# A Mobility Management Protocol for Wireless Sensor Networks

Jianjun Wen and Waltenegus Dargie

Chair for Computer Networks, Faculty of Computer Science, Technical University of Dresden, 01062 Dresden, Germany
Email:{jianjun.wen, waltenegus.dargie}@tu-dresden.de

***Abstract*—Wireless sensor networks supporting the free mobility of nodes can be useful for several applications. For example, in residential areas and rehab centres, sensors can be attached to subjects to monitor their movements, body exertions, and cardiac activities. These benefits, however, are also challenged by the difficulty of establishing reliable and stable links. In cellular networks, mobile stations are always associated with the nearest base station through intra- and inter-cellular handover. The underlying process is that the quality of an established link is continually evaluated and handover decisions are dully made by resource rich base stations. In wireless sensor networks, should a seamless handover be carried out, the task has to be accomplished by energy constraint, resource-limited, and low-power wireless sensor nodes in a distributed manner. In this paper we propose a sender-initiated mobility management protocol to enable seamless handover. We have fully implemented the protocol in TinyOS environment for the TelosB and Imote2 platforms, experiment results showing that our protocol achieves high reliability and triggers less handover requests (less than $50\%$ to $80\%$) compared to three state-of-the-arts. Furthermore, our protocol reduces the signalling overhead by up to $95\%$.**

***Index Terms*—Handover, MAC, mobility, mobility management, wireless sensor network**

## I. INTRODUCTION

A wide range of applications in wireless sensor networks require mobility support. Examples are healthcare applications [4], applications supporting independent living in residential areas [1], applications monitoring pollution in smart cities [15], and wildlife monitoring [10]. One of the main challenges in supporting mobility is the difficulty of establishing stable and reliable links when a continuous streaming of data is required. Independent studies have shown that wireless links established with low-power radios (i.e., those complying with the IEEE 802.15.4 specification) are often lossy and dynamic [8], [16], the fluctuation in link quality becoming significant for mobile nodes (some have reported $\pm 10$ dB fluctuation for distances less than 30 m) [19], [18], [5].

In cellular networks, the task of managing mobility (intra- and inter-cellular handover) is assigned to resource-rich base stations. Should the same feature be supported in wireless sensor networks, the management task should be undertaken by energy constraint, resource-limited, and low-power wireless sensor nodes. Furthermore, unlike cellular base stations, which are always powered on and active, the potential relay nodes with which a new link should be established before an existing one breaks, may be sleeping in order to save energy.

Ideally, a mobility management protocol in wireless sensor networks should be resilient to transient link dynamics but quick to react to persistent link quality degradations. This aspect entails:

- identifying the appropriate time to initiate a handover process,
- seamless discovery of candidate (neighbour) nodes, and
- selection of the most reliable relay node.

These steps have been addressed in the literature in different ways.

MRI-MAC [6] assumes that the relative location of a mobile node with respect to a stationary relay node can be estimated from the RSSI values of the packets it receives. In order to collect sufficient statistics from incoming ACK packets in short time, the protocol uses burst transmission. If the relative distance of a node with the current relay node is beyond a predefined threshold, the mobile node initiates a handover immediately. To discover candidate relay nodes, the mobile node eavesdrops on beacons transmitted by neighbour relay nodes.

MX-MAC [5] and MoX-MAC [2] extend X-MAC [3] to support a seamless handover. The former employs Least Mean Square (LMS) filter to predict the link quality of a mobile node and defines a threshold to trigger a handover, while the latter triggers a handover upon experiencing a single packet failure. SmartHop [9] transmits beacons in burst to discover candidate neighbours and estimates the relative link quality of its neighbours by evaluating received ACK packets. The handover decision is made by setting a predefined RSSI threshold with a hysteresis margin. The protocol is designed on the basis of an extensive study on the impact of key PHY and MAC parameters on the handover performance. However, the protocol does not support duty-cycling and assumes that candidate relay nodes are active all the time.

In this paper we propose a mobility management protocol to address the three features we identified above. Its typical features can be summarised as follows: It (1) enables mobile nodes to join a network quickly; (2) supports burst transmission in order to let a mobile node transfer as many packets as possible when the quality of a link is good and stable; (3) employs a Kalman filter in the background in order to predict the state of a mobile link with statistics obtained from received ACK packets; and (4) establishes the temporal evolution of all potential links during neighbour discovery in order to identify the best relay node to which a communication should be transferred.

The remainder of this paper is organized as follows: In Section III, we introduce our protocol and discuss its implementation detail. In Section IV, we present experiment results and quantitative comparisons with three state-of-the-art mobility management protocols. Finally, in Section V, we provide concluding remarks and outline future work.

## II. Seamless Handover

In a wireless sensor network supporting mobile nodes, the predominant traffic flow is from the mobile nodes to a remote base station via intermediate, stationary relay nodes. Hence, the main task of the relay nodes is assisting the mobile nodes. This is the case for many residential applications (healthcare, independent living) where sensor nodes are deployed on the bodies of people who nevertheless move freely and carry out everyday tasks while vital biomedical measurements are collected from them.

In order to support uninterrupted monitoring and a steady streaming of packets, we support a seamless handover. In our protocol, a mobile node initiates a handover when it perceives that the link it has already established with a stationary relay node is becoming bad. This can be done by evaluating physical and link layer parameters of received ACK packets. Moreover, a handover can be initiated and completed without first breaking an established link. This can be achieved by embedding handover requests into the MAC header of data packets.

### A. Protocol Design

Our protocol is a preamble-based MAC protocol [3] and supports burst transmission [7]. Hence, when a mobile node first attempts to join the network, it transmits a preamble until a nearby relay node responds with an acknowledgement. The preamble is *anycast*, in that all neighbour relay nodes can access and respond to it, as illustrated in Fig.1. With the arrival of a beacon (acknowledgement) packet, the join phase will be completed (this process will be explained in more detail shortly).

After establishing connection with a specific relay node, the mobile node begins transmitting packets in burst with unicast/ACK scheme, as depicted in the right part of Fig.1. While the transmission is still going on, the mobile sender estimates and predicts the link quality by continuously evaluating physical and link layer parameters in the background. In case of a steady link quality deterioration (characterised by persisting packet loss rate and poor RSSI values of incoming ACK packets), the mobile node initiates a handover request to all nearby relay nodes without actually breaking the data transmission with the current relay node. When it discovers a better relay node, it then transfers communication to this node and resumes burst transmission with unicast/ACK scheme. The cycle of burst transmission, handover trigger and neighbour discovery/selection is repeated until the bulk data transfer is completed.

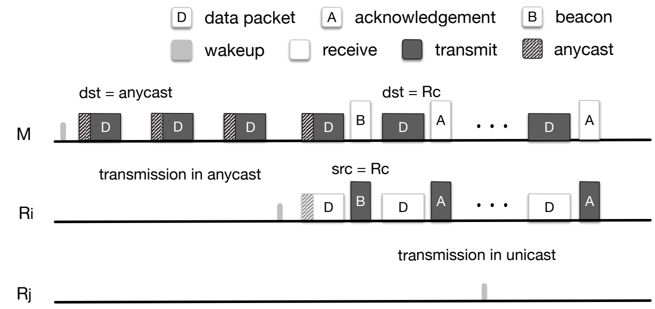


Fig. 1: MSI-MAC introduces *anycast* to discover and join the network with the first awoken receiver in the vicinity. The label M and R in the figure represent mobile sender and receivers respectively

Our design approach takes many of the requirements of low-power wireless sensor networks into account:

- In contrast to existing or proposed preamble-based MAC protocols, our protocol establishes a link by *anycasting* the first data packet and with a relay node which wakes up and acknowledges the earliest. This minimizes the number of packets transmitted as preamble and leads to a fast network joining.
- Unlike many mobility-aware MAC protocols, except for the network join, our protocol employs a *unicast* communication during the whole transmission, even during neighbour discovery and selection phase. As a result no data packet duplication is introduced and, therefore, no duplication suppression mechanism is required.
- Our protocol does not require extra control packets to manage a handover process thereby reducing the signalling overhead, for example, when compared with ME-Contiki [13] and SmartHop [9].
- Our protocol is compatible with duty-cycled operations.

### B. Fast Network Join

A mobile node may not have sufficient information about the relay node distribution in its surrounding. Therefore, it has to first search for an available relay node before it can transfer communication to a new link. In compliance with the IEEE 802.15.4 specification, it first performs clear channel assessment (CCA) to ensure that the medium is free. Then, it transmits the first data packet repeatedly in *anycast* mode, until it receives an acknowledgement from a nearby relay node. At the receiver's side, when a relay node receives an *anycast* data packet[1], it does not evaluate the entire packet in order to acknowledge it. Instead, it generates a beacon packet containing its own address and sends the beacon to the mobile node. The purpose is to simply indicate that it is available and ready to receive the remaining data packets. In case multiple relay nodes receive the *anycast* packet simultaneously, the

[1]In our implementation, we reserve the address 0x8000 as anycast address.

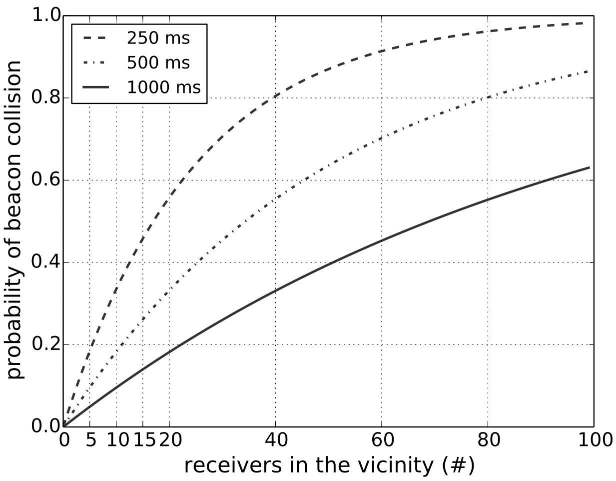


Fig. 2: The collision probability distribution function during *anycast* packet transmission within a single wakeup period (the wake up intervals is set between 250 ms to 100 ms).

probability of multiple beacons experiencing collision will be high.

To illustrate the impact of this: suppose that during the active period of a duty cycle, $T$, a mobile sender transmits at most $N$ packets with inter-packet interval $\tau$. We can express $N$ as $N = \lfloor \frac{\tau}{T} \rfloor$. Suppose also the wakeup times of the relay nodes are statistically independent and uniformly distributed between $(0, T)$. If we divide the duty cycle interval $T$ into $N$ uniform slots, the probability that a relay node awakes at any one of the $N$ slots to successfully receive a data packet and respond with a beacon at that slot is $1/N$. The beacon transmission collision occurs when at least two receivers awake in that slot. So the collision probability can be expressed as:

$$P_{collision} = 1 - \left[ \frac{1}{N} \left( 1 - \frac{1}{N} \right)^{m-1} + \left( 1 - \frac{1}{N} \right)^{m} \right] \quad (1)$$

where $m$ is the number of receivers in the neighbourhood of the mobile sender. Fig. 2 shows the beacon collision probability distribution for different wakeup intervals and number of receivers in the neighbourhood. By properly desynchronizing the wakeup time at set up time, the probability of beacon collision can be reduced to an acceptable level for small-scale, small duty-cycled networks (which is the case for residential areas, for example). For instance, the collision probability is between $8\%$ to $30\%$ when the wakeup interval is set from 1000 ms to 250 ms with 10 neighbours.

### C. Burst Transmission

Once a mobile sender discovers a relay node, it switches the communication mode back to unicast/ACK mode and transmits packets in burst, with no CCA between successive packets [8]. The idea is to enable the mobile node to transfer as many packets as possible before the link deteriorates. Hence, our protocol trades fairness for high throughout.

Two of the key components of our handover management protocol are the link quality estimation and handover trigger algorithm. The first continuously evaluates the fluctuation of link quality and whether this is a steady-state phenomenon. It is a realisation of the Kalman filter and takes as its input two link quality metrics from the physical and the link layer, namely, RSSI values and acknowledgement reception rate (ARR). The filter predicts whether the deterioration of a link quality is a steady phenomenon (and, therefore, whether a handover request should be triggered) or not. The second component triggers a handover request, collects beacons from its environment, and selects the best candidate to transfer a communication.

### D. Handover

Instead of broadcasting a sequence of control packets for discovering potential relay nodes during a handover request, our protocol keeps data transmission with current receiver as *unicast* but embeds a handover request in the MAC header, as illustrated in Fig. 3 (left part). Nearby relay nodes intercepting these packets need only evaluate the MAC header in order to determine whether the packets contain a handover request. Because surrounding relay nodes may be sleeping during this phase (as a consequence of duty-cycling), the mobile node should send multiple requests for a duration that equals the period of a single duty-cycle. Unlike the unicast/ACK scheme during a normal burst transmission, where packet transmission by the mobile node immediately follows the reception of an acknowledgement packet, the mobile node should now back-off after receiving an ACK packet from its current relay node. The reason is that those relay nodes which have intercepted the data packets and are ready to participate in a handover process have the possibility to transmit beacons to the mobile node. Fig. 4 illustrates this period.

Relay nodes, participating in a handover process should also back-off before they transmit beacons in order to minimise the probability of collision. In case more than two relay nodes wake up and respond to a handover request at the same time, beacon collision will occur and the probability distribution of this collision can be determined by using Equation 1.

*1) Neighbour Selection (NS-phase):* As we have already mentioned, a neighbour discovery lasts an entire duty cycle. Following this, the mobile node decides to elect one of them as its future relay node. This decision is made based on the feedback it gathers from each potential neighbour at the end of the neighbour discovery period. The feedback is gathered thus: At the end of a neighbour discovery period, the mobile node sends to all its neighbours a request for feedback, whereupon each candidate relay node sends a beacon containing its *unicast* address and bidding information. The bidding information consists of:

1) the averaged RSSI value for all the packets the relay node has intercepted since the beginning of a neighbour discovery phase;
2) the packet reception ratio; and,

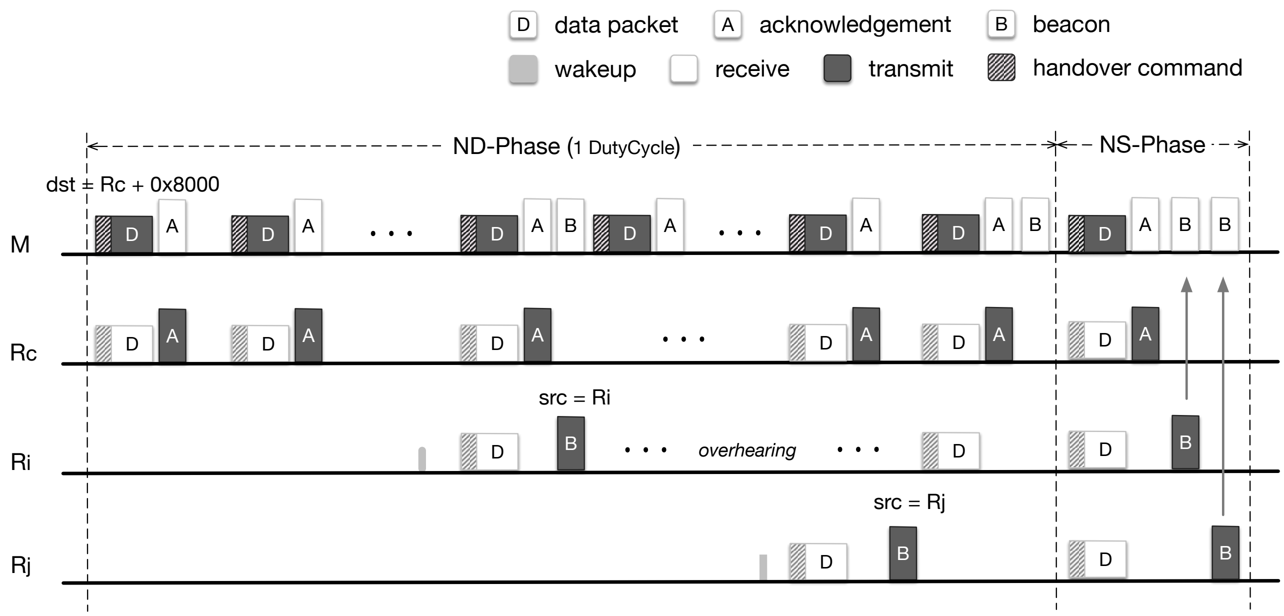


Fig. 3: Neighbour discovery (ND) and selection (NS) phases: The mobile sender (M) embeds a handover request in a data packet to probe potential receivers while keeping communication with the current receiver (Rc). In the meantime, nearby active relay nodes (Ri and Rj) overhear the handover request and estimate the quality of the link they establish between the mobile node and themselves.

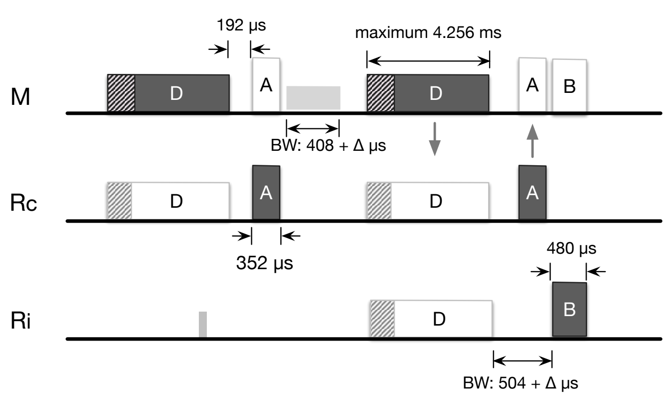


Fig. 4: The expected duration of a neighbour discovery phase. $\Delta$ is the small guard time added to the back-off window (BW).

3) the trend in the change of RSSI values in order to estimate whether the mobile node is moving towards the relay node or away from it. This phase is illustrated in Fig. 3.

The computation of these parameter is as follows:

**RSSI:** The RSSI values are collected by overhearing data packets in which a handover request is embedded. We apply an online moving average algorithm to amortize the calculation cost to each reception. The averaged RSSI value is calculated as:

$$\overline{r}_n = \overline{r}_{n-1} + \frac{r_n - \overline{r}_{n-1}}{n}, \text{where n > 1} \tag{2}$$

**Packet Reception Ratio (PRR):** The packet reception pattern is aggregated by a counter from the reception of the first packet containing a handover request to the arrival of the feedback request. The total number of handover request transmitted can easily be determined by examining the digital sequence number in the header.

**Mobility Trend:** The main aim of neighbour selection is to choose the most reliable next relay node to which the remaining packets of a mobile node can be transferred. To this end, estimating the mobility trend of the mobile node with respect to a potential relay node is necessary. Since neither the mobile nor the relay node has an explicit location information or mobility model, whether or no a mobile node is approaching or moving away from a relay node can only be estimated locally by the fluctuating pattern of RSSI values. For a short duration and distance, it is reasonable to model the change in the RSSI values as a linear function of time: $rssi(t) = at + b$. Then the changing rate can be estimated by a simple linear regression model[2], which results in:

$$a = \frac{cov(r,t)}{var(t)} \tag{3}$$

Thus, the link quality bidding metric $L_{bid}$ can be expressed as:

$$L_{bid} = \overline{r} + a \times s \times prr \tag{4}$$

where $s$ is the remaining number of packets which should be transmitted in burst and $prr$ is the packet reception ratio. Based on this input from each relay node, the mobile sender

[2]The constant $a$ is established by minimising the difference between $rsst(t)$ and its estimate $at + b$ in a mean square error sense.

chooses the one with the highest bidding value as its next relay node.

## III. Implementation

We implemented our protocol (henceforth called MSI-MAC) in TinyOS [11] for the TelosB [14] and IMote2 platforms [12], both of which integrate an IEEE 802.15.4 compatible radio (CC2420). A good portion of the code is hardware independent and can easy be ported to other platforms and operating systems.

*1) Anycast Communication:* The CC2420 radio chip does not support *anycast*, so we disabled the hardware address recognition and *auto-ack* features and delegated to the link layer the decision whether packet reception should be followed by the transmission of ACK packet or a beacon. This is implemented as follows: We reserved $0x8000$ as the *anycast* address. Any node receiving a data packet destined to this address and has a valid frame check sequence (FCS), responds with a beacon containing its address information without delay (without a CCA).

*2) Handover Request:* MSI-MAC does not introduce a new field in the MAC header but uses the most significant bit of the destination address to issue a handover request. For example, when a handover request is triggered, the destination address is set to $R_c+0x8000$, where $R_c$ is the current receiver address. This scheme has two benefits:

1) The unicast/ACK scheme remains intact during neighbour discovery. A designated relay node receiving the data packets in which a handover request is embedded can respond with an acknowledgement in the usual way by masking the handover request bit during the validation of the destination address.
2) All the other relay nodes, however, do not need to evaluate any part of the data packet except the header in order to determine whether this packet contains a handover request.

*3) Data and Beacon Format:* We extended the 802.15.4 MAC header with two additional fields, namely, "remains" and "opt" to encode the number of remaining packets in burst and the beacon's feedback during neighbour discovery, respectively. When a potential relay node responds to a handover request with a beacon, it randomly set one bit in the "opt" field. The mobile sender receives this beacon and sets the same bit in the "opt" field for next data transmission. If the relay node receives a handover request with the same "opt" bit, it will keep its radio on, continue overhearing handover request packets, but refrain from sending further beacons until the neighbour discovery period is over and the feedback request arrives. The "opt" field is used by relay nodes to determine the feedback transmission order during the neighbour selection phase. The beacon frame is varied from 13 bytes to 15 bytes and requires a maximum 480 $\mu s$ to transmit, the format is illustrated in Fig. 5.

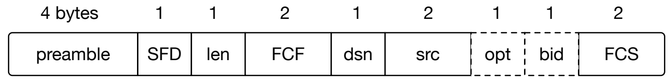


Fig. 5: The format of a beacon for responding to a neighbour discovery request.

## IV. Evaluation

We evaluated our protocol experimentally and compared it with three state-of-the-art protocols, namely, ME-Contiki, MX-MAC, and SmartHop (the authors provided us with the source code). We selected four metrics for our evaluation: packet success rate, the number of handover triggers, signalling overhead, and latency.

### A. Methodology

We performed the experiment with the MobiLab testbed [17] consisting of 3 to 10 TelosB and Imote2 nodes, depending on the specific experiments. One of these nodes was a mobile robot. The static nodes acted as relay nodes and were deployed in the corridor of our faculty along a straight line, with a separating distance of 5 m. The mobile sender carried by the robot moved at a constant speed of 0.13 m/s from one end of the corridor to the other end while transmitting packets in burst. The inter-packet-interval (IPI) is set to 10 ms which is the minimum interval between two outgoing packets that is currently supported by the TinyOS implementation. The transmission power is limited to -25 dBm. Each experiment is repeated 10 times. The following figures show the averaged results with error bars (standard deviation). For the performance comparison with the-state-of-the art, unless explicitly stated, we used 5 static relay nodes to minimise the probability of collision on beacons during neighbour discovery. The evaluation of more than four neighbours is shown in Table. I.

### B. Packet Success Rate

As illustrated in Fig. 6(a), the packet success rate of the three protocols are all above 97%, under different wakeup intervals. The reliability of MSI-MAC is a little bit lower than ME-Contiki, which is 98.77% against 97.57% for 1000 ms wakeup interval, while it is slightly higher than MX-MAC. The reason for the relatively low performance in this respect is that:

1) ME-Contiki is sensitive to a single packet failure and triggers handover upon a single packet loss, regardless of the link dynamics (i.e., irrespective of whether a mobile node faces a transient or a persistent link quality deterioration).
2) In contrast, MSI-MAC relies on two metrics coming from the physical and link layers to estimate the link quality fluctuation and exhibits a greater tolerance to transient link fluctuations. As a result, ME-Contiki experiences more handover oscillations (triggers), as depicted in Fig. 6(b).

TABLE I: Performance comparison: The wakeup interval of relay nodes was set to 1000 ms. The number of neighbours is the potential nearby relay nodes.

| Deployment | Protocol | PSR(%) | Handover Triggers (#) | Signaling Overhead (#) | Latency (ms) |
|---|---|---|---|---|---|
| neighbors: 2 spacing: 10 m | ME-Contiki | 98.79% | 63 (max: 83) | 1151.8 (max: 1529) | 11517.5 |
| | MX-MAC | 95.49% | 37.5 (max: 51) | 37.5 (max: 51) | 372.9 |
| | MSI-MAC | 95.97% | 11.5 (max: 15) | 46 (max: 60) | 293.3 |
| neighbors: 4 spacing: 5 m | ME-Contiki | 98.77% | 68.5 (max: 99) | 756.9 (max: 1166) | 7568.8 |
| | MX-MAC | 96.81% | 36.1 (max: 45) | 36.1 (max: 45) | 219.4 |
| | MSI-MAC | 97.57% | 8.6 (max: 13) | 69 (max: 104) | 228.6 |
| neighbors: 8 | ME-Contiki | 99.2% | 55.5 (max: 72) | 362.8 (max: 440) | 3627.5 |
| spacing: max 5 m | MX-MAC | 97.0% | 56 (max: 69) | 56.0 (max: 69) | 179.7 |
| min 2.5 m | MSI-MAC | 98.0% | 8.5 (max: 12) | 136 (max: 192) | 242.2 |

### C. Handover Triggers

Triggering a handover request at the appropriate time is essential to avoid unnecessary oscillations. If the handover trigger algorithm is too sensitive to link quality variations, more handover events are experienced, and consequently, the handover cost (signalling overhead, latency etc.) is high. On the contrary, if the algorithm is too tolerant to the link dynamics and fails to trigger a handover on time, the node may suffer from a considerable packet loss. Our results show that MSI-MAC reduces the number of handover triggers by about $12\%$ and $23\%$ compared to ME-Contiki and MX-MAC, respectively.

### D. Signalling overhead

The signalling messages are exchanged during neighbour discovery. Fig. 6(c) shows the number of signalling messages transmitted on average. ME-Contiki has the worst performance, because of the relatively poor neighbour discovery strategy it employs. To suppress the data packet duplication, instead of transmitting data frames, ME-Contiki *anycasts* a burst of control packets to search for a new receiver. This led to the highest signalling overhead and, as a result, a large number of handover triggers. In contrast, both MX-MAC and MSI-MAC embed handover requests within data packets. Hence, the only signalling overhead is due to the response beacons generated by potential relay nodes. The difference is that MX-MAC receives only one beacon from a relay node which responds the earliest whereas in MSI-MAC, each potential relay node transmits two beacons to express their readiness and to bid their suitability. As a result, the signalling overhead of MSI-MAC is almost twofold when compared with MX-MAC, and amounts to $9\%$ to $40\%$ of the overhead produced by ME-Contiki for different wakeup intervals. By contrast, MX-MAC introduces data packet duplication due to its data packet broadcasting scheme during neighbour discovery.

### E. Latency

In the context of seamless handover, latency is the time needed to establish a new link and resume burst communication via this link. Similar to the signalling overhead, latency is introduced during neighbour discovery. For ME-Contiki, latency arises due to the time spent during the transmission of the control packet and the waiting for acknowledgement.

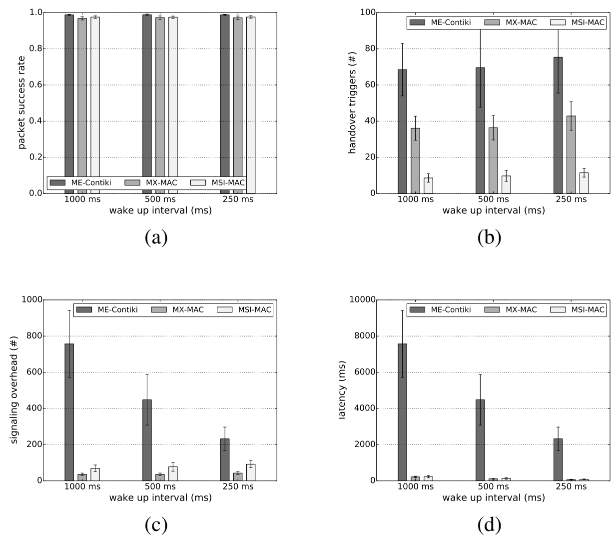


Fig. 6: Performance comparison: (a) packet success rate (b) the number of handover triggers (c) signalling overhead (d) latency averaged per handover trigger.

For MX-MAC and MSI-MAC, it is caused by the back-off time during the transmission of beacons by potential relay nodes. The neighbour selection phase in MSI-MAC contributes additional latency. Fig. 6(d) shows the averaged latency introduced by handover. As can be seen, the latency associated with MSI-MAC is significantly the smallest.

### F. Comparison with SmartHop

Fig. 7 shows the performance comparison between our protocol and SmartHop [9]. Since the duty cycling mechanism is not enabled in SmartHop, to make a fair comparison, we set the wakeup interval to 125 ms in MSI-MAC and the window size of neighbour discovery to 10 for SmartHop. In order to fix other parameters, such as the handover threshold and hysteresis margin for SmartHop, we performed a set of preliminary experiments and tuned them accordingly. SmartHop is a hard handover solution; in other words, the protocol first interrupts data transmission during neighbour discovery and associates a mobile node with an alternative relay node. Consequently, the signalling overhead and latency are significantly higher (4 to

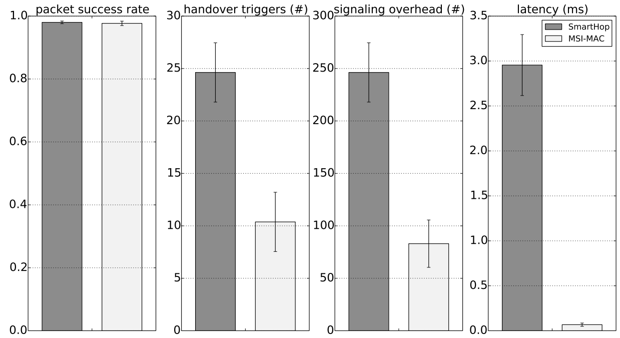


Fig. 7: Performance comparison with SmartHop. The wakeup interval for MSI-MAC is 125 ms. The RSSI threshold for triggering handover is set to -85 dBm for SmartHop, and the decision margin is 1 dB.

44 times greater) in SmartHop than those introduced by MSI-MAC. Furthermore, when the wireless link is highly dynamic, SmartHop performs even worse because it relies only on a single, unreliable metric (RSSI values) to estimate link quality fluctuation.

## V. Conclusion

In this paper we proposed a protocol for enable a seamless handover in wireless sensor networks supporting mobile nodes. The protocol carries out seamless handover by (1) enabling mobile nodes to quickly join the network, (2) continuously evaluating link quality and stability using PHY and MAC layer parameters and by implementing a Kalman filter, and (3) defining a bidding metric to select the best relay node amongst competing nodes. Moreover, our protocol supports burst transmission in order to enable a mobile node to transfer as many packets as possible when the link is stable. This approach also has the added benefit of collecting sufficient statistics for the Kalman filter, so that it can make reliable prediction pertaining to link quality fluctuation.

We implemented our protocol for TinyOS runtime environment and for TelosB and Imote2 platforms. Furthermore, we compared our protocol with three state-of-the-art protocols. Thus, through repeated experiments we demonstrated that our protocol was able to make reliable handover; reduce handover latency, overhead and oscillation; and deal with transient link quality fluctuations. In future, we are aiming to focus on 1) optimizing the protocol to further reduce the signalling overhead and latency by introducing mechanisms to quickly identify bad links, so that aimless handover attempts can be quickly aborted, and 2) accommodating and scheduling multiple mobile senders simultaneously.

## Acknowledgement


This work has been partially funded by the German Research Foundation (DFG) under project agreements: DA 1211/5-2 and DA 1211/7-1.


## References


[1] H. Alemdar and C. Ersoy. Wireless sensor networks for healthcare: A survey. *Computer Networks*, 54(15):2688–2710, 2010.

[2] P. D. Ba, I. Niang, and B. Gueye. An optimized and power savings protocol for mobility energy-aware in wireless sensor networks. *Telecommunication Systems*, 55(2):271–280, 2014.

[3] M. Buettner, G. V. Yee, E. Anderson, and 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*, pages 307–320. ACM, 2006.

[4] O. Chipara, C. Lu, T. C. Bailey, and G.-C. Roman. Reliable clinical monitoring using wireless sensor networks: experiences in a step-down hospital unit. In *Proceedings of the 8th ACM conference on embedded networked sensor systems*, pages 155–168. ACM, 2010.

[5] W. Dargie and J. Wen. A seamless handover for wsn using lms filter. In *LCN*, pages 442–445. IEEE, 2014.

[6] Q. Dong and W. Dargie. Analysis of the cost of handover in a mobile wireless sensor network. In *WWIC*, pages 315–322. Springer, 2012.

[7] A. Dunkels. The contikimac radio duty cycling protocol. 2011.

[8] S. Duquennoy, F. Österlind, and A. Dunkels. Lossy links, low power, high throughput. In *SenSys*, pages 12–25. ACM, 2011.

[9] H. Fotouhi, M. Zúñiga, M. Alves, A. Koubâa, and P. Marrón. Smart-hop: A reliable handoff mechanism for mobile wireless sensor networks. In *EWSN*, pages 131–146. Springer, 2012.

[10] A. Gonga, O. Landsiedel, and M. Johansson. Mobisense: Power-efficient micro-mobility in wireless sensor networks. In *DCOSS*, pages 1–8. IEEE, 2011.

[11] P. Levis, S. Madden, J. Polastre, R. Szewczyk, K. Whitehouse, A. Woo, D. Gay, J. Hill, M. Welsh, E. Brewer, et al. Tinyos: An operating system for sensor networks. *Ambient intelligence*, 35:115–148, 2005.

[12] L. Nachman, J. Huang, J. Shahabdeen, R. Adler, and R. Kling. Imote2: Serious computation at the edge. In *Wireless Communications and Mobile Computing Conference, 2008. IWCMC'08. International*, pages 1118–1123. IEEE, 2008.

[13] G. Z. Papadopoulos, V. Kotsiou, A. Gallais, P. Chatzimisios, and T. Noël. Wireless medium access control under mobility and bursty traffic assumptions in wsns. *Mobile Networks and Applications*, 20(5):649–660, 2015.

[14] J. Polastre, R. Szewczyk, and D. Culler. Telos: enabling ultra-low power wireless research. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 48. IEEE Press, 2005.

[15] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, et al. Smartsantander: Iot experimentation over a smart city testbed. *Computer Networks*, 61:217–238, 2014.

[16] K. Srinivasan, M. A. Kazandjieva, S. Agarwal, and P. Levis. The $\beta$-factor: measuring wireless link burstiness. In *SenSys*, pages 29–42. ACM, 2008.

[17] J. Wen, Z. Ansar, and W. Dargie. Mobilab: A testbed for evaluating mobility management protocols in wsn. In *Testbeds and Research Infrastructures for the Development of Networks and Communities*, pages 49–58. Springer, 2016.

[18] T. Zhiyong and W. Dargie. A mobility-aware medium access control protocol for wireless sensor networks. In *IEEE Globecom Workshops*, pages 109–114. IEEE, 2010.

[19] Z. Zinonos, C. Chrysostomou, and V. Vassiliou. Wireless sensor networks mobility management using fuzzy logic. *Ad Hoc Networks*, 16:70–87, 2014.