HAMA: A Herd-Movement Adaptive MAC Protocol for Wireless Sensor Networks

Eyuel D. Ayele, Nirvana Meratnia, Paul J.M Havinga
Pervasive Systems Group, University of Twente, Enschede, the Netherlands
Email: {e.d.ayele, n.meratnia, p.j.m.havinga}@utwente.nl

***Abstract*—In this paper we propose HAMA, i.e., a herd-movement adaptive medium access control protocol suitable for wireless sensor networks with mobile nodes. The specific focus of HAMA is wildlife monitoring applications, in which network protocol is required to adapt to the movement patterns of herds to make the communication more energy-efficient and reliable. The protocol is an extension of preamble sampling scheme with an adaptive sleep-interval based on network traffic conditions. We have implemented and evaluated HAMA on Contiki Cooja platform. Our simulation results show 22.28%-52.28% reduction of average network energy consumption as well as 11.65%-14.63% reduction of average end-to-end latency when HAMA is compared with A-MAC and X-MAC. The overall packet reliability of the gateway node(s) is also increased by up to 16.3%.**

Keywords*—*Wireless Sensor Network (WSN), MAC Protocol, Duty-Cycle, Sleep-time, Latency, Reliability

I. Introduction

Recent studies suggest that node mobility plays an important role in network topology dynamics because of its impact on the communication link quality [1, 2]. Even though existing approaches provide a short term solution for low node mobility in Wireless Sensor Networks (WSNs), they usually neglect the impact of high node mobility [3, 4]. Consequently, such high mobility will often introduce undesirable energy consumption and end-to-end latency [2].

The direct impact of node movement on radio link quality will lead to change of effective node degree, hence, changing the network topology accordingly [1, 2]. As far as MAC protocols are concerned, these variations in node degree could be translated into a change in incoming or outgoing data traffic locally at each sensor node. Higher number of received or transmitted packets means higher amount of energy consumption and communication overhead.

Despite the need to quantify node movement characteristics directly in terms of the packet reception and transmission rates to design a mobility-driven MAC protocol, not much attempt has been made in this direction so far. Existing efforts to mitigate the effect of node mobility often ignore the use of local packet statistics to predict state of nodes movement. They enable nodes to perceive a change in their surrounding only at the beginning of each active period. Consequently, there is a delay in packet transmission whenever topology changes. This delay is even higher in multi-hop networks.

In this paper we propose a highly adaptive mobility-aware medium access protocol called HAMA. The specific focus of HAMA is wildlife monitoring applications, in which network protocol is required to adapt to the movement patterns of herds to make the communication more energy-efficient and reliable. The ultimate objective of HAMA is to offer a reasonably high response time (low Latency), low power consumption, and high packet delivery ratio in mobile WSNs. To achieve this objective, HAMA locally calculates nodes' duty-cycle using their received and transmitted packet statistics.

This paper makes the following contributions: (i) HAMA protocol, which extends the B-MAC[3], by developing an adaptive duty-cycling scheme to adjust preamble polling times according to packet traffic conditions, and (ii) A light analytical model, which keeps track of the packet traffic changes to decide on the duty-cycle interval. It uses the $G/G/1/K$ queue model with feed-back control to predict the run-time duty-cycle. This makes HAMA protocol more suitable for supporting continuously changing network topology, as will be described in Section III-D.

The structure of this paper is as follows: Section II will further describe the current state-of-the-art. Section III presents our protocol design. Performance evaluation of HAMA is presented in Section IV, while concluding remakrs are presented in Section V.

II. Related Work

Performance analysis of existing MAC protocols have shown that asynchronous duty-cycled MAC protocols, such as [3–5], perform well in terms of energy consumption, latency, and throughput [2] for applications of WSN in which nodes move slowly. For high mobility WSN applications, the synchronized MAC protocols (TDMA based MAC protocols) will not suffice mainly because of the overhead associated with the synchronization process. In asynchronous duty-cycled MAC protocols, there is no need to continuously share synchronization information. They rather proactively sample the channel to determine when an activity can be performed. As presented in [3, 6], the overall effective active period for asynchronous MAC protocols could be much shorter than of synchronous protocols. As such their energy consumption may be significantly lower.

X-MAC [4] is a MAC protocol based on asynchronous listen-intervals. For each packet, X-MAC transmits a strobe of preambles, between which the receiver can signal reception-readiness with a so-called EarlyACK. X-MAC derives a formula for optimal wake/sleep intervals given traffic at a certain fixed rate and outlines a mechanism to let it adapt the duty cycle and the sleep/wake interval to best accommodated the traffic load in the network. Since the basic mechanism of X-MAC still requires a certain fixed minimum interval between two active intervals and a generally high overhead per-packet, both its latency and energy consumption are high. As such it is only applicable to WSNs with low node movement.

A-MAC [5] is a receiver-initiated duty-cycled MAC that aims at handling variable traffic. Receiver nodes frequently wake up and send out probing beacons for polling possible incoming data without using preamble packets. A-MAC enhances throughput by reducing medium occupation. However, it suffers from packet collisions when there are multiple senders. A-MAC throughput is acceptable for applications with many bursts or heavy traffic.

Wang et al. [7] proposed an off-line data-driven MAC protocol which is initialized by fixed duty-cycle duration for different packet rates as required by specific scenarios. For every packet arrival rate, it will set a specific duty-cycle duration and power consumption values. The duty-cycle and power consumption information are locally placed in a table, based on which every sensor node autonomously selects its duty-cycle at run-time. This procedure is not suitable for a network which experiences a high trend of topology dynamics.

pTunes [8] is a centralized protocol to adjust the MAC layer settings on-line in such a way that an appropriate trade-off between network lifetime and application requirements in terms of end-to-end reliability and latency can be found. It employs a flooding algorithm to disseminate information pertaining to the state of the network. The optimal MAC parameters are determined centrally out-side of the network. The disadvantage of pTunes is that its large communication overhead for centrally monitoring MAC layer parameters and flooding the adjusted settings back to the nodes.

The Mobile Cluster MAC (MCMAC) [9] is a schedule-based MAC protocol which extends LMAC [10] to support cluster mobility. Unlike most of the proposed mobility-aware MAC protocol, it is optimized for those nodes which travel in group. MCMAC categorizes the sensor nodes into a static network and a mobile cluster and it then defines a Reference Point Group Mobility (RPGM) model and a Random Waypoint Mobility (RWM) model to mimic the movement characteristics of mobile clusters and the individual node movement within cluster. Static nodes communicate with each other during their active time by dynamically occupying a unique transmission slot in their two-hop neighborhood. The group membership of the nodes and topology changes caused by the inter-cluster mobility leads to the disconnection of the mobile node from the network.

To overcome the high energy consumption of the MS-MAC protocol [11] due to its inappropriate mobility prediction model produced using only RSSI, in [9], an enhanced MS-MAC protocol named EMS-MAC was introduced to predict the nodes movement more accurately using both received signal strength indication (RSSI) and link quality indication (LQI).

Li et al. [12] proposed a distributed algorithm to control the duty-cycle duration of nodes by employing convex-optimization. Nodes adjust their sleep time locally by exchanging their current duty-cycle interval and energy consumption with their neighbors. The algorithm is self-adaptive for different traffic loads. Authors of [13] and [14] proposed effective queueing models in a feedback control system to dynamically adjust the duty-cycle interval of a node. When the pre-defined queue size is reached, the MAC protocol sets a duty-cycle that leads to low energy consumption and delay. Although analytic and simulation results confirm usefulness of these types of models, they are often too complex.

III. Protocol Design

A. Design Overview of HAMA

HAMA inherits several features from B-MAC [3], such as initialization of its access scheme by the sender and preamble listening. It additionally introduces features such as adaptation to polling and duty-cycle duration. HAMA has three modes of operation, i.e., (i) transmission, (ii) reception, and (iii) adaptive duty-cycling. The main difference between HAMA and B-MAC is related to the third mode, as transmission and receiving modes are similar to B-MAC. In what follows, we explain each of these three modes of operation.

B. Transmission

Figure 1 shows sender initiated MAC protocol operation, in which packet transmission is done by performing a series of Clear Channel Assessments (CCAs). Transmitting nodes send a preamble with the header identifying the respective receivers. They first send a polling preamble and then transmit the packet to the receiver. Applying preambles will allow implementation of adaptive low power listening.

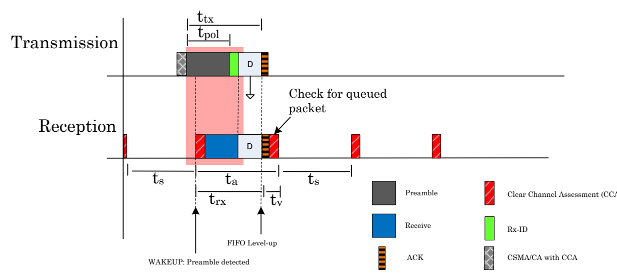


Fig. 1: HAMA protocol, t_s is the sleep duration, t_v the variable time spent after reception of the first packet, its duration depends on the next status of the transceiver. t_{pol} is the length of the transmitter's polling preamble, t_{tx} is the time interval needed to complete transmitting a packet.

C. Reception

As shown in Figure 1, receiving nodes periodically sample the CCA after waking up from sleep at the end of every sleep-time t_s. If transmitter's preamble is detected, the receiver(s) will keep their radio on until the transmitter finishes sending the preamble, even if the receiver(s) has already sensed the preamble signal. After this, receiver nodes determine whether they were the intended receiver by decoding the address header. If a receiver realizes that it is not the targeted receiver, it goes back to sleep immediately. By doig so, it frees the carrier channel and prevents any possible collision due to contention of the channel.

When there is an ongoing active packet exchange, the target receiver node stays awake for a period of t_a to finish receiving the packet after which it sends back an acknowledgement. The total time a node spends to receive a packet is expressed by $t_a = t_{rx} + t_v$, t_{rx}, where t_v shortens or extends the active period depending on availability of packets queued for next transmission or reception. Practically, t_v value is directly dictated by incoming packets and/or nodes own packet generation rates as describe in Section III-D. While t_{pol} is the length of the transmitter's polling preamble, t_{tx} is the time interval needed to complete transmitting a packet.

D. Adaptive Duty-cycling Mechanism

As shown in Figure 1, a sensor node becomes active after sleep-time t_s. It then will either try to send data from its buffer or sample the channel for any activity to receive packets. After active time is expired, the sensor node goes to sleep again for t_s time period. We aim to analytically determine the optimal sleep-time duration (t_s) by analysing the packet activity pattern in the queue model. To do so, we model the MAC layer as G/G/1/K queue system with idle-times. The model is used to drive stochastic estimation for sensor nodes sleep-time (t_s).

Algorithm 1 summarizes our basic steps towards computing sleep time and consequently the duty-cycle.

There are N regenerative cycles in a single control period (T_{cp}). A control period also serves as an observation window for collecting the statistics for predicting the next sleep duration of a node.

For the busy cycle of $G/G/1/K$ queue, packets arrive at time epochs λ_n^{-1}, $(n = 0, 1, 2, ...)$, and λ_n^{-1} be random variables. Let the packet transmission time of the n^{th} packet be S_n, and $S_n = (\mu^{-1}{}_n)$, $(n = 1, 2, ...)$ is random variables. Their mean values are denoted by, $E(\lambda_n^{-1}) = \frac{1}{\lambda}$ and $E(S_n) = \frac{1}{\mu}$, respectively. W_n $(n = 1, 2, ...)$ is the waiting time of the n^{th} packet. Within a single control period, Algorithm computes λ_n^{-1} and S_n for all incoming and outgoing packets.

After N regenerative cycles are being completed, Algorithm 1 computes the actual estimated idle-time ($Ti_{s,i}$) for the i^{th} control period $T_{cp,i}$, $(i = 1, 2, ...)$, which is calculated using the expected mean value of the individual idle-times ($X^{(r)}$):

$$Ti_{s,i} = E[X^{(r)}] = \frac{\sum_{r=1}^{N} X^{(r)}}{N} \tag{1}$$

where $X^{(r)} = -min(0, W_n^{(r)} + S_n^{(r)} - \lambda_n^{-1(r)})$ [15]. The optimal sleep time can be calculated using:

$$t_{s,i} = Ti_{s,i} + \varepsilon(K - K_i) - \xi(K_i - K_{i-1}) \tag{2}$$

where K is the max threshold queue size empirically determined (to be described in Section IV). $Ti_{s,i}$ is the current estimated sleep, and $t_{s,i}$ is the next estimated sleep-time duration for the i^{th} control period $T_{cp,i}$. ε and ξ are queue stability parameters. To have a stable queue system controller, (ε , ξ) should satisfy the following criteria:

Algorithm 1 Optimal duty-cycle Estimation

Input: N, K, λ_n^{-1}, S_i, W_n
Output: μ^{-1}, λ^{-1}, X^r, K_{i+1}, $T_{cp,i}$

```
procedure SLEEP TIME COMPUTATION
    i ← 1  //Initializing the Regenerative Cycle
    r ← 1
    if r < N then  //The r^th Regenerative Cycle is over
        λ_n^{-1} ← λ_n^{-1}
        S_n ← S_n
        X^(r) ← −min(0, W_n^(r) + S_n^(r) − λ_n^{-1(r)})
        r++
    end if
    if r = N then  //The i^th Control period is over
        β ← μ − λ
        Ti_{s,i} ← (Σ_{r=1}^{N} X^(r)) / N
        K_i ← max(0, (λ − μ)(T_{cp,i} − N Ti_{s,i}))
        ε ← ε  //Selecting the values for
        ξ ← ξ //the controller from the stable range
        t_{s,i+1} ← Ti_{s,i} + ε(K_{i−1} − K_i) − ξ(K_i − K_{i−1})
        r ← 1
        i ← i + 1  //Start monitoring for the next T_{cp,i}
    end if
end procedure
```

$$0 < (\varepsilon + \xi) < \Omega \tag{3}$$

$$0 < \xi < \frac{1}{(N\beta - \beta + 1)} \tag{4}$$

$$\Omega = \frac{2(1 + \beta)}{\beta(N\beta - \beta + 1)} \tag{5}$$

where $\beta = (\lambda - \mu)$, and $\Gamma = (\xi + \varepsilon)\beta$, ε and ξ are the control parameters. Since λ, N, and μ are known, β and Ω, and consequently ε and ξ can be computed on run-time from Equation 3 and 4. The proposed controller will adjust the sleep time by first computing values for Ω and then choosing appropriate stable values for ε and ξ in real-time. By doing so $(\varepsilon + \xi)$ will be in a stable range as expressed by Equation 3.

IV. Evaluation and Implementation

A. Simulation set-up

We evaluate performance of HAMA in simulation using the Contiki Cooja platform. The transmission range of each node was set to 750m and the transmission frequency was set at 2.4GHz. The default transmission power was set at maximum available radio output power level in Cooja, which is 31 (0dBm or 1mW). The Unit Disk Graph Medium (UDGM) [16] distance loss was used to run the simulation. We simulate 40 nodes moving in a defined trajectory in a grid area of 5000mx5000m.

To reflect real mobility scenarios as much as possible, we consider two mobility models. We first generate a mobility

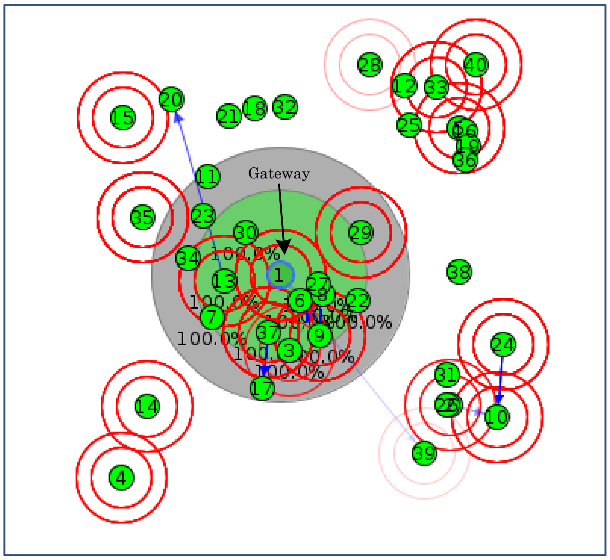


Fig. 2: Simulation set-up. The blue arrows illustrate data packet exchange, the red circles identify the transmission by sender nodes. In this case node-1 is the gateway node, however, the network could be set-up to have more than one gateway nodes.

model based on GPS data collected from collared Zebras in a game park. GPS data of these Zebras was collected every 10min during a foraging activity. To ensure that our mobility model is correct, we show through simulation that it generates an output which with 2hr resolution, produces roughly the same movement patterns (in terms of distance and turning angle distributions) as real Zebras dataset. In this mobility scenario, animals also consider conspecifics in their choices, thus there exits herding or clustering behaviour to some degree.

We generated a second mobility model called RWP using BonnMotion tool [17], in which mobile nodes move with Random-Way Point [18] mobility trajectory and are set to move at random speed range of [10,30 Km/h] with max-pause = 5s.

The summarized simulation parameters are shown in the Table I.

The network was operating in a tree network topology to simulate a more practical data collection scenario. In this topology, a number of sender nodes generate packets to be relayed by intermediate nodes towards the gateway. To form a tree topology, we deployed the collection tree protocol (CTP) [19] to collect data packets towards the gateway(s). CTP maintains a tree-based routing network topology using expected transmissions (ETX) to be defined as a link cost metric.

The nodes randomly generated data message with 20 bytes payload at inter-packet interval IPI(ms) or ($\lambda^{-1} = 2000ms$). We recorded the packet generation time as well as the time

Simulation	5000x5000m^2
Simulation Duration	200 packets
MAC Protocol	HAMA, X-MAC and A-MAC
Routing Protocol	Collection Tree Protocol (CTP)
frequency	868MHz
Max transmission power level	31(0dBm)
Transmission Range	250m
Number of nodes	40
Mobile Node Speed Range	[10,30]km/h
Mobility Model	Random Way Point (RWP) and Zebra GPS Data Set
RWP Pause Time	5 Seconds
Max Buffer Size (K)	10
Regenerative Cycle (N)	10
Inter-Packet Interval (IPI)	$\lambda^{-1} = 2000ms$

TABLE I: Simulation Parameters

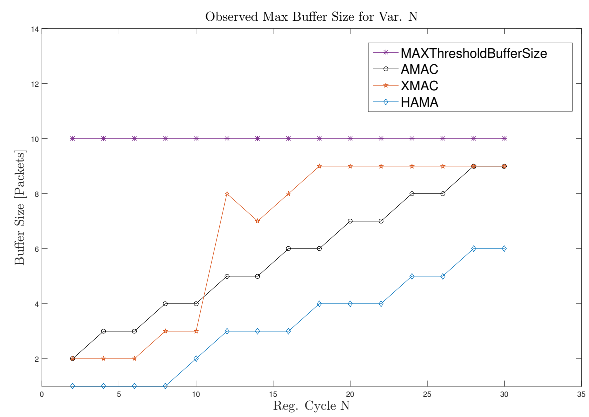


Fig. 3: Modeled queue buffer size with respect to various generation cycle values. Each point is the max buffer size value observed for various regenerative cycle (N) at $\lambda^{-1} = 200ms$.

when they are received at the gateway(s). We measure them for 200 packets simulation time.

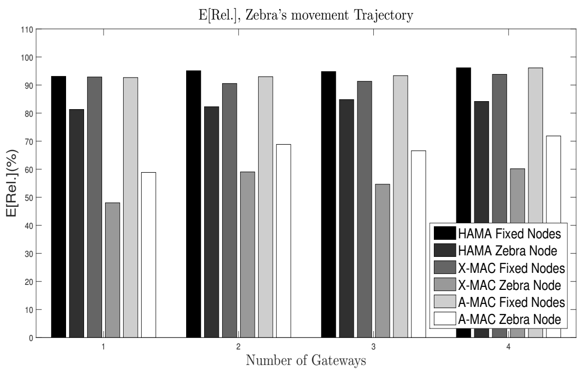
The appropriate value for regeneration cycles N and max buffer size K (to be used for Equation 2 and 3) are determined empirically. Figure 3 shows the maximum observed queue buffer length for different regeneration cycles ($N = 1, 2, 3, ...30$) run at IPI ($\lambda^{-1} = 200ms$). A relatively faster IPI is selected to reflect the worst case scenario, in which nodes will be generating packets in case of burst communication. As shown in Figure 3, the maximum buffer size is 9 packets in case of X-MAC and A-MAC and only 6 packets in case of HAMA (for significantly high traffic).

To evaluate response time and computational overhead, without lose of practicality, for the following simulation, we set $N = 10$ and $K = 10$ (see Figure 3).

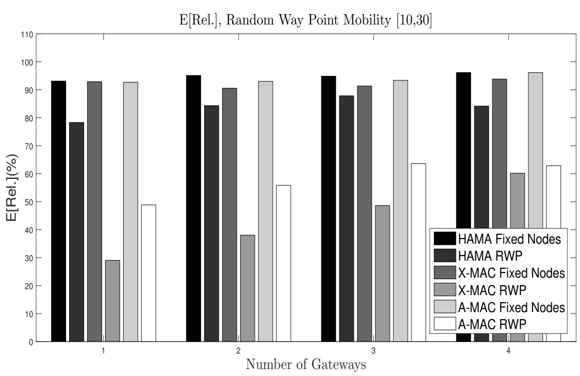
We evaluated performance of HAMA considering these two mobility scenarios as well as the case when all nodes are stationary. In addition we compared the performance HAMA with X-MAC and A-MAC protocols. We used three metrics to evaluate HAMA's performance: (i) Reliability, (ii) Average power consumption, and (iii) End-to-End Latency. In the following sections, we discuss our evaluation results.

B. Reliability

Reliability is measured by counting number of transmitted packets and number of successfully received packets by the gateway. As shown in Figure 4, compared with mobile networks, reliability is higher for the network with stationary nodes. In case of having mobility, it can be seen that as expected when the node speed increases, the reliability decreases. This effect is illustrated in Figure 4a and Figure 4b. However, for both mobility scenarios, HAMA performs better compared to X-MAC and A-MAC. One may notice that as amount of mobility increases, performance of X-MAC and A-MAC degrades significantly while HAMA remains robust against mobility. This is particularly true in case of mobile nodes with BonnMobility Random Way point [10,30]Km/h.



(a) Averg. Reliability for the zebra's mobility and fixed topology, in this case animals depict a herding or clustering scenario.



(b) Average reliability of BonnMobility Random Waypoint [10,30]Km/h and fixed topology.

Fig. 4: The reliability percentage (divided into Zebras and RWP mobility scenarios) compared to those of fixed network topology.

Overall, packet reliability of HAMA is at least 12.3% (for mobile nodes with Zebra's Trajectory) and at most 16.3% (for for mobile nodes with BonnMobility Random Way point [10,30]Km/h) higher than of X-MAC and A-MAC. Furthermore, one can see that increasing number of gateways leads to decreasing the difference between the reliability performance of all protocols. This is because an increase in the gateway density counters the effect of mobility and consequently increases the reliability.

C. Average energy consumption

All MAC protocols consume a comparable amount of energy at smaller IPI (higher packet rate), which is plausible since nodes are active most of the time in high traffic scenario.

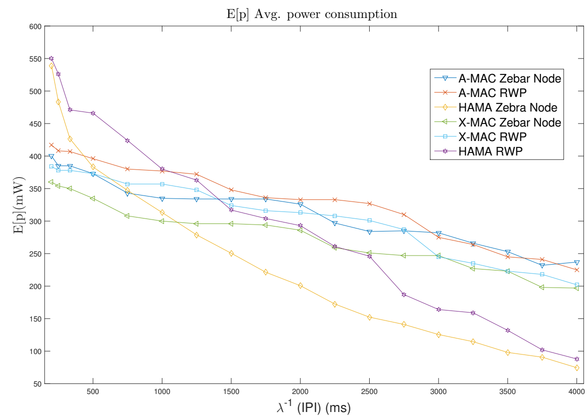


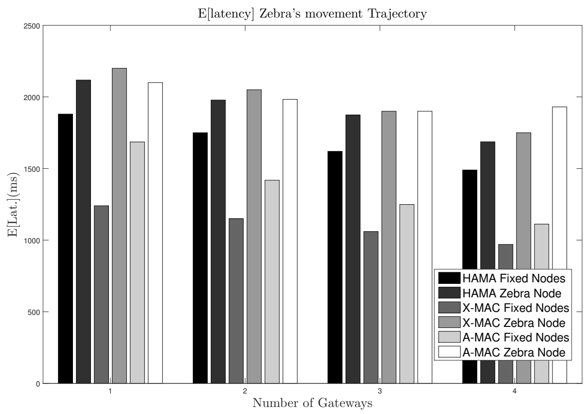
Fig. 5: Network wide average power consumption of with respect to different inter-packet intervals (IPI). Measured for Zebra and RWP mobility scenarios.

Figure 5 shows the average energy consumption of the entire network at various IPIs ($200 \leq \lambda^{-1} \leq 4000\ ms$).

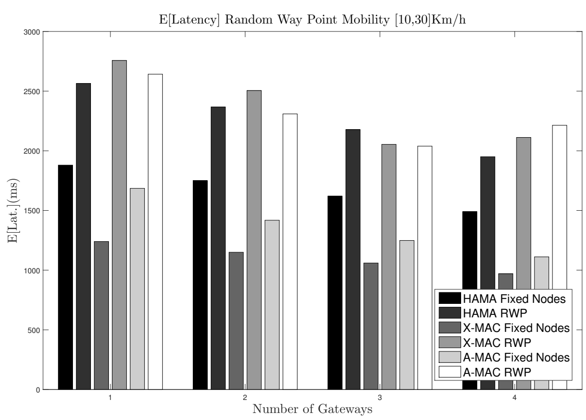
As can be seen in Figure 5, for both Zebra and RWP mobility scenarios, the energy consumption of the HAMA protocol increases as network packet arrival rate increases. For both X-MAC and A-MAC, energy consumption remains relatively the same as data packet arrival rate increases. However, A-MAC consumes comparably less energy than X-MAC. HAMA has a relatively high energy consumption compared to X-MAC or A-MAC for IPI ranges of $200 \leq \lambda^{-1} \leq 500\ ms$. This is due to implementation of its adaptive mechanism. As packet transmission and reception rates increase, duration of each epoch becomes smaller. Consequently the control period becomes shorter and the HAMA frequently sets smaller sleep-time to cope with increasing packet rate. However, as the packet reception rate decreases, HAMA sets longer sleep-times, which will significantly reduce the energy consumption. Overall, compared with fixed networks and X-MAC and A-MAC, energy consumption of HAMA increases by 22.28% for high packet traffic ($\lambda^{-1} \leq 1500\ ms$) and decreases by at least 52.28% for low packet traffic ($\lambda^{-1} \geq 1500\ ms$). This implies energy efficiency of HAMA for high mobility WSN applications.

D. Average end-to-end latency

Average end-to-end latency is calculated for all received packets based on the average time difference between when a packet was transmitted and when it was at the gateway. It is apparent from Figure 6 that in general higher number of gateways in the network will normally contribute to having a lower latency. The maximum latency observed is 2200ms and 2750ms for X-MAC in case of Zebras mobility and BonnMobility Random Way point [10,30]Km/h mobility having one gateway node. As expected the average latency is higher when node's mobility speed increases (as it can be seen from results of Figure 6.b). Overall, the average latency is decreased by at-least 11.65 - 14.63% when HAMA is used.



(a) Zebras mobility



(b) BonnMobility Random Way point [10,30]Km/h and fixed topology

Fig. 6: Comparison of average end-to-end latency: Top: Zebras mobility scenario. Bottom: RWP mobility scenarios

V. Conclusion

As commonly implemented, setting a fixed duty-cycle duration for mobile network topology often leads to low reliability, high energy consumption and latency. This is especially the case for high mobility networks. We presented HAMA to cope with highly frequently changing network topologies. Improving the B-MAC protocol, HAMA's feed-back stability controller analyzes the received and transmitted packets to determine an optimal sleep-time interval at run-time. To evaluate HAMA under as realistic as possible mobility and dynamic network topology scenarios, we used two mobility scenarios based on Zebras forging activity and BonnMobility Random Way Point. Our evaluation results show that using HAMA protocol the average network energy consumption is reduced by 22.28%-52.28%, providing an additional decrease in the average end-to-end latency by 11.65%-14.63% compared to A-MAC and X-MAC. The overall packet reliability is also increased by upto 16.3%.

Acknowledgment

Authors would like to thank Henjo de Knegt for providing the Zebra GPS dataset used for mobility modeling. This research is supported by the Smart Parks Project, funded by the Netherlands Organisation for Scientific Research (NWO).

References

[1] SA Hameed, EM Shaaban, HM Faheem, and MS Ghoniemy. Mobility-aware mac protocol for delay-sensitive wireless sensor networks. In *Ultra Modern Telecommunications & Workshops, 2009. ICUMT'09. International Conference on*, pages 1–8. IEEE, 2009.

[2] Qian Dong and Waltenegus Dargie. A survey on mobility and mobility-aware mac protocols in wireless sensor networks. *Communications surveys & tutorials, IEEE*, 15(1):88–100, 2013.

[3] Joseph Polastre, Jason Hill, and David Culler. Versatile low power media access for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 95–107. ACM, 2004.

[4] Michael Buettner, Gary V Yee, Eric Anderson, and Richard 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.

[5] Prabal Dutta, Stephen Dawson-Haggerty, Yin Chen, Chieh-Jan Mike Liang, and Andreas Terzis. Design and evaluation of a versatile and efficient receiver-initiated link layer for low-power wireless. In *Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems*, pages 1–14. ACM, 2010.

[6] Gertjan P Halkes, Tijs van Dam, and KG Langendoen. Comparing energy-saving mac protocols for wireless sensor networks. *Mobile Networks and Applications*, 10(5):783–791, 2005.

[7] Jiliang Wang, Zhichao Cao, Xufei Mao, and Yunhao Liu. Sleep in the dins: Insomnia therapy for duty-cycled sensor networks. In *INFOCOM, 2014 Proceedings IEEE*, pages 1186–1194. IEEE, 2014.

[8] Marco Zimmerling, Federico Ferrari, Luca Mottola, Thiemo Voigt, and Lothar Thiele. ptunes: runtime parameter adaptation for low-power mac protocols. In *Proceedings of the 11th international conference on Information Processing in Sensor Networks*, pages 173–184. ACM, 2012.

[9] Majid Nabi, Milos Blagojevic, Marc Geilen, Twan Basten, and Teun Hendriks. Mcmac: An optimized medium access control protocol for mobile clusters in wireless sensor networks. In *Sensor Mesh and Ad Hoc Communications and Networks (SECON), 2010 7th Annual IEEE Communications Society Conference on*, pages 1–9. IEEE, 2010.

[10] Lodewijk Van Hoesel and Paul Havinga. Collision-free time slot reuse in multi-hop wireless sensor networks. In *Intelligent Sensors, Sensor Networks and Information Processing Conference, 2005. Proceedings of the 2005 International Conference on*, pages 101–107. IEEE, 2005.

[11] Huan Pham and Sanjay Jha. An adaptive mobility-aware mac protocol for sensor networks (ms-mac). In *Mobile Ad-hoc and Sensor Systems, 2004 IEEE International Conference on*, pages 558–560. IEEE, 2004.

[12] Zhenjiang Li, Mo Li, and Yunhao Liu. Towards energy-fairness in asynchronous duty-cycling sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 10(3):38, 2014.

[13] Eyuel D Ayele, Jianjun Wen, Zeeshan Ansar, and Waltenegus Dargie. Adaptive sleep-time management model for wsns. In *Computer Communication and Networks (ICCCN), 2015 24th International Conference on*, pages 1–7. IEEE, 2015.

[14] Heejung Byun and Jungmin So. Queue-based adaptive duty cycle control for wireless sensor networks. In *Algorithms and Architectures for Parallel Processing*, pages 205–214. Springer, 2011.

[15] Jeremiah F Hayes and Thimma VJ Ganesh Babu. *Modeling and analysis of telecommunications networks*. John Wiley & Sons, 2004.

[16] Hugo V Bitencourt, Adriano B da Cunha, and Diogenes C da Silva. Simulation domains for networked embedded systems. In *Student Forum on Microelectronics, SForum*, 2012.

[17] Nils Aschenbruck, Raphael Ernst, Elmar Gerhards-Padilla, and Matthias Schwamborn. Bonnmotion: a mobility scenario generation and analysis tool. In *Proceedings of the 3rd International ICST Conference on Simulation Tools and Techniques*, page 51. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2010.

[18] Tracy Camp, Jeff Boleng, and Vanessa Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[19] Omprakash Gnawali, Rodrigo Fonseca, Kyle Jamieson, David Moss, and Philip Levis. Collection tree protocol. In *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, SenSys '09, pages 1–14.