A Mobility-Aware Medium Access Control Protocol for Wireless Sensor Networks

Tang Zhiyong and Waltenegus Dargie
Chair for Computer Networks, Faculty of Computer Science
Technical University of Dresden
01062 Dresden, Germany
Email: mail2zhiyong@gmail.com, waltenegus.dargie@tu-dresden.de

***Abstract*—This paper introduces a light-weight mobility-aware medium access control protocol for wireless sensor networks (MA-MAC). In many respects, the protocol is similar to preamble based, low power listening MAC protocols. It defines a duty-cycle to let nodes sleep when they have no packets to transmit. During active communication, the protocol estimates mobility by evaluating the received signal strength of incoming ACK packets and initiates seamless handover by embedding neighbour discovery requests in data packets. Neighbour nodes randomly wake up from their sleep to participate in a handover process. Moreover, at the beginning of each active cycle, they first listen for handover request before they transmit. A prototype was developed and tested with TinyOS and Micaz platforms. The paper reports the experiment results.**

***Index Terms*—Handover, MAC protocol, mobility, neighbour discovery, wireless sensor networks**

I. Introduction

A large number of the applications proposed for wireless sensor networks (WSNs) require static deployments, even though the topologies of the networks may change over time due to node failures or changing link quality. Consequently, existing or proposed MAC protocols are mainly intended for static deployments. But there are applications in which the nodes are really mobile. For example, in healthcare applications WSNs monitor the activities of nurses [3], [7] and the movements of patients [4], [13]; and in supply-chain management, the safety and whereabouts of products are monitored [11], [12]. In such applications, the data collection in some vital circumstances should not be interrupted.

While mobility affects the entire communication stack, the medium access control and routing layers are the ones which are mainly responsible to deal with mobility. Therefore, MAC and routing protocols should be able not only to accommodate the coming and going of nodes but also provide a mechanism to provide an unbroken end-to-end communication link despite the mobility. Due to the complexity of the problem and space limitation, this paper focuses only on solutions at the MAC layer.

Proposed MAC protocols deal with mobility to some extent. For example, schedule-based protocols such as SMAC [17] enable nodes to synchronise their sleeping schedule periodically. This enables nodes to update their knowledge about their neighbours. However, exchanging and updating schedules are costly. Moreover, mobility is perceived only at the beginning of schedule synchronisation. Preamble based protocols such as XMAC [2] avoid the need for periodic synchronisation, enabling the responder to send ACK packets to a preamble whenever it completes its sleeping phase. The Receiver-Initiated MAC protocol (RI-MAC) [16] even avoids the cost of preamble by enabling a potential receiver to send a beacon to potential transmitters whenever it completes a sleeping phase. These approaches implicitly assume that the receiver is already known to the transmitters. However, with slight modification of the protocols, it is possible to accommodate mobile nodes. The main problem, however, is that these, too, perceive mobility only at the beginning of the listening phase. If a link is broken in the midst of a communication, there is no way to gracefully deal with it.

MS-MAC [14] extends SMAC to support mobility. Each node discovers the presence of mobility within its neighbourhood based on the received signal strength of periodical SYNC messages from its neighbours. A change in signal strength is perceived to be due to the mobility of either the neighbour or the receiving node itself. The level of the change in the received signals is also related to the speed of the mobile node. This information is used to create an active zone around a mobile node when it moves from one cluster to another cluster, so that the mobile node can expedite connection setup with new neighbours before it loses all its neighbours. In the active zone, nodes run the synchronization periods more often, resulting in higher energy consumption, but the time it takes to create new connections is lower. MS-MAC is not well developed and suffers from the same shortcomings of the SMAC protocol.

MMAC [1] is an extension to the TRAMA protocol [15]. TRAMA is a distributed TDMA-based MAC Protocol in which the size of a frame as well as slot distribution takes place dynamically. The protocol divides a time frame into two parts: a random access period and a schedule access period. The random access period is used to collect neighbour information. Each node uses an adaptive election algorithm to determine the slot which can be used to transmit packets. The schedule access period is then used to announce the schedule and perform the actual data transmission. MMAC uses a probabilistic autoregressive model to predicate the mobility of two-hop neighbours. It adjusts the time frame and random access time according to the mobility of nodes. This protocol handles both weak mobility and strong mobility where nodes physically move through the network, however, the algorithm is computation intensive.

Our protocol (MA-MAC) extends the XMAC protocol – a contention based protocol, which divides a preamble into multiple strobes to reduce the cost of preamble. MA-MAC detects mobility through the received signal strength of ACK packets during communication and switches from unicast to broadcast to interleave data communication with neighbour discovery. Unlike all the protocols discussed above, MA-MAC recognises deterioration in a link quality while communication takes place; and attempts to seamlessly handover a communication to a better link.

The rest of this paper is organised as follows: In Section II, the concept of the MA-MAC is introduced. In Section III, the prototype implementation of the MA-MAC protocol is discussed. In Section IV, the evaluation of the protocol is presented. Finally, in Section V, concluding remarks are given.

II. Concept

Conceptually, when MA-MAC detects mobility while a transmission is not completed, one of the following conditions occurs:

1) The node completes transmitting all the data before a link breaks, hence, there is no need to deal with mobility.
2) The node negotiates with the receiving node for dynamic rate adaptation, so that it can send all the data at a higher rate and complete transmission before the link breaks.
3) The node initiates and completes a handover before the link breaks.

These decisions require the link layer to rely on information from the application, network, and physical layers. The application/network layer provides information pertaining to data size. With it and additional information about the speed of travel and the transmission rate of the radio, it is possible to estimate the time required to transfer the data ($t_s = \frac{D_s}{r}$ and $t_s \leq \tau$; where D_s is the data size; r is the transmission rate; t_s is the time required to send the data; and τ is the remaining time before a link breaks).

The physical layer provides information about the transmission rates that can be supported by the radio. MA-MAC defines two distance thresholds to support seamless handover. The first threshold prompts it to initiate a handover while the second threshold sets an upper limit to the distance that should be travelled before the mobile node "binds to" a new intermediate node (i.e., before a handover is completed). Between the two thresholds, MA-MAC broadcasts data packets in which handover requests are embedded. Accordingly, the transmission mode of both nodes changes from unicast to broadcast to accommodate neighbour discovery. Neighbouring nodes randomly wake up from their sleep state to participate in a handover. But this wake up duration is markedly smaller than the active state defined by the duty cycle, and it is a tuneable parameter that depends on the network density. Additionally, at the beginning of their active state, nodes listen to neighbour discovery packets to participate in a potential handover process.

The currently available wireless standards (for example, the IEEE 802.15.4 [9]) and technologies [5] do not support rate adaptation in wireless sensor networks. Therefore, the main focus of the remaining part of this paper will be on the third aspect only.

A. Finite State Machine

Figure 1 illustrates MA-MAC with a finite state machine. As can be seen, a node can be found in one of the five states, namely, sleep, receive, send, discover, or handover.

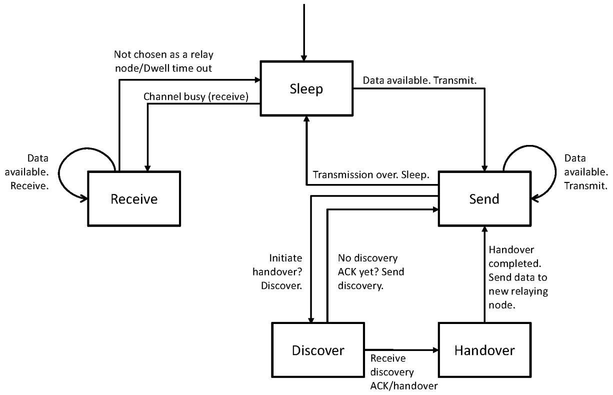
Not chosen as a relay node/Dwell time out
Sleep
Data available. Transmit.
Channel busy (receive)
Data available. Receive.
Receive
Transmission over. Sleep.
Send
Data available. Transmit.
Initiate handover? Discover.
No discovery ACK yet? Send discovery.
Handover completed. Send data to new relaying node.
Discover
Receive discovery ACK/handover
Handover

Fig. 1. The finite state machine of MA-MAC

Initially, a node is found in a `sleep` state after successfully booted. It may wake up at any time if it has data to send, in which case it enters into a `send` state. It remains in this state until it has transmitted all pending data[1]. Another condition for leaving the sleep state is when the node either randomly wakes up to participate in a handover process or begins the normal active period. Both conditions cause it to transit into a `receive` state – the former only for a brief amount of time. If a transmitting node detects that either the receiver or itself has crossed the first mobility threshold, it enters into a `Discovery` state in which the transmitting node searches for an intermediate neighbour before the actual link breaks. If it receives ACK packets from a neighbour before it reaches the second mobility threshold, it enters into the `Handover` state to "bind to" the newly discovered node and to update its routing configuration. If the node cannot discover any neighbouring node, it sends out discovery packets until the second mobility threshold is crossed. If the handover attempt is unsuccessful by then, it enters into a sleep state to avoid inefficient communication (this is not shown in the figure). Otherwise, it enters into a send state to resume unicast transmission via the new intermediate node.

B. Mobility Model

Our mobility model takes human activities and movements into account. The movement of people in most places can

[1] Not shown in the figure is the periodic transitions between send and receive states during the entire period of data transmission and acknowledgements.

be characterised as an intermittent and slow movement (with an average speed of 1.5 mps). In rehabilitation centres and hospitals the movement of people is even slower (0.5 mps). Physical surroundings and the presence of other people around constrain mobility. Hence, the model we consider makes the following assumptions:

1) Deterioration of a link's quality due to mobility is a gradual but a steady process;
2) Mobile nodes are surrounded by some quasi-static nodes.

Similar to all low duty cycle MAC protocols, MA-MAC enables a node to sleep most of the time and periodically switch on the radio to listen for incoming packets. It has two operational modes: static and mobile. In the static mode, MA-MAC performs similar to XMAC. When it operates in a mobile mode, however, the RSSI of the incoming ACK packet is evaluated to assess the quality of a link (assumed to be degraded as a result of mobility). If a mobile node crosses the first threshold, the transmitter begins to embed handover requests into the MAC protocol data unit (MPDU), sets the address to broadcast mode and sends the data packets in this fashion until an acknowledgement is received from a neighbour node.

C. Mobility Estimation

There is a substantial body of work on mobility and distance estimation in wireless sensor networks. Most approaches exploit RSSI or SNR measurements. Zaidi and Mark [18] propose an autoregressive model to predict the mobility of a node from its past mobility history. It gives the mobility state of a node at the current time in terms of the position, velocity, and acceleration. But the model is too computationally intensive to be implemented on a wireless sensor node. Farakas et al. [6] apply cross-correlation and a pattern matching algorithm to predict link quality variations. The combined computation cost of correlation and pattern matching makes the technique likewise expensive. Ji and Zha [8] apply multidimensional scaling and coordinate alignment techniques to estimate the position of nodes. However, their approach requires the presence of a large number of anchor (reference) nodes.

MA-MAC does not require precise information about the direction and magnitude of mobility. It is sufficient to know whether the quality of a link is being gradually and steadily degraded so much so that communication between two nodes at a given packet arrival rate cannot be supported. Neighbour nodes participate in a handover request only if they are along the way to the base station. One such model is proposed by Texas Instruments for the Chipcon CC2420 radio [5]:

$$RSSI = -\left(10\gamma \cdot log_{10}d + A\right) \quad (1)$$

where γ is the signal's propagation constant; d is the distance between the transmitter and the receiver; and A is the received signal strength of a line-of-sight link at 1 m distance. The main problem with this model is its disregard of the impact of multi-path scattering and non-uniform propagation characteristic of the radio frequency. This is particularly the case in indoor environments. We slightly modified this model to avoid unnecessary oscillation. Even though the model is simple and error prone, it was sufficient to demonstrate the operation of MA-MAC. We are still working for a more robust and accurate model.

D. Mobility mode of data transmission

Figure 2 illustrates the mobility mode of MA-MAC during data transmission. After a CCA, the sender, *S1*, transmits a series of short preambles frames until the receiver *R1* replies with an ACK frame. At this moment, *S1* does not have enough mobility information to carry out mobility estimation. Therefore, it sends out a new data packet to *R1*. Meanwhile, the mobility estimation scheme at the background accumulates the RSSI values of incoming ACK packets until a queue fills (the size of the queue depends on the sampling rate, the speed of mobility and the minimum distance that should be reached to initiate a handover – at present, this is determined at compilation time) and then decides whether or not a handover should be initiated. In case a handover is necessary, the protocol enters into a handover state. In this state, *S1* embeds handover requests[2] in the outgoing data packets and sends out the packets in a broadcast mode. Upon receiving the first broadcast packet, *R1* will keep refrain from sending for a brief moment to enable *S1* to collect the handover reply. If there is no active node at the time, *S1* resumes sending out handover requests until the second threshold is crossed. Following the first handover request (assuming the request is intercepted), *R2* and *R3* will send back a handover reply at a random. *S1* picks up the first relay node (say from *R2*) and sends a new data packet with the control field `Handover` set to FALSE. At this time *R1* attempts to transfers all state information (if there is any) to the relaying node, but if it cannot reach it, it simply enters into a sleep state. It will, however, wait a "dwell time[3]" to receive packets from other nodes. Meanwhile, *R2* receives the broadcast data packet from *S1* and learns that it has been selected as a new relay node. It keeps its radio on and prepares to relay packets from *S1*. *R3* overhears the data packet too, and learns that it has not been chosen as a relay node. Therefore, it enters back into the sleep state to save energy.

III. Implementation

MA-MAC extends the XMAC implementation, UPMA [10] and inherits several features of the TinyOS message header – frame length, frame control field, data link sequence number, and address – and introduces additional control headers to support handover. A higher-level protocol implementation is shown in Figure 3. Even though the IEEE 802.15.4 frame specification permits 4 - 20 bytes of the address field, this is fixed at 4 bytes in the TinyOS implementation. Since the moteID in a TinyOS message refers to the destination address,

[2] This is a neighbour discovery request with the aim to directly use the responding node as an intermediate node.

[3] This parameter is defined in the XMAC protocol.

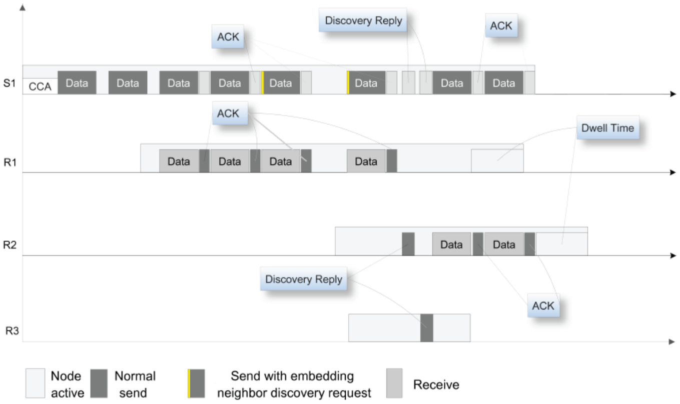

Fig. 2. Mobility mode for data transmission in MA-MAC

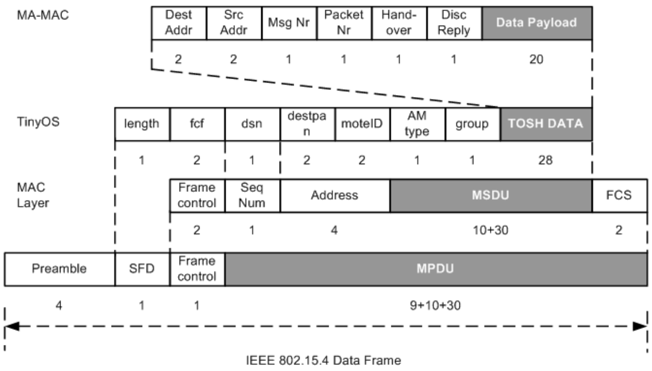

Fig. 3. Placement of the MA-MAC header in the TinyOS radio stack

in MA-MAC, the source node's address is embedded in the TOSH_DATA field. Moreover, TinyOS defines the default size of a message payload to be 28 bytes. During handover, 7 of these bytes are used to embed handover request, i.e., the handover control fields.

We ported the UPMA implementation onto the Micaz platform, as the presently available implementation runs on the TelosB platform. This includes porting the code running on the MTS serial microcontroller to ATmega128 serial microcontroller.

`Dest addr` and `Src addr` are used to identify the message which may be split into several packets. The `MsgNr` field is used to re-construct individual received packets into a complete message. The `PacektNr` indicates the order of the packets. The Boolean field `Handover` is useful for an implicit neighbour discovery request. If the `Handover` field is TRUE and the destination address is not the same as the local node which receives the data packet, this local node would reply to the discovery request and keep its radio on until a confirmation packet from the handover initiating node arrives. Once a handover is completed, the `Handover` field is set back to FALSE. The `Disc_Reply` field is a Boolean field and serves as a response to a handover (neighbour discovery) request. When a nearby neighbour replies to the handover request, it sets this field as TRUE and leaves all the other bytes empty.

The Micaz platform integrates an IEEE 802.15.4 compatible radio, which is based on a digital direct sequence spread spectrum baseband modem and provides a spreading gain of 9 dB and an effective data rate of 250 kbps [5]. A TinyOS implementation of this hardware consists of many layers that reside between the application and the radio hardware. The higher-level components in the radio stack modify the data and control headers in each packet as it progresses towards the low level components.

The link layer relies on `Active Messages`, which are packets that specify a handler ID in their header. They are called so because they trigger the invocation of a named handler upon receipt, pre-empting any ongoing computation. The active messages provide an unreliable single-hop datagram service. We keep this layer as the highest layer in our implementation. The `CSMA (Carrier Sense Multiple Access)` layer is responsible for defining IEEE 802.15.4 FCF (Frame control Field) byte information in outgoing packets. It provides a default back-off time when the radio detects a channel in use and defines the power-up/power-down procedure for the radio. We keep this as the lowest layer in our implementation to connect to the radio hardware via the `ReceiveP` and `TransmitP` components.

Between the `Active Message` layer and the `CSMA` layer, we insert the `MA-MAC` layer. This layer consists of the modules which are responsible for low power listening, mobility estimation, work mode adaptation and data link handover.

IV. Evaluation

Six Micaz sensor nodes were used in the experiment. One of these nodes was connected to a laptop computer and served as a base station. Three of the nodes were placed along a straight line (L1) that was about 20 m away from the base station. Two mobile nodes moved along a straight line (L2) that was about 40 m away from L1. The length of L2 was about 60 m. There were two main objectives for conducting the experiments: (1) To investigate the factors that contribute to the deterioration of a link's quality and quantify their contributions; and (2) to determine the impact of a handover on the performance of the network – in terms of packet loss and latency.

The nodes setup a 2-hop network in which intermediate nodes were chosen based on the quality of links they establish with the mobile nodes. We carried out the experiment first without a handover mechanism and then with the handover mechanism to observe the packet arrival rates in both scenarios. We consider the following parameters: sending interval (30 ms and 250 ms); speed of mobility (fast and slow movements); and distance of communication (when L2 was 40 m away from L1 and when L2 was 20 m away from L1). The relationship of each parameter with the average RSSI was examined by keeping all the other parameters constant. The experiment without the handover mechanism was used to determine the two RSSI thresholds which were useful for the handover.

The mobile nodes moved in and out of the communication range of the static nodes while sending and receiving data

Role	ROM(Bytes)	RAM(Bytes)
Base Station	15244	1726
Relay Station	12178	956
Mobile node	12368	314

TABLE I
THE MEMORY FOOTPRINT OF THE MOTES IN THE EXPERIMENT

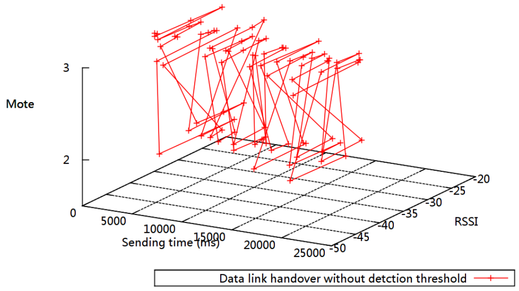

Fig. 4. Handover without the oscillation avoidance mechanism.

packets. When the received signal strength was below the set threshold, the mobile nodes initiated a handover. In order to focus on the handover strategy, we disabled some of the features of the MA-MAC protocol, such as neighbour table management. Moreover, we separated data transmission into sending and receiving features. The intermediate nodes and the base station were deployed with both features enabled whereas the mobile nodes were deployed with only the sending feature enabled. This does not mean, however, the mobile node did not receive packets: it received ACK and handover reply packets. The intermediate nodes received data packets from the mobile nodes and forward them to the base station.

Table IV shows the memory footprint of the application code, according to the features that were enabled. Because the intermediate nodes managed a queue, their RAM usage is larger than the mobile nodes. The base station has the largest RAM usage, because it has two queues: one is for the wireless link and the other for the serial port.

A. Oscilliation

One of the problem of relying on a simple model of mobility, such as the one discussed in Section II is that the handover oscillation is high. This is rather undesirable, as the magnitude of the RSSI fluctuates even when the user is not actually moving. To avoid unnecessary oscillation, a simple algorithm based on the mean crossing rate was implemented in the MA-MAC protocol. The algorithm simply counts how often a mean value in a sliding window is crossed and outputs a “True” or “False” value according to which a handover request is triggered. The mean value crossing rate is empirically defined. Figure 4 and 5 shows the different in handover ocilliation with and without the algorithm, respectively.

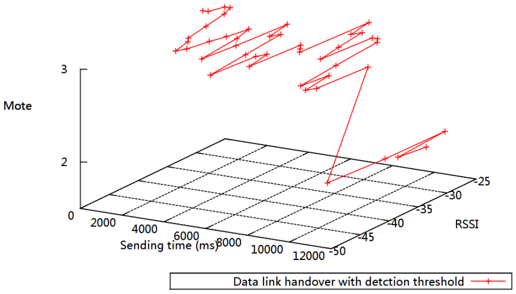

Fig. 5. Handover with the oscillation avoidance mechanism.

B. Packet loss

Apart from the oscillation, experiments were conducted to investigate the impact of handover on the packet arrival rate and latency. In this paper, only packet loss will be further considered. The packet loss was evaluated by varying the sending interval, the RSSI threshold (implicitly, the distance of travel), and the speed of mobility. All the other settings were the same as above. The minimum RSSI threshold for establishing a link was set to either -40 or -30 dBm.

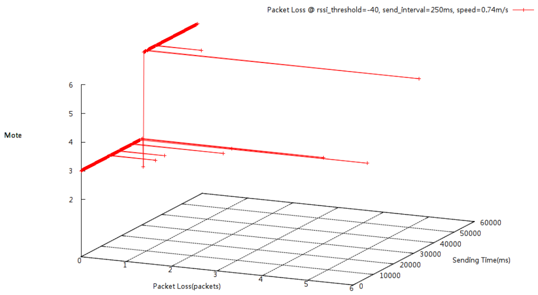

Fig. 6. Packet loss with sending interval = 250 ms; RSSI threshold = -40 dBm; seed = 0.74 m/s; and distance = 20 m.

Figure 6 shows the packet loss profile for interval = 250 ms; RSSI threshold = -40 dBm; seed = 0.74 m/s; and distance = 20 m. The x-axis shows the number of unaccounted packets; the y-axis shows the ID of the nodes between which a handover takes place; and the z-axis shows the sending time in ms. Intuitively, it may appear that a packet loss rate is highly handover dependent. But this is not necessary true. As can be seen in the figure, the worst packet loss occurred when Node-3 was used as an intermediate node and the link was stable. Figure 7 demonstrate that packet loss is not much affected by the speed of mobility as much as it is affected by the sending interval. The worst case is observed when all the three factors, distance between the nodes, speed of travel and sending interval are combined. This is shown in Figure 8).

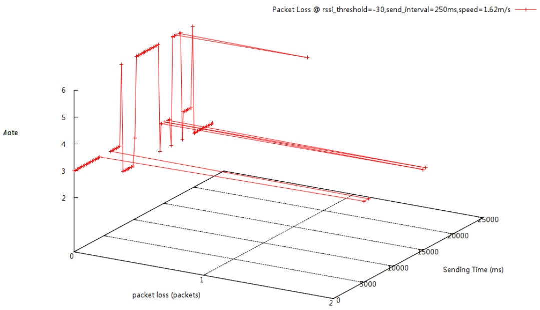

Fig. 7. Packet loss with sending interval = 250 ms; RSSI threshold = -30 dBm; seed = 1.62 m/s; and distance = 20 m.

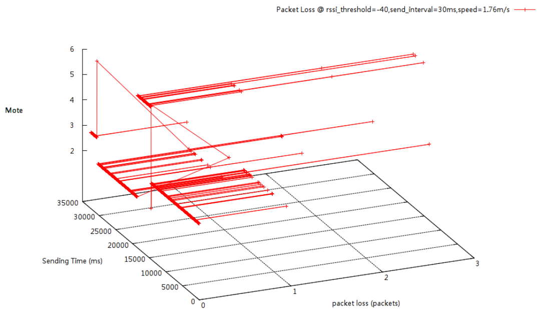

Fig. 8. Packet loss with sending interval = 30 ms; RSSI threshold = -40 dBm; seed = 1.76 m/s; and distance = 40 m.

V. Conclusion

In this paper a mobility-aware MAC protocol is introduced. Its main contribution is estimating mobility while communication still goes on and seamlessly initiating a handover when the link quality deteriorates. The Protocol embeds neighbour discovery requests inside data packets, which is then broadcasted to all recipients. Sleeping neighbour nodes randomly wake up for a brief amount of time to participate in a handover process. This wakeup time should not be confused with the wakeup time defined by the duty cycle. The random wakeup is used only to support a handover; for how long and how often a node should interrupt its sleep depends on the mobility scenario defined by the application and the network density.

MA-MAC extends XMAC and its implementation extends UPMA, developed by Klues et al. The prototype was deployed on the Micaz platform that runs the TinyOS runtime environment. Experimenting with the MA-MAC enables us to observe several open issues. The first problem is to develop an accurate and yet computation efficient mobility estimation model. The model used at present is unreliable. Secondly, The IEEE 802.15.4 specification does not allow the implementation of XMAC as it has been defined by the authors, i.e., it is difficult to define small size strobes to minimise the cost of preamble packets. Apart from these, work remains to quantify the probability of successful handover, since node wake up randomly.

References

[1] M. Ali, T. Suleman, and Z. A. Uzmi. Mmac: a mobility-adaptive, collision-free mac protocol for wireless sensor networks. In *24th IEEE International conference on Performance, Computing, and Communications*, pages 401–407, 2005.

[2] M. Buettner, G. V. Yee, E. Anderson, and R. Han. X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks. In *SenSys '06: Proceedings of the 4th international conference on Embedded networked sensor systems*, pages 307–320, New York, NY, USA, 2006. ACM.

[3] M. Cheng, M. Kanai-Pak, N. Kuwahara, H. I. Ozaku, K. Kogure, and J. Ota. Dynamic scheduling based inpatient nursing support: applicability evaluation by laboratory experiments. In *Casemans '09: Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, pages 48–54, New York, NY, USA, 2009. ACM.

[4] S. Dagtas, Y. Natchetoi, and H. Wu. An integrated wireless sensing and mobile processing architecture for assisted living and healthcare applications. In *HealthNet '07: Proceedings of the 1st ACM SIGMOBILE international workshop on Systems and networking support for healthcare and assisted living environments*, pages 70–72, New York, NY, USA, 2007. ACM.

[5] R. de Paz Alberola and D. Pesch. Avroraz: extending avrora with an ieee 802.15.4 compliant radio chip model. In *PM2HW2N '08: Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pages 43–50, New York, NY, USA, 2008. ACM.

[6] K. Farkas, T. Hossmann, F. Legendre, B. Plattner, and S. K. Das. Link quality prediction in mesh networks. *Comput. Commun.*, 31(8):1497–1512, 2008.

[7] T. Inomata, F. Naya, N. Kuwahara, F. Hattori, and K. Kogure. Activity recognition from interactions with objects using dynamic bayesian network. In *Casemans '09: Proceedings of the 3rd ACM International Workshop on Context-Awareness for Self-Managing Systems*, pages 39–42, New York, NY, USA, 2009. ACM.

[8] X. Ji. Sensor positioning in wireless ad-hoc sensor networks with multidimensional scaling. In *INFOCOM*, 2004.

[9] E. Karapistoli and F.-N. Pavlidou. An overview of the ieee 802.15.4a standard1. *IEEE Communication magazine*, 48(1), 2010.

[10] K. Klues, G. Hackmann, O. Chipara, and C. Lu. A component-based architecture for power-efficient media access control in wireless sensor networks. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 59–72, New York, NY, USA, 2007. ACM.

[11] M. Malinowski, M. Moskwa, M. Feldmeier, M. Laibowitz, and J. A. Paradiso. Cargonet: a low-cost micropower sensor node exploiting quasi-passive wakeup for adaptive asychronous monitoring of exceptional events. In *SenSys '07: Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 145–159, New York, NY, USA, 2007. ACM.

[12] R. Marin-Perianu, M. Marin-Perianu, P. J. M. Havinga, and H. Scholten. Movement-based group awareness with wireless sensor networks. In *Pervasive*, pages 298–315, 2007.

[13] M. J. Mathie, A. C. F. Coster, N. H. Lovell, and B. G. Celler. Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement. *Physiological Measurement*, 25(2):1–20.

[14] H. Pham and S. Jha. An adaptive mobility-aware mac protocol for sensor networks (ms-mac).

[15] V. Rajendran, K. Obraczka, and J. J. Garcia-Luna-Aceves. Energy-efficient, collision-free medium access control for wireless sensor networks. *Wirel. Netw.*, 12(1):63–78, 2006.

[16] Y. Sun, O. Gurewitz, and D. B. Johnson. Ri-mac: a receiver-initiated asynchronous duty cycle mac protocol for dynamic traffic loads in wireless sensor networks. In *SenSys '08: Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 1–14, New York, NY, USA, 2008. ACM.

[17] W. Ye, J. Heidemann, and D. Estrin. Medium access control with coordinated adaptive sleeping for wireless sensor networks. *IEEE/ACM Trans. Netw.*, 12(3):493–506, 2004.

[18] Z. Zaidi and B. Mark. Mobility estimation for wireless networks based on an autoregressive model. In *Global Telecommunications Conference (GLOBECOM 04)*, volume 6, pages 3405–3409, 2004.