CORA Cork Open Research Archive
Cartlann Taighde Oscailte Chorcaí

Title	ER-MAC: A hybrid MAC protocol for emergency response wireless sensor networks
Authors	Sitanayah, Lanny;Sreenan, Cormac J.;Brown, Kenneth N.
Publication date	2010-07
Original Citation	Sitanayah, L., Sreenan, C. J. and Brown, K. N. (2010) 'ER-MAC: A Hybrid MAC Protocol for Emergency Response Wireless Sensor Networks', Fourth International Conference on Sensor Technologies and Applications (SensorComm), Venice, Italy, 18-25 July, pp. 244-249. doi: 10.1109/SENSORCOMM.2010.45
Type of publication	Conference item
Link to publisher's version	10.1109/SENSORCOMM.2010.45
Rights	© 2010 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.
Download date	2026-11-05 19:51:24
Item downloaded from	https://hdl.handle.net/10468/8960



University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

ER-MAC: A Hybrid MAC Protocol for Emergency Response Wireless Sensor Networks

Lanny Sitanayah
Mobile & Internet Systems Laboratory
Department of Computer Science
University College Cork, Ireland
Email: ls3@cs.ucc.ie

Cormac J. Sreenan
Mobile & Internet Systems Laboratory
Department of Computer Science
University College Cork, Ireland
Email: cjs@cs.ucc.ie

Kenneth N. Brown
Cork Constraint Computation Centre
Department of Computer Science
University College Cork, Ireland
Email: k.brown@cs.ucc.ie

***Abstract*—This paper introduces ER-MAC, a hybrid MAC protocol for emergency response wireless sensor networks. ER-MAC is designed as a hybrid of the TDMA and CSMA approaches, giving it the flexibility to adapt to traffic and topology changes. It adopts a TDMA approach to schedule collision-free slots. Nodes wake up for their scheduled slots, but otherwise switch into power-saving sleep mode. When an emergency occurs, nodes that participate in the emergency monitoring change their MAC behaviour by allowing contention in TDMA slots to achieve high delivery ratio and low latency. ER-MAC offers a synchronised and loose slot structure to allow nodes to join or leave the network. Simulations in ns-2 show that ER-MAC outperforms Z-MAC with higher delivery ratio, lower latency, and lower energy consumption.**

***Keywords*-MAC protocol; wireless sensor networks; fire emergency.**

I. Introduction

Wireless sensor networks (WSNs) for emergency applications such as monitoring fires in buildings must be traffic and topology adaptive. In our specified application, the communication protocol can be delay tolerant during normal monitoring and designed for energy efficiency. However, when an emergency event occurs, energy efficiency is less important than high packet delivery ratio and low latency, and the communication protocol should adapt in response.

Some traffic adaptive medium access control (MAC) protocols have been designed. S-MAC [1], T-MAC [2], B-MAC [3] and X-MAC [4] are contention-based protocols that adapt to both traffic and topology changes, but suffer from collisions, idle listening and overhearing. Hybrid MAC protocols such as Z-MAC [5], PMAC [6], Crankshaft [7] and EB-MAC [8] are not designed for emergency monitoring, and so none of them are both traffic and topology adaptive.

In this paper, we propose ER-MAC, a hybrid MAC protocol for emergency response WSNs. While our scenario assumption is the fire monitoring in buildings, this protocol is also useful in a range of WSN emergency applications. The contributions of this paper are:

- ER-MAC allows contention in TDMA slots to cope with large volumes of traffic. This scheme trades energy efficiency for higher delivery ratio and lower latency.
- ER-MAC maintains two priority queues to separate high priority packets from low priority packets.
- ER-MAC offers a synchronised and loose slot structure, where nodes can modify their schedules locally. This allows nodes to join or leave the network easily.
- Simulation results validate ER-MAC's performance, which outperforms Z-MAC [5] with higher delivery ratio and lower latency at low power consumption.

The remainder of this paper is organised as follows. We formulate the problem definition in Section II. In Section III, we review the related work on traffic adaptive MAC protocols. We present the proposed ER-MAC protocol in Section IV. We show our simulation results in Section V. Section VI concludes the paper and presents our future work.

II. Problem Definition

In this section, we describe some assumptions for the network and identify the requirements for our MAC protocol.

A. Assumptions

We assume a pre-deployed WSN that has a connected finite set of sensor nodes and one or more base stations, which are static. We also assume that there are two types of packets: high and low priority. For example, data from video sensors can be tagged as high priority. In addition, we do not assume symmetric communication. We have two different situations: no-fire and in-fire. When a node or a group of nodes senses fire, we assume autonomous control over the MAC protocol to change the MAC behaviour.

B. Requirements for MAC

When designing the MAC protocol for emergency response, there are several factors to be taken into account:

- Traffic load is light during normal monitoring, but increases significantly when an emergency occurs.
- Energy efficiency is important, but can be sacrificed for low latency and high delivery ratio in emergency.
- Normal monitoring is delay tolerant, but emergency monitoring is not.
- The MAC protocol has to achieve high delivery ratio in both normal and emergency situations.

III. Related Work

S-MAC [1] introduces a fixed duty cycle, which periodically puts nodes into sleep to reduce idle listening, but increases latency under heavy traffic. Timeout-MAC (T-MAC) [2] tries to improve on S-MAC by using an adaptive duty cycle, which dynamically adjusts nodes' sleep and active cycles. Unfortunately, it is common to all contention-based protocols, including T-MAC, that the chance of collision increases rapidly during high traffic loads. B-MAC [3] uses adaptive preamble sampling to reduce the duty cycle and minimise idle listening. But, idle listening still occurs when the node wakes up but there is no activity in the channel. X-MAC [4] tries to solve the problems of B-MAC by introducing a series of short preambles with target addresses to avoid overhearing and reduce the energy expenditure on non-target receivers. X-MAC utilises random back off, but this does not solve the hidden terminal problem.

Z-MAC [5] dynamically switches between carrier sense multiple access (CSMA) and TDMA depending on the traffic. Nodes execute a distributed slot selection algorithm to get a collision-free slot. A slot begins with a small contention period. If a node has data to send and it is the owner of the slot, it back offs within T_O period, else it back offs between T_O and T_{NO}. Under *low contention level* (LCL), nodes in the network can compete in any time slots, but under *high contention level* (HCL), only the owner and one-hop neighbours of the owner can compete for the slot to reduce collision. In high contention networks, Z-MAC uses explicit congestion notification (ECN) messages to reduce hidden terminals. When a node detects heavy traffic, it propagates the ECN message to its two-hop neighbourhood. Z-MAC builds a TDMA structure on top of B-MAC's back off mechanism, clear channel assessment (CCA) and low power listening (LPL). Hence, it inherits B-MAC's limitation.

PMAC [6] is a hybrid protocol that adaptively adjusts the sleep-wakeup schedules based on local traffic. But, using traffic patterns makes PMAC prone to error, because some nodes may receive incorrect patterns due to interference signals. Crankshaft [7] schedules receive slots and allocates one unicast slot in every frame for a node. Crankshaft is suitable for long-lived monitoring applications, where throughput can be traded for energy efficiency. EB-MAC [8] is tailored for event based systems that can handle high and low traffic conditions. The schedule of EB-MAC is calculated according to the received signal strength (RSS) of the detected event. The higher the RSS reading, the earlier the slot is given to a node. EB-MAC uses B-MAC's CCA and LPL techniques, so it inherits B-MAC's limitation.

Since none of the existing MAC protocols above are designed for emergency response, none of them address all of our MAC protocol requirements. Hence, we design ER-MAC as a hybrid of the TDMA and CSMA approaches, giving it flexibility to adapt to traffic and topology changes.

IV. ER-MAC Protocol Design

This part of work was first presented in [9]. ER-MAC initially communicates using CSMA/CA with a random-access mechanism. During the startup phase, the data gathering tree and TDMA schedules are created.

A. Topology Discovery

The base station initiates the tree construction using a simple flooding mechanism. Our process is similar to the hop tree configuration of the PEQ routing protocol [10]. In our context, the goal of the topology discovery is not only to setup a routing tree, but also to find neighbours and to track changes in the tree. The base station generates a *TOPOLOGY_DISCOVERY* message, which consists of *hop_count*, *new_parent_id* and *old_parent_id*. This message is broadcast by a node to find its prospective children, as well as a reply to its parent and a notification to its previous parent when it wants to change parent. In this phase, each node records its hop count to the base station, its parent ID, a list of its children and its one-hop neighbour list.

B. TDMA Slot Assignment

In this phase, nodes perform slot assignment and exchange schedules, so no two nodes within a two-hop neighbourhood use the same slot. Our TDMA slot assignment follows a bottom-up approach, where a leaf node (a node with no children) starts the slot assignment. Our purpose of starting the slot assignment from the leaf nodes is to have a transmission schedule that can support message flow towards the base station. A non-leaf node (except the base station) waits until all of its children report their schedule before assigning one unicast slot to send its own data, several unicast slots to forward its descendants' data and a broadcast slot to synchronise its children. The slot assignment phase ends when the base station receives *SCHEDULE_NOTIFICATION* messages from all of its children. The base station switches to TDMA by sending the first *SYNCHRONISATION* message. When a child receives the message, it switches to TDMA and synchronises its children using its broadcast slot.

C. Local Time Synchronisation

ER-MAC manages local time synchronisation using parent-children broadcast synchronisation similar to the root-neighbours synchronisation of FTSP [11]. This simple mechanism is sufficient for our approach because each child only needs to have the same clock as its parent to ensure that the parent is in receive mode when the child transmits data to it and vice versa. A *SYNCHRONISATION* message consists of *sender_ID*, *current_slot* to help synchronising new nodes, *highest_slot* to report the TDMA frame length, *clock* and *hop_count* to help a new node to select its prospective parent.

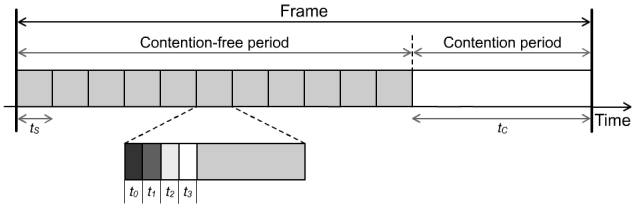


Figure 1. ER-MAC's frame structure

D. Priority Queue

ER-MAC has a pair of queues to separate high priority from low priority packets. In a queue, a packet is ordered based on its *slack*, i.e., the time remaining until the packet deadline expires and is part of the packet header. The slack is updated at each hop by subtracting the queuing and transmission delays from it. The basic rule is the high priority packets are transmitted first until the high priority queue is empty. If a queue is full, we drop a packet with the shortest slack because it is most likely to miss its deadline. We modify the queue by considering fairness over the packets' sources, so the base station can have a balance of information from all nodes. When the reporting frequency increases, a node may have lots of its own data. If the node always takes a packet from the head of the queue, it may happen that it sends its own data more than its descendants'.

E. MAC Prioritisation

The ER-MAC frame consists of contention-free slots with duration t_s each and a contention period with duration t_c as depicted in Fig. 1. In each contention-free slot, except for the synchronisation slot, there are sub slots t_0, t_1, t_2 and t_3, which only appear in emergency mode for contention. Note that in the emergency mode, the period of t_s - (t_0 + t_1 + t_2 + t_3) is sufficient to carry a packet and a sub slot is big enough to carry a MAC header (a source, a destination and a flag). In the normal mode, a transmitter occupies a slot from the beginning of the slot and sleeps after transmitting a packet. We include a contention period at the end of a frame to support new node addition.

In normal monitoring, communication follows the nodes' schedules. To further conserve energy, a sender turns off its radio if it has no data to send and a timeout forces a receiver back to sleep if it does not receive any packets. When fire is detected by some nodes' sensors, only nodes affected by the fire change their MAC to emergency mode. These nodes are nodes caught in fire that send *FIRE* messages, their one-hop neighbours that receive the messages, their ancestors toward the base station that receive data packets with an emergency flag and the ancestors' one-hop neighbours, while other nodes remain in the normal mode. A node changes the MAC with the following rules:

1) An owner of a slot wakes up in the beginning of its transmit slot. If it has a high priority packet to send, it transmits the packet immediately, but otherwise it allows its one-hop neighbours to contend for the slot.
2) All non-owners of the slot wake up in the beginning of every slot for possible contention or reception of packets. If a non-owner with a high priority packet senses no activities in the channel during t_0, it contends for the slot during t_1 by sending a *SLOT_REQUEST* message to the slot owner. The owner replies with a *SLOT_ACKNOWLEDGEMENT*.
3) The owner of the slot with low priority packets can only use its own slot if during t_0+t_1 it does not receive any *SLOT_REQUEST* from its neighbours.
4) A non-owner with low priority packets can contend for the slot if during $t_0 + t_1 + t_2$ it senses no activities. It contends for the slot during t_3 by sending a *SLOT_REQUEST* to the slot owner. The owner replies by sending a *SLOT_ACKNOWLEDGEMENT*.

To prevent a node sending an emergency packet to a sleeping parent, the first emergency packet is sent in a scheduled slot. This allows the ancestors of the node to switch their MAC when they receive the packet. The latency of the first emergency packet is the same as in normal situation. If a false alarm happens, the node that thinks it detects the fire will broadcast a *FALSE_ALARM* message to allow its one-hop neighbours to change their MAC back to the normal mode. The ancestors of the node on the route to the base station that are in the emergency mode will change their MAC back to the normal mode if they do not receive any emergency packets after n gathering cycles.

F. New Nodes and Dead Nodes

A new node has to listen to its neighbours' *SYNCHRONISATION* and data messages for at least one gathering cycle to select a parent with the lowest hop count and to synchronise its clock. The slot assignment for a new node is similar to the slot assignment during the initial setup phase as described in Section IV-B. The parent allocates one slot to forward the new node's data. If the parent had no children before, it has to assign another slot to synchronise its new child. The parent then performs schedule exchange in the next contention slot. The process of allocating new slots is carried out along the new node's route toward the base station. It takes approximately $(k+1) \times t$ seconds until the slot assignment reaches the base station after the new node is deployed, where k is the new node's hop count and t is one gathering cycle period.

The addition of new slots lengthens the TDMA frame and all nodes have to apply these changes simultaneously. For this purpose, a count down timer, set to be $k_{max} \times t$ seconds, is piggybacked in *SYNCHRONISATION* messages and propagated to the whole network. k_{max} is the highest hop count of the network. As the timer expires, all nodes simultaneously use the new schedules. The process of disseminating the new frame length proceeds until all nodes

Table I
ER-MAC SIMULATION PARAMETERS IN NS-2.

Simulation parameters	Default value
Transmission range	10 m
Transmit power	52.2 mW
Receive/idle listening/transition power	59.1 mW
Sleep power	0.003 mW
Transition time	580 μs
ER-MAC TDMA slot size	50 ms
ER-MAC TDMA sub-slot size	5 ms

change their TDMA frame length and takes at most k_{max} gathering cycle periods. Hence, the total time needed for a new node to operate in TDMA mode after it is deployed is $(k_{max}+k+1)\times t$ seconds. Frame length inconsistencies will not happen, since the synchronisation slots are collision-free.

A node is dead if it runs out of battery or is destroyed by fire. If a parent does not receive any data during all scheduled receive slots of a child after n gathering cycles, it assumes that the child is dead. It removes the child from its children list. It also removes m scheduled receive slots and m transmit slots that are associated with that child. If the child is the only child of that parent, it also removes the synchronisation slot. The parent informs its ancestors to do the same process by piggybacking the information on the data packet sent in the immediate data gathering slot. All of the unused slots are then informed within the two-hop neighbourhood during the contention slot period. When a node does not receive *SYNCHRONISATION* messages after n gathering cycles, it may assume that its parent is dead. The orphan node finds a new parent by following the same procedure as the new node deployment. The orphan node will report its transmit slots' schedule to its new parent, so the descendants of the orphan do not need to rebuild their schedules. The parent then assigns new transmit slots to forward its new descendants' data.

V. SIMULATION AND RESULTS

We implemented ER-MAC in ns-2 [12]. Our simulation results are based on the mean value of five different network deployments that are simulated five times each using random seeds, enough to achieve a 95% confidence interval. The network consists of 100 nodes deployed within randomly perturbed grids, where a node is placed in unit grid (8 m x 8 m) and the coordinates are slightly perturbed. The location of the base station was fixed at the top-left of the network. We use a simple wireless channel using the two-ray ground radio propagation model. We also randomly select up to n links and for each drop up to m packets, where m is large enough to model unreliable links. Our simulation parameters presented in Table I were based on Tmote sky hardware [13].

A. Protocol Comparison

We compared the performance of ER-MAC with Z-MAC, because this protocol has several similar characteristics with ours, such as hybrid designs and allowing contention in TDMA slots when the traffic load increases. We followed the Z-MAC ns-2 installation manual [14] and configured Z-MAC according to the settings shown in Table 1 in [5], except that we use a 10 m transmission range. In each experiment, we simulated a data gathering for 300 seconds, where every node except the base station generates packets with fixed intervals. We considered no-fire and in-fire situations for ER-MAC, and forced Z-MAC to operate in either LCL or HCL. For the in-fire situation, we assume all nodes are in fire from the beginning of the simulation.

Fig. 2 shows that ER-MAC consumes less energy than Z-MAC. This is because in ER-MAC, the owner of the slot does not need to contend to access the channel. However, in Z-MAC, the owner of the slot has to contend before sending data. The figure also shows that during the in-fire situation, ER-MAC spends more energy than the no-fire situation, because nodes wake up in every slot for possible contention. The energy consumption of ER-MAC during the in-fire situation is high when the traffic load is low (less than 0.1 packets/node/sec) because more nodes contend for their one-hop neighbours' slots. In our simulation, the network's peak load is around 0.2 packets/node/sec. Hence, the energy consumption above the peak load is stable as nodes always have data to send, so the possibility of contention is minimal.

To compare the delivery ratio of high and low priority packets, we force source nodes to generate the two kinds of packet at the same time. Fig. 3 shows that ER-MAC's high priority packets always achieve better delivery ratio. Even though the delivery ratios of ER-MAC's high priority packets decrease gracefully above the peak load, its delivery ratio in the in-fire situation is slightly higher than the no-fire situation. This phenomenon is caused by contention to prioritise the high priority packets during the emergency.

We expect the latency to rise as the traffic load increases. However, when the traffic load goes up, the base station receives fewer packets and most of them are from nodes near it. This low delivery ratio corresponds to the low average latency that will be explained below. Fig. 4 shows that ER-MAC's high priority packets generally have lower latency compared to Z-MAC's. The figure also shows that below the peak load, the latency of ER-MAC's high priority packets during the no-fire situation is predictable because each of a node's routers has already been preassigned a slot to forward the node's data. However, for the in-fire situation, the delay is reduced because nodes can propagate data quickly. Since ER-MAC prioritises high priority packets, the latency of low priority packets is high. When the traffic load increases, the latency for Z-MAC's packets and ER-MAC's low priority packets drop because fewer packets are received at the base station and most of them are from nodes near it. This argument is validated by the low delivery ratio in Fig. 3.

As explained in Section IV-D, we implement priority queues by considering fairness over the packets' sources.

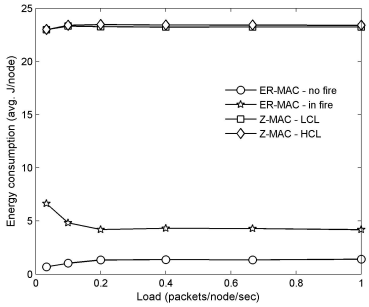


Figure 2. Energy consumption

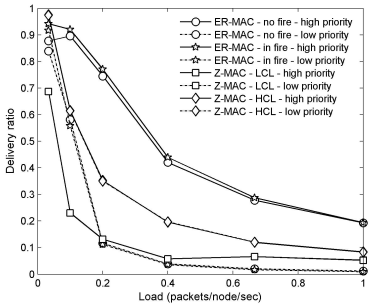


Figure 3. Delivery ratio

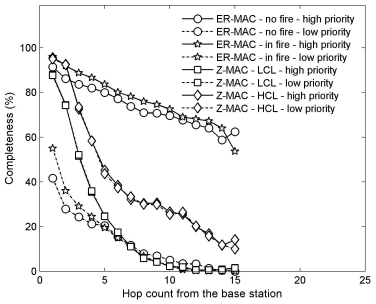


Figure 5. Completeness

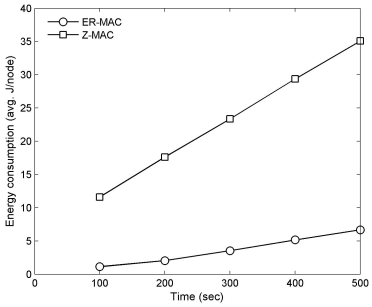


Figure 6. Energy consumption under variable traffic load

Fig. 5 shows the completeness of the packets received at the base station when the network reaches its peak load, i.e., 0.2 packets/node/sec. We measure the completeness as the percentage of the average delivery ratio per hop basis. The graph shows that the completeness of ER-MAC's high priority packets for no-fire and in-fire situations are higher than Z-MAC's packets and ER-MAC's low priority packets.

B. Behaviour Under Variable Traffic Load

In this simulation, we vary the traffic load during 500-second simulations. The traffic changes every 100 seconds. It jumps from 0.1 to 0.4 packets/node/sec, then drops to 0.1 packets/node/sec, and so forth. When a node generates more traffic, the MAC changes its behaviour. Fig. 6, 7 and 8 show the comparison of ER-MAC against Z-MAC when the traffic changes over time in terms of average energy consumption per node, packet delivery ratio and average per packet latency, respectively. In Fig. 7 and 8, the delivery ratio and latency of Z-MAC's high and low priority packets overlap because Z-MAC only uses one queue and sends the high and low priority packets one after another. That is why the results are almost the same.

C. Behaviour When Topology Changes

We want to show that ER-MAC is topology adaptive. In this simulation, we increase the number of dead nodes from five to 20 and calculate the average energy consumption and time needed to reconfigure the network. The energy

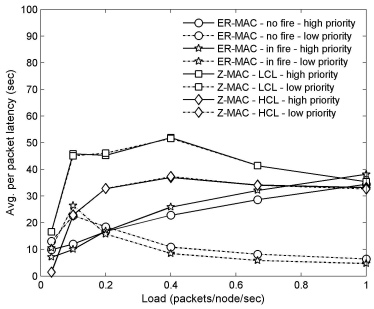


Figure 4. Latency

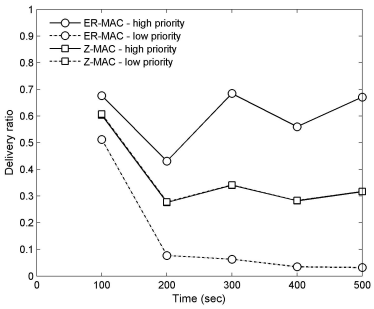


Figure 7. Delivery ratio under variable traffic load

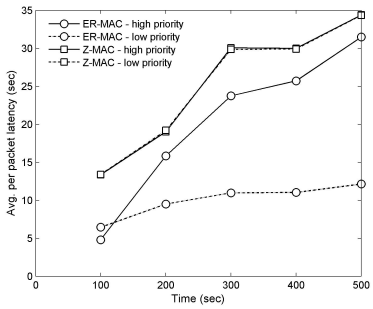


Figure 8. Latency under variable traffic load

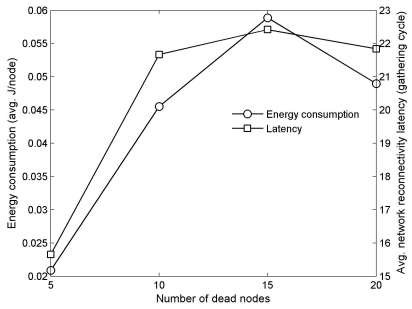


Figure 9. Energy consumption and latency for network reconnectivity

consumption to reconfigure the network is the amount of energy spent by orphan nodes to find their new parents and announce new schedules in contention slots. The network reconnectivity latency is calculated from the time a node knows that its parent is dead until it uses its new TDMA schedules. These simulation results are depicted in Fig. 9. The energy consumption and latency to reconfigure the network decrease when the number of dead nodes goes over 15 because the network gets partitioned as the number of failed node increases. Hence, we only measure the energy expenditure and time to reconfigure the network from the remaining nodes that still form a connected network to the base station.

VI. Conclusion and Future Work

We present ER-MAC, a hybrid MAC for emergency WSNs with flexibility to adapt to traffic and topology changes. Our ns-2 simulation results demonstrate the scalability of ER-MAC and show that it achieves higher delivery ratio, lower latency, and lower energy consumption compared to Z-MAC. Our current work includes ER-MAC implementation in Contiki running on our Tmote sky testbed.

Acknowledgment

This research is fully funded by the NEMBES project, which is supported by the Irish Higher Education Authority PRTLI-IV research program.

References

[1] W. Ye, J. Heidemann, and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. 21st Ann. Joint Conf. IEEE Computer and Communications Societies (INFOCOM'02)*, vol. 3, 2002, pp. 1567–1576.

[2] T. van Dam and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. 1st Int'l Conf. Embedded Networked Sensor Systems (SenSys'03)*, 2003, pp. 171–180.

[3] J. Polastre, J. Hill, and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," in *Proc. 2nd Int'l Conf. Embedded Networked Sensor Systems (SenSys'04)*, 2004, pp. 95–107.

[4] M. Buettner, G. Yee, E. Anderson, and R. Han, "X-MAC: A Short Preamble MAC Protocol for Duty-Cycled Wireless Sensor Networks," in *Proc. 4th Int'l Conf. Embedded Networked Sensor Systems (SenSys'06)*, 2006, pp. 307–320.

[5] I. Rhee, A. Warrier, M. Aia, and J. Min, "Z-MAC: A Hybrid MAC for Wireless Sensor Networks," in *Proc. 3rd Int'l Conf. Embedded Networked Sensor Systems (SenSys'05)*, 2005, pp. 90–101.

[6] T. Zheng, S. Radhakrishnan, and V. Sarangan, "PMAC: An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks," in *Proc. 19th IEEE Int'l Symp. Parallel and Distributed Processing (IPDPS'05)*, 2005, pp. 65–72.

[7] G. Halkes and K. Langendoen, "Crankshaft: An Energy-Efficient MAC-Protocol for Dense Wireless Sensor Networks," in *Proc. 4th European Conf. Wireless Sensor Networks (EWSN'07)*, 2007, pp. 228–244.

[8] Z. Merhi, M. Elgamel, and M. Bayoumi, "EB-MAC: An Event Based Medium Access Control for Wireless Sensor Networks," in *Proc. 2009 IEEE Int'l Conf. Pervasive Computing and Communications (PerCom'09)*, 2009, pp. 1–6.

[9] L. Sitanayah, C. Sreenan, and K. Brown, "Poster Abstract: Emergency Response MAC Protocol (ER-MAC) for Wireless Sensor Networks," in *Proc. 9th ACM/IEEE Int'l Conf. Information Processing in Sensor Networks (IPSN'10)*, 2010, pp. 364–365.

[10] A. Boukerche, R. Pazzi, and R. Araujo, "Fault-Tolerant Wireless Sensor Network Routing Protocols for the Supervision of Contex-Aware Physical Environments," *Journal of Parallel and Distributed Computing*, vol. 66, no. 4, pp. 586–599, 2006.

[11] M. Maroti, B. Kusy, G. Simon, and A. Ledeczi, "Robust Multi-Hop Time Synchronization in Sensor Networks," in *Proc. Int'l Conf. Wireless Networks (ICWN'04)*, vol. 1, 2004, pp. 454–460.

[12] VINT, "The Network Simulator - ns-2," Apr. 2010. [Online]. Available: http://www.isi.edu/nsnam/ns/

[13] Moteiv, "Tmote Sky Datasheet," Apr. 2010. [Online]. Available: http://www.eecs.harvard.edu/~konrad/projects/shimmer/references/tmote-sky-datasheet.pdf

[14] I. Rhee, "Z-MAC: Hybrid MAC for Wireless Sensor Networks," Apr. 2010. [Online]. Available: http://www4.ncsu.edu/~rhee/export/zmac/software/zmac/zmac.htm