Mobility in Wireless Sensor Networks — Survey and Proposal

Ricardo Silva*, Jorge Sa Silva, Fernando Boavida

Department of Informatics Engineering, University of Coimbra, Polo II - Pinhal de Marrocos 3030-290, Coimbra, PORTUGAL

Abstract

Targeting an increasing number of potential application domains, wireless sensor networks (WSN) have been the subject of intense research, in an attempt to optimize their performance while guaranteeing reliability in highly demanding scenarios. However, hardware constraints have limited their application, and real deployments have demonstrated that WSNs have difficulties in coping with complex communication tasks — such as mobility — in addition to application-related tasks. Mobility support in WSNs is crucial for a very high percentage of application scenarios and, most notably, for the Internet of Things. It is, thus, important to know the existing solutions for mobility in WSNs, identifying their main characteristics and limitations. With this in mind, we firstly present a survey of models for mobility support in WSNs. We then present the Network of Proxies (NoP) assisted mobility proposal, which relieves resource-constrained WSN nodes from the heavy procedures inherent to mobility management. The presented proposal was implemented and evaluated in a real platform, demonstrating not only its advantages over conventional solutions, but also its very good performance in the simultaneous handling of several mobile nodes, leading to high handoff success rate and low handoff time.

Keywords: wireless sensor networks, mobility, 6LoWPAN, proxies

1. Introduction

Wireless Sensor Networks research has intensely addressed performance, reliability and capacity optimization, in an attempt to shorten the gap that separates them from conventional networks. However, WSNs are largely constituted by resource-constrained devices, whose characteristics are still far from those required by most applications. Advanced routing algorithms, neighbor and service discovery mechanisms, security, mobility and debugging, among others, are just examples of features that researchers are trying to implement in WSNs. Even though it is possible to install and evaluate them individually, the integration of all of these features with the aim of developing a reliable, complete system will, on one hand, limit the algorithms' complexity due to ROM and RAM restrictions and, on the other hand, contribute to a decrease in the lifetime of each mote due to added energy requirements.

While working on the GINSENG project [1], an European project whose main objective was the deployment of performance-controlled WSNs in critical scenarios, we faced this problem when we tried to include all features we considered fundamental in a real, deployed WSN, whose target was Petrogal's oil refinery in Sines, Portugal. In this case, the adopted solution was to remove some features and simplify the software installed in each mote, in order to still achieve the necessary performance without negatively affecting the network lifetime.

In [2] the authors also arrived at a similar conclusion, namely that motes must be relieved from the routing process and must become as simple as possible, acting just as end nodes and delegating routing procedures on more powerful entities. Basically, the authors advocate the separation of the sensing activity from the network operation activity. A similar line was taken in [3], in which the authors study the enhancement of mobile networks by adding infrastructure support, concluding that, in general, this kind of support is highly beneficial when mobility is concerned.

Since WSN nodes are frequently small, portable devices, which can be easily coupled to mobile entities such as vehicles or people, many applications require mobility support. Therefore, it is crucial to support efficient mobility mechanisms in WSNs, without compromising the main application operation and network lifetime.

Mobility in WSNs has been approached from several perspectives and targeting different goals, leading to a variety of solutions. In the first part of this paper we propose a WSN mobility classification and survey the main existing mobility approaches. This not only provides a broad view of the field, but also allows the reader to identify the potential and implications of the various options where mobility is concerned, constituting one of two main contributions of the paper.

Given the problems and limitations of the various mobility solutions, identified in the first part of the paper, in its second part we present and evaluate a WSN mobility support proposal, called Network of Proxies (NoP), designed to perform complex, time-consuming, processor-intensive and energy-demanding operations, such as mobility management operations, on behalf

*Corresponding author. tel: +351 962476436 fax: +351 239 701266

Email addresses: rnsilva@dei.uc.pt (Ricardo Silva), sasilva@dei.uc.pt (Jorge Sa Silva), boavida@dei.uc.pt (Fernando Boavida)

of WSN nodes. The Network of Proxies concept was originally proposed in [4] and [5], where we concluded that conventional node-based mobility solutions, such as MIPv6, could not meet the requirements of many WSN applications in terms of reliability and overall performance. NoP was then designed to overcome the problem, guaranteeing controlled end-to-end performance in the presence of high mobility while contributing to an extension of the WSN's lifetime. The entire NoP development process, assessment and final comparison with MIPv6 collectively constitute the second main contribution of this paper.

NoP's objective is to simplify the sensor network, moving the complexity from the motes to local proxies. These proposed proxies are machines without the stringent energy restrictions of sensor nodes, and with the ability to operate alone or to be part of a mesh network. They should be capable of monitoring each mote's link quality and determining when handoff should be done, taking care of it on behalf of mobile nodes. In this way, it is possible to keep mobile nodes as simple as possible, focusing their activity on sensing, and saving energy.

NoP was specifically designed for critical scenarios, such as GINSENG, in which the extra cost of a wireless mesh network infrastructure is largely compensated by the added reliability and performance control of the resulting system.

Although we are dealing with mobility support in this paper, the NoP concept can be used to support any other activity whose complexity level requires more powerful mechanisms, such as security.

This paper is organized as follows. The next section presents the general characterization of mobility in WSNs. Section 3 surveys WSN mobility support at the MAC layer, while section 4 surveys it at the Network layer. Section 5 details the concept of Network of Proxies and its application to mobility support in WSNs, presenting an overview of implementation and operation aspects and concluding with a presentation and discussion of the NoP's evaluation results. Section 6 surveys important, related projects in this research field. The conclusions and guidelines for further work are provided in section 7.

2. General characterization of mobility in WSNs

Mobility in wireless sensor networks can be classified considering the following aspects: the element that is mobile; the type of movement; the protocol level at which mobility is supported; and the entity who handles the mobility process. While the former two concern the physical aspects of mobility, the latter two regard the architectural aspects. The following subsections detail each of them.

2.1. Mobile element

Table 1 summarizes the mobility characterization in what concerns the WSN element that is mobile. As it can be seen in the table, two cases can occur: mobility of the sink node, and mobility of the sensor node.

Sink node mobility was introduced in [6] and [7], among others, with the objective of making sink nodes closer to each sensor node or sensor node cluster, in order to save the nodes' energy. A second objective was to avoid the high cost of maintaining long multi-hop paths.

Table 1: Mobile element

Sink node	Mobile Base Stations (MSB) Mobile Data Collectors (MDC) Rendezvous (Hybrid)
Sensor node	Weak Strong robotic Strong parasitic

Three classes of sink node mobility exist: Mobile Base Stations (MBS), Mobile Data Collectors (MDC), and Rendezvous-Based solutions (which is a hybrid of the former two classes).

With Mobile Base Stations the sink node is capable of moving across the network, increasing the coverage and decreasing the number of hops to reach each node. Reference [8] evaluates sink node mobility performance for various network topologies and types of movement.

Mobile Data Collectors (MDC), in turn, takes advantage of the capability of more powerful nodes (either sink nodes or other dedicated nodes) to perform on-demand collection, avoiding the need for data to travel through several hops. [9] introduced the concept of data mules, where mobile sink nodes move randomly, collecting data across the network. [10] proposed a solution where the trajectory of the Mobile Data Collector is not controlled but is known a priori, while [11] proposed a controlled MDC in real-time.

Rendezvous-Based solutions are a hybrid of the two previous classes of solutions: MBS and MDC [12]. Instead of uncontrolled mobility or on demand data gathering, [13] proposed a careful mobility/positioning of the sink node in order to better cover the network. The same author also introduced the concept of dynamically changing position, readapting to network changes.

Sensor node mobility can be classified into two basic modes [14]: weak mobility and strong mobility.

Weak mobility is the mobility forced by the death of some network nodes. Due to their intrinsic characteristics, namely hardware restrictions and battery operation, nodes have limited, often short lifetime. Consequently, new nodes must be added to replace dead nodes, thus leading to network topology changes.

Strong mobility, in turn, is the type of mobility associated with the movement caused by either an external agent (wind or water) or by an intrinsic characteristic of the sensor node. Strong mobility can be further subdivided into robotic and parasitic. In the former case, the sensor node has the capacity to move on its own. In the latter case, it is attached to a moving entity.

An example of robotic node mobility is Robomote [15], a wheel-equipped sensor node designed for easy deployment and low cost. Robomote was also equipped with two engines, one infrared sensor to detect obstacles and a sun-rechargeable battery. Despite the interest in and potential of Robomote, most existing applications are based on nodes attached to mobile bodies, i.e., on parasitic sensor node mobility. In [16] this issue is analyzed in depth, using various types of parasitism to classify the possible forms of association between motes and mobile bodies.

2.2. Types of movement

Mobility in WSNs can also be classified according to the type of movement of the moving entity. The following types are commonly referred to in the literature: random, predefined, and controlled. Random movement means that the moving entity (be it a sink node or a sensor node) moves randomly within the area under consideration. Predefined movement means that the entity moves along a specific path, with known speed, reaching each point of interest at known, specific points in time. Lastly, controlled movement means that the entity's movement is controlled by an external entity in real-time.
Independently of the type of movement, [17] and [18] studied node mobility as a way to increase the network coverage, and presented several energy-aware routing algorithms oriented towards the maximization of the covered area.

2.3. Protocol level

In what concerns the protocol level, WSN mobility can be handled at the medium access control (MAC) (sub-)layer or at the network layer.
When dealt with at the MAC layer, the challenge is to efficiently integrate mobility functionality in the already complex duty cycle mechanisms, keeping energy consumption and latency as low as possible. Although it is possible to support mobility at the MAC layer, there are several problems in doing so. On one hand, as mentioned before, the complexity of duty cycle algorithms that are tailored for latency and energy reduction is a serious obstacle to modifications required by mobility. On the other hand, there are critical situations where high performance and reliability are a must, which means that the MAC protocol must be optimized for these goals, and adding mobility support may compromise them. Hence, the alternative is to implement mobility at the network layer, in order to leave the MAC layer free from the added complexity.

One key aspect of WSN MAC layer solutions is their duty cycle, that is, the scheme that determines when sensor nodes should be awake or sleeping. The following aspects are relevant for the classification/analysis of duty cycle schemes not only in what concerns mobility, but also in general:

- **type of duty cycle** — duty cycles may be fixed, adaptive, or dynamic; in fixed duty cycles, wake-up and sleeping times do not vary and follow a predetermined order; adaptive schemes use a predetermined order but take into account the state of the network and/or application to determine the duration of awake/sleep periods; in dynamic duty cycles the time and duration of awake/sleep periods is neither fixed nor predetermined;

- **synchronization** — this controls the sensor nodes wake-up periods, so that nodes involved in a communication are active simultaneously; some schemes rely on some form of node synchronization; other schemes are asynchronous or hybrid, resorting to the use of preambles for node wake-up;

- **latency** — this is the maximum time a node must wait before initiating a transmission; this includes the time for waiting for the destination node to wake up; different duty cycle schemes have different impact on latency;

- **energy consumption** — intrinsic characteristics of duty cycles schemes (e.g., type of duty cycle, synchronization) lead to different energy consumption;

- **mobility awareness** — some duty cycle schemes are mobility-aware, meaning that they were specifically designed having mobility in mind; other do not take mobility into account.

When dealing with mobility at the network layer, the following aspects are relevant for the classification/analysis of WSN mobility solutions:

- **IP-based** — although operating at the network layer, some mobility solutions may not rely on IP mobility and, instead, use Zigbee or WirelessHart; other solutions may be based on IP mobility;

- **LLN awareness** — some mobility schemes are specifically designed having in mind the restrictions of Low power and Lossy Networks (LLN); others, like MIPv6-based solutions, do not take these restrictions into account;

- **latency** — mobility mechanisms always lead to communication latency, due to the handoff procedure; the lower the latency the more efficient is the solution;

- **signaling cost** — different mobility solutions use different mechanisms, which require signaling protocols/messages; signaling is overhead and should be minimized.

Wireless Sensor Networks and mobility have also appeared associated to indoor localization methods. For instance, in [19] [20] the authors present some localization solutions that explore well-known methods like references, signal strengths or triangulation, taking advantage of deployed WSNs to track entities. Localization methods can be used to detect or track mobile sink nodes, sensor nodes or users.

2.4. Mobility handler

One final aspect to take into account when considering the support for mobility in wireless sensor networks is the entity that handles the mobility procedures. As mentioned in the Introduction, an alternative to burdening sensor nodes with mobility functionality is to resort to some kind of infrastructure that handles mobility on their behalf. The foreseen possibilities are, thus, the following:

- **node-based mobility** — in this case, mobility is handled by the sensor nodes, in addition to other communication and application tasks; this does not require any assistance from the network side, at the cost of some negative impact on node complexity, performance, and energy consumption;
- **network-based mobility** — sensor nodes are relieved from mobility-related tasks (e.g., handoff, registration, route-optimization, etc.), which are performed on their behalf by an external entity, such as special network nodes or mobility agents; these entities either reside in the existing network infrastructure or in an additional, complementary infrastructure; the added complexity on the infrastructure side is compensated by lighter nodes, with benefits in terms of performance and energy consumption;
- **hybrid** — in hybrid solutions, mobility functions are partly carried out by sensor nodes and partly by the network infrastructure.

2.5. Sum up

In light of the presented mobility characterization framework, Table 2 presents a classification of some WSN mobility approaches that can be found in the literature. The common factor of all these solutions is that they approach mobility from an application perspective, regardless the complexity of implementing the needed communication support.

In contrast, in the next two sections we will address WSN mobility solutions from the communications perspective. Section 3 deals with WSN mobility solutions at the MAC layer, while section 4 addresses WSN mobility at the network layer.

3. Mobility at the MAC layer

MAC layer duty cycle is a fundamental aspect of any wireless sensor network, as it determines many of its features including its ability to support mobility. This section presents the main WSN duty cycle schemes and their impact on mobility support.

Duty cycle schemes have the main goals of saving node's energy, in order to increase the network lifetime, and enabling the nodes to communicate efficiently, notably with reduced overhead and latency. As seen in the previous section, there are two basic approaches for this: synchronous operation or asynchronous operation.
Synchronous duty cycle schemes require clock synchronization among all network elements belonging to the same cluster. The clock is used to wake up all nodes at specific, synchronous points in time and for specific periods. After this, the nodes revert to sleep mode until the following active period. This form of operation additionally requires several synchronization messages or synchronization flags and forces edge nodes to synchronize with the various clusters/networks to which they are attached.
Asynchronous mechanisms are typically based on preambles, long or short, that precede transmissions. Preambles constitute overhead, and so the price to pay for not relying on clock synchronization is a decreased efficiency in data transmission, i.e., more bits are transmitted to send the same amount of data.

Several proposals for WSN mobility support at the MAC layer were built on the adaptation of existing MAC layer solutions designed for static networks. Being one of the first MAC duty cycle schemes, Sensor MAC (S-MAC) [21] is the basis for many of the WSN mobility approaches described in this section. S-MAC is a synchronous duty cycle solution in which nodes are grouped into clusters. All cluster nodes switch to passive and active modes at the same time. The active mode is divided into three phases (SYNC, RTS, CTS). During each of these active phases nodes contend for the medium using a carrier sense multiple access (CSMA) scheme. The algorithm relies on the broadcasting of a synchronization message (SYNC) in order to synchronize the internal clock of each sensor node. All nodes broadcast SYNC messages periodically and when a node wants to communicate with another node, it sends a Request to Send (RTS) packet and waits for the respective Clear to Send (CTS) response. The duty cycle size is fixed and, thus, it does not adapt to the network load, which can cause an increase in latency when the load is high, and a non-optimal amount of transmitted data between nodes.

Based on S-MAC, several approaches to WSN mobility support were developed. Mobile S-MAC, MOB-MAC, AM-MAC, and MD-MAC are the most relevant ones, and their main characteristics are presented below.

Mobile S-MAC (MS-MAC) [22] was the first WSN MAC layer proposal that considered mobility and it is an extension of the base S-MAC protocol with the aim of supporting both stationary and mobile networks. With MS-MAC, when a mobile node enters a stationary network, the surrounding nodes within a range area, R, will form an active space with synchronized periods shorter than those of stationary operation. These shorter periods are useful for monitoring the movement of the new neighbor. MS-MAC was only tested by simulation and, as it is based on S-MAC, suffers of the same problems. MS-MAC defines a shorter Neighbor Discovery Period (NDP) in the mobile nodes, when compared to S-MAC. Moreover, it increases the complexity of sensor nodes by using the link quality variable for movement detection. It also requires an extra listening time for neighbors, mainly for the boundary nodes, in order to detect mobile nodes.

Mobile MAC (MOB-MAC) [23] was developed in order to solve the MS-MAC's retransmission problem, caused by the high frame loss rate, which increases energy consumption. MOB-MAC solves this problem by introducing adaptive frame sizes, considering the link quality. If the link quality is poor then the frame size is reduced, thus leading to less energy consumption per transmitted frame and decreasing the error probability, which is proportional to the frame size. Conversely, when link quality improves, the frame size gets increased.

Table 2: Classification of WSN mobility approaches.

	Mobile element		Type of movement	Protocol level	Mobility handler
Wang et al. 2005 [6]	Sink node	MBS	Predefined	n/a	Network
Kim et al. 2003 [7]	Sink node	MBS	Random	n/a	Network
Stevanovic and Vlajic 2008 [8]	Sink node	MBS	Random/predefined	MAC	Network
Shah et al. 2003 [9]	Sink node	MDC	Random	n/a	Network
Chakrabarti et al. 2003 [10]	Sensor node	Parasitic	Predefined	n/a	Node
Somasundara et al. 2004 [11]	Sink node	MDC	Controlled	n/a	Network
Ekici et al. 2006 [12]	Sink node	Hybrid	Controlled	n/a	Network
Akkaya et al. 2007 [13]	Sink node	MBS	Controlled	n/a	Network
Dantu et al. 2005 [15]	Sensor node	Robotic	Controlled	MAC	Node
Srinivasan and Chua 2007 [17]	Sensor node	n/a	Predefined	n/a	Node
Liu et al. 2005 [18]	Sensor node	n/a	Random	n/a	Node

MOB-MAC may have problems when fixed size frames are used, as is the case in most real application scenarios.

Adaptive Mobility MAC (AM-MAC) [24] is a MAC protocol based on S-MAC in which each node has a listen-sleep schedule that it periodically broadcasts through a SYNC message. AM-MAC supports mobility based on two mechanisms: secondary listening period and smart scheduling adaptation. Nodes using the same schedule constitute a virtual cluster. Each virtual cluster has its own schedule. Border nodes are deployed between virtual clusters, and operate according to the schedules of the clusters they belong to. When a mobile node (MN) receives a SYNC message from a border node, whose schedule does not belong to its virtual cluster, the MN learns that it will probably handoff to another virtual cluster, using a secondary listening period to quickly perform the scheduling adaptation. On the other hand, if the MN does not receive a SYNC message of a border node and enters directly in another virtual cluster, it will perform schedule adaptation to the new schedule. Although AM-MAC outperforms S-MAC, it energetically depletes border nodes, like in the MS-MAC solution.

MD-SMAC [25], Mobility aware Dynamic S-MAC, is a combination of MS-MAC and DS-MAC [26], in an attempt to obtain a mobility-aware solution capable of handling delay-sensitive applications, and maintaining low energy consumption. MD-SMAC proposes some changes to the respective base protocols. In the case of DS-MAC, MD-SMAC proposes that once a pre-defined energy level threshold is reached, the mobile node (MN) must revert the duty cycle to its initial format (note that DS-MAC doubles the duty cycle to improve latency), thus improving nodes lifetime. Regarding MS-MAC, MD-SMAC removes the old schedule when the MN is in the next cluster (MS-MAC keeps both schedules by default, which consumes more energy). Besides, MD-SMAC also reverts the Neighbor Discovery frequency to its initial form as soon as the MN is in the new cluster with the new schedule. In MS-MAC this frequency changes from 5 minutes to 30 seconds when the MN is moving between clusters, but it never reverts to the original form. Although this solution improves the network lifetime, it negatively affects latency and, therefore, the support for delay-sensitive applications is compromised.

Collision Free Mobility Adaptive MAC for WSNs (CFMA) [27] is a MAC protocol that outperform S-MAC, MOB-MAC, AM-MAC, MS-MAC, MD-MAC and DS-MAC by modifying the way in which back-off values are allocated. Usually, MAC protocols compute a random value to decide the access period of each node. However, CFMA-MAC proposes the use of pre-defined delays, assigned by the Cluster Coordinator (thus imposing the use a cluster-based network organization), implementing different levels of priority for each mote based on the delay values. For instance, when a new node enters the cluster it has the maximum priority, which means the shortest delay, to assure its quick inclusion in the network. Following each transmission, the node asks the coordinator the delay for the next transmission. In the case of a mobile node (MN), it must also monitor the signal strength of the adjacent clusters. When that signal strength is increasing, the MN asks the coordinator for a delay to access the adjacent cluster.
Thus, when arriving at that cluster the MN can communicate as quickly as the delay allows it, which is faster than requesting the delay only after the movement. CFMA-MAC was analyzed and compared with the mentioned MAC layer protocols via MATLAB only. Although the results show it to be superior, neither implementation nor evaluation exists in real platforms. Besides, all the protocols with which CFMA-MAC was compared are based on S-MAC, which does not allow any conclusions on how it will compare to other types of MAC protocols. Furthermore, forcing a cluster-based network organization may not be adequate for some scenarios and thus limits its applicability.

For critical scenarios where CSMA access might cause some uncontrolled performance, TDMA solutions are usually preferred. The Traffic-Adaptive Medium Access protocol (TRAMA) [28] was one of the first TDMA solutions for WSNs and, like S-MAC, it has also served as a base for newer contributions. TRAMA is a TDMA MAC protocol capable of adapting transmission schedules to traffic needs. TRAMA uses not only scheduled slots for transmission but also contention-based slots (CSMA) for node admission and network management.

Based on TRAMA, the Mobility-Adaptive Collision-Free MAC Protocol for Wireless Sensor Networks (MMAC) [29] is a schedule-based MAC protocol capable of adapting the frame time, transmission slots and random-access slots (a frame is composed of scheduled access slots and random access slots). This solution adapts the frame time according to node movement conditions, thus allowing a more efficient synchronization of nodes. With fixed frame time (as in the case of TRAMA), when there are several mobile nodes each node may have to wait a considerable time until it synchronizes and joins the network. However, using adaptive frame time, it is possible to reduce the frame time as a function of the number of mobile nodes, and thus speed up the joining process. The problem of MMAC is the highly complex scheduling algorithm used to calculate the transmitter of each slot in the frame. Besides, MMAC assumes that it is possible to predict each nodes' position, based on their mobility pattern. This assumption may not be valid in many real scenarios, which limits the applicability of this solution.

In an attempt to avoid the need for nodes' synchronization, required in the cases of S-MAC, TRAMA, and its derivatives, several asynchronous MAC solutions were developed. WiseMAC [30], B-MAC [31] and X-MAC [32] are the most popular unsynchronized sender-initiated solutions, the latter currently being one of the most used MAC layers. On the other hand, RI-MAC [33] and A-MAC [34] are well-known unsynchronized receiver-initiated solutions.

Asynchronous solutions resort to the use of *low power listening* (LPL) and to the transmission and reception of preambles in order to avoid the need for clock synchronization. In sender-initiated solutions, preambles are sent by the sender to detect when the receiver is active, following which the communication can start. On the other hand, in receiver-initiated solutions the receivers send a beacon at wakeup. Senders hold in silence until listening the receivers' beacon, starting the transmission only after the beacon's reception.

WiseMAC and B-MAC suffer from the same problem: the overhearing caused by long preambles, which affects the entire network. In contrast, X-MAC replaces the long preamble by a sequence of short preambles and introduces an acknowledge message from receiver to sender, in order to notify the latter that the preamble was already received and therefore the transmission can start. Thus, there is no need for both entities to wait for the preamble to end. Moreover, in order to avoid neighbors to become affected by preambles of messages targeting other destinations, each short preamble contains the receiver's ID. With this type of preambles, neighbors that wake up and receive a short preamble will check the ID to see whether they are the intended receivers. If so, they will send an acknowledgement message, the sender will stop the preambles and subsequently sends the packet. Otherwise they just go to sleep again.

Both RI-MAC and A-MAC are affected by the long duty cycle of senders, with A-MAC minimizing this issue by introducing a control channel and a multi-channel mechanism for data transmission. However, A-MAC uses a single control channel that can itself suffer from interference and traffic jam, thus compromising data transmission. EM-MAC [35] proposed a similar multi-channel solution that overcomes the A-MAC limitation by creating a dynamic channel selection also for the control channel. In EM-MAC, each node has its own channel switching and wakeup scheme, which is predicted by the sender based on its knowledge of the receiver's function used to generate channel and wakeup times.

Although these solutions were not specifically designed for mobility support, the fact of not requiring synchronization makes them more flexible, which eases the addition of mobile nodes to the network. Benefiting from this, an adaptation of X-MAC for mobility support was developed, named MoX-MAC [36]. In X-MAC, whenever a node wants to transmit it sends a set of short preambles until it receives an acknowledgement notifying that the target is awake and the transmission can start. Applying X-MAC to mobile nodes would affect a potentially large number of nodes, namely those along the MN's path. Therefore, with MoX-MAC MNs first listen to any ACK to check whether there is a transmission going on. By learning the schedule through the received ACKs, MNs know when they can start sending data. MoX-MAC was compared with X-MAC, but the results show a slight advantage only in some cases.

In an attempt to profit from the positive aspects of synchronous and asynchronous MAC solutions, some hybrid solutions were developed. These mix synchronized nodes in the network core with unsynchronized nodes at network boundaries. MAMAC and MMH-MAC, described next, are two relevant hybrid synchronization proposals developed with mobility support in mind.

The Mobile Adaptive MAC (MAMAC) [37] proposes a simple mechanism where nodes, either mobile or fixed, do not use synchronized clocks. Instead, each node wakes up at random points in time and sends an acknowledgement beacon. When a node wants to transmit, it starts listening to the environment until it receives the acknowledgement beacon from the destination node, at which point the node starts the transmission. In what concerns channel occupation, this protocol is very efficient because each transmission only requires the ACK message and the data transmission, instead of RTS and CTS used, for instance, in S-MAC. According to its authors, MAMAC is an effective solution for high mobility scenarios, although they do not provide any evaluation results supporting this claim.

Also a hybrid solution, the Mobile Multimode Hybrid

MAC protocol (MMH-MAC) [38] aims at maintaining the low energy consumption of asynchronous solutions and the high throughput of synchronous solutions, even in the presence of mobile nodes. In order to achieve this, it implements both the synchronous mode for stationary nodes and the asynchronous mode for mobile nodes. In the asynchronous mode, it uses preambles and a frame called SHUT-UP for stopping them. SYNC messages are sent in order to synchronize the nodes' internal clock and to serve as beacon to detect neighborhood changes. When a mobile node arrives at a new cluster it starts sending preambles until it gets a SHUT-UP message. Subsequently, the mobile node receives a SYNC message and synchronizes with the neighboring nodes. This solution was both simulated and implemented in a real environment.

Table 3 presents an overview of the various MAC protocols discussed in this section, using the classification proposed in sub-section 2.3. The *State* column indicates whether the protocol was implemented and deployed or simply simulated.

We can see that many MAC proposals are still based on simulation only. This is more so for the proposals that support mobility. So, it is apparent that many issues are still open and/or must be further evaluated prior to any implementation.

In addition, several other considerations should be done concerning these MAC proposals. MS-MAC is well established but has several limitations. X-MAC is the most used one but was not designed for mobility. Nevertheless, X-MAC with short preambles can also support mobility, avoiding the complexity of MS-MAC and MAMAC, by a simple adaptation as proposed by MoX-MAC. Moreover, mobility aware algorithms at the MAC layer require complex handoff mechanisms, which makes them heavier and more difficult to implement. Furthermore, in the cases of MAC protocols that can achieve considerably efficiency, they do so for specific types of applications or networks, not for the general case, which limits their applicability.

In view of the above, and in order to enable interoperability among different WSNs, most authors defend that mobility should be implemented at the network layer using data from the MAC layer for mobility estimation only. Network layer mobility is the subject of the following section.

4. Mobility at the Network Layer

Although some network layer WSN mobility solutions do not rely on IP and, instead, use Zigbee or WirelessHart, most of them are IP-based. Thus, in order to address WSN mobility support at the network layer, we start by providing, in sub-section 4.1, an overview of conventional IP mobility approaches (i.e., not specifically developed for WSNs) that could be used or adapted for WSN mobility support, while in sub-section 4.2 we analyze and compare existing IP mobility approaches specifically developed for 6LoWPAN [39].

6LoWPAN is an IETF Working Group (WG) created with the objective of adapting the IP technology to 802.15.4-based networks. The support of IPv6 over IEEE802.15.4 will allow WSNs to take advantage of well-known IPv6 capabilities, such as Neighbor Discovery (ND) or Mobile IP (MIPv6). 6LoWPAN aims also at enabling the use of upper layer protocols from conventional networks in LoWPANs (e.g., UDP, TCP, HTTP; etc.), thus providing the basis for an efficient interoperability of WSNs and the Internet.

4.1. Native MIP-based solutions

This section presents native IP Mobility proposals that can potentially be applied to 6LoWPAN.

Managing mobility at the network level is not novel in WSNs. In the past, authors in [40] [41] have analyzed the pros and cons of doing it, and surveyed the respective scenarios. In this section we briefly present the main characteristics of the most popular *Mobile IP* solutions, with the objective of finding the most suitable approach for WSNs.

The main objective of *Mobile IP* is to guarantee the connectivity of mobile devices, irrespectively of the network they are in, in a way that is completely transparent to the upper layers. This protocol was initially developed for IPv4 (MIPv4) [42], and later adapted and included in IPv6 (MIPv6) [43].

MIPv4 introduced the concepts of *Home Agent* (HA), *Foreign Agent* (FA), *Correspondent Node* (CN), *Mobile Node* (MN), *Care of Address* (CoA), *Mobility Binding Table*, and *Visitor List* (VL). A Home Agent is a local entity responsible for the management of local Mobile Nodes (MN), keeping their mobility information in a *Mobility Binding Table* (MBT). A *Foreign Agent*, in turn, is an external entity, located at the foreign network to which a *Mobile Node* (MN) moved. Whenever an MN arrives at a foreign network, the FA assigns it a *Care of Address* (CoA), registers that information in its VL, and notifies the HA, which, in turn, updates the corresponding entry in the local MBT. CNs are static or mobile nodes located on the Internet, with which an MN is communicating.

When a HA receives a packet destined to an MN it checks in its MBT whether the MN is in the local network. If so, the HA just delivers the packet. Otherwise, the HA gets the CoA, encapsulates the packet and forwards it to the FA, through an IP tunnel. When the FA receives the packet it checks the CoA in the VL, de-encapsulates the packet and delivers it to the MN. In the opposite direction the FA may forward the message from the MN to the CN using conventional routing mechanisms or using the FA-HA tunnel.

HAs and FAs periodically broadcast *Agent Advertisement* messages in their respective networks in order to detect changes in the existing devices. Whenever an MN moves to a new network it can wait for an *Agent Advertisement* or, instead, it can send an *Agent Solicitation*, to quickly announce itself to the new FA.

Table 3: Duty cycle optimized schemes.

	Type of duty cycle	Synchronization	Latency	Energy consumption	Mobility	State
S-MAC	Fixed	YES	Medium	Medium	NO	Deployed
MS-MAC	Adaptive (Active area)	YES	Medium	Medium-High	YES	Simulated
MOB-MAC	Adaptive	YES	Medium-	Medium-Low	YES	Simulated
AM-MAC	Adaptive	YES	Medium	Medium-High	YES	Simulated
MD-SMAC	Adaptive	YES	Low	Medium- Low	YES	Simulated
DS-MAC	Dynamic	YES	Low	Medium	NO	Simulated
CFMA-MAC	Adaptive	YES	Low	Medium-Low	YES	Simulated
TRAMA	Adaptive	YES	High	Medium	NO	Deployed
MMAC	Adaptive	YES	Medium	Medium	YES	Simulated
WiseMAC	Dynamic (Long Preambles)	NO	Medium	Medium-Low	NO	Deployed
B-MAC	Dynamic (Long Preambles)	NO	Medium	Medium-Low	NO	Deployed
X-MAC	Dynamic (Short Preambles)	NO	Low	Low	NO	Deployed
RI-MAC	Dynamic (No Preambles)	NO	Low	Medium	NO	Deployed
A-MAC	Dynamic (No Preambles)	NO	Low	Low	NO	Deployed
EM-MAC	Pseudorandom/Predicted	NO	Medium	N/A	NO	Deployed
MoX-MAC	Dynamic (Short Preambles)	NO	Low	Low	YES	Simulated
MAMAC	Dynamic	NO	High	Medium	YES	Simulated
MMH-MAC	Dynamic	YES/NO	Medium	Medium- Low	YES	Deployed

MIPv6 introduced significant changes in this mobility model, exploring native functionality in IPv6. Implemented as a specific message type of IPv6, MIPv6 becomes more flexible and easy to use. On one hand, the use of more than one address, global and link-local, associated with the address configuration mechanisms of IPv6, made it unnecessary to have FAs, as MNs can self-configure their own addresses. On the other hand, each MN (and CN) has its own *Binding Cache*, which is used for keeping information about other MNs/CNs. It is the responsibility of each MN to maintain updated information in its HA agent and in the *Binding Caches* of the MNs/CNs with which it is communicating, using *Binding Update* messages.
When a CN wants to communicate with an MN, it first checks whether a CoA exists in the *Binding Cache*. If so, the CN directly sends the packet to that CoA, using the IPv6 Routing Header Option. Otherwise, it will send the packet using the conventional mechanism, which will force the HA to encapsulate the packet and create the IP tunnel to forward it, like in the MIPv4 case. When the MN receives the encapsulated packet it will then notify the CN about the new CoA, in order to guarantee the direct communication in the next transmission and perform route optimization.

In addition to the Mobile IP protocol, there are many other solutions to support mobility in conventional networks. In the following, we present a brief description of the most important ones.
Hierarchical Mobile IPv6 (HMIPv6) [44] was specifically designed to minimize the handoff time in the cases of micro or intra mobility. *Mobile IP Fast Authentication* (MIFA) [45] allows local authentication, thus reducing the handoff load in MNs, moving it to the network side. PMIPv6 [46] presents a solution based on additional entities in the network, capable of handling the handoff on behalf of MNs. *Fast Handovers* (FMIPv6) [47] is based on Mobile IPv6 and PMIPv6 but it reduces handoff delay through the pre-configuration of the CoA in MNs prior to their movement. Finally, *Mobile SCTP* [48] uses several IP addresses to identify MNs. Each of these solutions is described in the following paragraphs.

4.1.1. HMIPv6

This handoff mechanism was developed to improve the performance of MIPv6, by implementing the concepts of local and global mobility.
HMIPv6 defines the existence of a new entity, named *Mobility Anchor Point* (MAP). This entity is basically a local HA capable of supporting several *Access Routers* (AR), which define the range limits of the MAP network and is responsible for continually announcing itself by broadcast. In this solution, MNs have two types of CoA: the *Regional Care of Address* (RCoA) and the *On-link Care of Address* (LCoA). While the former is the address used to identify MNs outside the visited network, the latter is used to identify MNs in the visited network. This mechanism allows MNs to freely move within the visited network without informing their HAs or CNs.
The MAP entity keeps a *Mobility Binding Table* containing information about MNs, associating their RCoA with the respective LCoA. ARs continually announce the MAP service so that whenever an MN enters the network it can use the information contained in those announcements to self-configure its RCoA. Each time an MN moves within the MAP network it will only handoff between ARs, which only requires an LCoA update, and does not affect the RCoA address. Movements inside the MAP network are, thus, local mobility, whereas movements between different MAP networks are global mobility.

4.1.2. MIFA

The *Mobile IP Fast Authentication* (MIFA) protocol explores the performance improvement of the registration process between MNs and HAs, The method consists in the delegation of the MN registration process in the FA, allowing the MN to authenticate locally. When an MN sends a *RegRqst* to a FA, the latter replies with a *Registration Reply* message, acknowledging the success of the process. Then a tunnel between the FA and the previous FA is recursively created until the HA is reached, thus leading to a single tunnel from the HA to the actual FA.

In order to guarantee the authenticity of the MN, the FA must be a member of a recognized FA group, known as *Layer 3 Frequent Handoff Region* (L3-FHR). This group can be dynamically or statically configured through specific algorithms. The L3-FHR region is generally constituted by a limited number of FAs.

Security mechanisms among FAs can be statically defined, for instance by the network administrator, or dynamically by the network. The MNs local authentication is guaranteed through notification messages sent by FAs to their L3-FHR group while the MNs are connected to them. These messages contain security data concerning the MNs, the FAs of the L3-FHR group, and the HAs. This security information is kept by each FA of the group, and is eliminated when used by one of them. Additionally, these messages can contain information regarding HA attributes and/or authentication data needed by MNs in their next FA registration.

When an MN sends a registration request to a new FA, the latter will verify the local authenticity of the MN through the security information sent by the previous FA to the L3-FHR group. Furthermore, the FA uses the extra information about the HA to check whether it corresponds to its requirements. Once the authenticity is confirmed, the new FA sends a notification to the previous FA, asking it to forward all packets whose destination is the MN. Afterwards, the FA replies to the MN registration request. Once the local registration is completed, the FA sends a notification message to the HA informing it about the new registration. In response to such notification the HA establishes an IP tunnel to the new FA. This type of mechanism allows the support of global mobility, like in Mobile IP, and also local mobility, like in HMIPv6.

4.1.3. Proxy MIPv6

Proxy MIPv6 (PMIPv6) was standardized in RFC5213 and describes a method to support node mobility using a network-based procedure. To do so, PMIPv6 defines two extra elements, namely the *Local Mobility Anchor* (LMA) and the *Mobility Access Gateway* (MAG). All the traffic to/from the MN is forwarded via the LMA, which creates a tunnel with the current MAG. Upon arrival at a new network the MN sends a Router Solicitation that is received by the local MAG. After receiving the solicitation, the MAG will send a *Proxy Binding Update* (PBU) to the MN's LMA, containing the MN ID and the profile, which includes, among others, the MAG address (proxy-CoA). The MNs LMA, in turn, will create a *Binding Cache Entry* (BCE), which includes the MN-ID, the MAG address and the new MN prefix, and send a *Proxy Binding Acknowledge* (PBA) to the MAG. When the PBA is received, the MAG creates the bi-directional tunnel to the LMA and sends a Router Advertisement to MN, containing its *Home Network Prefix* (HNP). Hence, almost all the process runs on the network side, with the MAG operating on behalf of the MN. Nevertheless, when moving from one MAG to another, the MN needs first to detach from the previous one before attaching to the new MAG, which might lead to packet losses.

4.1.4. FMIPv6

FMIPv6 is an enhancement of PMIPv6 with the objective of improving the performance of mobile IPv6 handoffs in what concerns latency and packet losses. For this, mobile nodes are not directly involved in the handover procedure, which is carried out by the previous network access router (PAR), the new network access router (NAR) and the HA. FMIPv6 has two basic modes of operation. In the predictive mode, low layer information is used in order to initiate the handoff procedure before the mobile node attaches itself to the new network. By establishing a bidirectional tunnel between the PAR and the NAR prior to the arrival of the node, the handoff time can be substantially reduced. In the reactive mode this tunnel is established after the node's attachment to the new network. In any case, to avoid losses during handoff, buffering is performed by the PAR and/or NAR, so that packets can be handed in to the MN when the handoff is complete.

4.1.5. Mobile SCTP

This handoff mechanism is an extension of the SCTP protocol in order to provide mobility support. One of the main characteristics of the SCTP protocol is multi-homing, which allows nodes to be reached through several IP addresses. One of those addresses is used as the main identifier, while the remaining addresses are used as a backup, for activation whenever the main address fails. The handoff process was thus designed for the case where an MN, supporting several addresses, is communicating with a static host. During the handoff process the main address is dynamically changed. When the MN arrives at a foreign network it first obtains the new IP address and then, using the mSCTP control message, adds that address to the host association table, making it the main address.

Based on the handoff mechanisms presented above, we can draw some conclusions considering packet losses, packet reordering, blackout period and the addition of extra elements. Although packet loss also depends on the type of application, by analyzing the handoff mechanisms we can conclude that MIPv6 and HMIPv6 can lead to significant packet losses. On the other hand, in order to avoid packet losses and guarantee a soft-handoff mSCTP makes use of multiple IP addresses and FMIPv6 may lead to temporary multiple paths. Both situations might cause packet reordering. Nevertheless, this does not occur with MIFA, because in this case the HA continues to forward data using the old path until receiving a new MN

registration and defining the new path. The blackout period in the handoff mechanisms is only avoided by the mSCTP protocol, which, as previously mentioned, uses several addresses to identify the same MN. Another aspect concerning the presented schemes regards the need for additional devices in the network. This is the case of HMIPv6, PMIPv6, and FMIPv6.
Table 4 summarizes the comparison of the various Mobile IPv6 solutions presented above.

As already mentioned, the mechanisms presented in this subsection were all designed and developed for unconstrained networks. In contrast, some alternatives considering IP mobility in WSNs have appeared in the literature. The next sub-section surveys the most relevant ones.

4.2. Mobile IP in 6LoWPAN

Most of the mobility solutions presented in the previous sub-section suffer from the same problem: intense signaling mechanisms that, if used in WSNs, would require considerable energy expenditure. Moreover, in order to prolong battery lifetime, there is a large consensus that mobility in WSNs should be handled on the network side, with as little involvement of sensor nodes as possible.

In [49] the authors proposed LowMOB, a network-based solution capable of handling mobility at the 6LoWPAN adaptation layer. LowMOB assumes a network structure constituted by Static Nodes (Full Functional Devices, FFDs), each one equipped with an antenna array used to determine MNs' directions. Authors completely defined a handoff protocol between static nodes and the gateway, and MNs and static nodes, implemented as a 6LoWPAN extension. In addition, a distributed LowMOB version was proposed, introducing the concept of Mobility Support Points (MSPs) capable of supporting mobility and also optimizing routing mechanisms. MSPs act like a cluster head of static nodes. The main drawback of LowMOB is that it relies on angle of arrival (AoA) and direction prediction in an attempt to predict the next sensor node that will handle the MN. This method is hard to implement in real deployments and its efficiency is not proven.

Another well-known proposal is inter-MARIO [50], an inter-PAN handoff solution based on pre-configuration. In this proposal, the objective is to send MN's information to the whole neighborhood and, at the same time, to send information about the neighborhood to the MN, every time mobility is detected. Thus, when the MN arrives at a new PAN the network has already dealt with its configuration. On the other hand, by receiving the neighborhood information the MN does not need to scan all channels to find another network, because it already knows the frequency of the surrounding networks. The inter-MARIO solution assumes that mobility estimation is available, regardless the method and its efficiency. The authors of inter-MARIO compared it with MIPv6 and with our μMIPv6 adaptation solution [51] in terms of signaling cost, concluding its superiority when pre-configuration is used. An evaluation using typical WSN scenarios and network topologies is missing.

In [52] the authors proposed another network-based mobility support scheme for 6LoWPAN. In this proposal, mobility support is achieved by organizing the WSN into cluster trees, in which each node maintains a ”mobile-cluster associate-node” table. This scheme assumes that each cluster head is capable not only of calculating the distance to other cluster heads but also of computing the relative position using AoA. Using this relative position and the information from the mobile-cluster associate-node table, it is possible to determine the next associate node. Unfortunately, as we mentioned before, AoA is not natively supported in most WSNs radio transceivers, and thus, this solution, would require specific hardware (directional antennas, antenna arrays, etc.) to make its usage possible.

Following a more realistic approach, authors in [53] presented a node mobility scheme for IP and non-IP WSNs using 6LoWPAN. Using a combination of host-based and network-based mobility, this scheme bases the handoff process on a protocol between the home edge router, the new edge router and the MN. Following a simplistic approach, authors demonstrated a slight superiority of their scheme when compared to MIPv6 and PMIPv6, regarding signaling messages. A proof-of-concept was also provided, demonstrating its feasibility. Nevertheless, the carried out evaluation is quite shallow and does not address reliability or efficiency, and the proposal never reached a mature state.

In [40] the authors presented an MIPv6-based solution, introducing the concept of Mobile Responsible Sensor Router (MRSR). This new concept aims at distributing the role of the home agent, splitting the responsibilities and guaranteeing information redundancy.

The Sensor Proxy MIPv6 (SPMIPv6) [54] is a WSN PMIPv6-based solution that, by using Sensor network based Localized Mobility Anchors (SLMA), extends PMIPv6 with the capability of interconnecting distinct WSNs through a shared backbone structure. In this proposal, SLMAs provide AAA mechanisms and nodes' reachability, while Sensor MAG (SMAG) act as edge-routers, detecting mobility and triggering the handoff process on SLMAs. When compared to PMIPv6 and MIPv6, SPMIPv6 demonstrated to be slightly superior regarding signaling and mobility costs. Nevertheless, the obtained results are not impressive and do not compensate the cost of adding an entire infrastructure that includes SLMAs, SMAGs, and the PMIPv6 domain. In addition, centralizing the entire action in the PMIPv6 domain creates a bottleneck and a central point of failure.

In an attempt to overcome these problems, in [55] the authors proposed Cluster Sensor PMIPv6 (CSPMIPv6). This approach is an enhancement of the SPMIPv6 scheme that groups SMAGs into clusters, each cluster being controlled by a cluster Head MAG (HMAG). The HMAG main function is to reduce the dependency on LMAs by moving to a

Table 4: Comparison of Mobile IP solutions

	Packets reordering	Blackout period	Additional infrastructure	Major feature
MIPv6	No	Yes	No	Mobility control
HMIPv6	No	Yes	Yes	High performance in local handoff
MIFA	No	Yes	No	Reliability
PMIPv6	No	Yes	Yes	Reliability, lightweight
FMIPv6	Yes	Yes	Yes	Faster handoffs
mSCTP	Yes	No	No	Reliability

distributed solution in which HMAGs are capable of handling intra-mobility signaling by themselves. Although CSPMIPv6 represents a significant improvement on its former solution, relying on a typical static tree-based backbone structure makes this solution hard to deploy in large-scale networks. Moreover, adopting a similar dependency on LMA for the CN tunneling establishment leads to the same drawbacks in terms of latency, failure points and bottlenecks previously observed in SPMIPv6.

Table 5 presents a summary comparison of the various 6LoWPAN mobility approaches mentioned in this sub-section, using the classification proposed in sub-sections 2.3 and 2.4. The *State* column indicates whether the approach was studied and/or evaluated by simulation, by analytical modeling, or by implementation.

5. A proposal for assisted mobility

As we could learn so far, mobility support in WSNs is not an easy matter, irrespectively of the level at which it works, involving and affecting all network components. None of the analyzed proposals demonstrated an efficient mobility support associated with a reliable service. Supporting mobility at the MAC layer has demonstrated to be inefficient and extremely complex due to the implications on duty cycle schemes. On the other hand, supporting mobility at the network layer simplifies the MAC layer, is energetically more efficient, and has the added value of being capable of controlling the entire handoff process along the MNs path. Unfortunately, most existing network layer mobility solutions require the nodes to handle their own mobility procedures, consuming precious energy and processing resources. Moreover, as we have concluded in [5], supporting MIPv6 and derived solutions in WSNs leads to considerable problems due to the fact that MIPv6 was not engineered for constrained networks.

The referred problems and limitations were the main driver for the development of the WSN assisted mobility proposal being presented in the current section, which evolved from the Network of Proxies concept proposed in [4] [5]. Being a network layer and network-based WSN mobility solution (see sections 2.3 and 2.4), it not only avoids complex duty cycle schemes but also relieves MNs from mobility management tasks and procedures.

5.1. Motivation

Deploying WSNs in real environments poses several challenges, especially when critical scenarios are targeted, for which performance control, reliability, robustness and efficiency are a must. In this context, complex duty cycle schemes to save energy, cognitive radios, advanced topology control modules, high-efficiency routing protocols, security mechanisms, and mobility support, are, among many others, examples of modules that have been designed for inclusion in sensor nodes and networks, often overlooking the fact that these are resource constrained systems.
However, real deployments, such as the one in Petrogal's oil refinery, in the scope of the GINSENG project [1], have shown us that, in general, sensor nodes are not capable of supporting all of those features together due to hardware constraints and energy restrictions.
In the case of the WSN deployed in Petrogal's refinery, the solution was to simplify the motes by removing features that were not crucial to the application at hand, focusing the sensor nodes' operation in what they were originally designed for: sensing and reporting.
Nevertheless, although we have concluded that adding complex features, such as features based on heavy communication protocols, requiring considerable amounts of memory (ROM and RAM) and CPU, should be avoided, efficient mobility support was a key requirement of the scenario despite the fact that such feature is tremendously demanding, as we demonstrated in [5]. As such, a feasible mobility management solution had to be developed.
In this context, we proposed a new concept called Network of Proxies, or simply NoP, a flexible and transparent overlay whose main objective is to seamlessly free WSN motes from complex, resource-demanding tasks, without interfering with the normal WSN behavior.
Naturally, the use of an NoP implies additional costs. Nevertheless, although NoPs were mainly designed for critical scenarios, the concept can be used in any WSN for which the extra investment is justified by the benefits, of which seamless mobility, quality of service, robustness and efficiency are examples.

5.2. Proposal Overview

The Network of Proxies (NoP) is a wireless mesh network overlay constituted by resource-unconstrained nodes (i.e., nodes not having the stringent limitations of normal sensor

Table 5: Comparison of 6LoWPAN (IP-based) mobility approaches

	Latency	Signaling Cost	Mobility handler	State
LowMOB [49]	Medium-Low	Medium-High	Network-based	Simulated
Inter-Mario [50]	Medium-Low	Medium	Hybrid	Simulated
Wang et al. [52]	Low	Medium-Low	Network-based	Simulated
Sinniha et al. [53]	Medium	N/A	Hybrid	Proof of Concept
Camilo et al. [40]	N/A	Medium-Low	Host-based	Simulated
SPMIPv6 [54]	N/A	Medium-Low	Network-based	Analytical model
CSPMIPv6 [55]	N/A	Medium- Low	Network-based	Analytical model

nodes), capable of handling energy/processing/time-consuming operations on behalf of sensor nodes. NoP was proposed in the scope of the GINSENG project [1] and was designed to assist constrained networks in critical scenarios, such as healthcare systems, oil refineries or power plants, in which reliability and performance are essential.

In its early phases, the concept relied on the existence of a shared backbone [4]. Then, the idea evolved and in [5], along with an evaluation of MIPv6 in WSNs, we presented the first version of the Network of Proxies and assessed its positive impact on the WSN side. Later, the concept was optimized, and an implementation was developed, leading to the results and evaluation presented in this paper.

In Figure 1 we can see a scenario in which an NoP is superimposed on a dense WSN. The figure separately depicts the WSN and the NoP in the upper part, and the two networks after superimposition, in the bottom part.

Several configurations are possible. A single NoP can be installed to cover the entire WSN, several WSNs or just a specific section of one or more WSNs, depending on the requirements. As previously mentioned, NoPs are transparent overlays, which means that these networks do not take part in or interact with the WSNs at the application level, and that the WSN's operation is unaffected by them.

Nodes supervised by NoPs are named Assisted Motes (AMs), while WSNs supervised by NoPs are named Assisted WSNs or just AWSNs. NoPs can be installed in both, new and/or existing WSNs.

Typically, an NoP is deployed following the Manhattan style, assuring that adjacent covered areas overlap. This leads to different AWSN areas being covered by different proxies and also in a different way. For instance, considering Figure 1 we can observe that there are motes covered by one proxy only, which means that they are located at the edge of the NoP, while there are other motes covered by as much as four proxies, which means that they are located somewhere in the middle of the NoP. This may be explored by the NoP in order to determine the areas in which the motes are, which can, in turn, be used for mobility support as explained in the next sub-section.

It should be emphasized that the NoP is an overlay structure that does not interfere with the WSN itself. As already mentioned, it can be used for assisting motes while they move through a WSN, but this does not mean that the WSN topology and/or multi-hop structure is affected. This is illustrated in Figure 2, where we can see a proxy assisting a mobile node in choosing its parent node while it roams through the WSN. Applying NoPs on WSNs brings new possibilities and perspectives, such as, new mobility estimation paradigms.

The following sub-section further discusses this example, as it is dedicated to explaining the use of NoP for mobility support.

When used for mobility support, each NoP proxy is responsible for monitoring the WSN area within its range and take note of the various motes' received signal strength indicators (RSSI). Depending on their location, motes are heard with different signal strengths by each proxy. By sharing the gathered information on each mote's RSSI among the proxies, the NoP is able to build what we call a "signature" for each mote. By analyzing the "signature" of mobile nodes, it is then possible to dynamically decide which is the best parent node for each mobile node, manage the handoff processes, and instruct mobile nodes to change their configuration accordingly.

Table 6 depicts an example of the signatures of various motes in an AWSN.

Table 6: Example of data stored and shared by each proxy in an NoP

Mote	Parent	HA proxy ID	HA RSSI (dBm)	P8 RSSI (dBm)	P4 RSSI (dBm)	P5 RSSI (dBm)
1	7		-50			
2	3	3	-55		-70	
3	4	3	-60	-60		
4	10	5	-65	-70	-68	-70
5	4	1	-70	-65		
1	7		-52			
7	14	5	-46	-70	-80	-65
...	...	...	...	...	...	...

As we can observe in the table, for each node in range, each proxy maintains information on the parent ID (the parent node of the mote in the WSN), the Home Agent proxy ID (the proxy responsible for the mote), the RSSI between the HA proxy and the mote (HA RSSI), and, in the case the mote is under more than one proxy, it also keeps the RSSI between each of the other proxies (Foreign Agent proxy, FA proxy) and the mote. This information is shared in an ad hoc fashion among proxies using the NoP overlay. While RSSI and parent information are obtained by sniffing the motes' communications, the other items are obtained by sharing information among the NoP proxies.

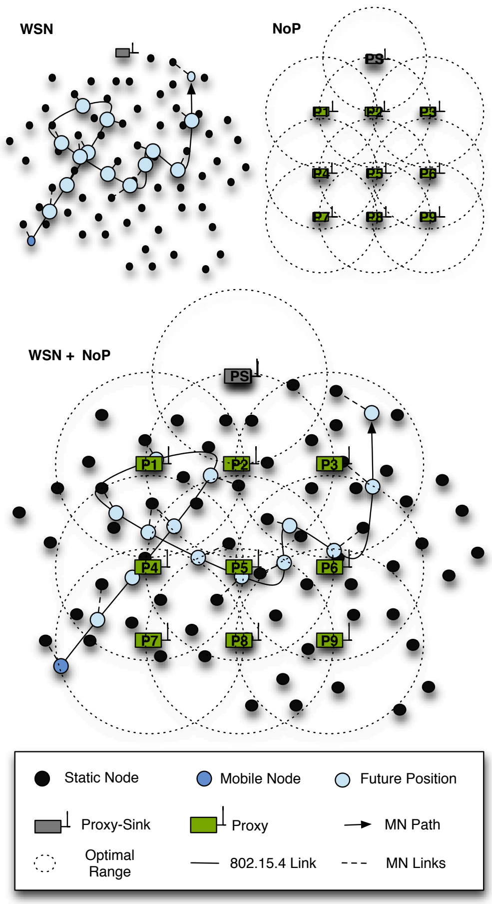

Figure 1: WSN partial assisted by a Network of Proxies.

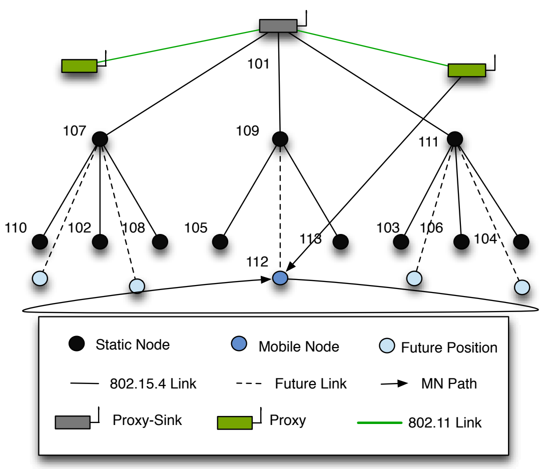

Figure 2: A proxy assisting a mobile node in choosing its parent node. This scenario was taken from GINSENG project.

As an example, considering node 3, we know from the table that in the last communication this mote was heard by the HA proxy with an RSSI of -60dBm, and also with an RSSI of -60dBm by FA proxy P8, which allows to draw the conclusion that this mote is somewhere between the HA proxy and FA proxy P8. Considering motes 4 and 7, we can observe that both are covered by the same proxies and also that both have similar RSSI values, which allows to conclude that both motes are in the same region.

Based on the signature information, proxies can easily detect when a node is moving, and manage the WSN connections along the path, either in intra-mobility or in inter-mobility scenarios. Moreover, the proxies can choose the best parent node of each mote and execute all handoff procedures on behalf of the motes. The process is totally transparent to the motes, which, on completion of the handoff, simply receive a command to update their configuration (parent, address, channel, etc.), interpreting these actions as simple topology updates performed by request of topology control modules.

An RSSI threshold is also used to define the optimal range of each proxy. Motes out of this optimal range will not be assisted.

As a final remark, although we are aware that RSSI might be unstable and that radio ranges are not symmetric, our experience shows that the described RSSI monitoring and decision method is effective, and that in the vast majority of cases the assisted motes are connected to the best parent. RSSI is not only considered a reliable metric in WSNs [56], but also a recent study concluded that we can use LQI to assess RSSI [57], making this solution yet more robust. Furthermore, NoPs allow evolving traditional mobility estimation methods to methods where the average RSSI can be analyzed, filtered and compared, such as the method presented in this section.

5.3. Implementation and Operation

In their simplest form, proxies can be implemented by common, embedded boards running Linux, WinCE or Android, or entry-level laptops running a standard Linux distribution such as Ubuntu. ARM [58] provides a wide range of embedded solutions capable of fitting different needs, including the ones of NoP implementation.

This section presents several aspects of the NoP implementation directed to mobility support. These include the proxy software architecture, the mobility estimation method and the handoff protocol.

Based on the described implementation, several sets of experiments were carried out to evaluate the behavior and performance of the NoP solution in different scenarios. The results of these experiments will be presented in section 5.4.

5.3.1. Proxy Architecture

An NoP is set up when two or more proxies establish a wireless mesh network among themselves, using their IEEE 802.11 interfaces for this purpose. Since proxies have all the necessary features, they can also act as sink nodes, which

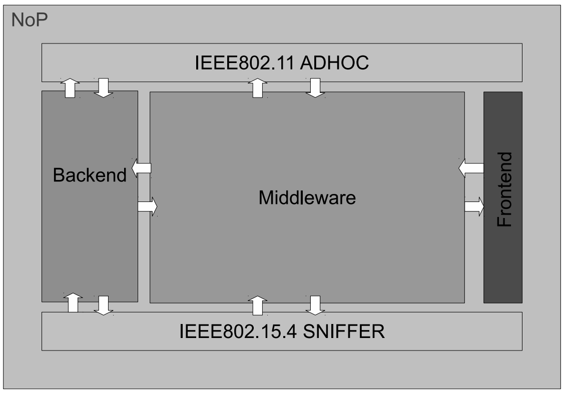

Figure 3: Proxy software architecture.

means that an NoP may contain both Proxies and Proxy-Sinks. Proxies can also act as Edge-Proxies when deployed between different WSNs.
To be part of an NoP, each proxy must be, on one hand, capable of communicating with other proxies via IEEE 802.11 and, on the other hand, capable of monitoring one or more WSNs (in the case of Edge-Proxies), via IEEE802.15.4. This characteristic poses several challenges to the design and implementation of these devices, forcing the use of, at least, two different radios and a mechanism for interconnecting them (note: this is also the approach taken in [59], for instance).
Nevertheless, more than being a simple sniffer, a proxy needs intelligence, as it must be able to decide what actions should be taken based on real-time information. For instance, in the case of mobility, proxies must be capable of not only deciding when a specific mobile node (MN) should handoff but also of performing all the handoff procedures on behalf of that MN.
Hence, to support this, the proxy's architecture (Figure 3) includes not only the mentioned radio interfaces but also a central unit (middleware), connected to a local data backend, and accessible through a configuration frontend.

Proxies constantly monitor the networks to which they are attached — the NoP mesh network(s) and the WSN network. After real-time processing of all packets by the middleware module, the data obtained from these networks is recorded in the local database, as presented in Table 6. The middleware module uses the proxy's communication interfaces to trigger or perform the necessary actions. The frontend module is the interface for configuration and management of the proxy software. This module is outside the scope of the current paper and will not be further described. The NoP architecture requires approximately 180MB of ROM and 60MB of RAM, when in full operation.
In summary, the NoP concept led to several requirements that were accommodated in the proxies' architecture, allowing them to:

- establish IEEE 802.11 ad-hoc connections with other proxies;
- detect and communicate with at least one IEEE 802.15.4 network;
- store, manage and share data regarding all the proxy neighbors and all the sensor nodes in range;
- process data, and communicate between both sides (WSNs and NoP) in real time;
- dynamically perform handoffs based on real time information.

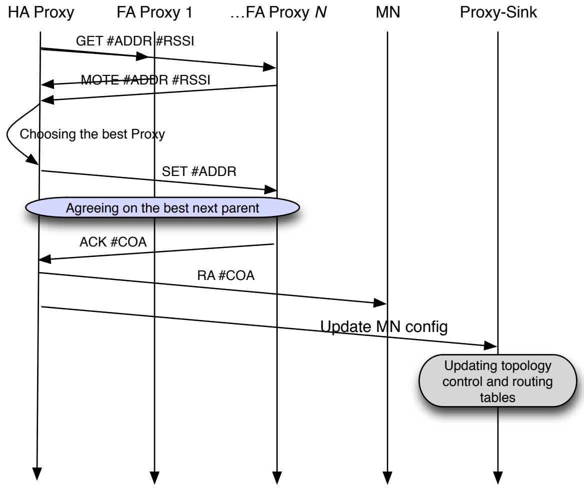

Figure 4: Mobility estimation and handoff protocol performed via proxies.

5.3.2. *NoP handoff protocol*

In the current implementation, the main goal of the proxy software is to monitor in real time the RSSI evolution of each mote included in the database and marked as home node, in order to perform the mobility protocol on behalf of the sensor nodes. It is the responsibility of this application to timely detect that a specific mote is significantly changing its signal strength, and to perform the necessary mobility estimation and handoff procedure using the IEEE802.11 ADHOC mesh network.
In our proposal, mobility estimation is done by the comparison of the motes' signatures, in which similar signatures will mean that the respective motes can connect to the same parent or that the MN can connect to the mote with the closest signature. To minimize wrong decisions, in our experiments mobility estimation is done based on the average of the last ten RSSI measurements of a specific node.
When the middleware module detects that the average RSSI is changing significantly (varying +/-10dBm), it executes the handoff procedure, using the protocol presented in Figure 4. This value can be adapted depending on the requirements of the scenario.

The protocol presented in Figure 4 uses UDP. In addition, to improve the performance we used two dedicated UDP ports (one for receiving and another for sending) per mote listed in the database.
The protocol is started by the home agent proxy (HA proxy) sending a GET multicast request message to all proxy neighbors. Although Table 6 contains the shared information, this

request will guarantee the most updated values from each proxy at the time of the latest MN's movement.
When a foreign agent proxy (FA proxy) receives a GET message, it checks if the specified mote exists in its database. If the mote exists, the proxy will then compute the average of the previous ten RSSI values for that particular mote and compare it with the RSSI value in the received GET message. If the computed result is higher, then it will reply with a MOTE #ADDR #RSSI message, which contains the calculated RSSI average value. This method is fundamental to decide which proxy is in better position to deal with the handoff.
On the other hand, the HA proxy stores all received RSSI values (contained in the MOTE messages sent by the various foreign agent proxies). After a time period proportional the number of neighbors, the home agent proxy will send a SET message to the proxy that reported the best RSSI value, which might be itself. That proxy will choose the next best parent to attach the MN to, based on the signatures' comparison, responding then with an ACK message providing the care-of address (CoA) that the mote should use from now on and the new parent id. When receiving the ACK, the home agent proxy extracts the CoA, assembles a Router Advertisement (RA) message and sends it to the mote via the IEEE802.15.4 interface. In turn, when the node receives the RA message, it loads the new configuration, thus completing the handoff procedure. In the case where the best proxy is located in a different WSN, the RA message also contains the channel to which the mote should move. When the best proxy to deal with the handoff is the HA Proxy, the SET #RSSI and ACK #COA messages are not used.
After concluding the handoff, the HA proxy updates its local Sink-Proxy (the WSN Sink-Node), which in turn updates its internal tables or forwards that same information in case of inter-mobility. Topology control modules and routing protocols should then be responsible for maintaining the consistency of the entire network.
Hence, the middleware module can operate in two distinct modes: as a HA proxy, when it starts the handoff procedure; or as a FA proxy, when it is acting in response to a HA proxy request. Figure 5 and Figure 6 present the state diagram for each proxy mode. As we can observe, in addition to the description given above, there exist several timeout mechanisms to avoid deadlock situations. given above, there exist several timeout mechanisms to avoid deadlock situations.

5.4. Evaluation

The implementation described in the previous sub-section was used to evaluate the Network of Proxies proposal, using various operating scenarios. Several experiments were done, not only to assess the intrinsic performance, regarding feasibility and load capacity, of the NoP concept, but also to compare this performance with non-NoP, node-based mobility solutions using MIPv6, whose feasibility in WSN we previously evaluated in [5].
Five scenarios were defined, differing in the number of proxies, number of MNs and movement pattern. These are presented in the next sub-section, whereas sub-sections 5.4.2 to 5.4.4 present and discuss the results.

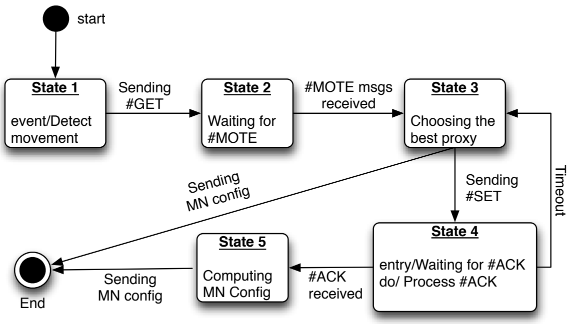

Figure 5: HA Proxy state diagram.

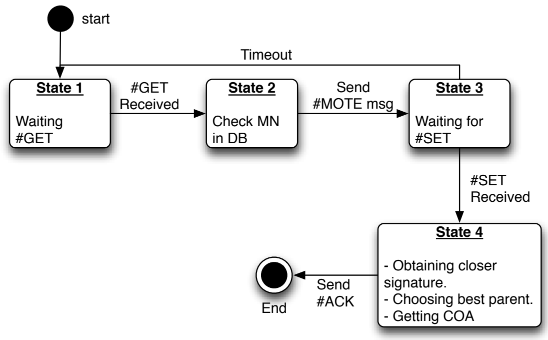

Figure 6: FA Proxy state diagram.

5.4.1. Scenarios

The scenarios used for evaluation of the NoP proposal are presented from Figure 8 to Figure 12. These scenarios were based on the GINSENG project. For illustration purposes, two photos of the deployment environment are presented in Figure 7.

While moving in the oil refinery, the employees can move alone, in small groups or in large groups. In general, their movements are limited to specific paths, which, on the one hand, increases the probability of large groups of employees moving together but, on the other hand, excludes random mobility. This case study thus poses several challenges to the existing infrastructures, requiring efficient mobility support between different WSNs. NoP was designed targeting this type of cases, and the scenarios used in this evaluation represent the most common situations in GINSENG.

Scenario 1, presented in Figure 8, represents one or a group

Figure 7: Photos of the GINSENG deployment.

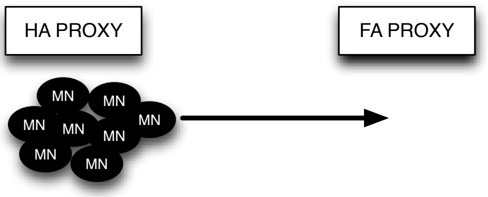

Figure 8: Scenario 1: 1 HA proxy, 1 FA proxy and N motes moving from one to the other.

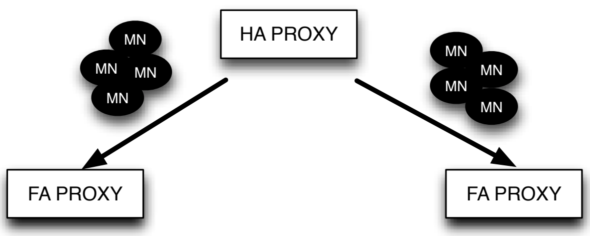

Figure 9: Scenario 2: 1 HA proxy, 2 FA proxies and N/2 motes moving to one FA proxy while the other N/2 are moving to the other FA proxy. Each sub-group of MNs is only accessible from the closest FA proxy.

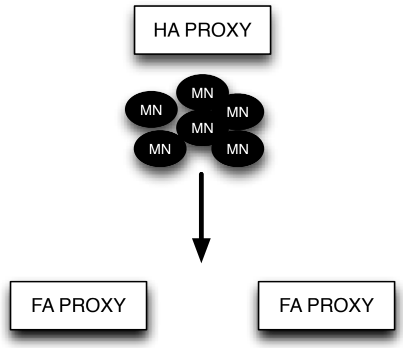

Figure 11: Scenario 4: 1 HA proxy, 2 FA proxies and N motes moving to an area accessible from both FA proxies, with different RSSI values.

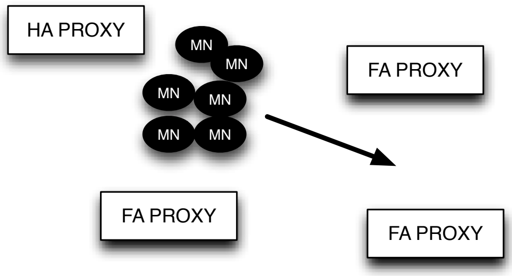

Figure 12: Scenario 5: 1 HA proxy, 3 FA proxies and N motes moving to an area accessible to all FA proxies, but with different RSSI values.

of mobile nodes (MNs) moving from one location under the control of their HA proxy to another location near one FA proxy. The objective of this scenario is to evaluate the basic configuration, when one HA proxy deals with just one neighbor, performing the handoff of one or more MNs.

Scenario 2 (Figure 9) introduces two novel aspects when compared to scenario 1. In this scenario we have one additional FA proxy and the MNs are split into two separate sub-groups. This means that a FA proxy is not able to detect packets from MNs under the responsibility of the other FA proxy and, thus, the HA proxy will only receive one answer per MN, avoiding the need to choose the answer with the best RSSI.

Unlike scenario 2, in scenario 3 (Figure 10) each FA proxy can listen to all the MNs, although with different RSSIs. Consequently, when asking for information on MNs (GET message), the HA proxy will receive two answers (MOTE messages) for each request, one from each FA proxy. Therefore, it will have to choose the one with the best RSSI. In the case of using just one MN it will move toward one of the FA proxies.

Scenario 4 (Figure 11) is similar to scenario 3, with a simple difference: while in scenario 3 the MNs move in separate groups, leading to different RSSIs for each FA proxy, in this scenario they move all to the same region, maintaining the same RSSI to each FA proxy. While in scenario 3 the HA proxy needs to randomly deal with one of the two FA proxies, depending on the one reporting the best answers, in this scenario it will analyze the two messages anyway, but reporting always to the same FA proxy.

Finally, scenario 5 (Figure 12) introduces one more FA proxy However, the setup is similar to scenario 4. The MNs are moving together and all to the same area. The HA proxy will have to choose between 3 answers per mote, all of them with different RSSI values, although equal for all MNs.

These scenarios target the evaluation of the NoP load capacity and the time taken to perform the handoff since the middleware module determines the need to initiate the handoff until the MN receives the RA message carrying the new CoA. For each scenario, we varied the number of MNs from 1 to 90, with intervals of 10, i.e., 1, 10, 20, 30, 40, 50, 60,70, 80 and 90 MNs. Because of logistic reasons we could not handle several

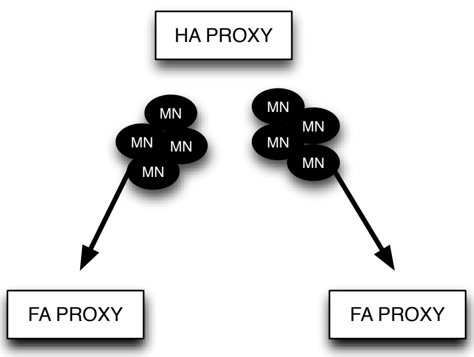

Figure 10: Scenario 3: 1 HA proxy, 2 FA proxies and N/2 motes moving to one FA proxy while the other N/2 are moving to the other FA proxy, although both sub-groups are accessible from both FA proxies.

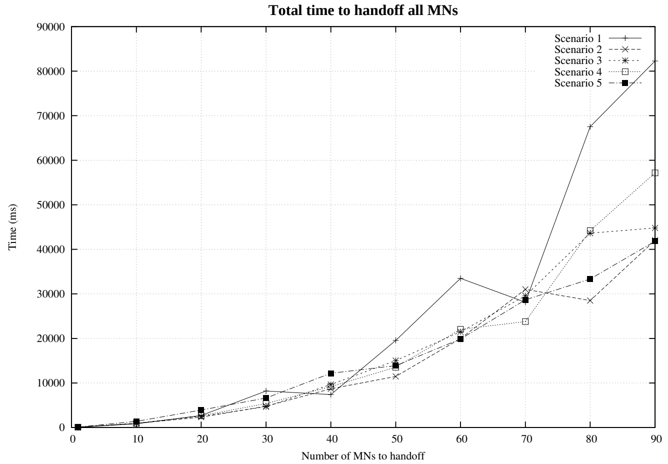

Figure 13: Total time to handoff.

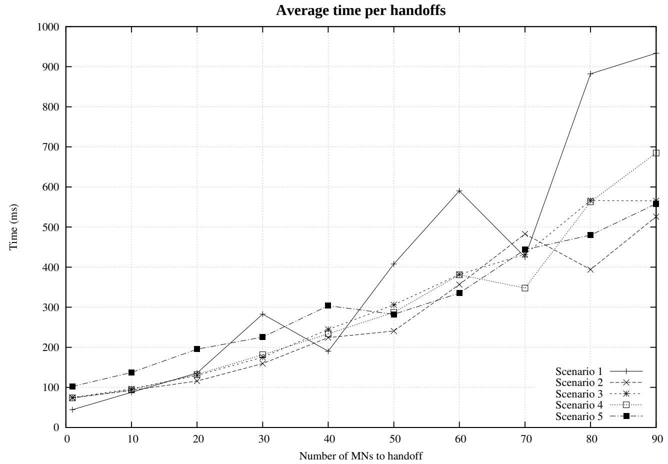

Figure 14: Average time per mote to handoff .

MNs moving simultaneously and therefore, we were forced to emulate the handoff requests. All the MNs were automatically inserted in the database and all of them required to handoff at the same time, emulating the mobility estimation process, as well.
For each case we performed the handoff 100 times and all the results presented in the next sub-section are the average of those 100 runs.
The proxies were laptops with different hardware capabilities, but all of them running Ubuntu 11.04. The used MNs were all TelosB (MSP430 and CC2420) [60]. Because of logistic limitations, all evaluations with more than 10 MNs were done by emulation, in which the MNs were emulated in software while the proxies were maintained in hardware. This emulation comprised the record of simulated RSSI values in each proxy and the trigger of the handoff process for each MN, originating the handshake protocol of Figure 4 among all real proxies.

In addition to the time taken by the NoP infrastructure to perform handoffs, we also present the total handoff time, i.e., the time up to and including the MN's processing of the RA message and loading of the new configuration.
Last but not least, we compare the NoP solution with the conventional solution (i.e., MIPv6-enabled WSN nodes), using values determined in [5].

5.4.2. Handoff time and success rate

For each experiment, we measured the total time that the set of MNs took to handoff, the number of handoffs successfully completed, the number of unsuccessful handoffs, the success ratio, and the average time per handoff.
Figure 13 presents the total time taken by the proxies to complete the simultaneous handoff of all MNs, while Figure 14 presents the average time per MN to handoff in the different situations.

Analyzing Figure 13 we can first conclude that, as expected, increasing the number of MNs simultaneously requiring handoff increases the total time needed to complete the task, independently of the scenario. However, looking in detail we can observe that scenario 1, the simplest one, requires more time than the other scenarios to complete the handoffs when the number of MNs is higher than 40. As mentioned before, more neighbors in the NoP means that the HA proxy will increase the waiting time when requesting information. However, this increase, fundamental to guarantee the reception of all responses, is not meaningful when the number of MNs to handoff is high. Furthermore, through these results we can observe that increasing the number of proxies will help when the number of MNs to handoff is relatively high. We can also observe the trend that, in general, increasing the number of proxies (in the experiments, from two to three or four) leads to a decrease in the total time to handoff all MNs.

Despite the number of proxies, the way the MNs moved also created some differences. We can observe the results of scenarios 4 and 5, where the MNs moved to the same area, and compare them with scenarios 2 and 3, where the MNs are split into two sub-groups, and conclude that when the MNs move altogether the total time to handoff is more predictable, and therefore controlled, than when they move in groups. The increase in scenario 5 is almost linear, and scenario 4 follows the same trend, with some small variations, as opposed to the remaining scenarios, for which the linear increase is not apparent.
A similar trend can, naturally, be observed if we look at the average time to handoff per MN, as depicted in Figure 14.

Looking at Figure 14, it is important to note that not even in the worst and extreme case of the simultaneous handoff of 90 emulated MNs did we have handoff times of 1 second or higher. Moreover, if we consider scenario 5, in which the number of proxies is higher, we can see that the average time per handoff when simultaneously performing the handoff of 90 MNs was below 600ms, which is a very good result when compared with the NoP solution with the conventional node-based approach, as we will see later in this paper.
Because the total time to handoff all MNs by all proxies and the

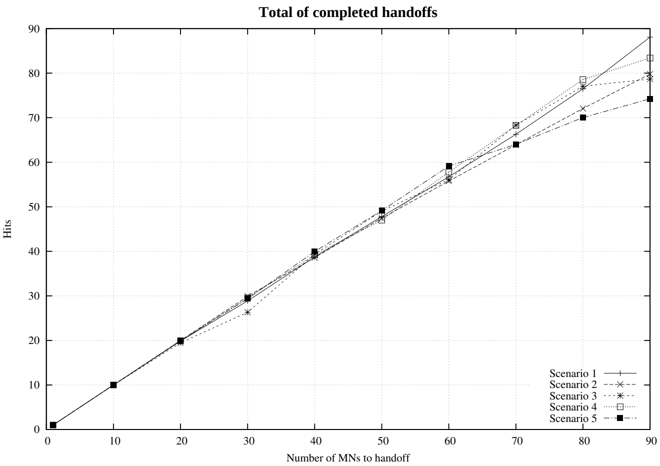

Figure 15: Total number of completed handoffs.

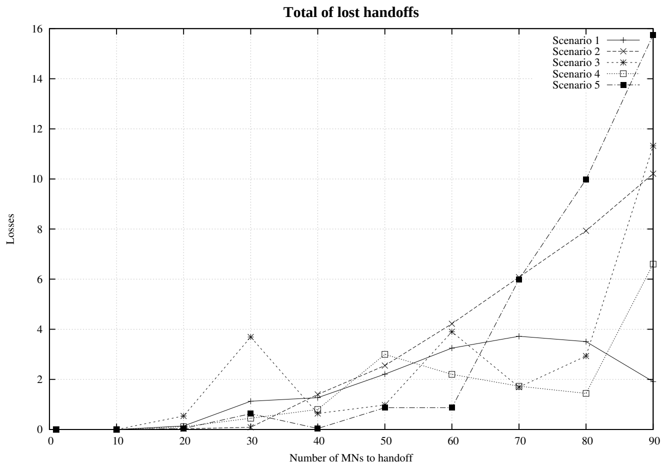

Figure 16: Total number of lost handoffs.

average time per handoff and per MN are not enough to completely depict the success of the operation, we also measured the number of successful handoffs, losses and the respective ratio per experiment.
Handoff failures can occur due to a variety of reasons, such as system overload or radio interferences. When loading a system up to a point close to one hundred MNs, even when assigning individual sockets to the communication between each MN and the proxies, some losses can occur.
Figure 15 presents the number of successful handoffs, which means the number of processes completed, for all MNs, since the proxy initiates the handoff until the MNs receive an RA message with the new configuration. On the other hand, Figure 16 presents the number of lost handoffs, and Figure 17 combines the results in Figure 15 and Figure 16, showing the success ratio for each experiment.

Looking at Figure 15, we can observe that, in general, until 40 MNs, all handoff requests were successfully completed. Under this mark there is only a slight difference in the 30 MNs case for scenario 3. From this point on, we can observe a small relative decrease in the number of completed handoffs, starting with the 50 MNs case until the 90 MNs case. Starting with the 80 MNs case, the relative decrease becomes distinctly noticeable. If, on one hand, we conclude from the handoff time analysis that increasing the number of proxies (e.g., as in scenario 5) will increase the performance, on the other hand we can now observe that scenario 5 completed relatively less handoffs after the 70 MNs point, when compared to the other scenarios. However, under the 70 MNs point this scenario outperformed all others.

Analyzing Figure 16 we can confirm the conclusions presented above. The loss increase when increasing the number of simultaneous MNs handoffs is justified by the increase in the number of exchanged messages. This is corroborated by the peak in scenario 5, that can be justified by the increase in the number of responses (MOTE messages) caused by the increase in the number of proxies. Although more proxies can improve the handoff time, as demonstrated in Figure 14, this will also

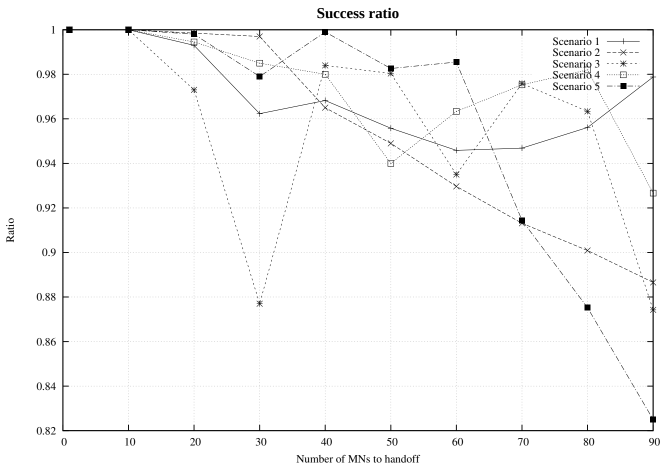

Figure 17: Handoff success ratio per scenario and per number of MNs.

increase the probability of congestion and consequent losses. Looking at these results across all scenarios, we can see that the increase in unsuccessful handoffs is not linear, in general. Under the 60 MNs point the average losses will not be higher than 4 handoffs and above the 60 MNs point we can observe a general upward trend, although the number of lost handoffs is always below 20.

The high handoff success ratio can be better observed in Figure 17, where it is apparent that until the 70 MNs point there is a success rate higher than 90% in the first handoff, with the exception of the 30 MNs case in scenario 3. It is important to mention that, in the case of handoff failure, the NoP middleware module will repeat the handoff attempt after 3 seconds. Figure 18 provides a complementary perspective of the results, presenting the relation between the average time to handoff and the handoff success rate, considering the average of the five scenarios. These are the two metrics that better depict the overall performance of the NoP proposal, as they convey latency and reliability information.

Looking at Figure 18, it is clear that, although there is no direct relation between the average time to handoff and the hand-

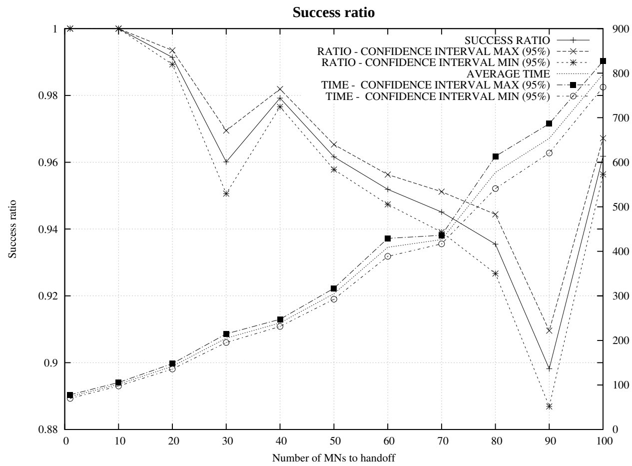

Figure 18: Relationship between average time to handoff and handoff success rate. Average of the five scenarios.

```
> timeaov <- aov(data$time~data$mn)
> summary(timeaov)
            Df  Sum Sq Mean Sq F value    Pr(>F)
data$mn      9 1780311  197812  23.226 3.255e-13 ***
Residuals   40  340682    8517
> rateaov <- aov(data$success~data$mn)
> summary(rateaov)
            Df   Sum Sq   Mean Sq F value    Pr(>F)
data$mn      9 0.046032 0.0051147  5.1993 0.0001107 ***
Residuals   40 0.039349 0.0009837
```

Figure 19: One-way ANOVA for Average Time and Success Rate per number of MNs.

off success rate, increasing the number of MNs that simultaneously require handoff affects both metrics as expected.

5.4.3. *Analysis of Variance*

In order to further assess the obtained results, we used the one-way analysis of variance (one-way ANOVA) technique, applying it to all scenarios, to check whether or not the impact of the varying number of MNs on the success rate and on the average handoff time was significant. Figure 19 presents the results.

Considering a significance level of 0.05, in Figure 19 we can see that both average time and success rate are affected by the number of MNs, their mean values being significantly different. The difference between the average times is indeed very high, with a *p-value* of 3.255×10^{-13}, while for the success rate that difference is not so visible, with a *p-value* of 0.001107. In order to see where the difference lies, we ran the Tukey's Honestly Significant Difference (Tukey HSD) test for both cases.
Analyzing Figure 20 we can conclude that, in general, considering a significance level of 0.05, the average handoff time is significantly affected after the 50 MNs point. This means that increasing the number of MNs simultaneously performing handoff over 50 will significantly affect the average time of each operation.

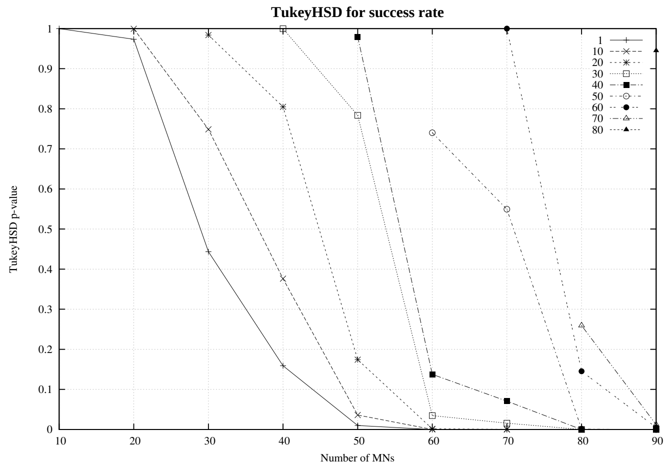

Figure 20: Tukey HSD results for the average time.

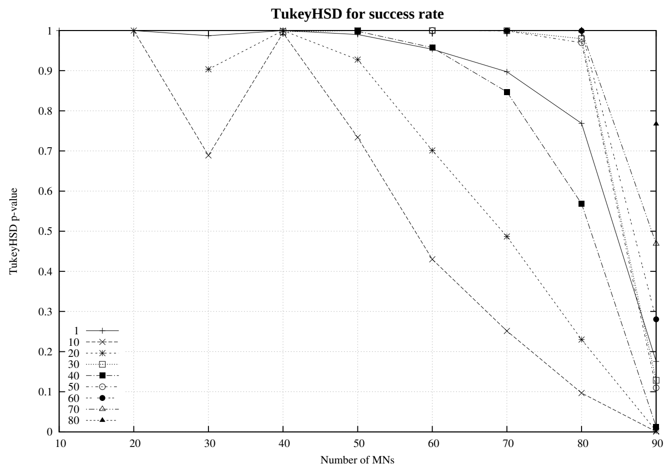

Figure 21: Tukey HSD for the success rate.

Regarding the success rate, we can see in Figure 21 that only the case of 90 MNs significantly affects the average.
Hence, to conclude this analysis about the significance of the impact of the number of MNs to handoff at the same time, we can conclude that using the proposed NoP architecture, we can perform up to 50 simultaneous handoffs without significantly affecting the average time per handoff, and up to 90 simultaneous handoffs without significantly affecting the success rate, considering a level of significance of 0.05.
In order to also analyze the impact of the different scenarios on the obtained results, we also used the one-way ANOVA test. In the previous sub-section we did a visual analysis of the results and we could observe some differences between the scenarios. However, we did not know yet whether such differences were significant or not.

As we can observe in Figure 22, the obtained *p-value* of 0.6996 for the average handoff time and 0.7284 for the handoff success rate demonstrate that the different scenarios did not create a significant difference in the final results. Although we could visually observe some differences, as mentioned before, they do not significantly affect the overall performance. Hence,

```
> summary(scentimeaov)
              Df  Sum Sq Mean Sq F value Pr(>F)
data$scenario  4   98955   24739  0.5506 0.6996
Residuals     45 2022038   44934

> summary(scenrateaov)
              Df   Sum Sq    Mean Sq F value Pr(>F)
data$scenario  4 0.003705 0.00092623  0.5103 0.7284
Residuals     45 0.081676 0.00181503
```

Figure 22: One-way ANOVA to analyze the scenarios' impact on the average time and success rate.

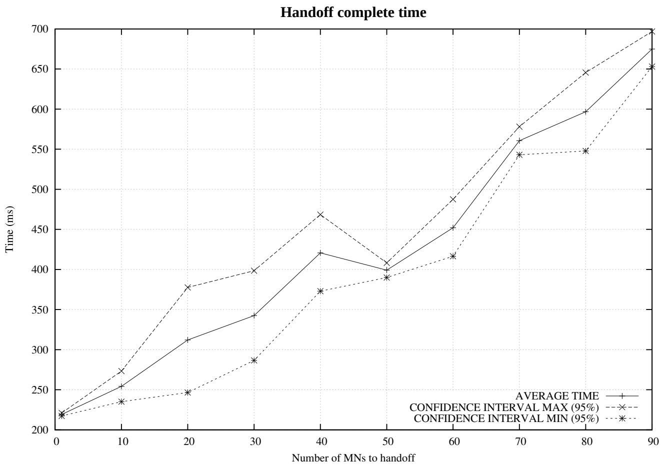

Figure 23: Average time needed to complete the handoff until the MN loads the new configuration.

we can conclude that the performance of the proposed NoP solution is quite stable, independently of the used scenario. Nevertheless, we aim to extend this study in future work in order to increase the proxies' density and study its impact.

5.4.4. NoP versus node-based handoff

It is important to analyze how good the proposed NoP architecture is when compared with the conventional solutions. In conventional WSN solutions there are no proxies and the MNs must execute the handoff process by themselves. To do so, we compared the results presented above with the results obtained in a previous study [5], which analyzed the impact of using standard MIPv6 over WSNs.

In [5] we determined that each MN took an average of 116.8ms to receive the RA message and load the new configuration. Therefore, if we sum this value to the average handoff processing time by the NoP infrastructure, we achieve the total time to handoff (i.e., the time taken by the NoP since it detects the need to handoff until it sends the RA message to the mobile node, plus the time spent by the mobile node to receive this message and load the newly assigned configuration). Figure 23 presents these values, for a varying number of MNs simultaneously performing handoff.

In [5] we also determined the average time taken by a single node running MIPv6 to perform one handoff: 771.90ms. Figure 24 depicts this result along with the best and worst NoP cases.

As we can observe, even considering the worst NoP case (i.e., 90 MNs performing handoff at the same time), we can achieve better results than MIPv6 doing just one handoff.

Regarding energy consumption on the WSN side, since in the presence of the NoP the MN only receives the RA message and loads the new configuration, the energy it spends is considerable less than the energy it would spend if the node was running MIPv6. To conclude, Figure 25 presents this comparison.

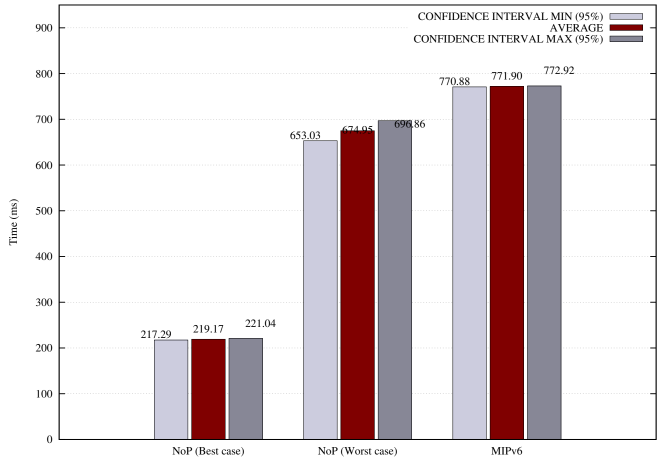

Figure 24: NoP versus the conventional solution MIPv6.

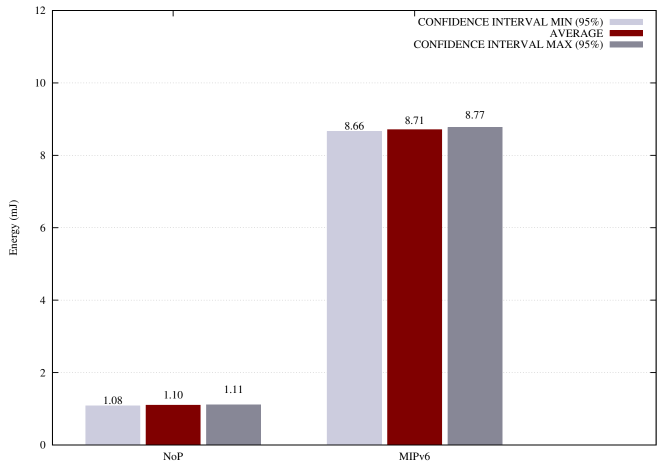

Figure 25: Comparison of the energy spent by MNs during MIPv6 node-based handoff and NoP-based handoff.

6. Related Projects

This section presents projects that deal with or are related to WSN mobility.

AWARE [61] was a European project whose main objective was to develop middleware for the support of communications between aero vehicles, such as helicopters, and ground sensor networks. Those networks comprise mobile nodes, carried by ground vehicles or people, which means that both sink nodes and sensor nodes can be mobile. Moreover, nodes are heterogeneous and, thus, the middleware must support several

different devices and operate as an abstraction layer to the application layer.
Regarding mobility, this project does not approach handoff techniques. Instead, the project considers that mobile nodes must connect to new points of attachment as fast as possible, and then the middleware layer deals with connectivity issues. An adaptable routing protocol was developed to assure nodes reachability [62].

CitySense [63] was a project from Harvard University and BBN Technologies, sponsored by Microsoft and the American National Science Foundation, whose main objective was to deploy about 100 sensor nodes dispersed through the city of Cambridge, MA. Static and vehicular sensor nodes collected data about weather and pollution conditions. This project aimed also at providing an open urban large-scale platform for researchers worldwide, allowing them to test their own applications. CitySense is an example of a project where mobility could improve coverage and therefore efficiency, through the deployment of mobile nodes in vehicles with pre-defined and controlled routes, such as city buses. CitySense did not handle mobility itself but aimed at interfacing with mobile and low power sensors, obtaining data from heterogeneous mobile devices, such as cellular phones or vehicular sensors.

WISEBED Wireless Sensor Network Testbeds [64] was an FP7 European project whose main objective was to provide a large-scale, well-organize structure considering heterogeneous nodes. This project aimed at dealing with hardware, software, algorithms and data, and also at making the resulting, final structure available to other research groups. WISEBED considered dynamic and heterogeneous nodes, and therefore the project outcome can be a useful platform for mobility evaluation.

CHOSeN was also a European project [65], launched to develop specific applications for real, large-scale wireless sensor networks in critical scenarios such as automotive and aeronautic. The CHOSeN consortium was composed of partners with wide experience, from chips design to application software development, including also MAC, network and transport layers protocol design.
CHOSeN aimed at smart wireless sensor networks composed by heterogeneous nodes. The resulting smart network should be interoperable with other types of networks, such as vehicular networks. Therefore, mobility was an important requirement, considering the project objectives. Nevertheless, no significant developments have come out of the project in this area.

e-Sense [66] was a project that aimed at the exploration of Ambient Intelligence through wireless sensor networks. The idea was to create a context-capturing framework that accepted and integrated different inputs, focusing on energy efficient operation of the network. The considered scenarios were equipped with multiple sensors and used heterogeneous, partly mobile nodes in highly variable numbers. Besides, the e-Sense consortium also developed some related work, mainly regarding sink mobility [67]. The e-Sense framework also specified middleware components and the integration with beyond 3G mobile communication systems. e-Sense used a pre-planned deployment approach and even had a tool for simple network planning. e-Sense used a TDMA-based network setup.

RUBICON [68], an FP7 European project started in April 2011, aimed at developing a self-learning robotic network, constituted by sensors, effectors, and robots. The objective was to develop robots capable of adapting to the environment, learning through the different interactions. This self-learning robotic ecology could be used in several target application areas, such as assisted living and security, among others. By using their mobility and sensing capability, robots could intelligently adapt to each situation.

Aiming at delivering a global wireless sensor and actuator network (WSAN) framework capable of providing services and applications through universal interfaces, SENSEI an FP7 European project [69] started in January 2008. This project focused on a highly scalable architectural framework, an open service interface and corresponding semantic specification, an efficient WSAN island solution, and a Pan European test platform for long term evaluation of WSANs in the Future Internet. Although SENSEI's main objectives did not target mobility directly, some related work was carried out in the scope of this project, targeting mobility of multiple sink nodes [70], routing of mobile elements [71], and study of the impact of mobility on SENSEI traffic [72].

HOBNET [73], another FP7 European project, targeted the development of solutions for Future Internet applications focused on energy efficiency and smart buildings. The project resorted to large scale, heterogeneous wireless sensor networks for monitoring and controlling entire buildings, with the aim of maximizing their energy efficiency. HOBNET used an all-IPv6 approach, through 6LoWPAN. The project had a specific work package on network protocols and architectures, including heterogeneous and mobile devices. This project resulted in several scientific contributions regarding mobile sinks, mobile data propagation, and indoor tracking based on mobile nodes.

As it can be seen from the above, some projects and initiatives consider mobility scenarios. Nevertheless, the ones that took mobility research further typically consider sink mobility, not node mobility. This points to the facts that, on one hand, the research community considers that mobility support at sensor node level is too complex and too demanding for constrained sensor nodes and, on the other hand, there are no widely accepted WSN mobility solutions.

7. Conclusion

Despite all the technological development witnessed in the last few years, hardware restrictions of wireless sensor nodes keep being incompatible with the large amount of features that many researchers insist on including in WSNs, such as advanced routing algorithms, security mechanisms, data fusion methods, and mobility, among many other. The research community is starting to recognize this and is looking at complementary solutions that can relieve sensor nodes from tasks they were not designed for.

In this paper we surveyed mobility in WSNs from different perspectives. After a general characterization of mobility in wireless sensor networks, we presented and discussed WSN mobility solutions implemented at the MAC layer. Subsequently, network layer WSN mobility was addressed, considering solutions that were not specifically developed for low power and lossy networks, and solutions that were. Despite the existence of many proposals, the conclusion is that they can hardly be used in real deployments due to a variety of factors.

This conclusion was the main motivation for the proposal of the Network of Proxies (NoP) concept and solution, which relieves sensor nodes from performing complex mobility tasks by moving them to the network side. NoP is an overlay structure that relies on a self-configured wireless mesh network of proxies capable of handling processor, memory, and energy-intensive operations on behalf of the sensor nodes.

Using mobility as case study, and through implementation, we demonstrated that NoP is capable not only of triggering and performing the handoff of sensor nodes faster than conventional solutions (i.e., solutions in which the same operations are the responsibility of sensor nodes), but also that it is capable of handling several mobile sensor node handoffs at the same time, under different scenarios. Specifically, the experiments led to the conclusion that the implemented NoP, based on standard, off-the-shelf hardware, allowed the handling of up to 50 simultaneous handoffs without significantly affecting the average permote handoff time, and up to 90 simultaneous handoffs without significantly increasing the probability of failing the handoff. Moreover, when compared to node-based MIPv6 conventional handoff, the NoP solution is greatly beneficial in terms of handoff time and node energy consumption, which demonstrates that the use of an additional structure is largely compensated by the improvement in overall WSN performance. NoP is, thus, fundamental in critical scenarios, such as GINSENG, where reliability and performance control are key requirements.

As future work we plan to explore the NoP solution in scenarios with more sensor nodes and proxies per area, and with several WSNs, in order to further characterize the benefits of the proposed approach and explore its potential.

Acknowledgments

The work presented in this paper was partially financed by the IST FP7 0384239 European Project GINSENG - Performance Control in Wireless Sensor Networks, and by project iCIS (CENTR-07-ST24-FEDER-002003)

References

[1] C. Sreenan, Ginseng performance control in wireless sensor networks, http://www.ict-ginseng.eu/ (Sept. 2012).

[2] T. Schmid, R. Shea, M. B. Srivastava, P. Dutta, Disentangling wireless sensing from mesh networking, in: Proceedings of the 6th Workshop on Hot Topics in Embedded Networked Sensors, HotEmNets '10, ACM, New York, NY, USA, 2010, pp. 3:1–3:5. doi:http://doi.acm.org/10.1145/1978642.1978646.

[3] N. Banerjee, M. D. Corner, D. Towsley, B. N. Levine, Relays, base stations, and meshes: enhancing mobile networks with infrastructure, in: Proceedings of the 14th ACM international conference on Mobile computing and networking, MobiCom '08, ACM, New York, NY, USA, 2008, pp. 81–91. doi:http://doi.acm.org/10.1145/1409944.1409955.

[4] R. Silva, J. Sa Silva, F. Boavida, A proxy-based mobility solution for critical wsn applications, in: Communications Workshops (ICC), 2010 IEEE International Conference on, 2010, pp. 1 –5. doi:10.1109/ICCW.2010.5503908.

[5] R. Silva, J. S. Silva, F. Boavida, A proposal for proxy-based mobility in wsns, Computer Communications 35 (10) (2012) 1200 – 1216. doi:10.1016/j.comcom.2012.03.005.

[6] Z. Wang, S. Basagni, E. Melachrinoudis, C. Petrioli, Exploiting sink mobility for maximizing sensor networks lifetime, in: System Sciences, 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference on, 2005, pp. 287a–287a. doi:10.1109/HICSS.2005.259.

[7] H. S. Kim, T. F. Abdelzaher, W. H. Kwon, Minimum-energy asynchronous dissemination to mobile sinks in wireless sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03, ACM, New York, NY, USA, 2003, pp. 193–204. doi:10.1145/958491.958515.
URL `http://doi.acm.org/10.1145/958491.958515`

[8] D. Stevanovic, N. Vlajic, Performance of ieee 802.15.4 in wireless sensor networks with a mobile sink implementing various mobility strategies, in: Local Computer Networks, 2008. LCN 2008. 33rd IEEE Conference on, 2008, pp. 680–688. doi:10.1109/LCN.2008.4664265.

[9] R. C. Shah, S. R. 0001, S. Jain, W. Brunette, Data mules: modeling and analysis of a three-tier architecture for sparse sensor networks., Ad Hoc Networks 1 (2-3) (2003) 215–233.

[10] A. Chakrabarti, A. Sabharwal, B. Aazhang, Using predictable observer mobility for power efficient design of sensor networks, in: Proceedings of the 2Nd International Conference on Information Processing in Sensor Networks, IPSN'03, Springer-Verlag, Berlin, Heidelberg, 2003, pp. 129–145.
URL `http://dl.acm.org/citation.cfm?id=1765991.1766001`

[11] A. A. Somasundara, A. Ramamoorthy, M. B. Srivastava, Mobile element scheduling for efficient data collection in wireless sensor networks with dynamic deadlines, in: Proceedings of the 25th IEEE International Real-Time Systems Symposium, RTSS '04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 296–305. doi:10.1109/REAL.2004.31.
URL `http://dx.doi.org/10.1109/REAL.2004.31`

[12] E. Ekici, Y. Gu, D. Bozdag, Mobility-based communication in wireless sensor networks, Communications Magazine, IEEE 44 (7) (2006) 56–62. doi:10.1109/MCOM.2006.1668382.

[13] K. Akkaya, M. Younis, W. Youssef, Positioning of base stations in wireless sensor networks, Communications Magazine, IEEE 45 (4) (2007) 96–102. doi:10.1109/MCOM.2007.343618.

[14] A. Raja, X. Su, Mobility handling in mac for wireless ad hoc networks, Wirel. Commun. Mob. Comput. 9 (3) (2009) 303–311. doi:10.1002/wcm.v9:3.
URL `http://dx.doi.org/10.1002/wcm.v9:3`

[15] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, G. Sukhatme, Robomote: enabling mobility in sensor networks, in: Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on, 2005, pp. 404–409. doi:10.1109/IPSN.2005.1440957.

[16] M. Laibowitz, J. A. Paradiso, Parasitic mobility for pervasive sensor networks, in: Proceedings of the Third International Conference on Pervasive Computing, PERVASIVE'05, Springer-Verlag, Berlin, Heidelberg, 2005, pp. 255–278. doi:10.1007/11428572_16.
URL `http://dx.doi.org/10.1007/11428572_16`

[17] W. W. V. Srinivasan, K.-C. Chua, Trade-offs between mobility and density for coverage in wireless sensor networks, in: Proceedings of the 13th Annual ACM International Conference on Mobile Computing and Networking, MobiCom '07, ACM, New York, NY, USA, 2007, pp. 39–50. doi:10.1145/1287853.1287860.
URL `http://doi.acm.org/10.1145/1287853.1287860`

[18] B. Liu, P. Brass, O. Dousse, P. Nain, D. Towsley, Mobility improves coverage of sensor networks, in: Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '05, ACM, New York, NY, USA, 2005, pp. 300–308. doi:10.1145/1062689.1062728.
URL `http://doi.acm.org/10.1145/1062689.1062728`

[19] Z. Chaczko, R. Klempous, J. Nikodem, M. Nikodem, Methods of sensors localization in wireless sensor networks, in: Engineering of Computer-Based Systems, 2007. ECBS '07. 14th Annual IEEE International Conference and Workshops on the, 2007, pp. 145–152. doi:10.1109/ECBS.2007.48.

[20] D.-H. Chae, K.-H. Han, K.-S. Lim, K.-H. Seo, K.-H. Won, W.-D. Cho, S.-S. An, Power saving mobility protocol for sensor network, in: Software Technologies for Future Embedded and Ubiquitous Systems, 2004. Proceedings. Second IEEE Workshop on, 2004, pp. 122–126. doi:10.1109/WSTFES.2004.1300426.

[21] W. Ye, J. Heidemann, D. Estrin, An energy-efficient mac protocol for wireless sensor networks, in: INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Vol. 3, 2002, pp. 1567–1576 vol.3. doi:10.1109/INFCOM.2002.1019408.

[22] H. Pham, S. Jha, Addressing mobility in wireless sensor media access protocol, in: Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004. Proceedings of the 2004, 2004, pp. 113–118. doi:10.1109/ISSNIP.2004.1417447.

[23] P. Raviraj, H. Sharif, M. Hempel, S. Ci, Mobmac - an energy efficient and low latency mac for mobile wireless sensor networks, in: Systems Communications, 2005. Proceedings, 2005, pp. 370–375. doi:10.1109/ICW.2005.56.

[24] S.-C. Choi, J.-W. Lee, Y. Kim, An adaptive mobility-supporting mac protocol for mobile sensor networks, in: Vehicular Technology Conference, 2008. VTC Spring 2008. IEEE, 2008, pp. 168–172. doi:10.1109/VETECS.2008.47.

[25] S. Hameed, E. Shaaban, H. Faheem, M. S. Ghoniemy, Mobility-aware mac protocol for delay-sensitive wireless sensor networks, in: Ultra Modern Telecommunications Workshops, 2009. ICUMT '09. International Conference on, 2009, pp. 1–8. doi:10.1109/ICUMT.2009.5345591.

[26] P. Lin, C. Qiao, X. Wang, Medium access control with a dynamic duty cycle for sensor networks, in: Wireless Communications and Networking Conference, 2004. WCNC. 2004 IEEE, Vol. 3, 2004, pp. 1534–1539 Vol.3. doi:10.1109/WCNC.2004.1311671.

[27] B. Khan, F. Ali, Collision free mobility adaptive (cfma) mac for wireless sensor networks, Telecommunication Systems 52 (4) (2013) 2459–2474. doi:10.1007/s11235-011-9566-5.
URL `http://dx.doi.org/10.1007/s11235-011-9566-5`

[28] V. Rajendran, K. Obraczka, J. J. Garcia-Luna-Aceves, Energy-efficient collision-free medium access control for wireless sensor networks, in: Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys '03, ACM, New York, NY, USA, 2003, pp. 181–192. doi:10.1145/958491.958513.
URL `http://doi.acm.org/10.1145/958491.958513`

[29] M. Ali, T. Suleman, Z. Uzmi, Mmac: a mobility-adaptive, collision-free mac protocol for wireless sensor networks, in: Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International, 2005, pp. 401–407. doi:10.1109/PCCC.2005.1460597.

[30] A. El-Hoiydi, J.-D. Decotignie, Wisemac: an ultra low power mac protocol for the downlink of infrastructure wireless sensor networks, in: Computers and Communications, 2004. Proceedings. ISCC 2004. Ninth International Symposium on, Vol. 1, 2004, pp. 244–251 Vol.1. doi:10.1109/ISCC.2004.1358412.

[31] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proceedings of the 2Nd International Conference on Embedded Networked Sensor Systems, SenSys '04, ACM, New York, NY, USA, 2004, pp. 95–107. doi:10.1145/1031495.1031508.
URL `http://doi.acm.org/10.1145/1031495.1031508`

[32] M. Buettner, G. V. Yee, E. Anderson, R. Han, X-mac: A short preamble mac protocol for duty-cycled wireless sensor networks, in: Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, SenSys '06, ACM, New York, NY, USA, 2006, pp. 307–320. doi:10.1145/1182807.1182838.
URL `http://doi.acm.org/10.1145/1182807.1182838`

[33] Y. Sun, O. Gurewitz, D. B. Johnson, Ri-mac: A receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks, in: Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems, SenSys '08, ACM, New York, NY, USA, 2008, pp. 1–14. doi:10.1145/1460412.1460414.
URL `http://doi.acm.org/10.1145/1460412.1460414`

[34] P. Dutta, S. Dawson-Haggerty, Y. Chen, C.-J. M. Liang, A. Terzis, Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless, in: Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, ACM, New York, NY, USA, 2010, pp. 1–14. doi:10.1145/1869983.1869985.
URL `http://doi.acm.org/10.1145/1869983.1869985`

[35] L. Tang, Y. Sun, O. Gurewitz, D. B. Johnson, Em-mac: A dynamic multichannel energy-efficient mac protocol for wireless sensor networks, in: Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '11, ACM, New York, NY, USA, 2011, pp. 23:1–23:11. doi:10.1145/2107502.2107533.
URL `http://doi.acm.org/10.1145/2107502.2107533`

[36] P. Ba, B. Gueye, I. Niang, T. Noel, Mox-mac: A low power and efficient access delay for mobile wireless sensor networks, in: Wireless and Mobile Networking Conference (WMNC), 2011 4th Joint IFIP, 2011, pp. 1–6. doi:10.1109/WMNC.2011.6097246.

[37] L. Bing, Y. Ke, Z. Lin, Z. Huimin, Mac performance and improvement in mobile wireless sensor networks, in: Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference on, Vol. 3, 2007, pp. 109–114. doi:10.1109/SNPD.2007.267.

[38] L. Bernardo, H. gua, M. Pereira, R. Oliveira, R. Dinis, P. Pinto, A mac protocol for mobile wireless sensor networks with bursty traffic, in: Wireless Communications and Networking Conference (WCNC), 2010 IEEE, 2010, pp. 1–6. doi:10.1109/WCNC.2010.5506145.

[39] C. Bormann, G. Mulligan, Ipv6 over low power wpan (6lowpan), http://www.ietf.org/html.charters/6lowpan-charter.html (Sept. 2012).

[40] T. Camilo, P. Pinto, A. Rodrigues, J. Silva, F. Boavida, Mobility management in ip-based wireless sensor networks, in: World of Wireless, Mobile and Multimedia Networks, 2008. WoWMoM 2008. 2008 International Symposium on a, 2008, pp. 1–8. doi:10.1109/WOWMOM.2008.4594824.

[41] M.-K. Shin, H.-J. Kim, L3 mobility support in large-scale ip-based sensor networks (6lowpan), in: Advanced Communication Technology, 2009. ICACT 2009. 11th International Conference on, Vol. 02, 2009, pp. 941–945.

[42] C. Perkins, Ip mobility support for ipv4, IETF RFC3344 (Aug. 2002).
URL `http://www.ietf.org/rfc/rfc3344.txt`

[43] D. Johnson, C. Perkins, J. Arkko, Mobility support in ipv6, IETF RFC3775 (Jun. 2004).

[44] H. Soliman, C. Castelluccia, K. E. Malki, L. Bellier, Hierarchical mobile ipv6 mobility management, IETF, RFC 4140 (Aug. 2005).
URL `http://www.ietf.org/rfc/rfc4140.txt`

[45] A. Diab, A. Mitschele-Thiel, J. Xu, Performance analysis of the mobile ip fast authentication protocol, in: Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '04, ACM, New York, NY, USA, 2004, pp. 297–300. doi:10.1145/1023663.1023715.
URL `http://doi.acm.org/10.1145/1023663.1023715`

[46] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, Proxy mobile ipv6, IETF, RFC 5213 (Aug. 2008).
URL `http://tools.ietf.org/rfc/rfc5213.txt`

[47] R. Koodli, Mobile ipv6 fast handovers, IETF, RFC 5568 (Jul. 2009).
URL `http://tools.ietf.org/rfc/rfc5568.txt`

[48] M. Riegel, Q. Xie, M. Tuexen, Mobile sctp, draft-riegeltuexen-mobile-sctp-09.tx, Internet-Draft (Nov. 2007).

[49] G. Bag, M. T. Raza, K.-H. Kim, S.-W. Yoo, Lowmob: Intra-pan mobility support schemes for 6lowpan, Sensors 9 (7) (2009) 5844–5877. doi:10.3390/s90705844.
URL `http://www.mdpi.com/1424-8220/9/7/5844`

[50] M. Ha, D. Kim, S.-H. Kim, S. Hong, Inter-mario: A fast and seamless mobility protocol to support inter-pan handover in 6lowpan, in: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 2010, pp. 1–6. doi:10.1109/GLOCOM.2010.5683658.

[51] R. Silva, J. S. Silva, An adaptation model for mobile ipv6 support

in lowpans, Internet Engineering Task Force, Internet-Draft draft-silva-6lowpan-mipv6-00 (May 2009).

[52] X. Wang, S. Zhong, R. Zhou, A mobility support scheme for 6lowpan, Computer Communications 35 (3) (2012) 392 – 404. doi:http://dx.doi.org/10.1016/j.comcom.2011.11.001.

[53] G. R. Sinniah, Z. S. Kamalurradat, U. Sarwar, K. H. Teo, Node mobility scheme for ip and non-ip wireless personal area network nodes using 6lowpan, in: SENSORCOMM 2013, The Seventh International Conference on Sensor Technologies and Applications, 2013, pp. 83–89.

[54] M. Islam, M. Hassan, E.-N. Huh, Sensor proxy mobile ipv6 (spmipv6) - a framework of mobility supported ip-wsn, in: Computer and Information Technology (ICCIT), 2010 13th International Conference on, 2010, pp. 295–299. doi:10.1109/ICCITECHN.2010.5723872.

[55] A. Jabir, S. Subramaniam, Z. Ahmad, N. Hamid, A cluster-based proxy mobile ipv6 for ip-wsns, EURASIP Journal on Wireless Communications and Networking 2012 (1) (2012) 1–17. doi:10.1186/1687-1499-2012-173.
URL `http://dx.doi.org/10.1186/1687-1499-2012-173`

[56] K. Srinivasan, P. Levis, Rssi is under appreciated, in: In Proceedings of the Third Workshop on Embedded Networked Sensors (EmNets, 2006.

[57] S. J. Halder, W. Kim, A fusion approach of rssi and lqi for indoor localization system using adaptive smoothers, Journal of Computer Networks and Communications 2012. doi:10.1155/2012/790374.

[58] Arm embedded linux boards. [online] available: (2012).
URL `http://www.embeddedarm.com/products/arm-sbc.php`

[59] S. Bouckaert, E. De Poorter, P. De Mil, I. Moerman, P. Demeester, Interconnecting wireless sensor and wireless mesh networks: Challenges and strategies, in: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, 2009, pp. 1–7. doi:10.1109/GLOCOM.2009.5425861.

[60] Telosb datasheet. [online]. available: (2012).
URL `www.willow.co.uk/TelosB_Datasheet.pdf`

[61] A. O. et al, Aware project (Jul 2009).
URL `http://www.aware-project.net/`

[62] A. Erman, L. Hoesel, P. Havinga, J. Wu, Enabling mobility in heterogeneous wireless sensor networks cooperating with uavs for mission-critical management, Wireless Communications, IEEE 15 (6) (2008) 38–46. doi:10.1109/MWC.2008.4749746.

[63] G. H. et al, Citysense an open urban-scale sensor network testbed (Jul 2009).
URL `http://www.citysense.net/`

[64] S. F. et al, Wisebed wireless sensor network testbeds (Jul 2009).
URL `http://www.wisebed.eu/`

[65] R. H. et al, Cooperative hybrid objects sensor networks (Sept 2009).
URL `http://www.chosen.eu/`

[66] L. Herault, D. Babb, R. Tafazolli, e-sense - capturing ambient intelligence for mobile communications through wireless sensor networks (Sept 2006).
URL `www.ist-esense.org`

[67] S. Basagni, A. Carosi, E. Melachrinoudis, C. Petrioli, Z. Wang, Controlled sink mobility for prolonging wireless sensor networks lifetime, Wireless Networks 14 (6) (2008) 831–858. doi:10.1007/s11276-007-0017-x.
URL `http://dx.doi.org/10.1007/s11276-007-0017-x`

[68] G. OHare, M. Dragone, G. Amato, M. Broxvall, A. Renteria, R. Tarazon, M. McGinnity, S. Chessa, A. Micheli, C. Clissman, G. D. Vito, Rubicon self-learning robot ecology (Sept 2011).
URL `http://fp7rubicon.eu/`

[69] L. HERAULT, A. GLUHAK, F. BOURDEL, Sensei integrating the physical with the digital world of the network of the future (Sept 2007).
URL `http://www.ict-sensei.org`

[70] M. Conti, R. Di Pietro, L. V. Mancini, A. Mei, Emergent properties: Detection of the node-capture attack in mobile wireless sensor networks, in: Proceedings of the First ACM Conference on Wireless Network Security, WiSec '08, ACM, New York, NY, USA, 2008, pp. 214–219. doi:10.1145/1352533.1352568.
URL `http://doi.acm.org/10.1145/1352533.1352568`

[71] S. Basagni, M. Nati, C. Petrioli, R. Petroccia, Rome: Routing over mobile elements in wsns, in: Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE, 2009, pp. 1–7. doi:10.1109/GLOCOM.2009.5425430.

[72] I. Tomi?, S. Kr?o, D. Vu?kovi?, A. Gluhak, P. Navaratnam, Sensei traffic impact on mobile wireless networks, in: Towards the Future Internet - Emerging Trends from European Research, Books Online, IOS Press, 2010.

[73] S. Nikoletseas, S. Krco, Z. Shelby, A. Ruzzeli, D. Arvid, J. ArvindL, Hobnet holistic platform design for smart buildings of the future internet (Sept 2010).
URL `http://www.hobnet-project.eu/`