AUTHOR QUERY FORM

Dear Author,

Please check your proof carefully and mark all corrections at the appropriate place in the proof (e.g., by using on-screen annotation in the PDF file) or compile them in a separate list. Note: if you opt to annotate the file with software other than Adobe Reader then please also highlight the appropriate place in the PDF file. To ensure fast publication of your paper please return your corrections within 48 hours.

For correction or revision of any artwork, please consult http://www.elsevier.com/artworkinstructions.

Any queries or remarks that have arisen during the processing of your manuscript are listed below and highlighted by flags in the proof. Click on the 'Q' link to go to the location in the proof.

Location in article	Query / Remark: **click on the Q link to go** **Please insert your reply or correction at the corresponding line in the proof**
Q1	Please confirm that given names and surnames have been identified correctly.

Thank you for your assistance.

A proposal for proxy-based mobility in WSNs

Ricardo Silva *, Jorge Sa Silva, Fernando Boavida

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

ARTICLE INFO

ABSTRACT

Inability to meet the key requirement of efficient mobility support is becoming a major impairment of wireless sensor network (WSN). Many critical WSN applications need not only reliability, but also the ability to adequately cope with the movement of nodes between different sub-networks. Despite the work of IETF's 6lowPAN WG and work on the use of MIPv6 (and many of its variants) in WSNs, no practical mobility support solution exists for this type of networks. In this paper we start by assessing the use of MIPv6 in WSNs, considering soft and hard handoff, showing that, although feasible in small networks, MIPv6 complexity leads to long handoff time and high energy consumption. In order to solve these problems, we propose a proxy-based mobility approach which, by relieving resource-constrained sensor nodes from heavy mobility management tasks, drastically reduces time and energy expenditure during handoff. The evaluation of both MIPv6 and the proposed solution is done by implementation and simulation, with a varying number of nodes, sinks and mobility strategies.

© 2012 Published by Elsevier B.V.

1. Introduction

The use of IPv6 in WSNs is nowadays clearly identified as an added value. The capacity to establish communication with any single mote from any remote point extends the applicability of such networks and substantiates the concept of the Internet of Things (IoT), reducing the gap between the real and virtual worlds. Nevertheless, in order to effectively use IPv6 in WSNs, some adaptation is needed, so that the scarce energy and processing power resources of sensor nodes are not rapidly depleted. Although the work of IETF's 6lowPAN Working Group has led to clear advances in the area, mobility is still an open issue in WSNs, despite the fact that the use of IPv6 opens up the possibility of resorting to MIPv6 [1].

The fact is that current standards do not efficiently support mobility, and this poses considerable obstacles to their use in WSNs, especially if they are being used for critical applications requiring high levels of reliability and performance. If achieved, adequate mobility support would be fundamental for fulfilling the long-made promise of WSN deployment in scenarios as critical and important as military, health or transport scenarios, for which the number of real implementations is still very low.

In the scope of the GINSENG European Project, in which the authors are involved, WSNs are being used to monitor industrial processes within Petrogal's oil refinery located in Sines, Portugal. As this is a critical scenario, the underlying WSN must be completely reliable, controlled and fault-tolerant. One of the scenarios under consideration requires the constant monitoring of the employees' vital signs while they are working within hazardous areas, which poses several challenges, such as reliability, availability and, last but not least, mobility. The use of adjacent WSNs operating in different radio channels and using TDMA protocols, under 6lowPAN, leads to complex hard handoffs.

In order to address the referred challenges, guaranteeing a unified solution for critical and non-critical scenarios, in this paper we propose and evaluate the use of a proxy mesh network, complementary to the base WSN network, with the aim of assisting resource-constrained, mobile WSN nodes in their overall operation. With the purpose of clarifying the rationale for this proposal, the presented solution is preceded by an evaluation of the feasibility and limitations of MIPv6 in WSNs. This approach allows not only assessing the viability of deploying MIPv6 in WSN motes, but also to evaluate and compare node-based mobility solutions with the proposed network-assisted, proxy-based mobility solution. The assessment of both solutions under consideration in this paper was done by implementation and simulation.

Having in mind the stated goals and approaches, this paper is organized as follows. The next section presents background on mobility in WSNs. Section 3 introduces the case study and associated problem statement. The Network of Proxies (NoP) proposal is presented in Section 4, followed by two extensive evaluation sections in which the MIPv6 node-based mobility solution and the NoP network-assisted mobility solution are successively studied by prototyping (Section 5) and by simulation (Section 6). Section 7 summarizes the obtained results and presents guidelines for further research. Appendices A and B present implementation and simulation details, respectively.

* Corresponding author. Tel.: +351 239 790000; fax: +351 239 701266.
E-mail addresses: rnsilva@dei.uc.pt (R. Silva), sasilva@dei.uc.pt (J.S. Silva), boavida@dei.uc.pt (F. Boavida).

2. Background

Mobility in WSNs has been approached from different perspectives. Some authors [2] proposed sink node mobility based on three distinct methods: Mobile Base Station (MBS), Mobile Data Collector (MDC) and Rendezvous-Based Solution. With MBS the sink node is capable of moving across the network, increasing the coverage and decreasing the number of hops needed to reach each node [3]. MDC, in turn, takes advantage of special nodes to perform on-demand collection, avoiding data travel through several hops [4]. The Rendezvous-Based Solution is a mix of MBS and MDC [5].

Regarding node mobility, the authors in [6] classified it into two types: weak and strong. Weak mobility is the forced mobility of nodes caused by deaths and replacements. Strong mobility occurs when nodes are attached to a mobile body [7] or when they move by themselves [8].

During its lifetime, a mobile node can move within the same network, which is known as intra-mobility, or between different networks, performing inter-mobility. In the latter case, the mobile node can move between networks operating in the same channel but in different domains, or between networks operating in different channels and domains. If the same channel is used, the node can perform a soft handoff, which means that the node is able to release the connection to the previous network only after having established a connection to the new network. However, if the new network operates in a different channel, the node is forced to perform a hard handoff and, thus, has to break the connection to the previous network before connecting to the new one. The two options mainly depend on the MAC and Network Layers.

By default, WSNs are IEEE802.15.4 [9] based, which does not provide mobility support.

In an attempt to save energy and maximize the network lifetime, several MAC protocols use complex duty cycle schemes. Although most of them are not prepared to handle mobility, MS-MAC [10], MAMAC [11], MH-MAC [12] and MMAC [13] are examples of MAC-layer protocols prepared to adapt duty cycle schemes to mobility needs. Analytically or by simulation, they were proven to behave better than standard MAC protocols in mobility contexts. Nevertheless, none of them was effectively implemented and used in a real context, as dealing with advanced duty cycle mechanisms and mobility at the same level proved to be prohibitive in terms of protocol complexity. Moreover, none of the referred protocols was designed to effectively guarantee the performance required by critical applications.

Due to the difficulty in efficiently handling mobility at the MAC layer, some proposals rely on a cross-layer approach, by sharing mobility data across layers [14]. It should be noted that Zigbee also provides mobility support for WSNs, although it does so in an inefficient way, as demonstrated in [15].

Well-known IP mobility protocols should also be taken into account. These can be split into two groups: mote-based mobility solutions, in which mobility functions are solely the responsibility of the WSN motes, and network-assisted mobility solutions, in which the network infrastructure performs some or all of the mobility tasks on behalf of sensor nodes, relieving them from this work.

Standard MIPv6, a mote-based solution, provides mobility management and, in line with the 6lowPAN concept, it can be applied to WSNs. In [16] some considerations on mobility in 6lowPANs are made, while in [17] we proposed an adaptation of MIPv6 to 6lowPANs, subsequently evaluated in [18].

In addition to the original version of MIPv6, several variations can also be considered, even though none of them was specifically designed for WSNs. Examples of mote-based variations are: FMIPv6 [19], which uses information from the MAC layer to accelerate the process; and MIFAv6 [20], which delegates the authentication process on the Access Routers (AR).

PMIPv6 [21], which, in line with the NETLMM IETF Working Group, is a network-assisted solution, relies on specific network entities to perform the handoff process, putting the main load of the process on the network side. The network-based approach is further explored in [22], in which the authors present a study on the benefits of enhancing mobile networks with complementary infrastructures. The conclusions were varied and dependent on the network type and size but, in general, the addition of mesh nodes proved to be beneficial. The authors in [23] arrive at a conclusion that corroborates the conclusions in [22], and which our own experience confirms: it is not feasible to include advanced routing mechanisms, security, mobility, debugging, etc., in each mote and therefore, the solution should be to delegate those functions on the network, leaving the motes only for sensing or actuating.

In line with this, in [24] a preliminary version of a proxy-based mobility solution was proposed, in which the mobility proxies were interconnected by a shared backbone, which, nevertheless, limited the flexibility of the whole solution. We then designed a solution that has some similarities with PMIPv6, although simpler, with fewer messages and optimized for WSNs. PMIPv6 was not designed for sensor networks and, therefore, does not take hardware and resource restrictions into consideration. This leads to a protocol that, even though it is a network-assisted solution, still requires several messages between mobile nodes (MN) and the network.

Thus the distinguishing features of the proposed NoP solution are: (1) network-assisted mobility, in order to free sensor nodes from resource-demanding, energy-expensive and time-consuming mobility tasks both at the MAC layer and at the IP layer, in line with the conclusions of recent research work from other authors [22] [23]; and (2) optimization of the proposed network-assisted mobility approach to the requirements of WSNs.

3. Case study

Without loss of generality, the proposals presented in this paper are being developed and tested in the scope of the GINSENG FP7 European project, which uses an oil refinery as case study.

Within the oil refinery, workers are exposed to hazardous environments, in highly critical areas. In this scenario, it is absolutely essential to resort to a real-time monitoring system in order to constantly follow the workers vital signs, during their daily activity at the refinery. All workers use a smart shirt capable of reading ECG data, breathing rate and position through a 3D accelerometer, the latter with the aim of quickly detecting any fallen worker.

Currently, the GINSENG project has two partially overlapping WSNs running at the refinery, monitoring an area of about 3.1 square kilometers. Fig. 1 depicts the deployed scenario.

Both networks are located in critical areas, monitoring pressures, flows and gases, in addition to controlling various valves. An industrial control room is located in the overlap area, so that sensor nodes from any of the two networks can communicate with it.

As real-time operation is a must, these networks use a TDMA MAC protocol, running on a tree topology [25] specifically designed for this purpose. To avoid long epochs, each GINSENG network is limited to 25 nodes and 3 tree levels. With this configuration, epochs have a duration of 1 s. The 25-node per network limitation led to the deployment of two separate networks instead of using only one bigger network. Therefore, these networks must operate in different channels, in order to assure that no interference occurs between them.

Workers can move freely in the whole area, thus generating intra- and inter-mobility events. The underlying mobility protocol must be capable of performing handoff between the two networks, operating in different radio channels. Consequently, mobile nodes need to reboot the transceiver to handoff from one network to the
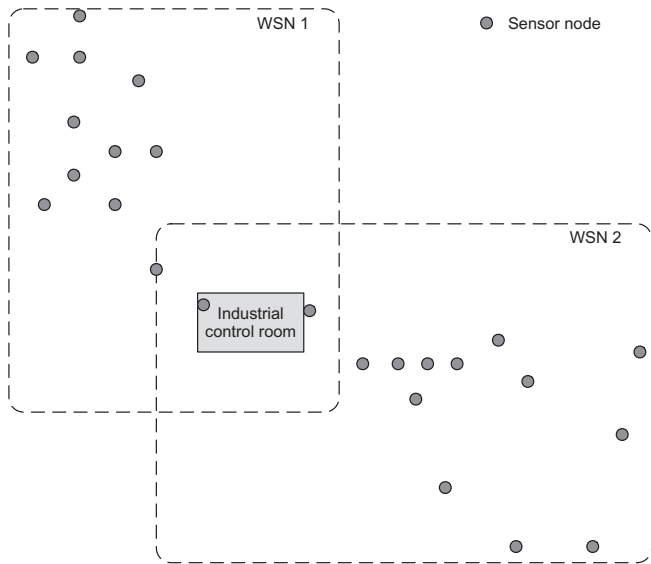
3

Fig. 1. GINSENG deployment at Petrogal's oil refinery.

other, performing a hard handoff. The main requirement is to per-form this process as fast as possible, minimizing the duration of the disconnected period. This is one of the aims of the Network of Proxies (NoP) proposal, presented in the next section.

4. Network of Proxies

4.1. Concept

Typically, WSNs are made of several resource-constrained nodes and one sink node. The sink node is not subject to the same energy, memory or processing power restrictions of sensor nodes and, thus, it is used as activity coordinator and central communica-tions hub, providing connectivity to the outside world.

The Network of Proxies (NoP) proposal extends the basic WSN concept by superimposing a flexible, wireless ad hoc mesh network made of resource-unconstrained proxies on the base WSN. These proxies are intended to be simple Linux-based embedded devices, using hardware platforms currently available in the market with a unit price range of 100–150 USD [26], equipped with ARM CPUs with at least 200 MHz, 32 MB of RAM and wireless communication.

This complementary wireless mesh network can use standard IEEE802.11 [27], thus facilitating its constitution and operation. IEEE 802.11 is a well-known technology, ease to use, already sup-ported for most mobile devices. By using a different technology in the right RF channels we avoid, on one hand, interferences with the WSN side, which is running IEEE802.15.4 by default and, on the other hand, we can support higher transmission rates.

Proxies are intended to operate in a seamless mode, capable of assisting WSNs without interfering with the application. Thus, proxies should operate in a plug-and-play, autonomous fashion, capable of performing self-integration in an existent NoP.

It should also be noted that the extra cost of deploying an NoP is perfectly acceptable when targeting critical applications, whose performance control is the main requirement. Nevertheless, the NoP concept is also applicable to any application scenario for which performance under mobility conditions is a requirement.

NoP is in line with and an evolution of the concepts presented in [21,22], adapted to WSNs. NoP proxies take care of time-consuming, communications-intensive and processor-demanding tasks – such as tasks related with mobility management – on behalf of the con-strained sensor nodes. The NoP concept is illustrated in Fig. 2.

If more than one network are placed together, the existing proxy mesh networks are capable of merging into one. Hence, the NoP concept comprises two different modes, namely, standalone mode and extended mode. Fig. 3 illustrates the NoP extended mode.

When dealing with mobility tasks, there are significant differ-ences between standalone mode NoP operation (Fig. 2) and ex-tended mode NoP operation (Fig. 3). In the former case, if motes move out of the domain, the mobility protocol needs to be per-formed through the sink node, which increases latency and, there-fore, the probability of packet losses due to longer disconnection periods. However, if there is the possibility to establish an extended NoP, proxies become able to exchange information and perform mobility management tasks in parallel with or in advance of node inter-domain movement. Once in a new domain, mobile nodes can already be prepared to quickly connect to the new parent, rebooting the transceiver if necessary, and loading the new configuration, which can also have been provided before they moved away from the previous network, thus optimizing inter-mobility operations.

In the presence of intra-mobility, the role of proxies becomes simpler, as they are only required to detect movement and update parent–child connections when needed. Additionally, in this case the existence of an extended NoP does not provide an added value, because it is not necessary to exchange information about the motes with other domains or to update high-level information.
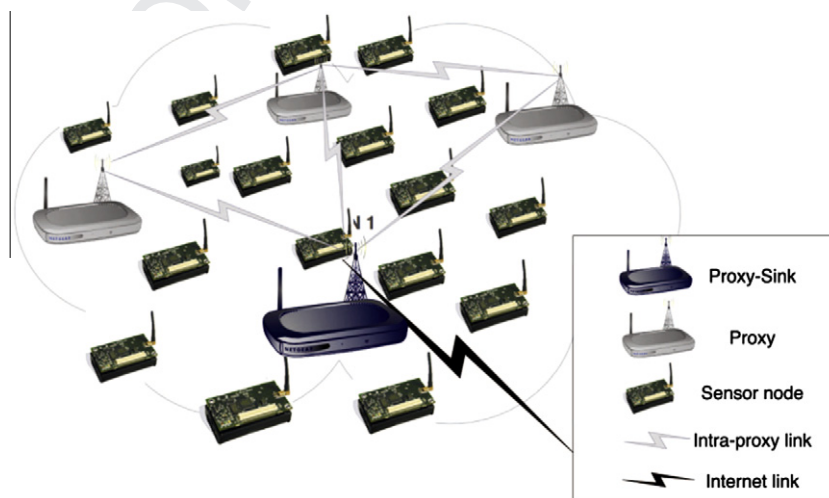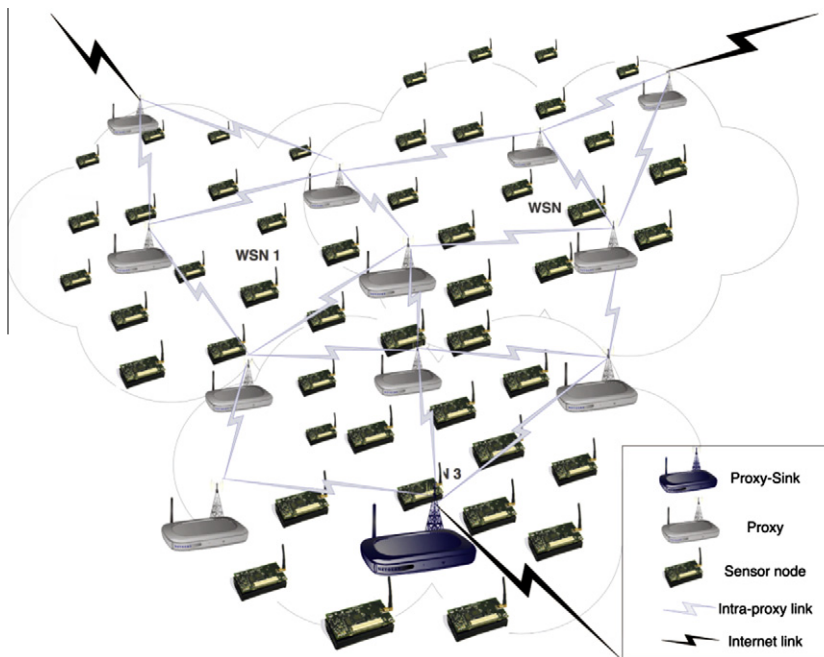

Fig. 2. The Network of Proxies concept.

R. Silva et al./Computer Communications xxx (2012) xxx–xxx
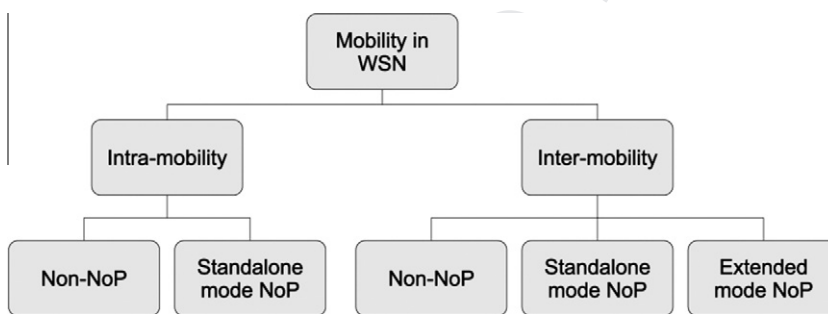

Fig. 3. Extended mode NoP.

Fig. 4. WSN mobility options using NoP concept.

280 With the NoP concept, we aim to assure that a network-based
281 mobility protocol will be capable of guaranteeing performance tar-
282 gets even in the presence of highly mobile nodes, by minimizing
283 handoff latency and, thus, avoiding packet losses and connection
284 disruption.

285 The proposed NoP approach is also designed to be interoperable
286 with all types of sensor networks, whether they are NoP-based or
287 not. Considering adjacent networks, we assume the possibility of
288 dealing with inter-domain mobility between a standalone NoP net-
289 work and a non-NoP network. Fig. 4 summarizes the possible
290 modes.

291 Each NoP proxy monitors a set of adjacent mobile sensor nodes
292 keeping a record of the links quality. Based on this record, each
293 proxy is responsible for deciding whether a specific sensor node
294 must handoff or not. Then, using a mobility protocol, such as
295 MIPv6, the proxy negotiates the handoff with the next network,
296 through the extended NoP. Once in the possession of all mobility
297 information, the proxy notifies the sensor node using just a single
298 message, to assure a fast handoff, even in a TDMA scenario. When
299 receiving the notification, the sensor node acts according to the re-
300 ceived information, and reboots the transceiver, if necessary.

301 In order to optimize the NoP operation, each proxy maintains a
302 table with information on each mobile sensor node located in its
303 range, and also information on sensor nodes under adjacent prox-

304 ies. Each proxy automatically broadcasts this data through the
305 mesh network.

4.2. Protocol

307 The use of a proxy capable of performing the heavier tasks on
308 behalf of mobile nodes can drastically reduce the energy consump-
309 tion and the time to handoff, even between networks operating in
310 different channels. The NoP concept allows the handoff procedure
311 to be entirely the responsible of the local proxy, i.e., the proxy that
312 has the best link to the specific MN. The local proxy is responsible
313 for the main mobility management tasks, which include: detecting
314 the need to handoff; requesting information from the next net-
315 work, such as channel and Care-of Address, through the Network
316 of Proxies; registering the new Care-of Address in the Home Agent
317 of the mobile node; and finally, providing that information to the
318 mobile node. Once in the possession of such data, the mobile node
319 can simply leave the current network and reboot the transceiver,
320 loading the new configuration.

321 On detecting a deteriorating link, a proxy must start by notify-
322 ing its proxy neighbors, requesting information on the quality of
323 the respective links to that specific mobile sensor node (MN). Proxy
324 neighbors with better link quality values reply to the request,
325 which allows the requesting proxy to decide on the next proxy
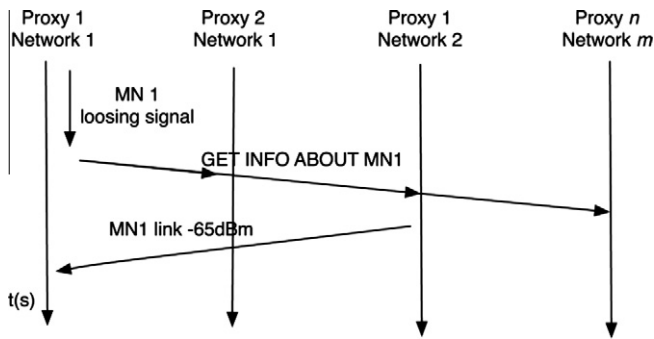
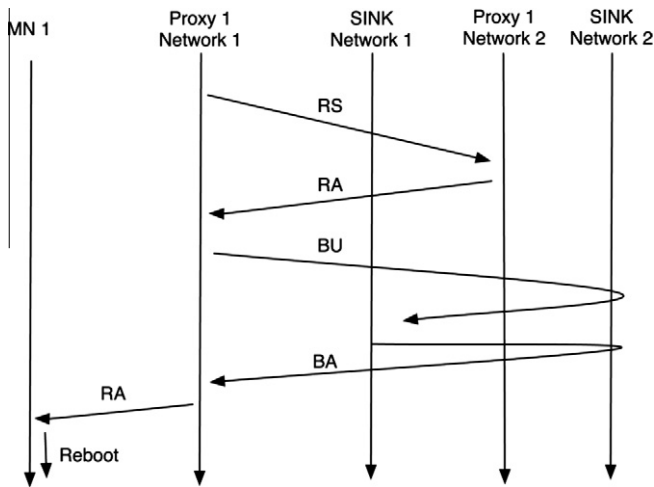Fig. 5. Extended NoP link quality info request.

Fig. 6. MIPv6 without Return Routability, performed by proxies on behalf of the MN.

and to start the handoff procedure with the network of the proxy that answered with the best link, on behalf of the MN. All the necessary exchanges are performed through the NoP mesh network. After the process is concluded, the former proxy notifies the MN to immediately change network.

Fig. 5 illustrates a procedure where proxy 1 of network 1 requests information on the quality of the links to MN1 from all its neighbor proxies. In this example, the proxy received the best answer from proxy 1 of network 2. Subsequently (see Fig. 6), proxy 1 of the network 1 initiates the handoff, which, in this example, is performed using MIPv6 without Return Routability.

In Fig. 6 we can see a set of Neighbor Discovery [28] messages, namely Router Solicitation (RS) and Router Advertisement (RA), and also Binding Update (BU) and Binding Acknowledge (BA) MIPv6 messages. The final notification to MN1 is done using an RA message. This message includes the channel information in the reserved field of the prefix information option.

Since the Home Proxy entirely takes up the role of the mobile node, the protocol stays basically the same. However, for the mobile node, the presence of an NoP drastically simplifies its handoff procedure, as it only receives one single message and then reboots the transceiver, loading the final configuration. The impact of mobility on sensor nodes is, thus, minimized.

4.3. Application to GINSENG

Before starting the NoP deployment, we ran several preliminary experiments to characterize radio communications in the demanding industrial environment of the case study [29].

Both mobility types – intra and inter-mobility – are required in the context of the GINSENG project. Nevertheless, as mentioned in the previous section, inter-mobility with hard-handoff is the most critical case. An Extended mode NoP was installed in order to cover all of the area presented in Fig. 1. Since GINSENG networks already operate with a topology control module, responsible for periodically checking the existing links, each proxy can act based on the information collected by this module. Note that proxies operate in promiscuous mode and do not have the energy restrictions of sensor nodes. As the WSN networks in the GINSENG project operate in different channels, hard handoff must be performed, and mobile nodes must reboot the transceiver to connect to a new network.

5. Prototype-based evaluation

While moving along a specific path, a mobile sensor node can cross several networks. Some of them might operate in the same channel and domain, while others might operate in different domains or even channels. From an application perspective, such details are irrelevant and only connectivity is important. Therefore, the handoff process must be completely seamless to applications, which requires a mobility management abstraction. This abstraction can be provided by well-known protocols, such as MIPv6 or any of its variants, e.g., FMIPv6, MIFAv6 or PMIPv6. Although MIPv6 is a well-known protocol that provides standard mobility support, its application to WSNs has not been sufficiently explored, mainly due to the fact that MIPv6 was not engineered for use in resource-constrained networks.

In this section we start by evaluating the use of MIPv6 in WSNs, measuring the required time to handoff between two networks and the energy spent by mobile nodes in order to perform this task. We chose not to evaluate packet losses in this context since this is an application-dependent metric.

In order to assess MIPv6 in WSNs, we implemented core MIPv6 functionality over the Contiki WSN operating system [30]. By core MIPv6 functionality we mean Router Advertisements, with the Home Agent flag activated, Binding Update, Binding Acknowledgment, Home Agents and respective Mobility Binding Tables. This implementation was done in C language and adapted to the Contiki net structure. To trigger the handoff we used Router Solicitation messages, as defined in Neighbor Discovery [28]. At this stage we did not implement the Return Routability procedure, which was left for future work.

The main goal of the first two rounds of tests was to evaluate the feasibility of node-based MIPv6 while performing soft-handoff and hard-handoff. In the first round we assessed MIPv6 when a single mobile node moved between two networks belonging to different domains, although operating in the same channel. In the second round, we implemented, in a lab environment, a setup equal to the one used in the case study, where adjacent networks operated in different domains and channels. This means that the handoff process required that mobile nodes rebooted the transceiver in order to load the new configuration in a new channel, which led to a disconnection period.

Although the results of the first two rounds of tests confirmed the feasibility of MIPv6 in WSNs, they showed that it is achieved at a high price in terms of handoff time and energy use. Thus, a third round of tests was performed, oriented towards the assessment of the NoP proposal.

All evaluations presented in this section were implemented and tested in a real platform, constituted by TelosB motes [31], MSP430-based [32], using the CC2420 transceiver [33]. In all tests, handoff was triggered when the RSSI parameter dropped below the -80dBm threshold. Further implementation details are provided in Appendix A.
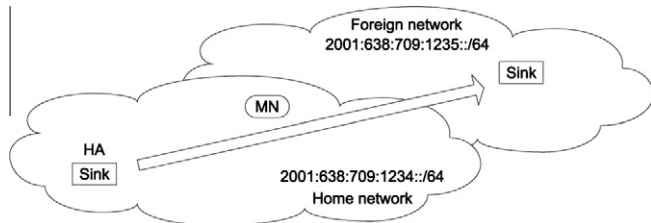
Fig. 7. Soft-handoff between different IPv6 domains.

It should be noted that the tests were done in a lab environment due to two reasons: on one hand, access to the refinery deployment obeys to strict rules and planning, which limited the type and amount of experiments we intended to carry out. On the other hand, according to the project schedule, NoP capabilities will only be fully deployed in the refinery in February 2012, after extensive testing in the lab. This, nevertheless, has no impact on the validity and usefulness of results, as the used setup is virtually equal to the one in the field.

5.1. MIPv6 soft-handoff

The first round of tests, consisting of 40 test runs, was intended to assess the behavior of MIPv6 in WSNs, by measuring the time and energy a mobile node took to handoff between two networks, operating in the same channel (in this case channel 11), but in different IPv6 domains.
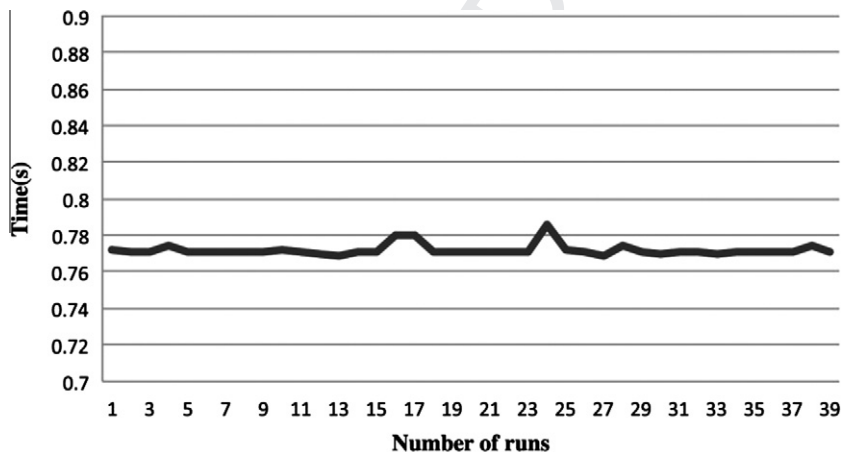
The mobile node was programmed to send an ICMP request message per second to simulate the application, to which the HA responded with an ICMP reply.

Fig. 7 depicts the network scenario, while Figs. 8 and 9 present the results for the handoff time and spent energy, respectively. The Contiki OS provides two alternatives for getting time values: clock time and rtimer. Although similar, rtimer has higher accuracy.

As we can observe in Fig. 8 in a graphical way and in Table 1 in a numerical way, the obtained average handoff time between two WSNs operating in different IPv6 domains and in the same channel is ~771 ms. Note that in this case, and due to the IPv6 capabilities, the handoff is simplified, as the node just adds another global address to its list of addresses. No transceiver reboots are needed and, therefore, the process is relatively fast and soft-handoff is possible.

Fig. 9 and Table 2 present the energy required to perform the handoff, in graphical and numerical terms, respectively. The energy values were obtained through the Contiki Energest module. Energest measures the time used by each component, which is then converted into energy based on the used hardware and using a reference of 3 V (the maximum, since motes were powered by USB). TelosB motes are composed of an MSP430 CPU, which consumes 1.8 mA, a CC2420 transceiver that requires 18.8 mA to receive and 17.4 to transmit, and a Low Power Mode (LPM), which consumes 0.426 mA.

As we can observe, to perform the handoff between two networks operating in the same channel, using MIPv6, the average energy expenditure was ~ 8.73 mJ.
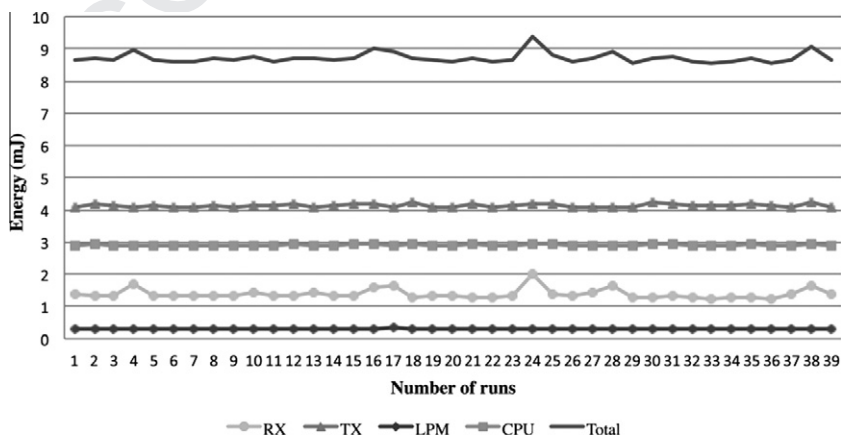

Fig. 8. MIPv6 soft handoff time.
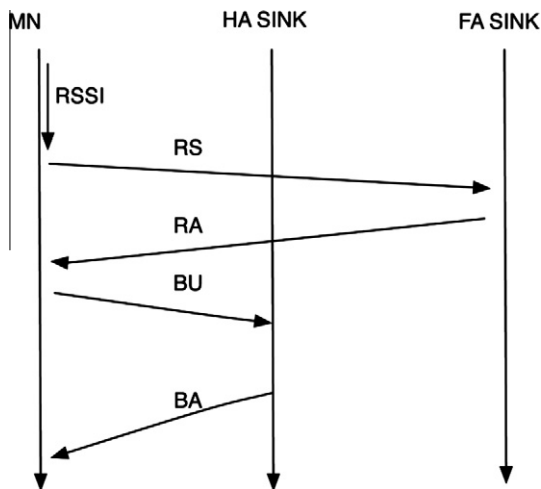

Fig. 9. Energy required by MIPv6 soft handoff.

7

Table 1
Summary of MIPv6 Soft-handoff time values.

	N	Range	Minimum	Maximum	Mean	Std. deviation
Time (ms)	40	17.5	768.7	786.1	771.9	3.3
Valid N	40					

Table 2
Summary of MIPv6 Soft-handoff energy values.

	N	Range	Minimum	Maximum	Mean	Std. deviation
RX (mJ)	40	0.7573	1.2255	1.9828	1.3957	0.1735
TX (mJ)	40	0.1466	4.0718	4.2183	4.1348	0.0457
LPM (mJ)	40	0.1585	0.2947	0.4532	0.3052	0.0244
CPU (mJ)	40	0.1734	2.8819	3.0553	2.8982	0.0289
Total (mJ)	40	0.9791	8.5528	9.5319	8.7340	0.2118
Valid N	40					

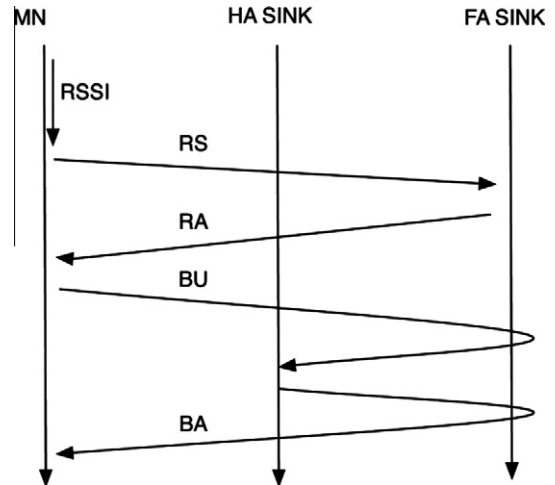**Fig. 11.** Hard handoff protocol exchanges.

Fig. 10. Soft handoff protocol.

The time and energy values presented above were obtained considering that the handoff process started at the moment the mobile node detected the RSSI level had reached the threshold value and ended when the Binding Acknowledge message had been received. Fig. 10 illustrates the handoff protocol.

From the first round of tests we can conclude that MIPv6 is not only feasible in WSNs, but also that MIPv6 allows relatively fast soft-handoffs when both networks operate in the same channel, as the handoff time is much lower than the usual data rate for continuous data monitoring applications, which is normally in the order of seconds.

5.2. MIPv6 hard-handoff

The second round of tests was intended to study the performance of MIPv6 when dealing with hard-handoff, i.e., when the mobile node moved between two networks operating in different channels, forcing a connection disruption. In this test suite we used the same application, protocol and methods to trigger the handoff. However, since each mote was only equipped with one transceiver, it was necessary to disconnect from the previous network, reboot the transceiver and connect to the new one. In addition, once operating in a different channel, the MN could not directly send a BU to the HA. Therefore, we implemented a routing mechanism between the two networks, in order to forward the Binding Update and Binding Acknowledge messages. Fig. 11 depicts this mechanism.

Note that in these tests we assumed the best case scenario, in which the mobile node knew the channel of the next network. In controlled deployments - such as the one corresponding to the case study, or in any scenario where there is the need to provide performance and reliability guarantees - such information can be previously provided to the sensor nodes. However, in other cases alternative methods must be implemented, in order to determine the next network channel. This, of course, will require additional time and energy expenditure.

In this scenario, network 1 was operating at channel 11 while network 2 was operating at channel 12. MN movement was performed from network 1 to network 2, and handoff was performed using the protocol depicted in Fig. 11, running on XMAC. Both clock time and rtimer were measured for 40 handoffs. The time to perform the complete handoff procedure, since the node detected that it had to handoff until it received the Binding Acknowledge, and the energy spent while performing such procedure, were measured and are presented in Figs. 12 and 13, respectively. Numerical values are presented in Tables 3 and 4, respectively.

In Table 3 we can observe that the average of ∼961 ms to handoff between different networks operating in different channels is approximately 25% higher than the average obtained to handoff between different networks operating in the same channel and different domains. However, depending on the application, this value might be as acceptable as the one in the previous case. For instance, considering the GINSENG case study, in which a worker in the refinery carries a sensor node that monitors his/her vital signs in real-time, at speeds lower than 10 m/s and networks with 100 m range each node stays a minimum of 10 s in each network, which is more than enough to handoff.

Analyzing the data in Table 4, we can observe that to perform a hard handoff, which includes a transceiver reboot, mobile nodes required an average energy expenditure of ∼11.76 mJ. We can also observe that the majority of the energy was spent on transmitting and processing tasks. Noting the total range of ∼4.3 mJ, and comparing it with the value in the previous round of tests (i.e., ∼0.97 mJ), we can conclude that, due to the routing mechanisms needed to complete the procedure, the time spent by the MN to perform handoff varies significantly. This variation might mean slightly higher or slightly lower values, occurred during normal operation, leading to a maximum energy expenditure of ∼14.43 mJ or a minimum of ∼10.12 mJ, which are within a perfectly acceptable range and prove the feasibility of the implemented MIPv6 version.
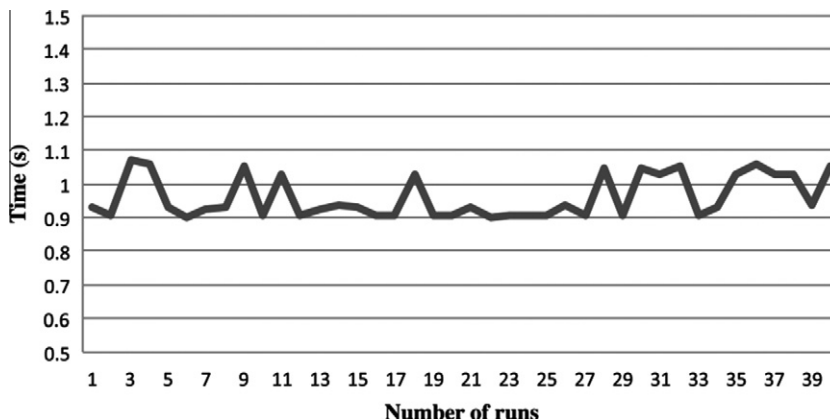
R. Silva et al./Computer Communications xxx (2012) xxx–xxx

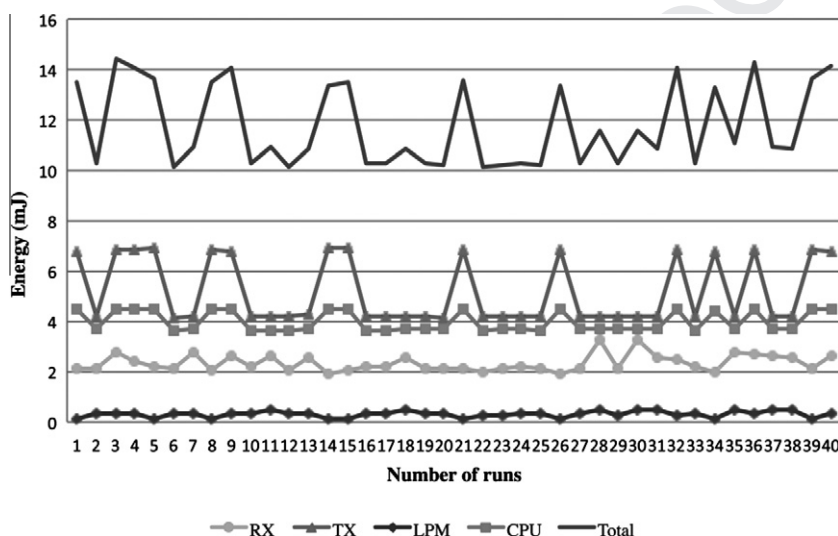**Fig. 12.** MIPv6 hard handoff time.

Fig. 13. Energy required by MIPv6 hard handoff.

Table 3
Summary of MIPv6 hard-handoff time values.

	N	Range	Minimum	Maximum	Mean	Std. deviation
Time (ms)	40	171.9	898.4	1070.3	961.3	64.2
Valid N	40					

Table 4
Summary of MIPv6 hard-handoff energy values.

	N	Range	Minimum	Maximum	Mean	Std. deviation
RX (mJ)	40	1.3701	1.8727	3.2427	2.3281	0.3351
TX (mJ)	40	2.7846	4.1100	6.8946	5.1792	1.2988
LPM (mJ)	40	0.3449	0.1242	0.4691	0.2891	0.1067
CPU (mJ)	40	0.8807	3.6024	4.4831	3.9619	0.3993
Total (mJ)	40	4.3083	10.1225	14.4308	11.7583	1.6227
Valid N	40					

5.3. MIPv6 hard handoff with NoP support

Although the previous experiments have shown that the use of a lightweight MIPv6 version in sensor nodes is feasible, if we extend it to include the return routability procedure or any other MIPv6 extension/improvement, the required time and energy will increase proportionally to the protocol complexity. Thus, in cases of frequent mobility, the use of MIPv6 in sensor nodes may lead to prohibitive energy expenditure, negatively impacting the nodes' and network lifetime.

In order to evaluate the performance of the proposed Network of Proxies approach and, consequently, confirm its expected benefits in terms of handoff time and energy, a third round of tests was made.

The obtained time and energy results are graphically presented in Figs. 14 and 15, respectively. The corresponding numerical results are presented in Tables 5 and 6.

As we can observe in Fig. 14 and in Table 5, the average time to perform a hard handoff was only $\sim$117 ms. Once in possession of the new configuration, the mote readily and simply rebooted the transceiver, switching to the new channel and self-configuring the new address. Naturally, this is much faster than in the traditional handoff case, presented in the previous sub-section, in which the mobile node spent an average of $\sim$961 ms to handoff. This dramatic difference can be easily understood when we realize that in the case of NoP-assisted mobility the mobile node only changes network when all mobility management tasks have been performed on its behalf by the NoP, while in the traditional mode it is the mobile node that has to deal with and complete all the procedure. As proxies use IEEE 802.11 to communicate between themselves (i.e., they benefit from much higher transmission rate than
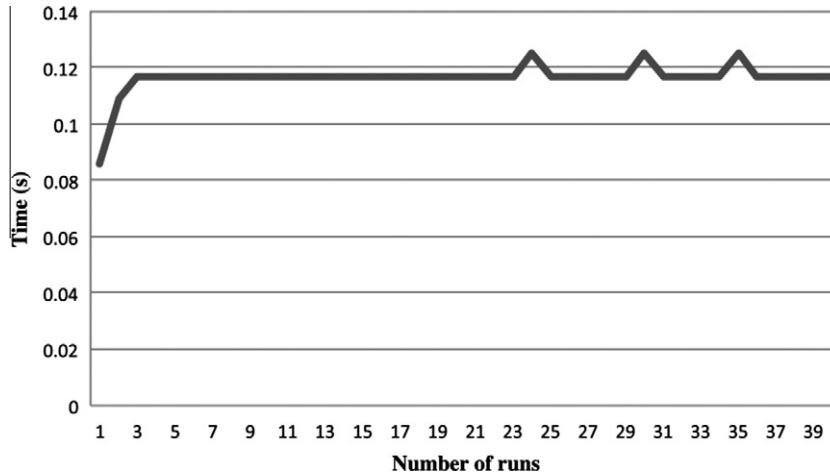
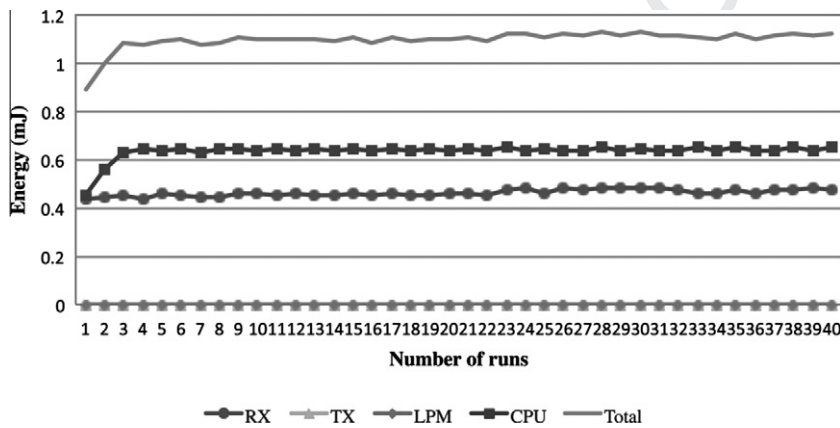**Fig. 14.** NoP-assisted hard handoff time.

Fig. 15. Energy required by NoP-assisted hard handoff.

Table 5
Summary of NoP – assisted hard-handoff time values.

	N	Range	Minimum	Maximum	Mean	Std. deviation
Time (ms)	40	39.1	85.9	125.0	116.8	5.6
Valid N	40					

Table 6
Summary of NoP – assisted hard-handoff energy values.

	N	Range	Minimum	Maximum	Mean	Std. deviation
RX (mJ)	40	0.0482	0.4337	0.4819	0.4628	0.0138
TX (mJ)	40	0.0000	0.0000	0.0000	0.0000	0.0000
LPM (mJ)	40	0.0000	0.0000	0.0000	0.0000	0.0000
CPU (mJ)	40	0.1958	0.4548	0.6506	0.6346	0.0323
Total (mJ)	40	0.2440	0.8886	1.1325	1.0974	0.0403
Valid N	40					

the one available to individual motes) and are not subject to energy and processing restrictions, they can perform the mobility procedures much faster than the mote-based mobility case.

Fig. 15, complemented by Table 6, presents the energy expenditure results.

We conclude that in the case of NoP-assisted handoff the mobile node only spent an average of ∼1.1 mJ to handoff between networks operating in different channels. This value sharply contrasts with the ∼11.75 mJ spent in the previous test suite, in which the mobile node performed all the mobility tasks. Fig. 16 depicts the difference between MIPv6 soft handoff, MIPv6 hard handoff and NoP-assisted handoff. As we can clearly observe, the latter requires much less time and consequently much less energy than the conventional solutions.

The obtained results clearly show the advantages of the NoP approach. Shorter handoff time decreases the disconnection period and reduces the loss probability. Concurrently, reduction in energy consumption reduces the number of dead nodes and, therefore, increases the network lifetime, reducing its maintenance and cost of ownership.

Recent work [34] has proved that, when applied to GINSENG, NoP is also capable of improving the received packet ratio (RPR) of mobile nodes from around 70% to 99%, under different mobility scenarios. This improvement clearly demonstrates the benefits of deploying NoP to assure performance control in critical scenarios, and justifies the additional cost of the whole deployment.

6. Simulation-based evaluation

The results presented in the previous section were obtained in a real platform, using controlled mobility and a limited number of motes, more specifically two sink nodes and one mobile node.

In this section our objective is to present the results of similar experiments, this time performed through simulation, now consid-
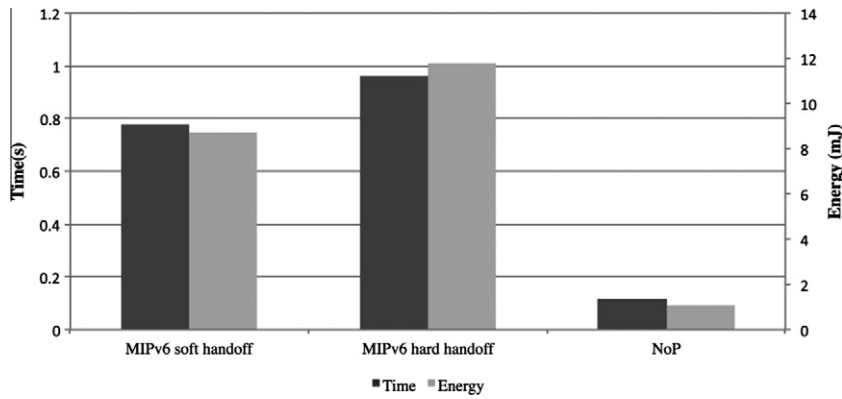
Fig. 16. Overview of time and energy expenditure for the three approaches.

ering there are additional sink nodes, a varying number of static, interfering nodes, and the mobile nodes use one of three possible mobility models.

6.1. Simulation overview

WSN mobility is classified into three different strategies [3]: random, predictable and controlled. In the first case, MNs move freely and randomly in a specific area. In the second case, the MNs paths are known in advanced, while in the third case the MNs' movement is controlled in real-time. The simulation-based evaluation of MIPv6 and NoP was performed using these three different strategies.

To simulate the mobility cases with the same code used in the implementation-based evaluation we chose the Cooja simulator [35], adapted with the mobility plugin. The main reasons for this choice were, on one hand, the fact that Cooja allowed us to re-use the code developed for the real experiment because it is also based on ContikiOS and, on the other hand, the fact that Cooja, along with the Castalia WSN simulator, are widely recognized as the best simulators/emulators for WSNs [36].

The mobility trace files were generated based on Bonnmotion [37]. For the random strategy we chose to use the Manhattan model, according to which a destination was randomly chosen from the various possible destinations in a grid representing the roads of our real scenario, i.e., the refinery in which employees and vehicles can freely move along the defined roads. For the predictable mobility strategy case, we generated a trace file in order to guarantee that the MN moved in a continuous fashion between two distant points $(x1,y1)$ $(x2,y2)$, like a ping-pong ball obliquely crossing the field, whereas for the controlled strategy we manually placed the MN

in the handoff area. The experiment area was limited to 10,000 m^2 and the MN speed was programmed to vary between 0.5 m/s and 2 m/s. To trigger the handoff we used layer 2 mechanisms, namely the RSSI value. The defined threshold was −80dBm.

For all three mobility strategies, we performed simulations with 2 and 5 sink nodes, and with 0, 2, 5, 10 and 15 additional static neighbor nodes, which generated background traffic. Applying these configurations to the three mobility approaches under evaluation – namely MIPv6 soft handoff, MIPv6 hard handoff and NoP – we obtained a total of 90 different scenarios. For each one we collected the results of 20 handoffs. Additional simulation details are provided in Appendix B.

6.2. Simulation results

Fig. 17 presents the overall results concerning the time required to handoff in each case. Starting by analyzing the controlled strategy section, we can observe that in the first situation – equivalent to the one used in the implementation-based evaluation, in which there were just two sink nodes (one in each domain) and no additional nodes – we obtained time values greater than 1.5 s, which were higher than the values obtained in the real platform using the same configuration. This difference was mainly caused by factors not present in the simple prototype scenario, such as sinks deployment, distances and noise. The difference is, in addition, a reminder that, even when using the same code – as we did – in the simulated and in the real environment, WSN simulations cannot yet replace implementation due to the influence of the great variety of factors (including device drivers and hardware types) that are present in wireless sensor networks. This is in line with [36], in which the author concluded that simulation results can
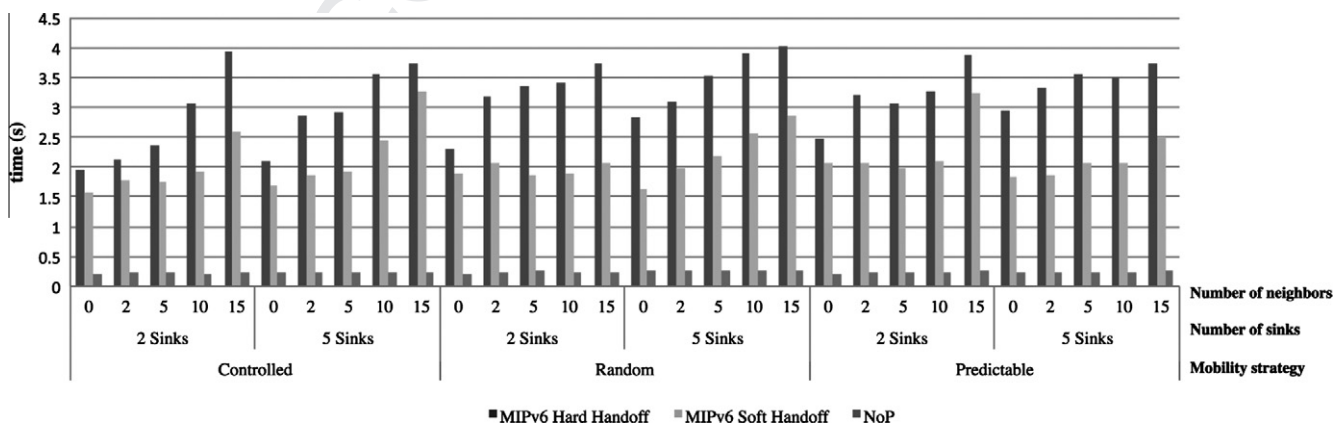

Fig. 17. Mean handoff time for each simulated scenario.

647 vary extraordinarily based not only on the simulator but also on
648 the used parameters. Nevertheless, as the main objective of the
649 presented study is the relative comparison of the node-based and
650 NoP-assisted mobility scenarios, and not the determination of
651 absolute values, the simulation results are perfectly valid.

652 One important observation is that, regardless the situation, the
653 MIPv6 hard handoff solution always took more time than the other
654 two solutions. Besides, we can also conclude that the proposed NoP
655 solution was not affected neither by the number of nodes and sinks
656 or by the mobility strategy. In contrast, the MIPv6 traditional solu-
657 tions were highly affected by the number of nodes, the number for
658 sinks and also by the mobility strategy.

659 Another interesting result was the non-linear behavior when
660 only 2 sink nodes were used. With the exception of the controlled
661 strategy, in which the values increase proportionally to the number
662 of nodes, in the random and predictable strategies the obtained
663 values did not change linearly with the number of nodes present
664 in the network. The constant change of positioning affected the

665 collision pattern and, consequently, retransmissions, which, in
666 turn, affected the time to handoff independently of the number
667 of neighbors present in the network. Hence, in general we can con-
668 clude that for the case of random and predictable mobility strate-
669 gies, in which the MN was continuously moving, an increase in the
670 number of sink nodes per domain led to a more predictable hand-
671 off time.

672 Figs. 18–20 present the energy expenditure results for each of
673 the three handoff types. The simulations confirmed that the NoP-
674 assisted handoff led to a dramatic reduction in the energy required
675 for mobility support, in addition to showing that this was not af-
676 fected by any other variable, such as, the number of nodes, sinks
677 or mobility strategy.

678 Looking at the base case – which is the soft handoff, controlled
679 strategy, 2 sink nodes and no neighbors – we can see that the MN
680 spent an average of ~25 mJ to handoff, while in the implementa-
681 tion-based evaluation it only reached ~8 mJ. On a closer look, we
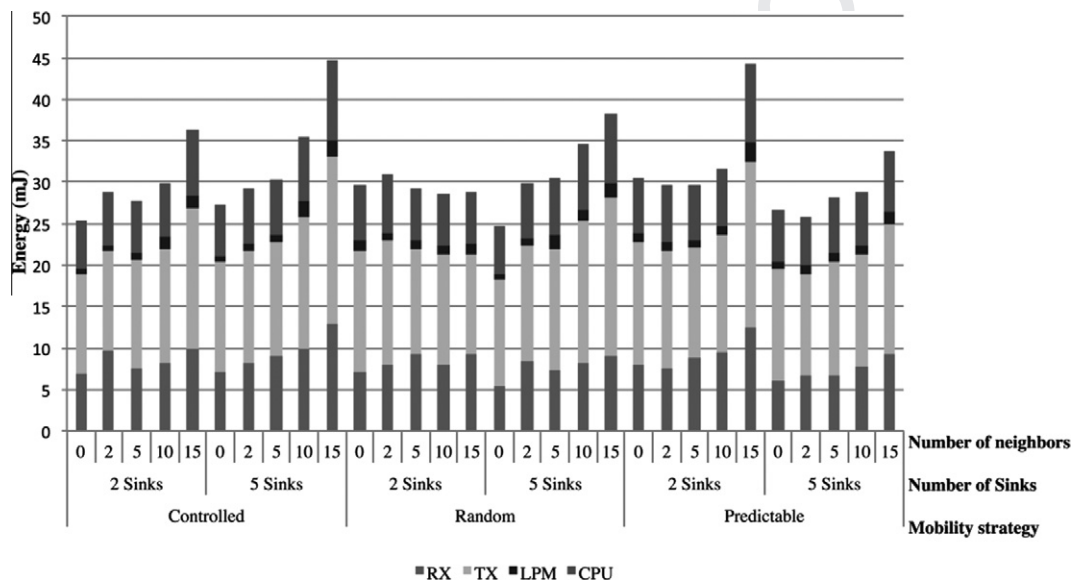682 can see that the MN spent roughly the same time in each task

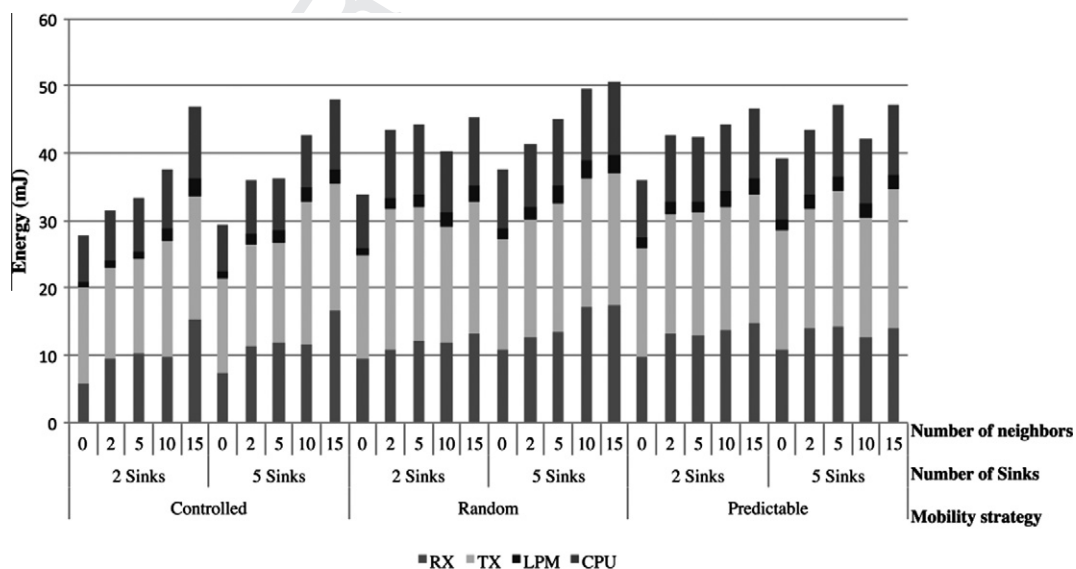Fig. 18. Simulation results for the energy required by MIPv6 soft handoff.

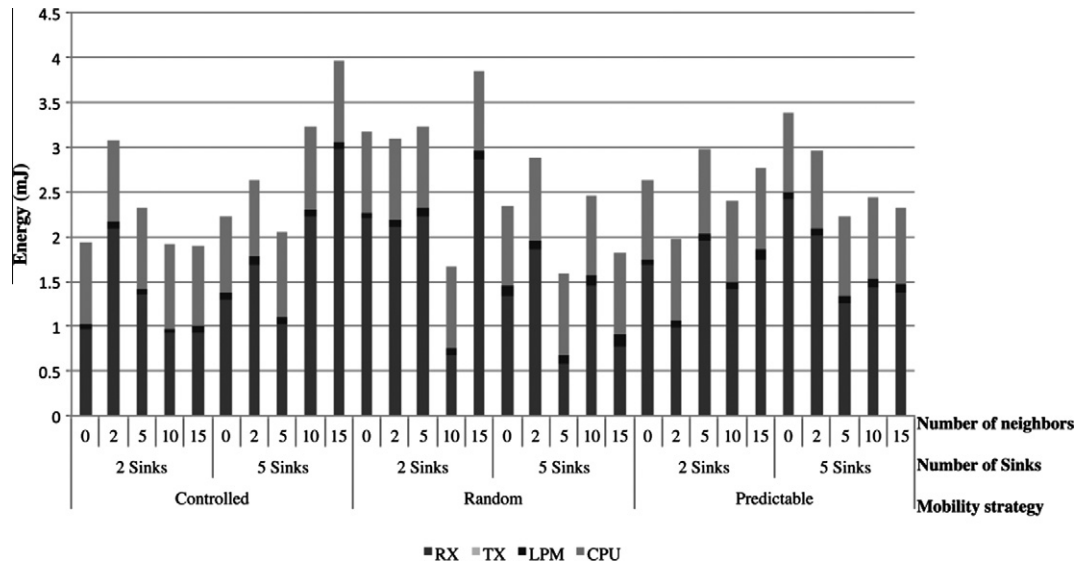Fig. 19. Simulation results for the energy required by MIPv6 hard handoff.

R. Silva et al./Computer Communications xxx (2012) xxx–xxx

Fig. 20. Simulation results for the energy required by the NoP-assisted handoff.

(receiving, transmitting, sleeping and processing). However, in the simulation more retransmissions occurred, namely of Router Solicitation and Binding Update messages.

Looking at the remaining results, we can observe the same type of behavior explained before in the context of the time analysis. In general, hard handoff required more energy, independently on the scenario. We can also conclude that the increase in the number of nodes directly affected not only the time (as previously observed) but also energy. Collisions and retransmissions are the main justifications for this.

7. Conclusion

As wireless sensor networks broaden their field of application, it is becoming apparent that the need for effective solutions for sensor node mobility is increasing.

In order to investigate the consequences and possible issues of using MIPv6 for mobility support in WSNs, we performed a set of tests with the objective of determining the handoff time and the energy expenditure incurred by MIPv6 node-based mobility. These tests were performed both by prototyping and by simulation, having as guidelines the requirements of a critical, real-life scenario, deployed in an oil refinery in the context of the GINSENG European project.

Although the test results have shown that deploying and using MIPv6 in sensor nodes is feasible, they also showed that this leads to non-optimal handoff time and to significant energy expenditure, even when using a simplified, lightweight MIPv6 implementation. This is mainly due to the complexity of the MIPv6 protocol, which is not adapted to the scarce resources generally found in WSN motes.

The considerable impact of MIPv6 on sensor nodes and, consequently, on the overall operation of wireless sensor networks, was the main motivation for the proposal of the Network of Proxies (NoP) concept, according to which an overlay mesh Network of Proxies is used to assist sensor nodes, by performing mobility management tasks on their behalf.

NoP was subject to a series of tests, similar to the ones used for MIPv6 node-based mobility, and the results have shown that the concept leads to a dramatic reduction in the handoff time and energy, thus proving that this is a solution well worth exploring in deployed WSNs.

In addition to the main conclusions referred above, two complementary conclusions can also be drawn from the real-life WSN deployment used as case study, and from the obtained results. On one hand, the performance control and high reliability requirements of most real-life, industrial WSN deployments cannot be achieved with currently available out-of-the-box solutions, and are not compatible with burdening sensor motes with complex tasks not directly related to sensing and actuating, such as new MAC protocols, debugging mechanisms, mobility, routing or security. Thus, the added cost of a complementary infrastructure like the proposed NoP is largely compensated by the performance benefits of the WSN motes. On the other hand, simulations and implementations are quite different matters. Because we recognized this at an early stage, we decided not to stick with implementations or simulations only and performed both. We believe that this provided a much more adequate view on the issue under study and is a much more honest approach.

As a final remark, the obtained results open up several lines for further research, including the study of fully-fledged MIPv6 mobility, the assessment of the NoP concept in larger scale scenarios and in other contexts, and the enhancement of the NoP functionality, which are already being explored.

Appendix A. WSN MIPv6 Implementation

This appendix provides details on the developed WSN MIPv6 implementation, used in all of the prototype-based evaluation experiments described in the paper.

Several MIPv6 implementations are available to the research community. However, none of them was specifically developed for WSNs, which means that it is not possible to run the existing code directly in motes due to their inherent constraints and limitations. Hence, in order to address the problems at hand, a Contiki-OS-adapted MIPv6 implementation had to be developed.

We focused our implementation on the strictly necessary, basic MIPv6 features, following RFC 3775 and 6lowPAN specifications as
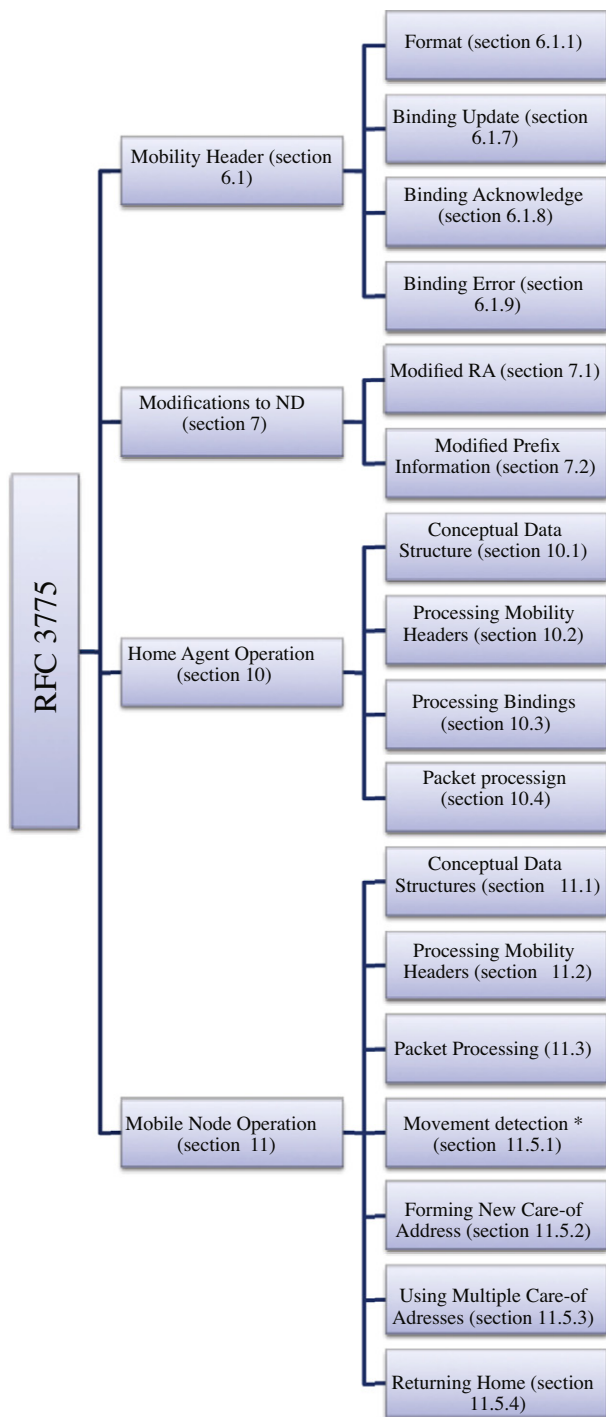
Fig. A.21. Implemented RFC 3775 functionality.

Table A.7
Implemented MIPv6 data structures.

```
struct uip_mip6_hdr {              struct uip_mip6_bu {
u8_t payload_proto;                u16_t seqno;
u8_t header_len;                   u16_t flagsreserved;
u8_t type;                         u16_t lifetime;
u8_t reserved;                     };
u16_t checksum;
};
(a) MIPv6 Header.                  (b) Binding Update.
struct uip_mip6_back {             struct binding_cache {
u8_t status;                       uip_ipaddr_t haddr;
u8_t flagsreserved;                uip_ipaddr_t coaddr;
u16_t seqno;                       u16_t lifetime;
u16_t lifetime;                    u8_t flags;
};                                 u16_t seqno;
                                   };
(c) Binding Acknowledge.           (d) Mobility Binding Table.
```

In what concerns Section 7, we implemented a modified Router Advertisement (RA) message, in order to include the MIPv6 flags needed for the NoP operations. In this case we based our implementation on the existing Neighbour Discovery (ND) implementation in ContikiOS.

The implementation contemplated a Home Agent capable of performing the basic MIPv6 operations and of dealing with the conceptual data structures specified in Section 10 of RFC 3775, including the mobility binding table (Table A.7 d).

The Mobile Node implementation (RFC 3775 Section 11) followed 6lowPAN and the concept of stateless configuration, in which the mote constructs a global address based on its 64-bit MAC address and on the network prefix, obtained via the received Router Advertisement. In order to allow for soft handoff, we also implemented Section 11.5.3 (Using Multiple Care-of Addresses). Dealing with multiple addresses has been simplified in ContikiOS version 2.4, through uip-netif.

In what concerns movement detection (RFC 3775 Section 11.5.1), we did not use Layer 3 mechanisms in order to avoid energy-expensive layer 3 broadcasts. Instead, Layer 2 RSSI-based mechanisms were used which, in addition to leading to lower energy expenditure, have the added benefit of being quicker. As explained in the paper, the RSSI threshold was -80 dBm.

As an example of the developed implementation, Fig. A.22 presents the construction of the Binding Update message, while Fig. A.23 shows the construction of the Binding Acknowledge.

In this implementation several macros were specifically created for MIPv6, following the style and approach of the remaining ContikiOS code.

Appendix B. WSN MIPv6 simulation

This appendix provides details on the developed WSN MIPv6 simulation, used in all of the simulation-based evaluation experiments described in the paper.

The objective of the simulations was to extend the evaluation previously performed by implementation, considering different mobility models and varying the number of sink nodes as well as of sensor nodes. For this, we chose to use the Cooja Simulator. Cooja is a java-based platform specifically developed for ContikiOS, that runs the same code as the actual applications.

By default, Cooja does not support mobility. Nevertheless, based on the fact that each deployed mote has its own location represented in a two-axis (x,y) system, a Cooja mobility plugin was developed by Marcus Lundn. This plugin is capable of loading specific mobility trace-files using the Interval Format. In this format each event originates specific waypoints for each mobile node,

closely as possible, using stateless configuration whenever possible. As it was not essential for the tests, the Return Routability procedure was not implemented at this stage. Fig. A.21 graphically lists the implemented RFC 3775 functionality. This is briefly explained below.

Concerning Section 6 of RFC 3775, we implemented the mobility header and the basic messages, such as the Binding Update, Acknowledgement and Error messages (Table A.7 a, b and c). The implementation contemplated not only the message structure but also all the necessary code to send and receive messages, integrating it with the Contiki OS uip6 stack.

```
u16_t flags;
u16_t sequence = 323;
flags = UIP_MIP6_BU_ACK;
/* IP header */
UIP_IP_BUF->vtc = 0x60;
UIP_IP_BUF->tcflow = 0;
UIP_IP_BUF->flow = 0;
UIP_IP_BUF->len[0] = 0; /* length will not be more than 255 */
UIP_IP_BUF->len[1] = UIP_ICMPH_LEN + UIP_MIP6_BU_LEN;
UIP_IP_BUF->proto = UIP_PROTO_MIP6;

UIP_IP_BUF->ttl = UIP_MIP6_HOP_LIMIT;

uip_ipaddr_copy(&UIP_IP_BUF->srcipaddr, &uip_netif_addr_lookup(NULL, 0, 0)->ipaddr);
uip_ipaddr_copy(&UIP_IP_BUF->destipaddr, &parentaddr);

/* MIP6 header */
UIP_MIP6_BUF->payload_proto=IPPROTO_NONE;
UIP_MIP6_BUF->type=UIP_MIP6_MH_TYPE_BU;
UIP_MIP6_BUF->header_len = UIP_MIPH_LEN;
UIP_MIP6_BUF->reserved = 0;
UIP_MIP6_BUF->checksum = 0;
UIP_MIP6_BUF->checksum = ~uip_icmp6chksum();

UIP_MIP6_BU_BUF->seqno= sequence;
UIP_MIP6_BU_BUF->flagsreserved=flags;
UIP_MIP6_BU_BUF->lifetime=63;
uip_len = UIP_IPH_LEN + UIP_MIPH_LEN + UIP_MIP6_BU_LEN;
```

Fig. A.22. Building a Binding Update message in ContikiOS.

```
create_ba:

uip_ext_len = 0;
flags = UIP_MIP6_BU_KEYM;

/* IP header */
UIP_IP_BUF->vtc = 0x60;
UIP_IP_BUF->tcflow = 0;
UIP_IP_BUF->flow = 0;
UIP_IP_BUF->len[0] = 0; /* length will not be more than 255 */
UIP_IP_BUF->len[1] = UIP_ICMPH_LEN + UIP_MIP6_BACK_LEN;
UIP_IP_BUF->proto = UIP_PROTO_MIP6;
UIP_IP_BUF->ttl = UIP_MIP6_HOP_LIMIT;
uip_ipaddr_copy(&UIP_IP_BUF->srcipaddr,
        &uip_netif_addr_lookup(NULL, 0, 0)->ipaddr);
uip_ipaddr_copy(&UIP_IP_BUF->destipaddr, &srcaddr);

/* MIP6 header */
UIP_MIP6_BUF->payload_proto = IPPROTO_NONE;
UIP_MIP6_BUF->type = UIP_MIP6_MH_TYPE_BACK;
UIP_MIP6_BUF->header_len = UIP_MIPH_LEN;
UIP_MIP6_BUF->reserved = 0;
UIP_MIP6_BUF->checksum = 0;
UIP_MIP6_BUF->checksum = ~uip_icmp6chksum();

UIP_MIP6_BACK_BUF->status = status;
UIP_MIP6_BACK_BUF->flagsreserved = flags;
UIP_MIP6_BACK_BUF->seqno = sequence;
UIP_MIP6_BACK_BUF->lifetime = 63;
```

Fig. A.23. Building a Binding Acknowledge message in ContikiOS.

817 which are also converted into specific steps. Each step represents
818 the (x,y) position of each mobile node in a specific point in time.
819 Fig. B.24 presents an extract of an Interval Format file that defines
820 the first 10 s of events for a Mobile Node 0.
821   In addition to the definition of each event, the trace file can also
822 include other parameters, such as the dimension of the scenario,
823 the duration of the simulation and the number of the mobile
824 nodes.

825 To generate the trace files in the Interval Format we use the
826 Bonnmotion application, as mentioned in the paper. Bonnmotion
827 supports not only several mobility pattern but also converts the
828 output for several formats, including the Interval Format, sup-
829 ported by Cooja, among other simulators.
830   The simulations were performed using three different mobility
831 models, namely, random, predictable and controlled. Even though
832 the controlled mobility model is simple implement, random and

```
#Node Time X Y
0 0.0 124.114701442512 82.13327649829307
0 1.0 124.03423056118321 81.63979454598056
0 2.0 123.95375967985443 81.14631259366806
0 3.0 123.87328879852562 80.65283064135554
0 4.0 123.79281791719683 80.15934868904303
0 5.0 123.71234703586804 79.66586673673052
0 6.0 123.63187615453926 79.172384784418
0 7.0 123.55140527321046 78.67890283210549
0 8.0 123.47093439188167 78.18542087979297
0 9.0 123.39046351055289 77.69193892748048
0 10.0 123.3099926292241 77.19845697516796
```

Fig. B.24. Sample interval format file.

predictable mobility models require substantially more complex settings. With Cooja, all three mobility models can easily be simulated and evaluated under different conditions, varying the number of sink nodes and of nodes per network.

In the case of the random mobility model, the respective trace files were obtained using Bonnmotion, with random movement limited to 10,000 m^2 and following the Manhattan grid model. Fig. B.25 shows the generating command and associated output.

As it can be seen in the figure, to generate the trace-file using Bonnmotion, we first must specify the name of the file, followed by the mobility model and then the specific attributes. In this case, the $-u$ and $-v$ attributes are specific to the ManhattanGrid model and are used to define the size of each grid square, the $-n$ parameter specifies the number of mobile nodes, the $-d$ parameter establishes the simulation time in seconds, and the $-x$ and $-y$ parameters specify the simulation area in meters. Once the mobility file is created, it is necessary to convert it to the Interval Format, in order for it to be readable by Cooja. Fig. B.26 depicts this command.

After conversion, a trace file with extension.if is created. Fig. B.27 presents an extract of the beginning of this file.

For the predictable mobility model simulations, as Bonnmotion does not provide a suitable model for this case we decided to write our own mobility trace file, necessarily complying with the Interval Format.

In this case our mobile node was programmed to move in a ping-pong fashion between the $(x1,y1)$ point, located in network 1, and the $(x2,y2)$ point, located in network 2. Fig. B.28 depicts four stages of this predictable movement scenario (from left to right and from top to bottom), in which a mobile node is moving from
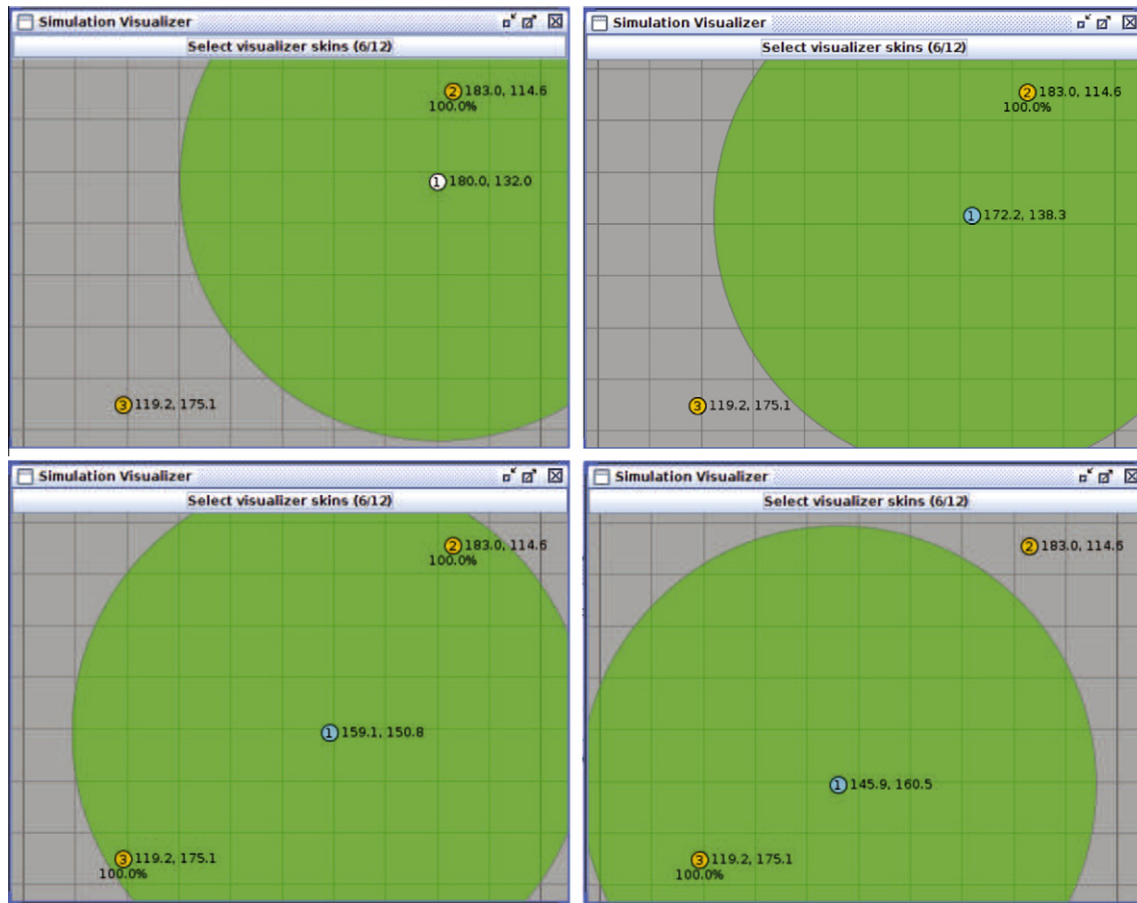
```
rnsilva@instant-contiki:~/Desktop/bonnmotion-1.5a/bin$ ./bm -f manhattan_scenario ManhattanGrid
-u 10 -v10 -d 200 -n 1 -x 100 -y 100
BonnMotion 1.5

Starting ManhattanGrid ...
Next RNG-Seed =1979738518953144356 | #Randoms = 1377
ManhattanGrid done.
rnsilva@instant-contiki:~/Desktop/bonnmotion-1.5a/bin$
```

Fig. B.25. Generation of random mobility trace file.

```
rnsilva@instant-contiki:~/Desktop/bonnmotion-1.5a/bin$ ./bm IntervalFormat -f manhattan_scenario
BonnMotion 1.5

Starting IntervalFormat ...
randomSeed (String):1303161850754
randomSeed (Long):1303161850754
IntervalFormat done.
rnsilva@instant-contiki:~/Desktop/bonnmotion-1.5a/bin$
```

Fig. B.26. Converting to the interval format.

```
rnsilva@instant-contiki:~/Desktop/bonnmotion-1.5a/bin$ more manhattan_scenario.if
#X 100.0
#Y 100.0
#Nodes 1
#Duration 200.0
#Waypoints node  0: 0.0 35.33721180562794 100.0 1.2551489858947207 36.59526379036521 100.0 18.87
171113855493 51.59526379036521 100.0 33.23540817426101 60.0 100.0 67.41541496433774 60.0 80.0 84
.50541835937611 70.0 80.0 135.7754285444912 70.0 50.0 147.04673660134267 76.59526379036521 50.0
168.99721652419566 90.0 50.0 171.6094876837651 90.0 51.59526379036521 186.3703270632077 90.0 66.
59526379036521 189.38432408755125 90.0 70.0 200.0 78.00809162154476 70.0
#Node Time X Y
0 0.0 35.33721180562794 100.0
0 1.0 36.33952467754607 100.0
0 2.0 37.229483175212216 100.0
0 3.0 38.080954641098366 100.0
0 4.0 38.93242610698451 100.0
0 5.0 39.78389757287066 100.0
0 6.0 40.6353690387568 100.0
0 7.0 41.486840504642956 100.0
0 8.0 42.33831197052909 100.0
0 9.0 43.18978343641524 100.0
0 10.0 44.04125490230139 100.0
0 11.0 44.89272636818753 100.0
0 12.0 45.74419783407368 100.0
```

Fig. B.27. Resulting interval format file, created according to the random mobility model.

Fig. B.28. Predictable mobility scenario.
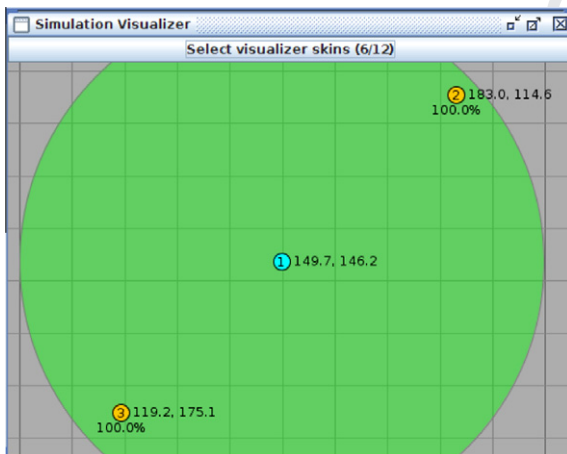
Fig. B.29. Handoff stage of a controlled mobility scenario.

position (180.0, 132.0), within range of Sink Node 2, to position (145.9, 160.5), within range of Sink Node 3. During its path, the mobile node crosses an area in which both networks are reachable, representing the handoff area. This predictable mobility scenario can represent several real-life cases, such as lifts, trams, trains or buses, in which the various positions can easily be anticipated.

Controlled mobility is the type of mobility that, in real scenarios, can correspondent to robotic systems in which there is a manual or automatic movement controller. For simulating this type of mobility no trace file was needed. Instead, we manually positioned the mobile node in the handoff zone, like we did in the real evaluation. The scenario is depicted in Fig. B.29.

Using the three types of mobility models described above, several simulations were performed. In addition to the use of different mobility models, the number of sink nodes and the number of sensor nodes per networks was varied.

Varying the number of sink nodes has direct impact on coverage, increasing the density and therefore increasing the number of possible attachment points during the handoff. Varying the number of nodes in the network impacts the network traffic, and therefore influences the handoff process.

It should be noted that the number of simulated sink nodes and sensor nodes in the network was limited by the processing capacity of the machine used for the simulations, this being the reason for not extending the tests to wider and denser networks.

References

[1] D. Johnson, C. Perkins, J. Arkko, Mobility support in ipv6, IETF RFC3775, June 2004.
[2] Z.M. Wang, S. Basagni, E. Melachrinoudis, C. Petrioli, Exploiting sink mobility for maximizing sensor networks lifetime, in: Proceedings of the 38th Annual Hawaii International Conference on System Sciences, IEEE Computer Society, Washington, DC, USA, 2005, p. 287.1.
[3] D. Stevanovic, N. Vlajic, Performance of ieee 802.15.4 in wireless sensor networks with a mobile sink implementing various mobility strategies, in: 33rd IEEE Conference on Local Computer Networks, 2008, LCN 2008, 2008, pp. 680 –688.
[4] R.C. Shah, S. Roy, S. Jain, W. Brunette, Data mules: modeling a three-tier architecture for sparse sensor networks, in: ACM Fourth International Conference on Information Processing in Sensor Networks (IPSN/SPOTS), IEEE/ACM, 2003, pp. 30–41.
[5] E. Ekici, Y. Gu, D. Bozdag, Mobility-based communication in wireless sensor networks, IEEE Communications Magazine 44 (7) (2006) 56–62.

[6] A. Raja, X. Su, Mobility handling in mac for wireless ad hoc networks, Wireless Communications in Mobile Computing 9 (3) (2009) 303–311.

[7] M. Laibowitz, J.A. Paradiso, Parasitic mobility for pervasive sensor networks, in: Proceedings of the 3rd Annual Conference on Pervasive Computing (Pervasive 2005), Springer-Verlag, 2005, pp. 255–278.

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

[9] Ieee std. 802.15.4, iEEE standard for information technology – wireless medium access control (mac) and physical layer (phy) specifications for wpan and lr-wpan (lowpan), August 2007. <http://ieee.org/>.

[10] H. Pham, S. Jha, Addressing mobility in wireless sensor media access protocol, in: Intelligent Sensors, Sensor Networks and Information Processing Conference, 2004, 2004, pp. 113–118.

[11] B. Liu, K. Yu, L. Zhang, H. Zhang, Mac performance and improvement in mobile wireless sensor networks, in: Proceedings of the Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, SNPD'07, IEEE Computer Society, Washington, DC, USA, 2007, pp. 109–114.

[12] L. Bernardo, R. Oliveira, M. Pereira, M. Macedo, P. Pinto, N.D. Lisboa, N.D. Lisboa, A wireless sensor mac protocol for bursty data traffic, in: IEEE PIMRC'07, 18th IEEE Annual International Symposium on Personal Indoor and Mobile Radio Communications, Athens, Greece, 2007, pp. 1–5.

[13] M. Ali, T. Suleman, Z.A. Uzmi, Mmac: A mobility-adaptive, collisionfree mac protocol for wireless sensor networks, in: Proceedings of the 24th IEEE IPCCC+05, 2005, pp. 401–407.

[14] M. Ali, T. Voigt, Z.A. Uzmi, Mobility management in sensor networks, in: Workshops Proceeding of 2nd IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS'06), San Francisco, California, 2006, pp. 131–140.

[15] L.-J. Chen, T. Sun, N.-C. Liang, An evaluation study of mobility support in zigbee networks, Journal of Signal Processing Systems 59 (2010) 111–122.

[16] G. Mulligan, C. Williams, Mobility considerations for 6lowpan, Internet Engineering Task Force, Internet-Draft draft-williams-6lowpan-mob, July 2010.

[17] 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.

[18] R. Silva, J.S. Silva, F. Boavida, Towards mobility support in wsns, in: 10+ Conferncia sobre Redes de Computadores (CRC2010), Braga, Portugal, 2010.

[19] R. Koodli, Fast handovers for mobile ipv6, IETF RFC4068, July 2005.

[20] 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.

[21] S. Gundavelli, K. Leung, V. Devarapalli, K. Chowdhury, B. Patil, Proxy mobile ipv6, IETF RFC5213, August 2008.

[22] 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.

[23] T. Schmid, P. 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.

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

[25] J.B.P. Suriyachai, U. Roedig, Poster abstract: a mac protocol for industrial process automation and control, in: the European Conference on Wireless Sensor Networks (EWSN 2010), IEEE, 2010.

[26] Arm embedded linux boards, 2012. <http://embeddedarm.com/products/arm-sbc.php>.

[27] Ieee std. 802.11, IEEE Standard for Information technology – Specific requirements – Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, June 2007. <http://www.ieee.org/>.

[28] T. Narten, E. Nordmark, W. Simpson, Neighbor discovery for ip version 6 (ipv6), IETF RFC2461, December 1998.

[29] T.-D. Tran, R. Silva, D. Nunes, J. Silva, Characteristics of channels of ieee 802.15.4 compliant sensor networks, Wireless Personal Communications, 2011, 1-1610.1007/s11277-011-0395-3.

[30] A. Dunkels, B. Gronvall, T. Voigt, Contiki – a lightweight and flexible operating system for tiny networked sensors, in: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks, LCN'04, IEEE Computer Society, Washington, DC, USA, 2004, pp. 455–462.

[31] Telosb datasheet, 2011. <www.willow.co.uk/TelosB_Datasheet.pdf>.

[32] The msp430 website, 2011. <http://focus.ti.com/mcu/docs/mcuhome.tsp?sectionId=101>.

[33] The cc2420 website, 2011. <http://www.supplyframe.com/datasheet-pdf/search/cc2420>.

[34] R. Silva, J.S. Silva, F. Boavida, Evaluation of nop solution in critical scenarios – technical report, February 2012. <http://ricardomendaosilva.pt.vu/>.

[35] F. Osterlind, A. Dunkels, J. Eriksson, N. Finne, T. Voigt, Cross-level sensor network simulation with cooja, in: Proceedings of the 31st IEEE Conference on Local Computer Networks 2006, 2006, pp. 641 –648. http://dx.doi.org/10.1109/LCN.2006.322172.

[36] A. Abdolrazaghi, Unifying wireless sensor network simulators, June 2009. <http://www.eeweb01.ee.kth.se/upload/publications/reports/2009/xr-ee-lcn_2009_005.pdf>.

[37] The bonnmotion website, 2011. <http://iv.cs.uni-bonn.de/wg/cs/applications/bonnmotion/>.