into account in this calculation.

$$P_{col}(Node1, Node2) = \left(1 - \frac{T_{sw} + T_{CCA}}{T_{cp}}\right)^2 \quad (2)$$

According to the protocol, the i^{th} node from all mobile clusters share the same MCS slots (the i^{th} slot). Let Γ be the set of mobile clusters that are in each other's interference range at a certain time frame. The size of this set ($\gamma = |\Gamma|$) is the *gathering likelihood* of mobile clusters ($1 \leq \gamma \leq N_{mc}$). To have a successful transmission by one mobile node in this situation, all random time points selected by mobile nodes but the earliest one should be at least $T_{sw} + T_{CCA}$ later than the earliest one. In this case, the earliest node succeeds in transmitting without colliding with other nodes. As different nodes select their t_r points independently, the probability of not colliding with $\gamma - 1$ nodes is the product of the probability of not colliding with each of them (Eq. (2)). To which node the earliest point belongs does not make a difference. Now, from the perspective of one competing cluster node c_j^i , the probability of selecting the earliest point ($1/\gamma$) and successfully transmitting (winning the lottery and no collision happening) in the current round is as follows.

$$P_{CSMA}(c_j^i) = \frac{1}{\gamma} \left(\prod_{c_j^k: C^k \in \Gamma, k \neq i} P_{col}(c_j^i, c_j^k) \right) = \frac{1}{\gamma} \left(1 - \frac{T_{sw} + T_{CCA}}{T_{cp}}\right)^{2(\gamma-1)} \quad 1 \leq \gamma \leq N_{mc} \quad (3)$$

A longer contention period leads to a higher chance of success. The appropriate value should be set considering the expected success probability (QoS requirements), the tendency of mobile clusters to gather in the target application, and the radio chip features (T_{CCA} and T_{sw}). $T_{transmit}$ is also an important factor. A long contention period is not reasonable for very short transmissions, as it imposes a high overhead.

4.3. Slotted ALOHA

The second contention-based scheme for accessing MCS slots is a slotted ALOHA mechanism. In this version, N_{AL} slots construct one *ALOHA superslot* and are dedicated to the j^{th} node of all clusters (instead of one single slot). The number of superslots in an MCS is then $|MCS|$. Mobile cluster nodes in each round randomly select one slot among

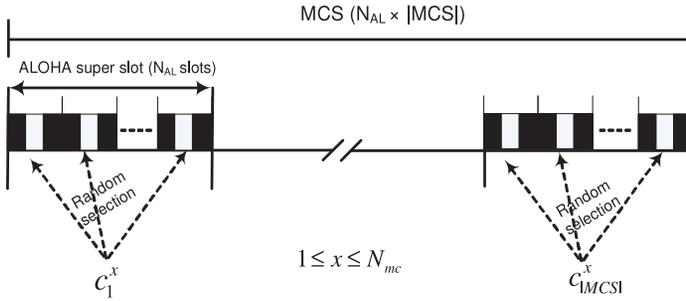


Fig. 5. The structure of a superslot prepared for performing slotted ALOHA in the MCS part of the frame.

N_{AL} in their ALOHA superslot and transmit their data in that slot. Figure 5 depicts the structure of the MCS part of the frame and its ALOHA superslots. No carrier sensing is performed by the mobile cluster nodes. Moreover, the structure of the time slots does not change (Figure 2). It makes the implementation of the protocol simpler than the CSMA approach.

Assume that in some frame, γ sensor nodes from different mobile clusters are in each other's range and want to transmit in the same ALOHA superslot. This means that each of the γ nodes selects a random slot out of N_{AL} slots. In such a case, a certain cluster node (c_j^i) successfully transmits if all other nodes ($c_j^k : C^k \in \Gamma$) select another slot than the one selected by that node. The probability that node c_j^i selects a certain slot is $\frac{1}{N_{AL}}$. It succeeds to transmit only if none of the other $\gamma - 1$ competing nodes selects the same slot. This happens with probability of $(1 - \frac{1}{N_{AL}})^{\gamma-1}$. Such competition may also occur for transmission in other slots out of N_{AL} slots. Therefore, the probability of a collision-free transmission for the given node (c_j^i) is calculated as follows.

$$P_{ALOHA}(c_j^i) = N_{AL} \times \frac{1}{N_{AL}} \times \left(1 - \frac{1}{N_{AL}}\right)^{\gamma-1} = \left(1 - \frac{1}{N_{AL}}\right)^{\gamma-1} \quad 1 \leq \gamma \leq N_{mc} \quad (4)$$

The controllable parameter here is the length of ALOHA superslots (N_{AL}). According to Eq. (4), a bigger value of N_{AL} increases the chance of success. Again the value of γ varies by movement of the clusters and depends on the likelihood of mobile clusters to gather in the target application. Notice that there is a slight difference with respect to the successful transmissions between these two contention-based approaches. In CSMA only one node may successfully transmit its data in each frame. In slotted ALOHA, however, there is some chance that multiple nodes are successful in one frame and do not collide with other nodes in the range.

4.4. Guidelines for Decision Making

Slotted ALOHA is easier to implement and does not need special support from the radio chip. When the transmit time ($T_{transmit}$) and accordingly slot length (T_{slot}) are short, this approach is indeed worth considering. In contrast, CSMA performs better for applications with longer transmissions. To have a quantitative comparison between these two approaches in an application, we investigate the amount of listening activities imposed on static nodes to listen to the MCS part. Assume that $T_{slot} = T_{transmit} + 2 \times T_{guard}$ is the length of a basic slot (Figure 2). The time duration of the MCS part of the frame (T_{MCS}) is given by Eqs. (5) and (6) for the CSMA and ALOHA versions, respectively. This is actually the time in which static nodes listen to the channel to possibly receive from the cluster nodes.

Table I. Values of the Parameters in Two Different Deployments

<i>metric</i>	<i>MyriaNed [van der Wateren 2011]</i>	<i>Mica2/Z-MAC [Rhee et al. 2008]</i>
<i>Data rate</i>	2 Mbps	19.2 Kbps
T_{CCA}	128 μ s	400 μ s
T_{sw}	130 μ s	200 μ s
<i>Slot length (T_{slot})</i>	764 μ s	50 ms
T_{cp} (Eq. (7))	2500 μ s	6 ms
N_{AL} (Eq. (8))	2	2
$T_{MCS}(CSMA)$	3400 μ s $\times MCS $	56 ms $\times MCS $
$T_{MCS}(ALOHA)$	1528 μ s $\times MCS $	100 ms $\times MCS $

$$T_{MCS}(CSMA) = [T_{slot} + T_{cp} + T_{sw}] \times |MCS| \quad (5)$$

$$T_{MCS}(ALOHA) = T_{slot} \times N_{AL} \times |MCS| \quad (6)$$

Comparing $T_{MCS}(CSMA)$ and $T_{MCS}(ALOHA)$ reveals which approach fits better in the target application. This comparison should be done when the parameters of the CSMA and ALOHA are set in such a way that both provide the same transmission success probability for a given cluster node in the same situation. T_{cp} and N_{AL} are controllable parameters of the CSMA and slotted ALOHA mechanisms and are given by Eqs. (7) and (8), respectively.

$$T_{cp} = (T_{sw} + T_{CCA}) \cdot \left(1 - [P_{CSMA} \cdot \gamma]^{\frac{1}{2(\gamma-1)}}\right)^{-1} \quad \gamma > 1 \quad (7)$$

$$N_{AL} = \left\lceil \left(1 - [P_{ALOHA}]^{\frac{1}{\gamma-1}}\right)^{-1} \right\rceil \quad \gamma > 1 \quad (8)$$

Equal values are set for the success probabilities ($P_{CSMA} = P_{ALOHA}$), so that both approaches are compared while providing the same transmit success probability. A fair comparison is then provided by inserting the values coming out from Eq. (7) and Eq. (8) in Eq. (5) and Eq. (6).

An example illustrates this procedure. We consider two deployments using different motes and TDMA structures. Assume that MyriaNed [van der Wateren 2011] is used in the first setting. Each slot contains one packet and leads to small slots (taking the high data rate into account). In contrast, consider Z-MAC [Rhee et al. 2008] settings using Mica2 motes. The data rate is much lower and so the slot length is much bigger. The values of different parameters of these two cases are shown in Table I together with the results of the calculations. We assume that the gathering likelihood is $\gamma = 2$ and the expected probability of successful transmission is 40%. Comparing the calculated value for $T_{MCS}(CSMA)$ and $T_{MCS}(ALOHA)$ allows us to decide which approach fits better for a setting. The results clearly show that in the first case using MyriaNed motes, the slotted ALOHA version performs better as $T_{MCS}(CSMA) \gg T_{MCS}(ALOHA)$. The reason is, indeed, the small size of the slots. In contrast, the CSMA mechanism is surely promising for the second case because $T_{MCS}(CSMA) \ll T_{MCS}(ALOHA)$.

5. SCHEDULING THE LISTENING TO MOBILE CLUSTERS

Power consumption is a central challenge in most WSN applications. Designing energy-efficient protocols is of great importance to provide an acceptable lifetime for typical energy-constrained sensor nodes. Idle listening is one of the power wasting activities in wireless communications and is to be minimized. In the MCMAC protocol, every static node listens to the MCS slots to receive data from sensor nodes in the mobile clusters

and route it to the sink nodes. However, when no cluster is around, the static node can stop listening to the MCS part to save energy. We fortify the MCMAC protocol with a mechanism with which the static nodes schedule their listening to mobile clusters. One of the reasons for having a shared contention-based part (MCS) in each frame is the possibility to minimize such idle listening when no cluster is around the static nodes.

The mechanism works as follows. Each static node continuously estimates the hop distance of the nearest cluster to itself. By following the variations of the estimated hop distance, the node tries to realize the movement behavior of the clusters. Based on that, the node decides how often it needs to listen to MCS slots of the frame. The objective is to reduce idle listening while avoiding data loss due to not listening to the clusters in the range. Both our experiments and simulations show a very considerable energy saving (around 70% on average) for static nodes without negatively influencing other performance metrics such as latency and delivery ratio. Clearly, the number of mobile clusters and their density in a deployment play an important role in the energy saving.

5.1. Hop-Distance Estimation

The goal is to have a distributed mechanism with which every static node estimates how far the clusters are away from it. As there is only one MCS part in the frame, static nodes only need to estimate the hop distance (d) from the nearest cluster. Each node in the network propagates its estimated distance by adding it into its packets. Cluster nodes always send zero ($d = 0$). Every static node that receives zero in a round realizes that there is at least one cluster in its one-hop neighborhood. Further, each node receives the hop distance of its direct neighbors in each round. It then estimates its own distance as the minimum distance of its neighbors plus one. In the case that a node did not receive any packet from any neighbor in a round, it keeps its estimate of the previous round. Figure 6 shows the value of d for all nodes in a given 6×5 static mesh network with two mobile clusters.

Note that the static nodes need to know how far the mobile clusters are to be able to schedule their listening. When the mobile clusters are far away, the node can relax from listening to the MCS part. Any *big* value of the hop distance has almost the same meaning; taking this into consideration we can specify a maximum value for the hop distance (d_{max}). It allows a limited small space for transmitting its value. Our experiments show that a value $d_{max} = 8$ is big enough. Therefore, three or four bits would be needed to transmit the parameter d . In comparison with the typical packet length (e.g., 32 bytes), the overhead of integrating hop distance d to the packets is low. Note that, the hop-distance value is added to the data packets in protocols with *frequent* periodic data exchange. In other protocols, very small control packets are added. Note that, in TDMA-based protocols, frequent data or control packets already exist for other purposes, in particular time synchronization.

The minimum hop distance from the mobile clusters is considered as an estimator of physical distance of the nearest mobile cluster to a certain static node. However, translation of the hop distance to the physical distance is not straightforward. Many issues play a role here. Nonuniform transmission patterns and heterogeneous transmit ranges of the sensor nodes, spatial and temporal link-quality variations, and available paths for movements influence the precision of the estimations. In particular, instantaneous connections and disconnections make the hop distance d vary over time even when no cluster is moving. Figure 7 shows the estimated hop distance for a sample static node in one of our experiments for a certain period of time. The general variation pattern of d reveals that the cluster goes far away and then approaches the node. However, the fluctuations due to wireless link variations are quite visible. Taking such issues into account, a conservative approach together with some averaging methods

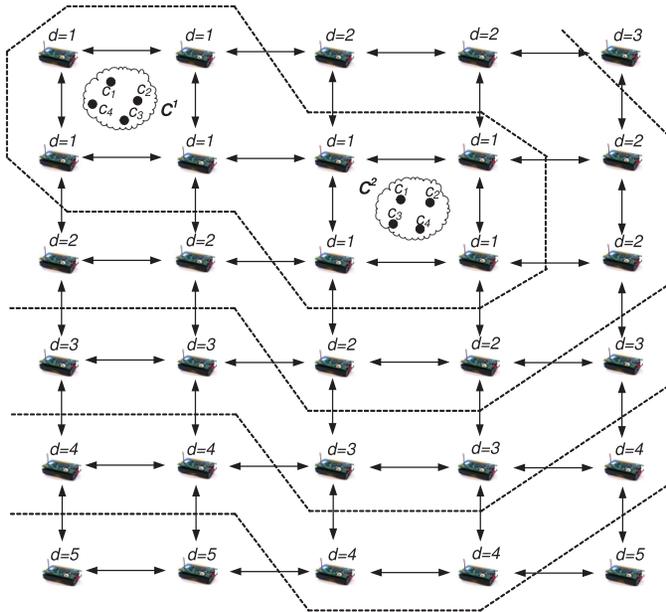


Fig. 6. Hop-distance estimation from the nearest cluster in a 6×5 nodes static WSN with two mobile clusters, each containing four nodes. Regions with the same d values are pointed out. The regular deployment is only to illustrate the hop-distance concept. The MCMAC protocol does not assume such a deployment.

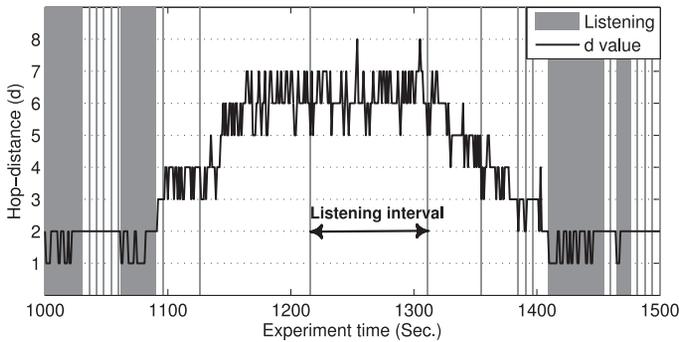


Fig. 7. The variations of the estimated hop distance d in a duration of a real experiment for a sample static node and its decision about listening to the MCS part.

seems needed for scheduling the listening activities of the static nodes. We discuss such an approach next.

5.2. Listening Schedule to Mobile Clusters

Algorithm 1 illustrates the behavior of a static node in the MCMAC protocol. The function `STATICOPT()` decides whether or not the node listens to the MCS part. Algorithm 2 presents this function. Every node maintains a listening interval time T_l that determines how often the static node listens to the MCS part of the frame. It is adjusted considering the value of parameter d and its history record. The node listens to the MCS part (`STATICOPT()` returns *true*) for one round in every T_l round(s). Figure 7 shows the listening interval of a static node in our experiments for a certain time duration

ALGORITHM 1: Behavior of a static node at round t .

```

for every TDMA frame  $t$  do
   $Lst \leftarrow \text{STATICOPT}(d(t));$ 
   $Rxd[i] = d_{max} + 1, 0 \leq i \leq |SAS|;$  /* reset values */
  for  $i = 1$  to  $|SAS|$  do
    if  $i \neq \text{ownTxSlot}$  then
       $RxPacket \leftarrow \text{Listen}();$ 
      if Received any then  $Rxd[i] = RxPacket.d;$ 
    else
       $TxPacket.d \leftarrow d(t);$ 
       $\text{Transmit}(TxPacket);$ 
    end
  end
  if  $Lst$  then
    for  $i = 1$  to  $|MCS|$  do
       $RxPacket \leftarrow \text{Listen}();$ 
      if Received any then  $Rxd[0] \leftarrow 0;$ 
    end
  end
   $d \leftarrow n\{ Rxd[i] \mid i : 0 \leq i \leq |SAS| \} + 1;$ 
  if  $d \leq d_{max}$  then
     $d(t+1) \leftarrow d;$ 
  else
     $d(t+1) \leftarrow d(t);$ 
  end
  Go to sleep mode until the next frame
end

```

ALGORITHM 2: The optimization function.

```

STATICOPT()
Input:  $d(t)$  /* the last estimated hop-distance */
Output:  $Lst$  /* if node listens to the MCS part */
calculate  $d_{avg}(t)$ ,  $\delta_d(t)$ , and  $\Delta d(t)$  using Eqn. 9,10,11 ;
switch  $\Delta d(t)$  do
  case  $\Delta d(t) > 0$   $T_l^t \leftarrow T_l^{t-1} + d(t);$ 
  case  $\Delta d(t) = 0$   $T_l^t \leftarrow T_l^{t-1} + 1;$ 
  case  $\Delta d(t) < 0$   $T_l^t \leftarrow \left\lfloor \frac{T_l^{t-1}}{2^{\delta_d(t)}} \right\rfloor;$ 
endsw
if  $T_l^t > T_{l,max} [d_{avg}(t)]$  then  $T_l^t \leftarrow T_{l,max} [d_{avg}(t)]$ 
 $IntervalCnt \leftarrow IntervalCnt + 1;$ 
if  $IntervalCnt \geq T_l^t$  then
   $Lst \leftarrow true;$ 
   $IntervalCnt \leftarrow 0;$ 
else
   $Lst \leftarrow false;$ 
end

```

and the decision of the static node about listening to the MCS part. The process for adjusting the listening interval should balance two aspects. First, it should minimize idle listening when there is no mobile cluster around. Second, it should minimize the packet loss caused by not listening to the MCS part, when there is a cluster in the neighborhood.

The intention is to reduce the idle listening while imposing minimum possible packet loss. An Additive Increase Multiplicative Decrease (AIMD) [Cai et al. 2003; Liu and Modiano 2005] mechanism is exploited as a conservative approach for adjusting the

value of T_l . To perform the AIMD, the hop distance d is considered from different angles: the last estimated value $d(t)$, an average over a limited history $d_{avg}(t)$, its last variation $\Delta d(t)$, and the deviation from the average $\delta_d(t)$. To reduce the effect of variations of d due to the link variations and to separate this effect from a real change of the value of d caused by the movement of a cluster, we use an exponentially weighted average over a limited history (H TDMA rounds) of the parameter d (Eq. (9)). The parameter α determines the gain of the weighted averaging.

$$d_{avg}(t) = \left\lfloor \frac{\sum_{k=1}^H [a_k \times d(t-k)]}{\sum_{k=1}^H a_k} \right\rfloor, \quad a_k = \left(\frac{1}{\alpha}\right)^k, \quad \alpha \geq 1 \quad (9)$$

$$\delta_d(t) = d(t) - d_{avg}(t) \quad (10)$$

$$\Delta d(t) = d(t) - d(t-1) \quad (11)$$

These parameters are used in Eq. (12) to adjust the value of the listening interval. T_l^{t-1} is the current listening interval. T_l^t is an intermediate version of the next value of the listening interval based on the AIMD paradigm. The method for adjusting the time interval is developed based on several experiments in various circumstances taking the observed variations of parameter d over time into account. The first two cases in Eq. (12) provide an additive increase of the listening interval when the nearest cluster is estimated to go farther or remain in the same place. While the hop distance remains fixed, it adds up one step (TDMA round). When the value of d increases, it can increase with bigger steps. We set such value as a factor (β) of the current hop distance. Farther clusters allow us to increase the interval with bigger steps without any risk. For the current implementation we set ($\beta = 1$). The last case performs the multiplicative decrease when d steps down. In this case, the value of the listening interval is divided by an exponential factor determined by the deviation of the hop distance to its average value ($2^{|\delta_d|}$). Such an exponential factor makes the same decrement for any decreasing pattern of the hop distance d .

$$T_l^t = \begin{cases} T_l^{t-1} + \beta \cdot d(t) & \Delta d > 0 \\ T_l^{t-1} + 1 & \Delta d = 0 \\ \left\lfloor \frac{T_l^{t-1}}{2^{|\delta_d(t)|}} \right\rfloor & \Delta d(t) < 0 \end{cases} \quad (12)$$

However, the listening interval should not be limitless. Consider a situation in which the cluster nodes stay fixed for a very long period of time. The listening interval additively increases. At a time, a mobile cluster starts moving. If T_l has become too big, decreasing that takes time even with the multiplicative method. Joining new mobile clusters is another issue that should be considered here. Thus, a maximum allowed value is set for the listening interval individually determined for each hop-distance average. $T_{l,max}[k]$ is the length limit of the listening interval when the weighted average hop distance is k ($d_{avg}(t) = k$).

$$T_l^t = \min\{T_l^t, T_{l,max}[d_{avg}(t)]\} \quad (13)$$

T_l^t is the new value of the listening interval. $T_{l,max}[1]$ is always set to one so that a node listens to the MCS part in all rounds when there is a cluster around. The proper maximum value for other cases depends on the density of the network and the average speed of the mobile clusters in the application.

5.3. Trade-Offs: An Empirical Exploration

The presented listening optimization mechanism has a couple of controllable parameters. In particular, the length of the averaging history (H), the gain of the weighted averaging (α), and the maximum allowed limits for the listening interval ($T_{l,max}[\cdot]$) are parameter settings that influence the performance of the mechanism. This section investigates the trade-off that these parameters make between energy consumption and packet losses in a real network setup.

To get insight in the variations of parameter d in a real-life network with movement specifications of a human, we performed 6 hours of experiment with the setup explained in Section 6 with only one mobile cluster (a human wearing four nodes) showing various mobility behavior. The goal is to realize how parameter d changes when a human approaches a static node or goes farther, and finally which kind of parameter settings work better for such movement specifications. Thus, only one human is used in this experiment to better investigate the network movement with respect to a human movement. During the experiment, the value of hop distance d and packet reception are logged in 60 static nodes. In this experiment, static nodes were always listening to the MCS part. By offline running of the listening scheduling mechanism with various configurations and using the logged data (hop distance and packet reception in each round), we can evaluate the efficiency of the mechanism with different settings. In particular, we considered parameters H , α , and $T_{l,max}[2]$. Ten different values were set for each parameters. Thus, in total, 1000 configurations have been tried. $T_{l,max}[2]$ is specifically considered because experiments show that its value has an important influence on the performance of the mechanism.

The observed performance metrics are the percentage of rounds with idle listening to the MCS part and nonlistening packet loss. When the offline running of function `STATICOPT()` returns *false* and a packet is received from a mobile cluster in the real experiment in that specific round, it is counted as a packet loss. Idle listening happens whenever `STATICOPT()` returns *true*, but no packet is received in the MCS part. As the obtained metrics vary over different static nodes, we calculate the average metrics over all static nodes.

Figure 8 depicts the explored trade-offs between the idle listening and packet loss in terms of the parameters of the mechanism. To investigate the influence of a parameter on the performance of the mechanism, in each iteration, various values for a parameter are tried while other parameters are fixed. Figure 8(a) shows the trade-off in terms of the maximum allowed listening interval when the average hop distance from the nearest mobile cluster is 2 ($T_{l,max}[2]$). Increasing the value of this parameter strongly decreases the idle listening at the cost of more packet loss from mobile cluster nodes and vice versa. Note that other limits of the listening interval ($T_{l,max}[n]$, $n > 2$) also make the same trade-off, but with a smaller influence on the performance metrics.

Figure 8(b) shows the trade-off in terms of the length of the averaging history (H). When a longer record is taken into account, the idle listening increases because the additive increase of the listening interval is triggered with more delay. However, the influence of history length on performance metrics is not as strong as the one from $T_{l,max}[2]$. Considering the averaging parameters (H and α), the conclusion is that using a smoother averaging (greater H and smaller α) so that the new values of d have softer influence on $d_{avg}(t)$, a lower idle listening is achieved. In such a case, the reaction of the mechanism is delayed when d drops, and so more packets are lost. If the averaging is set to be more dependent on the new values (smaller H and greater α), packet loss decreases and idle listening increases. This reflects the behavior of the AIMD mechanism.

Figure 9(a) depicts the average obtained metrics for all tried configurations. This clearly shows that some configurations are far from optimal. Figure 9(b) shows the

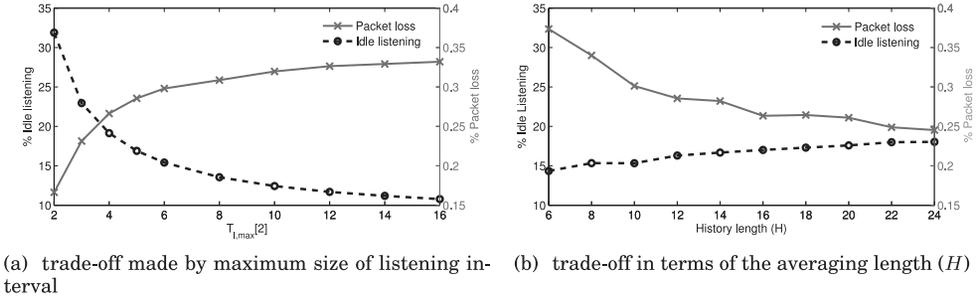


Fig. 8. Trade-off between idle listening to MCS part and packet loss in terms of the parameters of the listening scheduling.

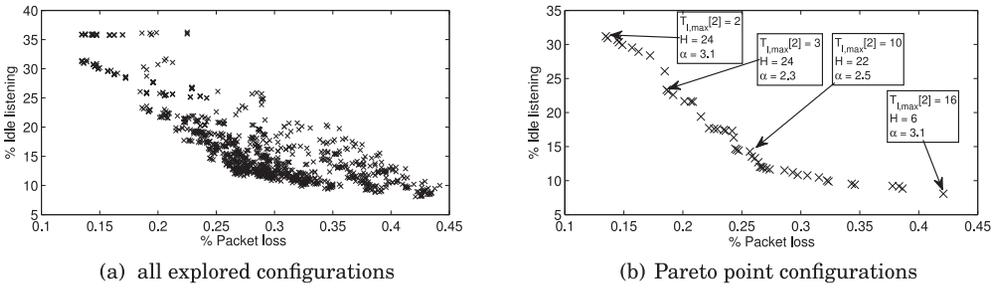


Fig. 9. Empirical exploration of the configuration space for listening scheduling mechanism.

Pareto points among all configurations. These are the settings that are not dominated by any other setting. A Pareto configuration should be finally selected based on the QoS constraints in the application scenario. When packet loss cannot be tolerated, the settings on the top-left side of the graph are selected. Points on the other extreme are proper when energy saving is an important requirement. Notice that a packet loss measured in a static node would not necessarily mean the loss of the packet from the cluster node. According to the network architecture, there can be other static nodes in the range that listen to the MCS part and so receive the packets. Taking this into account, we can say that packet loss might be tolerated to some extent.

6. PERFORMANCE EVALUATION

6.1. Experimental Setup

We investigate the applicability and performance of the cluster mobility support through performing various real-world experiments and simulations. We consider on-body sensor nodes in the form of WBANs as mobile clusters. During the real-world experiments, we used a rather large-scale indoor deployment using wireless motes. Real human behavior in a working environment, such as sitting in the office, walking, and gathering of several people wearing the nodes (mobile clusters), makes the experiments realistic. Experiments also reflect the effect of possible inaccuracy and failures in synchronization of TDMA frames of wireless nodes in the network, which makes the results more reliable and realistic. Various performance metrics are investigated during these experiments through in-detail analysis of the logged data.

Besides the experiments with real nodes, we also extensively simulated the protocol in several network setups. We run different simulations with and without performing listening scheduling to evaluate its influence on various QoS metrics. We also simulated network setups with various scales and cluster densities.

6.2. Protocol Stack

In the MAC layer of the protocol stack used in the experiments and simulations, slots in the SAS part of the TDMA frames are assigned to the static nodes based on the slot scheduling of the LMAC protocol [van Hoesel and Havinga 2005]. Each static node occupies a time slot to send its data packets that is unique in the node's two-hop neighborhood. In this sense, the communication between static nodes is in principle contention free.

An epidemic data dissemination mechanism, gossiping [Gavidia and van Steen 2008] is used by the static nodes to propagate data items generated by static nodes themselves as well as data received from mobile cluster nodes. Static nodes use a store-and-forward scheme to deliver sensor data to the sink node. On top of the data routing protocol, a monitoring application is used in which sensor nodes periodically sense some parameter and pass the sensed data to the lower layer. The exact physical parameter being sensed is not relevant here.

6.3. Real-World Experiments

Deployment setup. We use *MyriaNed* [van der Wateren 2011] wireless nodes for our experiments, which feature an ATMEGA128 microcontroller and a *Nordic nRF24L01* chip [Nordic Semiconductor 2007] as radio transceiver. The radio chip works in the 2.4 GHz ISM band using a data rate of 2Mbps and a 32-bytes fixed packet size. Taking the packet preambles into account, the transmit time of a packet is around $T_{transmit} = 164\mu s$. The radio can be set in RX, TX, or standby modes consuming 12.3mA, 11.3mA, and $22\mu A$, respectively. Nodes are equipped with a 4MByte Flash memory, which is used to log data (e.g., the radio activities) at each round of the experiment. After finishing each experiment, the logged data is downloaded from the nodes and is analyzed to extract efficiency metrics. Considering the specification of the radio chip, the slot length is $T_{slot} = 764\mu s$. The TDMA frame length is set to one second ($T_{frame} = 1s$). *MyriaNed* nodes use the decentralized frame synchronization mechanism explained in Section 3.4, using the median [Tjoa et al. 2004] algorithm for adjusting the TDMA frames among neighboring nodes.

We deployed a WSN testbed on two floors of the Electrical Engineering department building of the TU/e, which covers an area around $2400m^2$. In total 76 *MyriaNed* nodes are used, from which 60 nodes are static nodes and 16 nodes form four mobile clusters each having four nodes. Figure 10 shows node placement in our testbed. Although there are some (low-quality) direct links between nodes from two floors, we deployed several nodes in the stairways in both ends of the floor to ensure a reliable network connectivity. Four volunteers took part in the experiments to carry cluster nodes. The figure shows the working office of the volunteers where they normally sit and work. They perform their normal movement behavior such as meetings and coffee breaks. Besides that, they were asked to walk through the network (including the other floor) several times during each experiment to ensure a minimum level of cluster mobility. Node s_1 is the sink node and is connected to a laptop. Other static nodes are from 1 (e.g., s_2) to 7 (e.g., s_{60}) hops away from the sink node. Each node transmits one packet per frame.

Performed experiments. We performed four different experiments, each lasting three hours. The limiting factor on the duration of each experiment is the memory capacity of the *MyriaNed* nodes for logging network activities. After each experiment, all nodes are read to gather logged information. The first two experiments are done with only one mobile cluster and without performing listening scheduling. The goal of these experiments was to log hop distance of the mobile cluster to the static nodes (d) as well

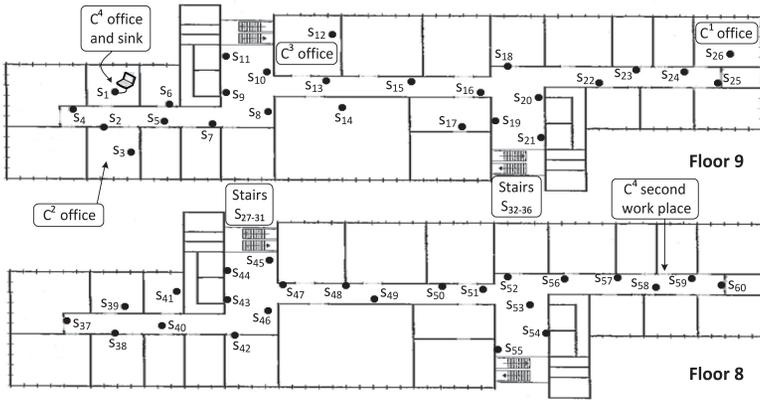


Fig. 10. Node deployment on two floors (8th and 9th) of the EE department of TU/e.

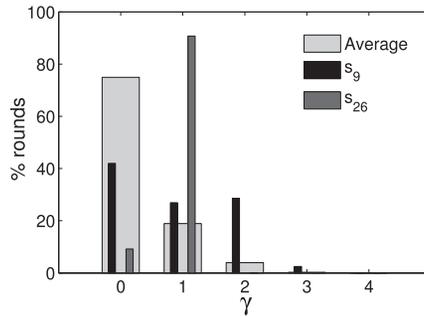


Fig. 11. The distribution of mobile clusters gathering likelihood (γ) in the experiments. The graph shows the average over all static nodes as well as the distributions in nodes s_9 (coffee corner) and s_{26} (C^1 office). For instance, in 73% of rounds, on average, the static nodes are not in the range of any mobile cluster ($\gamma = 0$).

as other network data to explore the trade-offs in the listening scheduling mechanism. The result was discussed in Section 5.3. The latter two experiments were done with four mobile clusters and with the full version of the protocol. Required information is logged in all nodes as well as the laptop connected to the sink to compute different performance metrics.

According to the results in Table I, in a network setup using the MyriaNed nodes, slotted ALOHA performs much better. Thus, we used slotted ALOHA for accessing the MCS slots. Considering the number of clusters in our experiments and their mobility behavior, we set the length of the ALOHA superslots to $N_{AL} = 2$. This value is taken according to the calculations in Table I. In these calculations, we used $\gamma = 2$ as the number of mobile clusters in each other’s interference range. Figure 11 shows the real distribution of the value of γ during our experiments. The vertical axis is the percentage of rounds that static nodes detect γ mobile clusters in their neighborhood. The graph presents the average over all static nodes. In around 75% of rounds, on average, no mobile cluster is in the range of the static nodes. This result confirms the power saving (70.5%) that is obtained by performing listening scheduling (as explained later). Only in less than 4% of rounds are multiple clusters together (and may cause collisions) from which the majority are for two-cluster gatherings. It follows that considering $\gamma = 2$ is a proper design choice. Note that in the cases that a static node is in the range of no

Table II. Average Values of Some Metrics over the Whole Experiments

Metric	static nodes	s_{24}	C^1	C^2	C^3	C^4
1-hop PRR %	-	-	99	98	96	94
DDR %	86	77	71	94	88	88
Latency (second)	27	37	46	14	20	20

cluster or only one mobile cluster ($\gamma < 2$), no collision happens. Thus these cases are not of our interest in the evaluation of the protocol.

Figure 11 also shows the distribution of γ in two nodes as examples of nodes that considerably deviate from the average because of their special locations. Node s_{26} is in the office of cluster C^1 and so in 90% of rounds senses one mobile cluster around. Node s_9 is located in the coffee corner where it is likely that several volunteers gather for some period of time. Thus the number of rounds in which multiple clusters are in its neighborhood is much greater than average.

Packet delivery performance. To investigate the performance of the protocol, we consider various application-level metrics and detailed low-level metrics. The important issue here is that we should properly differentiate the effects caused by the routing protocol and the network deployment (e.g., network coverage and congestion) from the MCMAC performance. Table II presents the achieved one-hop Packet Reception Ratio (PRR) over the whole experiment. The values are the average over the four nodes within each cluster. One-hop PRR for a mobile cluster node is defined as the percentage of the packets that have been received by at least one static node in the neighborhood. This is the main objective of the MCMAC protocol, to efficiently deliver mobile cluster node data to the static network. This metric is independent of the influences of the routing mechanism used in the static network because it only considers whether the packet enters the static part of the network. However, when a certain packet sent by a mobile cluster node is not received by any neighboring static node in the network, this may be caused by several issues, some outside the scope of MCMAC. From the MCMAC side, it may happen due to a wrong listening schedule of the static nodes around the mobile cluster, or because of a collision when several clusters are in each other's range. On the other hand, it may also happen due to other network circumstances such as link variations and interferences, or because the mobile cluster location is outside the coverage of the static network.

The achieved end-to-end Data Delivery Ratio (DDR) and latency (to the sink node) are also presented in Table II as high-level metrics. Although the routing mechanism and congestions in the static network have a very high influence on these performance metrics, they can still provide an impression of the overall behavior of the protocol. For comparison, the table also shows the average values of these metrics over all static nodes as well as node s_{24} which is near the office of cluster C^1 . The achieved metrics show that the DDR and latency of s_{24} is quite close to the values obtained for C^1 (considering the fact that C^1 is mostly one or two hops farther than s_{24} from the sink).

The distribution of the packet losses is also worth considering. Especially the length of bursts of losses is an important factor. It threatens the application-level performance when a cluster node cannot successfully deliver its packets to the static network for a long duration. Figure 12 provides the average and maximum value of the burst packet losses of the mobile clusters. Although a packet loss may be caused by issues other than MCMAC behavior, the achieved values match our expectation. Note that $N_{AL} = 2$; so if all four clusters gather together, the probability of successful transmission for a node in a round is $P_{ALOHA} = 1/8$. Therefore, the achieved maximum burst losses are

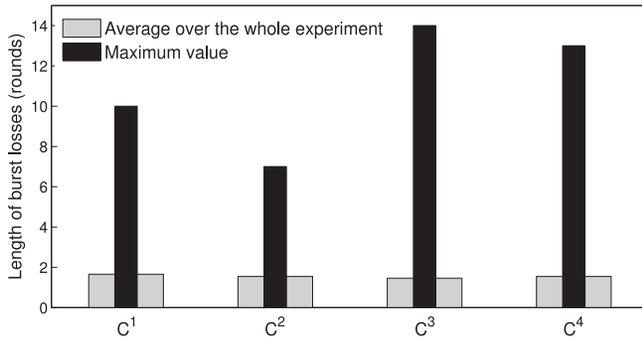


Fig. 12. The length of packet-loss bursts averaged over all nodes in each mobile cluster and over the whole experiment time. The graph also shows the maximum length of burst losses.

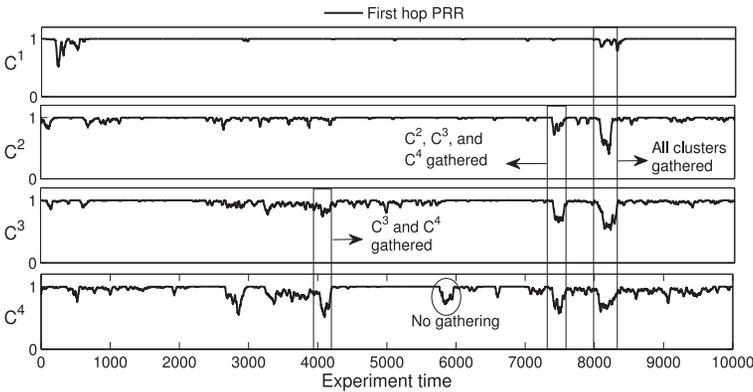


Fig. 13. One-hop PRR of the mobile clusters over a sliding window of size 20 frames.

as expected. Such values allow sufficient retransmission chances for the mobile nodes to deliver their generated data to a static node (considering the data generation period of one every 30 rounds). As Figure 12 shows, the average lengths of packet losses for different clusters are very close (around 1.5 frames). This is because all clusters are in a similar environment taking the whole experiment duration into account.

Figure 13 shows the calculated one-hop PRR over a sliding window of 20 rounds for four mobile clusters. It gives a good view of the network behavior during the experiment time. Several gathering durations of the mobile clusters are pointed out in the figure. The information about gathering clusters is extracted from logged data. For instance, in the last gathering (rounds 8000–8300), all clusters are participating. Clusters C³ and C⁴ started the meeting. Then cluster C² joined. It also leaves the meeting earlier. Cluster C¹ is not fully participating, but joins intermittently. There are also some durations in which the one-hop PRR of a cluster drops, but no cluster gathering takes place. This means that collisions are not the reason. One instance is pointed out in Figure 13 by a circle. Investigating this duration by checking other logged data reveals that cluster nodes have not received any packets from the static nodes either. It shows that the cluster was out of the coverage of the static network during that time.

Listening scheduling behavior. We consider several low-level metrics to evaluate the performance of the listening optimization mechanism. Assume that \mathbf{E} is the set of all TDMA rounds of an experiment, and $\mathcal{L}_i \subseteq \mathbf{E}$ is the set of rounds in which the static

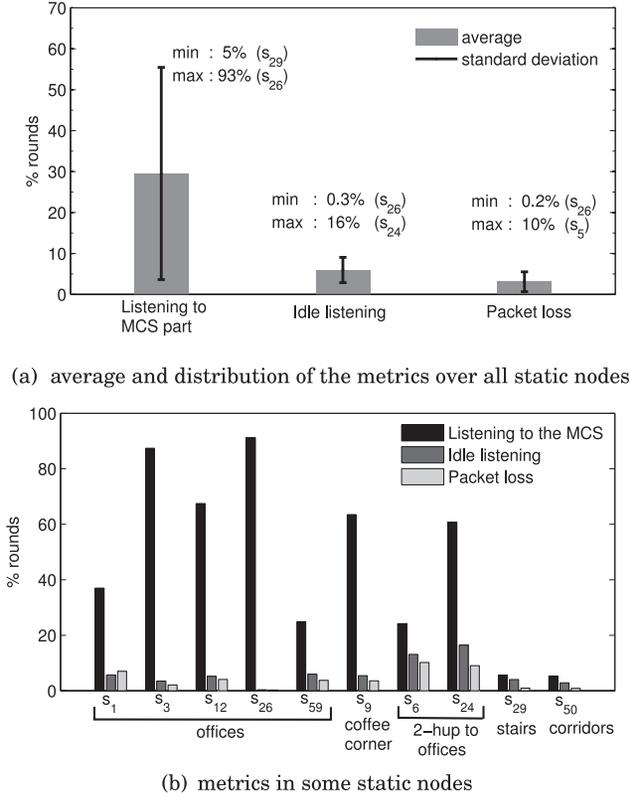


Fig. 14. The performance of the listening scheduling mechanism in the experiments.

node s_i listens to the MCS part of the frame. $\mathcal{L}'_i = \mathbf{E} - \mathcal{L}_i$ is then the set of all rounds in which the static node decides not to listen to the MCS part. Let \mathcal{R}_i be the set of rounds in which s_i is in the range of a mobile cluster and thus is able to receive packets from mobile cluster nodes. Note that, during our experiments, we did not really turn off the receivers of static nodes in MCS slots, in order to be able to compute \mathcal{R}_i . However, in the rounds where the node should not listen to the MCS part (according to the listening mechanism), the received packets are simply ignored. Accordingly, \mathcal{R}'_i are the rounds in which s_i is not in the range of any mobile cluster. These sets are extracted from the logged data for each node. Following are three metrics that we calculate for each node using these sets.

$$\text{listening gain} = \frac{|\mathcal{L}'_i|}{|\mathbf{E}|}, \quad \text{idle listening} = \frac{|\mathcal{L}_i \cap \mathcal{R}'_i|}{|\mathbf{E}|} \quad (14)$$

$$\text{non-listening packet loss} = \frac{|\mathcal{L}'_i \cap \mathcal{R}_i|}{|\mathbf{E}|} \quad (15)$$

Figure 14(a) depicts the overall results regarding these metrics. We set the parameters as $T_{l,max}[2] = 3$, $H = 24$, and $\alpha = 2.3$ to balance between idle listening and packet loss (a Pareto configuration in Figure 9(b)). On average, static nodes have listened to the MCS part of the frame in 29.5% of the rounds during the experiments. This means a listening gain of 70.5% for the static nodes. Average idle listening and packet loss are 6% and 3%, respectively. However, the deviation per static node from

the average listening ratio is considerable. In fact, the achieved listening gain depends on the location of the static nodes. Nodes in or close to the offices of the volunteers (e.g., $s_1, s_2, s_3, s_{12}, s_{26}$) perform the highest rate of listening to the MCS part because there is mostly one cluster in their neighborhood. Moreover, these nodes have the least idle listening. Nodes located in two-hop distance from the offices of the volunteers (e.g., $s_4, s_5, s_6, s_{15}, s_{24}$) have the most idle listening as they mostly anticipate a mobile cluster coming their way and so keep their listening interval low (that is why parameter $T_{l,max}$ [2] is so important). These nodes also have a relatively high packet loss. Nodes that are located in the corridors and stairs and not close to volunteer offices or meeting rooms (e.g., s_{29}, s_{49}, s_{53}) perform the least listening activity.

Figure 14(b) gives the obtained metrics for some selected static nodes in different locations. The volunteer wearing cluster C^4 has two working places (around nodes s_1 and s_{59}) and has a higher movement activity than the other volunteers. The graph shows that cluster C^1 has spent most of the time in its office. That is why node s_{26} has the highest rate of listening to the MCS part among all static nodes.

Comparison with M-LMAC behavior. There is no work in the literature that explicitly supports *cluster* mobility in WSNs. M-LMAC is a state-of-the-art protocol targeting *node* mobility; it is the best reference for comparison with MCMAC, to assess the impact of explicitly taking cluster mobility into account. Because the movement of mobile clusters in the experiments is based on daily activities of the volunteers, different experiments cannot have the exact same mobility patterns. Thus using different experiments to compare the performance of different protocols is hard considering the large statistical variation between similar experiments. We compare the behavior of our protocol with M-LMAC performance by exploiting the logged data from the real experiments and simulating the M-LMAC protocol. This way, the mobility pattern, nodes' neighborhood, and radio link status are the same as those of the real experiments done for MCMAC.

In M-LMAC [van Hoesel and Havinga 2008], communication with mobile nodes is initiated by the static nodes through transmitting an Announcement Message (AM). Mobile nodes that have data to transmit should listen to the channel to find an AM, and then try to send their data by performing carrier sensing. Therefore, the number of opportunities that a mobile node has in each TDMA frame is equal to the number (say ζ) of static nodes in its communication range. We extract this number at each frame for mobile cluster nodes from the logged data. All mobile nodes in a neighborhood compete in these ζ contention-based access periods to send their data. The number of mobile clusters in a neighborhood at each frame is also extracted. Using this information, we calculate the probability of collision-free transmission at each frame for each mobile cluster node. We run the simulations for 10 different sizes of the contention period, as it is an important factor in M-LMAC. Each size is a multiple of the size of one slot.

Figure 15 shows the one-hop PRR for different clusters and various sizes of the contention period using M-LMAC. Comparing this result with the obtained results for MCMAC in Table II (reproduced inside Figure 15) reveals the large gain in PRR obtained by our approach, which avoids collisions between nodes within a cluster. Note that, for the M-LMAC result, it is assumed that once a transmission without collision occurs, the packet will be received by the static nodes. It overestimates the PRR in M-LMAC as, in the reality, other parameters like interferences may further decrease the PRR. Despite that, the gap between the two protocols is significant.

Comparing only the PRR values is not enough, as there might be a trade-off between PRR and energy consumption. To have an impression of the energy efficiency of the protocols, Figure 16 shows the energy consumption of mobile cluster nodes only for transmitting their packets. In M-LMAC, a mobile node may need to retransmit its

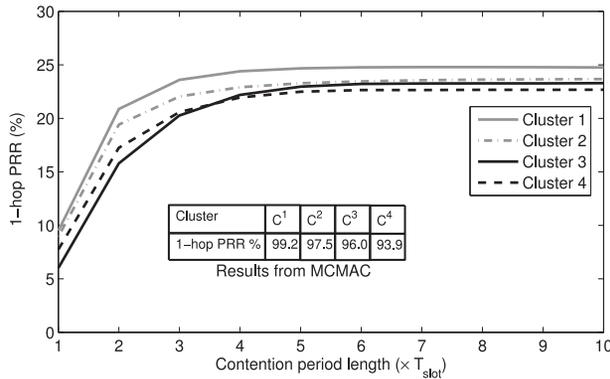


Fig. 15. The overall one-hop PRR for mobile clusters in the M-LMAC protocol using various lengths of the contention period. The table in the figure shows the one-hop PRR achieved in the experiments running the MCMAC protocol (taken from Table II).

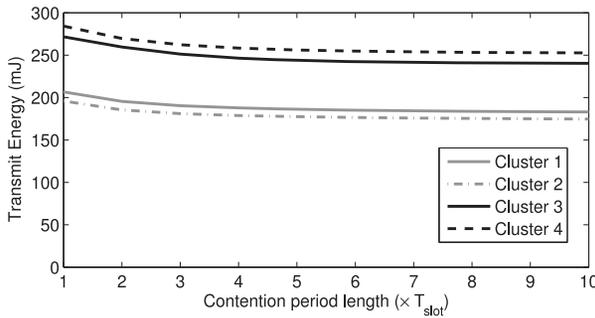


Fig. 16. Transmit energy consumption of mobile cluster nodes during the whole experiment, performing the M-LMAC protocol for various lengths of the contention period. The values are the average over the 4 nodes in each cluster. The transmit energy consumption of the mobile cluster nodes in MCMAC during the experiment is 70mJ.

packet if the previous attempt has not been successful due to a collision with another mobile node. Mobile nodes consume energy for carrier sensing, too. As both M-LMAC and MCMAC (in the CSMA version) have this energy consumption, we do not include it in our comparison. The figure reveals a higher transmit energy consumption for mobile nodes in M-LMAC. Note that we have not considered the energy that the mobile nodes consume in M-LMAC to scan the channel for announcement messages. We assumed that (like in MCMAC) mobile nodes are synchronized with the TDMA framing and know the time slots for each static node. Energy consumption of static nodes for receiving data from mobile cluster nodes is also worth considering. In M-LMAC, it heavily depends on the length of the contention period for each static node. We estimate the number of slots in which a static node listens to receive data from mobile nodes in the M-LMAC protocol. We use the information about the number of mobile nodes around the static nodes in each round, extracted from the experiment's logged data. Figure 17 presents the listening energy consumption of static nodes in the whole experiment to receive mobile nodes' data. In MCMAC, listening energy depends on the size of the MCS part of the frame.

In conclusion, MCMAC provides better packet reception ratios at lower power consumption than M-LMAC (which, as one should remember, is not intended for cluster

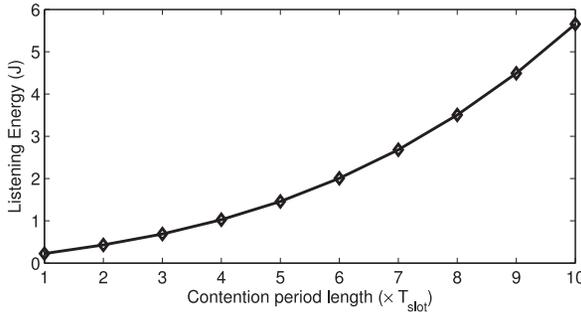


Fig. 17. Average energy consumption of static nodes for listening to mobile nodes during the whole experiment, performing the M-LMAC protocol for various lengths of the contention period. The average listening energy consumption of static nodes in an MCMAC experiment is 541mJ.

mobility). These results confirm that it is useful to provide specific support for cluster mobility in the MAC layer for WSN applications that exhibit this kind of mobility.

6.4. Simulations

We implemented the protocol in the MiXiM [Kopke et al. 2008] framework on top of the OMNeT++ 4 [Varga 2014] simulator. We considered three static network sizes of 36, 64, and 100 nodes. For each network, the static nodes are randomly distributed in a square area. To ensure a fairly even distribution across the area, we placed the nodes with a 10% variance around fixed square grid points (referred to as 6×6 , 8×8 , and 10×10 setups for networks with 36, 64, and 100 nodes, respectively). Because of the gossiping data dissemination that is used on top of the MAC layer, different network sizes cause different network loads. Thus, performance of the protocol is investigated for different data loads. While scaling the number of static nodes, the deployment area was scaled accordingly, such that the density of static nodes was equal for all networks. The static nodes in the corners were assumed to be the sink nodes. We also performed simulations having various numbers of mobile clusters ($N_{mc} = 1, 2, 3, 4$). Each cluster contains five nodes ($N_i = 5$) and uses the mobility model described in Section 3. The parameters for the mobility model were set to mimic human movement. Every node generates a new data item in each round (sampling rate of one second). The cluster nodes use the CSMA scheme to access MCS slots. The length of the contention period is set to $T_{cp} = 1ms$. To have statistically more reliable results, every experiment was repeated 10 times with different seeds for the random generator (with the same mobility pattern) and the shown results are the average over all runs.

Comparing the performance of M-LMAC and MCMAC in the experiments reported in the previous section, we learned that specific cluster mobility support is beneficial for certain applications. A qualitative reasoning says that LMAC may even perform better than M-LMAC in the networks that we are considering. In LMAC, nodes within a cluster will occupy different transmit slots (according to a unique slot assignment in the two-hop neighborhood) and do not collide with each other. We checked this statement hypothesis for one of our network setups by running simulations using both LMAC and M-LMAC. The percentages of transmissions by mobile cluster nodes that failed due to collision were 25.7% and 31.6% for LMAC and M-LMAC, respectively. Because of this, in the simulations of this section, we focus on a pure LMAC scheme for comparison. For each network size we tried the LMAC scheme, our protocol without listening scheduling (MCMAC⁻), and the full version of the MCMAC protocol.

In contrast to the experiments using wireless motes, the collisions can be counted during simulations in each node. This is done by checking if multiple packets have

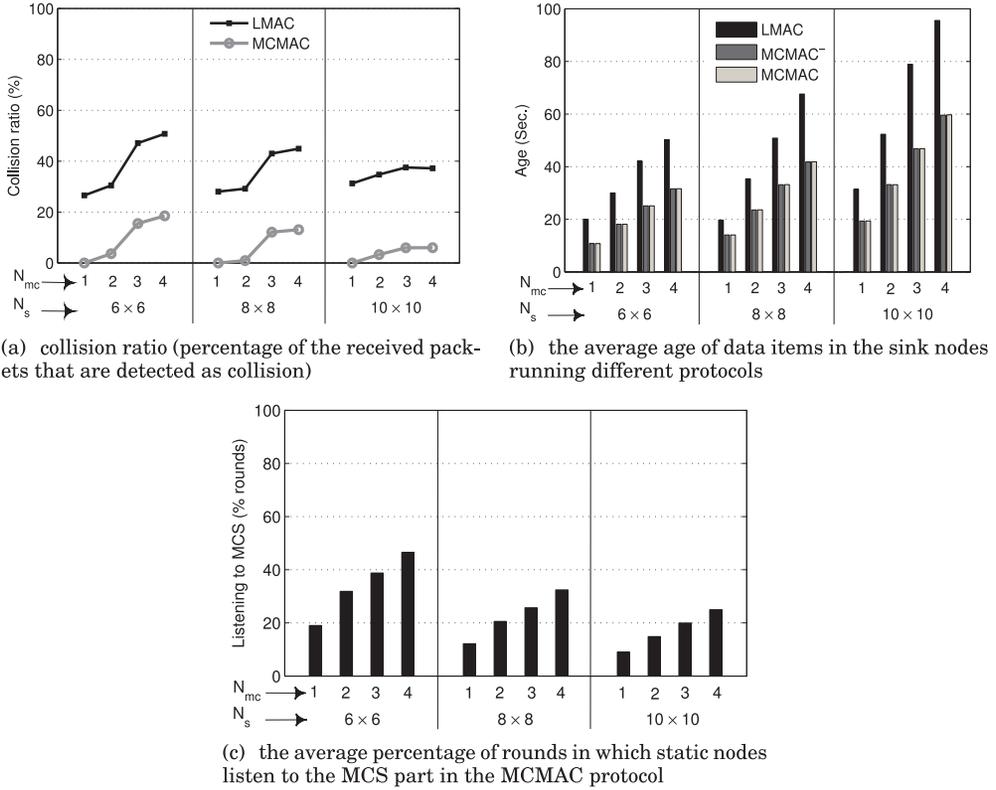


Fig. 18. Simulation results for different network setups and protocols.

arrived in a single slot from different nodes. If so, all packets are ignored for further processing since a collision has happened. Figure 18(a) shows the ratio of collisions detected by the mobile cluster nodes for all combinations of network sizes and cluster numbers. In general, the LMAC principle should provide a contention-free communication. But mobile cluster nodes may collide with static nodes due to their movement. In some runs, using LMAC, the average collision ratio is around 40–50%. There is also a chance of collisions in our MCMAC protocol when there are multiple clusters in a neighborhood, using the shared MCS slots for their data transmission.

We also calculate the application-level QoS metrics of the network. We specifically consider the *age* of data items in the sink node, that can reveal the effect of both latency and delivery ratio. The age of every data item in the sink node at every moment is calculated as the difference between the current simulation time and the sample time attached to the data item. Note that the last arrived data item from each node is maintained in the memory. Figure 18(b) shows the average age of the data items from cluster nodes. The first observation here is that the average age for cluster nodes using LMAC is around 56% higher, on average, than the achieved average age in the MCMAC protocol. The main reason for such higher age is the collisions that happen using LMAC. The second observation in Figure 18(b) is that our listening scheduling mechanism does not exacerbate the age. The obtained values of the metrics are almost the same as in MCMAC⁻ (the protocol without listening scheduling). As the third observation, Figure 18(b) shows the behavior of the protocols when the traffic load of the static nodes changes, which is caused by different sizes of the static network.

From the power consumption point of view, the savings achieved by the listening scheduling mechanism of MCMAC are considerable. The average percentage of rounds with listening to the MCS part over all static nodes is shown in Figure 18(c) for different setups. The listening energy saving varies over different static nodes based on their location. It also depends on the network size and the number of clusters in the network setup. The parameters of the listening scheduling are set as $T_{l,max}[2] = 11$, $H = 15$, and $\alpha = 1.25$ for all setups. These values are selected as a uniform near-optimum configuration for all setups. Static nodes have listened to the MCS part in 30% of experiment rounds on average over all simulated networks. This means that the listening scheduling mechanism decreased 70% of listening to MCS slots, without causing any QoS degradation.

7. DISCUSSION

7.1. Protocol Generalization: Multiple Contention Access Sections

So far in this article, we have assumed only one MCS part in each TDMA frame. It is, however, possible to adapt the protocol for different circumstances by setting the protocol parameters according to Eqs. (3) and (4) for CSMA and slotted ALOHA versions, respectively. The density of the mobile clusters in the deployment area, clusters' gathering likelihood, and data delivery constraints in the application scenario are taken into consideration. To maintain a certain success probability of transmission for mobile cluster nodes, we may increase the length of the contention period (T_{cp}) in CSMA or the number of slots per superslot (N_{AL}) in slotted ALOHA. However, this creates a trade-off between transmission success rate of the mobile cluster nodes and the energy consumption of the static nodes. Increasing these parameters is costly for static nodes as they listen to the MCS part.

Our vision here is to have multiple MCS sections in each frame for crowded networks (many mobile clusters). This way, more energy savings can be achieved because the static nodes can then optimize their listening activities separately for different MCS sections, and each MCS section would be shorter. Let's assume that, in an application, we use N_{MCS} different MCS sections. Mobile clusters are grouped to use specific MCS parts. Different grouping approaches may be applied here. In one approach, an estimation of the gathering likelihood of different clusters, may be used. The clusters that are more likely to gather are assigned to different MCS sections to decrease the chance of interference. A second approach is to classify clusters based on their size. Clusters with similar sizes are then put into one group. The size of the assigned MCS (|MCS|) for a group is set as the maximum cluster size in that group. Note that different MCS sections do not need to have the same length. The last approach can be based on QoS requirements for different clusters. The idea is to have smaller groups for clusters with stringent data delivery constraints. By this we decrease the chance of collisions for the nodes in those clusters. This approach can also be applied together with the first approach. These are the design-time decisions about the number of MCS sections and the length of each one, and the approach for grouping clusters.

With multiple MCS sections, every static node needs to estimate and propagate N_{MCS} different hop distances for scheduling its listening to the different MCS parts. Thus d_i will be the estimated hop distance from the nearest mobile cluster that uses the i^{th} MCS part. A larger N_{MCS} obviously imposes higher communication and computation overhead on the static nodes to estimate corresponding d values. Taking the computation limits of typical nodes in WSNs into account, computation complexity is a considerable issue. In this sense, the number of MCS sections cannot be freely increased and a proper decision should be made. We consider exploring the trade-offs made by such an extension and its performance in dense application scenarios as a future work.

7.2. MCMAC and Frame Synchronization

MCMAC provides efficient cluster mobility support for a low-duty-cycle TDMA-based MAC protocol. Time synchronization is an essential requirement for this protocol. Providing accurate and reliable synchronization for schedule-based communication is an ongoing research topic. This section discusses the influences that node mobility and the MCMAC mechanism may have on the synchronization performance.

As explained in Section 3.4, a decentralized synchronization mechanism is used in the base TDMA MAC protocol. It measures the frame phase error with respect to all sender nodes of which it has received a packet in the last frame. In Assegei [2008], several methods for frame adjustment based on the observed phase errors are investigated. It is pointed out that the synchronization algorithm has a faster convergence and a better stability in static networks, and dynamism such as node mobility increases the synchronization error. Adjusting the frame schedule to the median that can be derived from the observations is simple with low overhead, while a Kalman filter approach is computationally expensive. The evaluation results in Assegei [2008] reveal that the median method works well in a static network, when the guard times in the slots are sufficiently long to compensate small phase drifts. However, the Kalman filter algorithm provides better accuracy and is recommended especially for mobile networks.

The implementation of the MyriaNed nodes that we used in our experiments only supports the median method. This is the reason that a rather long guard time ($T_{guard} = 300\mu s$) is used in slots. The mechanism is applied in both the static and the mobile cluster nodes. It worked well in our experiment as the network never went out of synchronization during the experiments. Note that the Kalman filter could provide the same performance with shorter guard times, which leads to shorter slots and thus energy saving for the nodes. Results in Assegei [2008] show that guard times can be shortened to about half of the size for the median method, approximately $150\mu s$.

A possible improvement of the synchronization method that could be investigated in the future is the following. In the MCMAC protocol we know which nodes are static and which are part of mobile clusters. This allows us to avoid degradation of synchronization accuracy caused by cluster mobility, and to facilitate resynchronization of mobile clusters that have been isolated from the network for a while and want to join the network. Nodes can detect if the sender of a received packet is a static node or a mobile cluster node. The static nodes only synchronize with the other static nodes in their neighborhood. This way, the synchronization of the static nodes is not affected by larger phase errors in mobile cluster nodes and achieves the synchronization performance of a static network. The mobile cluster nodes, on the other hand, preferably synchronize with the static network when they are in the coverage area of the static network. Otherwise, they synchronize with the other mobile nodes in their cluster, if they need to communicate with each other in the running application. An efficient channel search mechanism is then necessary for mobile cluster nodes to be performed when they do not receive any packets from the static network for a while. In this way, mobile cluster nodes use the stable synchronization of the static network as a reference.

8. CONCLUSION

This article presents a protocol for supporting cluster mobility for TDMA-based Medium Access Control (MAC) protocols in Wireless Sensor Networks (WSNs). A network architecture consisting of (mostly) static nodes and several mobile clusters is envisioned. Health monitoring including wireless body area networks (WBANs) is a fitting application. The protocol dedicates a separate part in each TDMA frame to be used by sensor nodes in the mobile clusters. A scheduling mechanism is exploited by static nodes for collision-free communication while mobile cluster nodes access the shared part using a contention-based mechanism. CSMA and slotted ALOHA are considered as two

different paradigms for accessing the shared slots by the mobile cluster nodes. Guidelines for using the proper approach for a certain deployment and setup are provided. The behavior of the protocol is observed by performing extensive simulations with various network sizes and several real-world large-scale experiments using wireless motes.

A listening scheduling mechanism is proposed to avoid idle listening of the static nodes to the mobile clusters when they are not in the range of any mobile cluster. This is done by estimating the hop distance from the nearest mobile cluster and scheduling the listening activities to the mobile cluster slots. The evaluations show that this mechanism can substantially decrease the energy consumption due to idle listening to the mobile cluster slots, depending on the network setup and mobile cluster density. The measured energy saving is around 70% on average in our experiments. Moreover, this mechanism for reducing idle listening does not have a negative influence on the performance of the network.

ACKNOWLEDGMENTS

The authors would like to thank the volunteers who took part in the experiments. The authors also thank the anonymous reviewers for their constructive comments.

REFERENCES

- M. Ali, T. Suleman, and Z. Uzmi. 2005. MMAC: A mobility-adaptive, collision-free MAC protocol for wireless sensor networks. In *Proceedings of the 24th IEEE International Performance, Computing, and Communications Conference (IPCCC'05)*. 401–407.
- F. Assegei. 2008. Decentralized frame synchronization of a TDMA-based wireless sensor network. M.S. thesis, Eindhoven University of Technology, the Netherlands.
- M. Buettner, G. Yee, E. Anderson, and R. Han. 2006. X-MAC: A short preamble MAC protocol for duty-cycled wireless sensor networks. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys'06)*. ACM Press, New York, 307–320.
- S. Cai, Y. Liu, and W. Gong. 2003. Analysis of an AIMD-based collision avoidance protocol in wireless data networks. In *Proceedings of the 42nd IEEE Conference on Decision and Control*. 104–109.
- W.-P. Chen, J. Wei-Peng, and L. Sha. 2004. Dynamic clustering for acoustic target tracking in wireless sensor networks. *IEEE Trans. Mobile Comput.* 3, 258–271.
- T. Dam and K. Langendoen. 2003. An adaptive energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys'03)*. ACM Press, New York, 171–180.
- A. El-Hoiydi and J.-D. Decotignie. 2004. WiseMAC: An ultra low power MAC protocol for the downlink of infrastructure wireless sensor networks. In *Proceedings of the 9th International Symposium on Computers and Communications (ISCC'04)*. 244–251.
- A. Ephremides and O. Mowafi. 1982. Analysis of a hybrid access scheme for buffered users-probabilistic time division. *IEEE Trans. Softw. Engin. SE-8*, 1, 52–61.
- S. Ergen and P. Varaiya. 2006. PEDAMACS: Power efficient and delay aware medium access protocol for sensor networks. *IEEE Trans. Mobile Comput.* 5, 7, 920–930.
- F. Ferrari, M. Zimmerling, L. Mottola, and L. Thiele. 2012. Low-power wireless bus. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys'12)*. ACM Press, New York, 1–14.
- D. Gavidia and M. van Steen. 2008. A probabilistic replication and storage scheme for large wireless networks of small devices. In *Proceedings of the 5th IEEE International Conference Mobile and Ad Hoc Sensor Systems (MASS'08)*. 469–476.
- J. K. W. Heinzelman and H. Balakrishnan. 2002. Negotiation-based protocols for disseminating information in wireless sensor networks. In *Wireless Networks*, Vol. 8, Kluwer Academic Publishers, 169–185.
- W. Heinzelman, A. Chandrakasan, and H. Balakrishnan. 2002. An application specific protocol architecture for wireless microsensor networks. *IEEE Trans. Wirel. Comm.* 1, 1, 660–670.
- X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. 1999. A group mobility model for ad hoc wireless networks. In *Proceedings of the 2nd ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM'99)*. ACM Press, New York, 53–60.
- C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. 2003. Directed diffusion for wireless sensor networking. *IEEE/ACM Trans. Netw.* 11, 1, 2–16.

- A. Jhumka and S. Kulkarni. 2007. On the design of mobility-tolerant TDMA-based media access control (MAC) protocol for mobile sensor networks. In *Proceedings of the 4th International Conference on Distributed Computing and Internet Technology (ICDCIT'07)*. Springer, 42–53.
- A. Kopke, M. Swigulski, K. Wessel, D. Willkomm, P. L. Haneveld, T. Parker, O. Visser, H. Lichte, and S. Valentin. 2008. Simulating wireless and mobile networks in OMNET++ - The MIXIM vision. In *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems and Workshops (SIMUTools'08)*.
- K. Langendoen and A. Meier. 2010. Analyzing MAC protocols for low data-rate applications. *ACM Trans. Sensor Netw.* 7, 1, 10:1–10:34.
- C. Liu and E. Modiano. 2005. On the performance of additive increase multiplicative decrease AIMD protocols in hybrid space-terrestrial networks. *Comput. Netw. ISDN Syst.* 47, 5, 661–678.
- M. Nabi, M. Blagojevic, M. Geilen, T. Basten, and T. Hendriks. 2010. MCMAC: An optimized medium access control protocol for mobile clusters in wireless sensor networks. In *Proceedings of the 7th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks (SECON'10)*. 28–36.
- Nordic Semiconductor. 2007. nRF24L01 Radio transceiver. <http://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P>.
- J. Polastre, J. Hill, and D. Culler. 2004. Versatile low power media access for wireless sensor networks. In *Proceedings of the ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*. ACM Press, New York, 95–107.
- V. Rajendran, K. Obraczka, and J. Garcia-Luna-Aceves. 2003. Energy-efficient collision-free medium access control for wireless sensor networks. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys'03)*. ACM Press, New York, 181–192.
- I. Rhee, A. Warriar, M. Aia, J. Min, and M. Sichitiu. 2008. Z-mac: A hybrid MAC for wireless sensor networks. *IEEE/ACM Trans. Netw.* 16, 3, 511–524.
- I. Rhee, A. Warriar, J. Min, and L. Xu. 2009. DRAND: Distributed randomized TDMA scheduling for wireless ad hoc networks. *IEEE Trans. Mobile Comput.* 8, 10, 1384–1396.
- Texas Instruments. 2010. cc2420 radio transceiver. <http://focus.ti.com/docs/prod/folders/print/cc2420.html>.
- R. Tjoa, K. Chee, P. Sivaprasad, S. Rao, and J. Lim. 2004. Clock drift reduction for relative time slot TDMA-based sensor networks. In *Proceedings of the IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'04)*. Vol. 2, 1042–1047.
- F. van der Wateren. 2010. MyriaCore implementation details, the inside of myriaCore and gMAC. Tech. rep., Chess Company, the Netherlands.
- F. van der Wateren. 2011. Myrianed development guidelines, the art of developing WSN applications with MyriaNed. Tech. rep., Chess Company, the Netherlands.
- L. van Hoesel and P. Havinga. 2004. A lightweight medium access protocol (LMAC) for wireless sensor networks: Reducing preamble transmissions and transceiver state switches. In *Proceedings of the International Workshop on Networked Sensing Systems (INSS'04)*. Society of Instrument and Control Engineers, 205–208.
- L. van Hoesel and P. Havinga. 2005. Collision-free time slot reuse in multi-hop wireless sensor networks. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'05)*. IEEE, 101–107.
- L. van Hoesel and P. Havinga. 2008. Ideas on node mobility support in schedule-based medium access. In *Proceedings of the International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'08)*. IEEE, 539–544.
- A. Varga. 2014. OMNeT++. <http://www.omnetpp.org>.
- G. Welch and G. Bishop. 1995. An introduction to the Kalman filter. Tech. rep TR 95-041. http://www.cs.unc.edu/~welch/media/pdf/kalman_intro.pdf.
- W. Ye, J. Heidemann, and D. Estrin. 2002. An energy-efficient MAC protocol for wireless sensor networks. In *Proceedings of the Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM'02)*. IEEE, 1567–1576.
- W. Ye, F. Silva, and J. Heidemann. 2006. Ultra-low duty cycle MAC with scheduled channel polling. In *Proceedings of the Conference on Embedded Networked Sensor Systems (SenSys'06)*. ACM Press, New York, 321–334.
- M. Zonoozi and P. Dassanayake. 1977. User mobility modeling and characterization of mobility patterns. *IEEE J. Selected Areas Comm.* 15, 7, 1239–1252.

Received September 2012; revised July 2013; accepted October 2013