A new Energy-Efficient TDMA-based MAC Protocol for Periodic Sensing Applications in Wireless Sensor Networks

Shams ur Rahman[1], Mushtaq Ahmad[2], Shafaat A. Bazaz[3]

[1,2]Faculty of Computer Science and Engineering, GIK Institute of Engineering Sciences and Technology,

Topi, Pakistan

[3]Department of Computer Science, Centre for Advance Studies in Engineering (CASE)

Islamabad, Pakistan

Abstract

Energy efficiency is a major requirement in wireless sensor networks. Media Access Control is one of the key areas where energy efficiency can be achieved by designing such MAC protocol that is tuned to the requirements of the sensor networks. Different applications have different requirements and a single MAC protocol cannot be optimal for all types of applications. In this paper we present a TDMA-based MAC (TDMAC) protocol which is specially designed for such applications that require periodic sensing of the sensor field. TDMAC organizes nodes into clusters. Nodes send their data to their cluster head (CH) and CHs forward it to the base station. CHs away from the base station use multi-hop communication by forwarding their data to CHs nearer than themselves to the base station. Both inter-cluster and intra-cluster communication is purely TDMA-based which effectively eliminates both inter-cluster as well as intra-cluster interference.

***Keywords*–** *energy efficient MAC, periodic sensing, TDMA-based MAC, clusters, multi-hop*

1. Introduction

Wireless sensor networks consist of small-batteries-powered tiny nodes. Each node senses some environmental parameter, such as temperature, humidity, motion etc., and transmits its readings to some central point, called base-station or sink, using wireless means. The energy resources of these nodes are very limited. In most of the cases, recharging or replacing these batteries is not possible. In order that the network is operational for longer periods of time, it's mandatory that the energy is used optimally. Intense research is being conducted in various areas of the sensor networks – from node hardware to protocol design and nodes deployment – to achieve energy efficiency. Media Access Control (MAC) is one of the key areas where energy efficiency can be enhanced by reducing or eliminating causes of energy waste. The main causes of energy waste at MAC layer are collision, overhearing, idle listening and control packets overhead [1]. Collision occurs when transmissions of two or more nodes overlap in time which results in failure of the communication and requires retransmission. Overhearing is the case in which a node receives packets which are not destined for it. In idle listening, a node keeps its receiver on in the hope of receiving something while the channel has nothing for it. Control packet overhead is caused by all those packets communicated for network and link management purposes. MAC protocols aimed at energy efficiency must reduce or eliminate the energy waste caused by all these reasons.

A large number of MAC protocols have been proposed to overcome these energy waste problems. These protocols can be classified into two groups: contention-based protocols and schedule-based protocols [2]. In contention-based protocols, nodes compete for access to the communication medium when they have data to transmit. Contention-based protocols are usually simple in working and don't require topology information or synchronization etc. on part of sensor nodes. In schedule-based protocols nodes use schedules to communicate. Usually each node is assigned its slot(s) according to some criteria and nodes use those slots to communicate. Scheduled-based protocols overcome the problem of

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.

collision and message retransmission but have an additional overhead of clock synchronization [1]. Furthermore schedule-based protocols face the problem of scalability and cannot easily adapt to topology changes.

In this paper, we present a TDMA-based MAC (TDMAC) protocol. This protocol is targeted at applications that require periodic data transmission by every node to the sink. Nodes are organized in clusters. Time is organized into frames and each node in a cluster is assigned a time slot in each frame to transmit its data to the cluster head (CH). The protocol assumes that there will not be a more than a pre-defined number of nodes (or nodes plus previous hop CHs) in a cluster. A CH Ph is previous hop of CH Nh if Nh is next hop of Ph.

This rest of the paper is organized as follows: section 2 discusses related work, section 3 takes a detailed view of the proposed protocol i.e. TDMAC, in section 4 a mathematical model for packet delay is developed and in section V simulation results are discussed.

2. Related Work

A large number of MAC protocols have been proposed for wireless sensor networks [1]. SMAC [3] is one of the most discussed protocols. It is contention-based. SMAC derives some concepts from IEEE802.11 [4]. In SMAC nodes save energy by using listen and sleep cycles. A node keeps its radio turned off while sleeping. Nodes in a neighborhood keep the same listen and sleep schedules forming a kind of virtual clusters. The duration of listen interval is application-dependent and is fixed. RTS/CTS/DATA/ACK procedure is used to limit collisions and the hidden terminal problem. TMAC [5] is an improvement on SMAC and dynamically adjusts the length of listen interval according to traffic conditions. DSMAC [6] is another variant of SMAC which dynamically adjusts duty cycle according to traffic conditions and available energy resources.

In BMAC[7] nodes use independent sleep schedule and periodically sample the medium to see if any node is trying to communicate with it. Transmitting nodes first send preambles before transmitting the actual data. The length of preamble should be long enough so that the intended destination does not miss it while sampling the medium.

WiseMAC[8] uses the same preamble technique for message transmission but reduces energy consumption by having nodes remember sampling offsets of neighbors. Nodes utilize this knowledge in selecting optimal time for starting preamble transmission, effectively reducing the length of preamble transmission and hence saving energy.

ZMAC[9] is a hybrid TDMA/CSMA-based protocol. Nodes have their assigned slots which they use when they have data to send. Nodes can even utilize other nodes' slots, if free, by using prioritized back-offs. Nodes use back-offs before trying to use any slots, even their own. However, back-offs for own slots are shorter then for others' which ensures that nodes get their own slots when they need it.

μ-MAC[10] tries to achieve energy efficiency by high sleep ratios. Application-level knowledge is utilized for flow specification. The operation of μ-MAC alternates between contention period and contention-free period. During contention period, topology discovery and sub-channel initialization is performed. In topology discovery, every node gets to know about its two-hop neighbors which is necessary for collision-free transmission. A sub-channel is a collection of related time slots in the contention-free period. There is a single general-traffic sub-channel carrying interest from base station or routing setup information, and a number of sensor-report sub-channels carrying reports from sensor nodes.

DEE-MAC[11] is TDMA-based and organizes nodes in clusters. It divides time into session with each session consisting of a contention period and a transmission period. Nodes send their interest to the cluster head during the contention period and are assigned slots by the CH for the transmission period.

3. TDMAC (TDMA-based Media Access Control) PROTOCOL

In this section we describe our proposed protocol TDMAC. TDMAC rigorously attempts to reduce or eliminate all causes of energy waste. It is aimed at applications in which nodes periodically sense the sensor field and send their readings to the base station (Sink). TDMAC organizes nodes into clusters. Each node sends its periodic readings to the cluster head (CH). The CHs use multi-hop communication to forward the readings received from nodes to the base station. The protocol requires that there should not be more than pre-defined maximum number of nodes (N) (or nodes plus previous hop CHs) in a cluster.

The working of the protocol consists of two phases. 1. Setup phase, 2. Steady phase

3.1 Setup phase

Setup phase consists of three sub-phases: cluster-formation phase, next-hop identification by cluster heads (CHs) and offset selection by CHs

a) Cluster-formation: Setup phase involves formation of clusters which is done the same way as in LEACH [12]. Nodes that are to be cluster heads broadcast a packet inviting other nodes to join the cluster. Non-cluster heads send joining requests to the CH. A non-cluster head node may receive invitation packet from more than one CH. In such a case it selects the one with the strongest received signal strength (RSS). Each cluster has a unique ID and the cluster head will assign ID to each node of the cluster. The sink forms a special type of cluster, and only nearby CHs can be members of that cluster.

b) Next hop identification: Once the cluster-formation is complete, the sink broadcasts next hop discovery packet. There is a hop-count field in this packet which is set to zero by the sink. This packet is intended only for cluster heads. Any non-cluster head node will simply drop this packet. The cluster heads that receive this packet set the sink as their next hop and broadcast the same packet setting their own cluster ID as the source ID and increment the hop count by one. Other cluster heads will receive the broadcast and repeat the same process. At the conclusion of this phase, each CH knows its next hop cluster. After the next hop discovery, each CH informs its selected next hop cluster head that it (the CH) would forward its data to him (next hop CH) for transmission. The next hop cluster head assigns it an ID for that purpose. A cluster head would be treated as a normal node (with a slight difference, which we explain later) in the next hop cluster and will have a time slot like other nodes in the next hop cluster, which it will use to transmit its data to the next hop CH. This time slot is calculated using its Id assigned by the next hop cluster head.

c) Offset selection: Offsets are meant for avoiding interference among neighboring clusters and involves time shifting of slots. Each CH selects and offset that is different from all its neighboring CHs. The number of different offsets depends on the density of sensor nodes and clusters; however, in most cases, four different offsets will be sufficient.

Initially each CH sets a timer to a random value in the range 0 -- T_{max} and turns on its receiver. When the timer of a node fires, it selects an offset for itself from the set of available (unused) offsets and broadcasts this decision in a packet. All the CHs that receive this packet mark the offset mentioned in the packet as used and reset the timer to a random value in the range 0 -- $T_{max}/2^{nopr}$, where ‘nopr’ is the number of offset packets received. This process continues unless all the CHs have chosen an offset.

3.2 Steady-State phase

In the steady-state phase, the periodic sensing of the field and transmission of their readings to the base station takes place. Time is divided into frames.

a) Frame: In a TDMAC frame, each node gets a slot in which it sends its readings to the CH, and each cluster head gets a slot to send its data to the next hop CH. Additionally, there is a slot for broadcasting control information (if any needed) to all the nodes in the cluster. There are two more slots reserved for any newly arrived node to join the cluster.

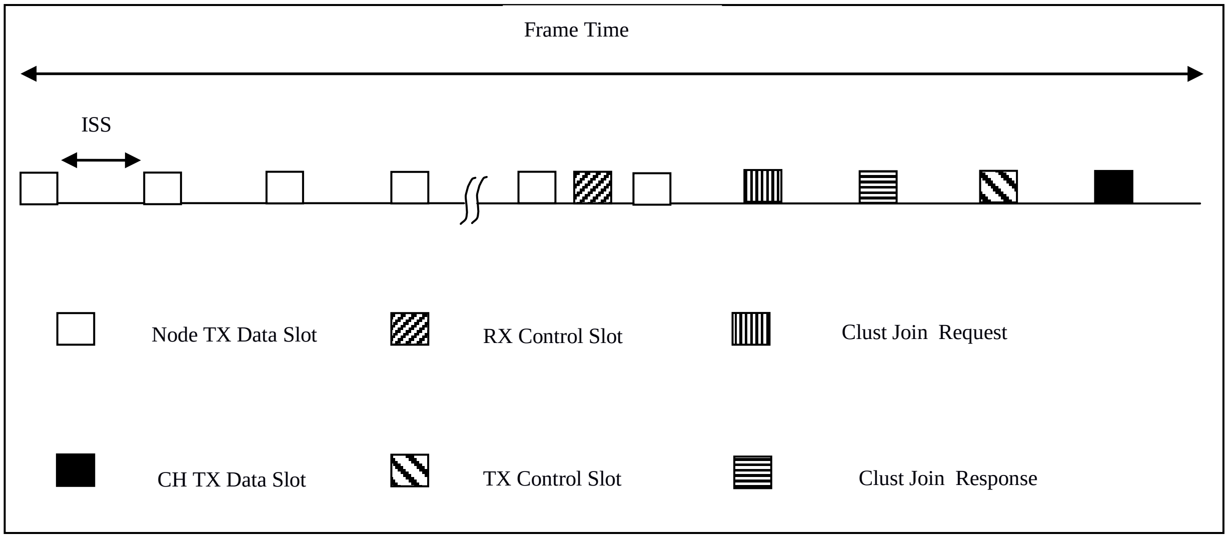

Fig. 1 TDMAC Frame Format

Figure 1 shows the format of a TDMAC frame. A frame consists of N + 4 slots of length `SlotDuration`. N is the maximum number of nodes that a cluster can have. The first N slots are for nodes, including previous hop CHs, to transmit their readings to the CH. Slots N+1 and N+2 are for newly arrived nodes to join the cluster and slot N+3 is for broadcasting control information to cluster nodes. Slot N + 4 is for transmitting data to the next hop CH. The length of the frame `(FrameTime)` is determined by the periodicity of sensing the field i.e. how frequently the field needs to be sensed. A node can calculate the start time of its slot using its node ID and frame start time by using the following expression.

$$\text{NodeSlotTime} = \text{FrameStartTime} + \text{NodeID}*(\text{ISS} + \text{SlotDuration}) \qquad (1)$$

b) Inter-Slot-Space (ISS): Inter-slot space is the separation between two consecutive slots in a frame. The length of ISS is carefully chosen so that it allows for already specified number of offsets and guard period. Guard period is the minimum separation, in time, between slots of neighboring clusters.

$$\text{ISS} = (\text{NumberOfOffsets} - 1)*\text{SlotDuration} + \text{numberOfOffsets}*\text{GuardTime} \qquad (2)$$

c) Start of frame calculation: The frame of a cluster should start earlier than its next hop so that its last slot (the slot in which it will forward to next hop) coincides with the slot allocated to it in its next hop's frame. Furthermore, in order to avoid interference among neighboring clusters, some offset is used to shift the slots in time relative to other neighboring clusters to make sure that slots in any neighboring cluster don't overlap with its own slots. Here is how the start of frame is calculated by a cluster head.

Suppose CH A is the next hop of CH B. Now if `NextHopFrameStart` is the start time of CH A's frame and `NextHopNodeID` is the ID assigned to B by CH A. Now using the slot time calculation expression (as used in case of node slot time calculation), slot in which CH B will transmit its data to CH A occurs at time:

$$\text{TnextHopSlot} = \text{NextHopFrameStart} + \text{nextHopNodeID}*(\text{ISS} + \text{slotDuration}) \qquad (3)$$

So by that time, CH B should have completed the N+3 slots of its frame so that its slots for transmission to next hop coincides with the slot allotted to it in the next hop (CH A's) frame. So the frame of CH B should start `(FrameTime – ISS – SlotDuration)` earlier than `NextHopFrameStart` (that is CH A frame start), that is,

$$\text{FrameStart} = \text{NextHopFrameStart} - (\text{FrameTime} - \text{ISS} - \text{SlotDuration}) \qquad (4)$$

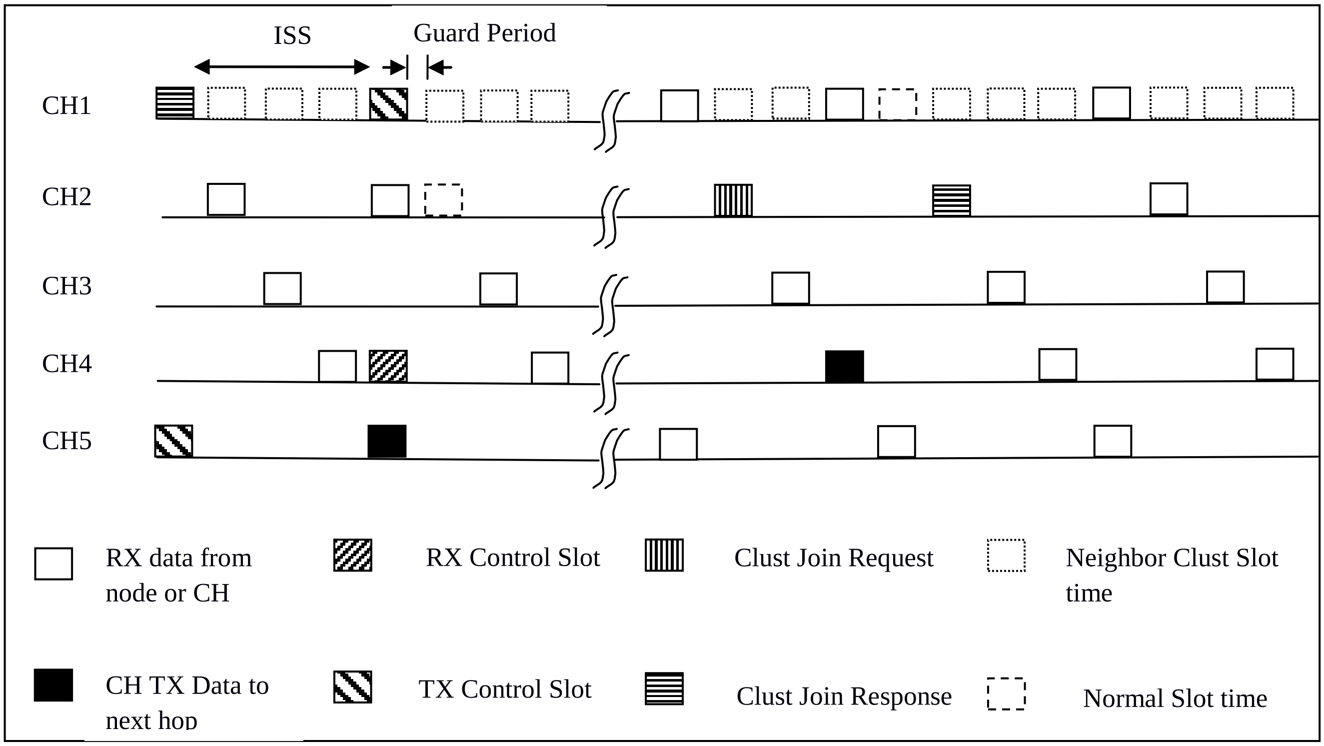

Fig. 2 Relative Frame Starts and Offsets

d) Frame start offset: Now if a CH (CH B for example) will start its frame at the time calculated according to equation 4, its slots will be perfectly synchronized with its next hop's (CH A for example) and hence interference will be caused. So to avoid this, the start of frame is shifted in time according to the offset type selected by the CH.

e) Data communication: Each node sends its data (readings) to the CH in each frame in the slot allotted to it. Nodes calculate their slot time using the equation 1. However, transmission by a CH to its next hop does not exactly occur according to the equation 3. Rather it occurs according to the sending CH's offset so that interference in another neighboring cluster of the sending CH which is using the same offset type as the next hop's is avoided. Consider for example Figure 2, which shows the time lines of four different clusters. CH1 is next hop of CH4 (CH1 and CH4 are cluster heads of their respective clusters). Furthermore CH5 is a neighbor of CH4 but not CH1 and hence it uses the same offset type as CH1 (and hence slots CH1 and CH5 occur at the same time). Now if CH4 were to forward its data to its next hop, that is CH1; its slot would occur at the time shown dashed on CH1's time frame. This would result in interference in CH5 since one of its slots occurs exactly at that time. So to avoid this interference CH4 sends its data to CH1 (the next hop) according its own offset as shown in the figure.

f) New node addition: Slots N+1 and N+2 are used for new addition to the cluster. When a new node arrives in the vicinity of a cluster, it keeps is receiver on to listen for cluster joining beacon which the cluster head transmits at the start of slot N+1. The new node immediately sends joining request. Upon receiving the request, the CH sends `NodeID`, start of frame and other control information to the new node in slot N+2.

g) Node removal: If the cluster head does not receive data from a node for a predefined number of times, it considers the node as dead or moved and removes it from the list of cluster nodes. Its ID is added to the list of unused IDs and may be assigned to any newly arrived node.

h) Time synchronization and other control information: TDMA-based protocols require that clocks of all nodes of a cluster be strictly synchronized with their CH. Due to clock drift, however, after some time, depending on drift rate, nodes may lose synchronization with their CH. Therefore, slot N+4 is reserved for broadcasting synchronization-related and other control information to all cluster nodes. All nodes of the cluster turn on their receivers when this slot arrives. For a CH, since the slots of the next hop cluster occur when it's ISS is in progress, it can safely receive synchronization and other control information.

4. Delay model for TDMAC

In this section we build a mathematical model of the delay that a packet undergoes before reaching the base station. If frameTime is the duration of one TDMAC frame and N is the maximum number of nodes (or nodes plus previous hop CHs) that a cluster can have; then the maximum delay that a packet undergoes before being forwarded to next hop is

$$\mathrm{MaxDelayInClust} = \frac{\mathrm{FrameTime}}{(N+4)} \times (N+3) \qquad (5)$$

And the minimum delay will be

$$\mathrm{MinDelayInClust} = \frac{\mathrm{FrameTime}}{(N+4)} \times 4 \qquad (6)$$

Now, to find the delay that the packet undergoes en route to the base station, let's suppose that a CH gets a slot no earlier than Slot S (this supposition is justified because a CH will allot their early slots to normal nodes by assigning then lower IDs). In that case the maximum delay per hop will be

$$\mathrm{MaxDelayPerHop} = \frac{\mathrm{FrameTime}}{(N+4)} \times (N-S+3) \quad (7)$$

And minimum delay per hop will be the same as minimum delay in cluster, that is,

$$\mathrm{MinDelayPerHop} = \frac{\mathrm{FrameTime}}{(N+4)} \times 4 \qquad (8)$$

So if there a H number of hops to the base station then the maximum delay before a data packet reaches the base station is,

$$\mathrm{MaxDelay} = H \times \frac{\mathrm{FrameTime}}{(N+4)} \times (N+3) + \frac{\mathrm{FrameTime}}{(N+4)} \times (N-S+3) \qquad (9)$$

Or

$$\mathrm{MaxDelay} = \frac{\mathrm{FrameTime}}{(N+4)} \times [H(N+3)+N-S+3] \qquad (10)$$

And the minimum delay will be

$$\mathrm{MinDelay} = \frac{\mathrm{FrameTime}}{(N+4)} \times 8 \times H \qquad (11)$$

And the average delay will simply be

$$\mathrm{AvgDelay} = \frac{\mathrm{MinDelay}\ + \mathrm{MaxDelay}}{2} \qquad (12)$$

5. Simulations and Results

Simulations of the proposed protocol were carried out in Castalia[13] and MATLAB. Castalia is based on OMNet++ [14] and is specially developed for wireless sensor network and body area networks. 44 nodes were randomly deployed over an area of 200m by 200m. Simulations were run, under the same conditions, for TDMAC and SMAC and performance compared.

Figure 3 shows a comparison of TDMAC and SMAC for various sample intervals. The figure clearly indicates that TDMAC consumes less energy than SMAC for all sample intervals. Furthermore, the energy consumed by TDMAC drops along as the sample interval increases whereas that of SMAC does not decrease much and in

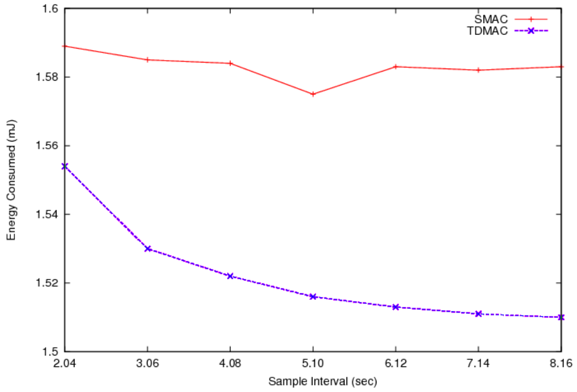

Fig. 3 TDMAC vs SMAC: Energy Consumption

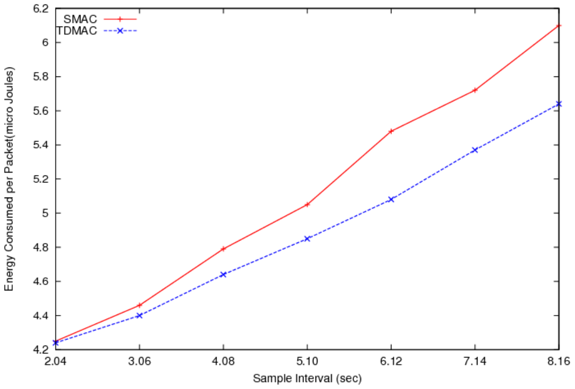

Fig. 4 TDMAC vs SMAC: Energy Consumption per data packet

fact it shows somewhat random behavior.

Figure 4 shows a comparison of energy consumed for a single data packet sent. Here again TDMAC does better than SMAC. One can see that the energy consumed per data packet rises as the sample interval increases. Ideally, the energy consumed per packet should be the same for all sample intervals. This happens because of the fact that Castalia provides realistic environment and nodes consume some energy even if in the sleep mode or doing processing. With longer sample interval, nodes send less frequently and hence more sleep mode energy consumption accounts for it.

Figure 5 shows the delay performance based on the mathematical model described in section IV. The model was implemented in MATLAB Here‘s’ is the slot number that a CH gets in the next hop cluster. The total number of slots was kept at 17 and frame duration was kept 1 sec. The figure clearly shows that as ‘s’ increases (that is CH is allotted later slot in the frame of next hop), the average delay drops. Thus one can easily conclude that CHs should be allotted later slots in the frame of next hop CH, so as to minimize the delay a data packet undergoes before reaching the base station.

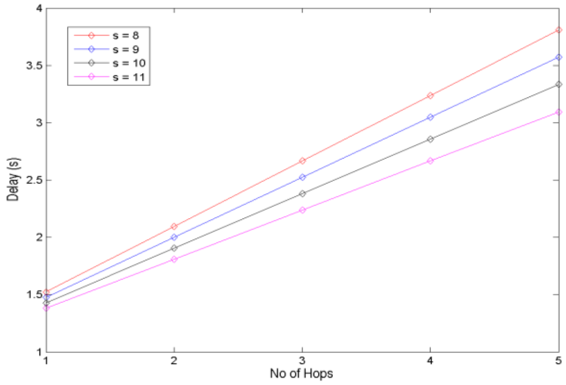

Fig. 5 Delay performance of TDMAC

6. Conclusions

A new energy-efficient TDMA-based MAC protocol was presented. The protocol was simulated in Castalia and its results compared with SMAC for various sample intervals. TDMAC performed better than SMAC and adjusted well to the requirements of periodic sensing applications. In future, we aim at coming up with another version of TDMAC which will assign slots dynamically so as to be fit for applications in which sensor nodes’ bandwidth requirements vary over time.

7. Acknowledgement

We are highly thankful to the GIK Institute for providing all the facilities for carrying out this research work.

References

[1] K. Kredo II, P. Mohapatra, “Medium access control in wireless sensor networks”, Computer Networks, vol. 51, 2007, pp. 961—994.

[2] K. Sohraby, et al., Wireless Sensor Networks: Technology, Protocols, and Applications, John Wiley and Sons Inc., 2007.

[3] W. Ye, et al., “An energy-efficient MAC protocol for wireless sensor networks”, Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (InfoCom), vol. 3, 2002, pp. 214–226.

[4] IEEE, IEEE Standard 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, 1999.

[5] T. van Dam, K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks”, Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2003, pp. 171–180.

[6] P. Lin, et al., “Medium access control with a dynamic

duty cycle for sensor networks", in Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC), vol. 3, 2004, pp. 1534–1539.

[7] J. Polastre, J. Hill, D. Culler, "Versatile low power media access for wireless sensor networks", in Proceedings of the International Conference on Embedded Networked Sensor Systems (SenSys), 2004, pp. 95–107.

[8] A. El-Hoiydi, J.-D. Decotignie, WiseMAC: an ultra low power MAC protocol for multi-hop wireless sensor networks, in: Proceedings of the International Workshop on Algorithmic Aspects of Wireless Sensor Networks (Algosensors), 2004, pp. 18–31.

[9] Rhee I, Warrier A, Aia M, et al. "Z-MAC: a Hybrid MAC for Wireless Sensor Networks", in IEEE/ACM Transactions, 2008, vol. 16(3), pp. 511 – 524.

[10] Barroso A, Roedig U, and Sreenan C., "μ-MAC: an energy efficient medium access control for wireless sensor networks", in Proc. of the Second European Workshop on Wireless Sensor Networks, Istanbul, 2005, pp. 70 – 80.

[11] Cho Sungrae, et al, "Dynamic energy-efficient TDMA-based MAC protocol for wireless sensor networks", in Autonomic and Autonomous Systems and International Conference on Networking and Services, Papeete, Tahiti, Joint International Conference, Oct. 2005, pp.48-48.

[12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Hawaii International Conference on System Sciences, 2000, pp. 3005-3014.

[13] Castalia User Manual, Available: http://castalia.npc.nicta.com.au/pdfs/CastaliaUserManual.pdf

[14] www.omnetpp.org

Shams ur Rahman got his B.S Computer Systems Engineering from the University of Engineering and Technology in 2007 and MS in Computer System Engineering from Ghulam Ishaq Institute of Engineering Sciences and Technology, Topi. He worked with GIKI as Computer

Engineer before enrolling to the MS degree. Currently he is working as Research Associate (Lecturer) at GIK Institute.

Dr. Mushtaq Ahmad got his B.Sc degree in Physics, Pure Mathematics and Applied Mathematics from Islamia College Peshawar, Pakistan. He got his M.Sc in Nuclear Physics from the University of Dhaka in 1965 and Ph.D in Computer Architecture and Computer Arithmetic from Victoria University Manchester in 1971. He has vast professional experience. He worked with Bradley University, Peoria, IL, USA as Associate Professor, Computer Science Department: from August, 1986 to May, 1988. Later on he moved to Engine Division Engineering, Caterpillar Inc., Mossville, IL, USA and worked as Computer Consultant from June, 1989 to December, 1989. He worked with Illinois Central College, East Peoria, IL, USA, Mathematics Department, and taught Mathematics courses from March, 1990 to March, 1996. From April 1997 to January 1998 he worked with Caterpillar Logistics System, Peoria, IL, USA as Systems Analyst. From July, 1999 to June, 2001 he worked at Choice Point Direct, Peoria, IL USA as CS/ MIS consultant. He worked with Infogenic Systems Inc., Peoria, IL USA as CS/ MIS consultant/Project Management from March 2001 to June 2006. Currently he has been working as professor at GIK Institute since August 2007. He has supervised two Ph.D and a number of MS students.

Dr. Shafaat A. Bazaz got his B.S. from NED University of Engineering and Technology Karachi in 1989 M.S. from Université de Franche Comté, Besanc¸ France in 1994 Ph.D. in Microelectromechanical Systems (MEMS) from Institute National des Sciences Appliquées (INSA) Toulouse, France in 1998. He worked in Technical University of Denmark (DTU) during 1999–2000 as Research Assistant Professor. There, he supervised CAD for MEMS project funded by Denmark Technical Research Council. From 2000 to 2006, he worked in CMC Mcirosystems in Canada as Senior Staff Scientist in Microsystem Engineering. His main responsibility was to provide design, fabrication and test services through fabless design manufacturing network to Canadian universities in the area of MEMS and Microfluidics. He served as a Visiting Professor, Department of Electrical Engineering, University of Waterloo, Canada from 2003-2006. In July 2006, he joined Ghulam Ishaq Khan Institute of Engineering Sciences and Technology as Foreign Professor under Higher Education of Pakistan's (HEC) program of Foreign Faculty Hiring Program. He has received numerous research grants (total of Rs. 20 Millions) from HEC and National ICT Fund to establish research in the area of MEMS. Presently, he is heading the Department of Computing in CASE and his research group is involved in the development of MEMS based microgrippers for biomedical applications, micro gyroscope and 3-axis accelerometers.

Copyright (c) 2012 International Journal of Computer Science Issues. All Rights Reserved.