Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 10 (2012) 78 – 85

Procedia Computer Science

The 3rd International Conference on Ambient Systems, Networks and Technologies (ANT)

Adaptive TDMA Slot Assignment Using Request Aggregation in Wireless Sensor Networks

Abdulaziz Y. Barnawi

Computer Engineering Department, King Fahd University of Petroleum and Minearls, Dhahran 31261, Saudi Arabia

Abstract

TDMA-based MAC protocols are considered an energy efficient solution to prolong wireless sensor network lifetime. However, their drawbacks such as complexity of slot assignment and schedule maintenance and adaptivity to varying traffic conditions are yet to be handled in an efficient way. In this paper we present On-demand Convergecast Scheduling-based (OCS) MAC protocol. It is a centralized and adaptive multihop scheduling-based TDMA protocol which supports convergecast applications. OCS adopts a novel requests aggregation mechanism for adaptive slot assignment such that time slots are assigned on-demand to currently active sources as well as relays. The performance of OCS is compared to existing protocols based on simulations in ns-2. Results show that OCS outperforms protocols such as Z-MAC, S-MAC, and others in terms of delay, throughput and energy efficiency.

Keywords: Wireless Sensor Networks, MAC, TDMA, Scheduling, Energy efficiency, Delay.

1. Introduction

Wireless Sensor Networks (WSN) are being deployed to support many different real-world applications, e.g. event detection, target tracking and field monitoring. Data communications in most of these applications are from sources to sink (*convergecast*) [1], [2], [3], [4]. In addition, it is expected that these networks remain active for a long period of time (i.e. have a long lifetime) while maintaining a specific level of confidence (accuracy) in the collected data. However, WSN utilize sensors that are usually inexpensive, small in size and weight, and most importantly battery powered. These characteristics lead to the requirement of designing protocols that aim to conserve energy consumption and extend network lifetime [5].

In the area of MAC design for WSNs, duty cycle and on-demand wakeup MAC protocols owe in part their performance gain to the use of CSMA as their underlying access mechanism. However, for the same reason, they might experience performance degradation under high contention and/or high traffic rate. On the other hand, TDMA-based MAC protocols where each node is assigned specific time slots to send, receive data or sleep, inherit the advantage of energy conservation due to the existence of the built-in duty cycle.

Email address: barnawi@kfupm.edu.sa (Abdulaziz Y. Barnawi)

doi:10.1016/j.procs.2012.06.014

Furthermore, they can avoid the hidden terminal problem by scheduling transmission of interfering nodes at different times.

Nevertheless, a conventional TDMA protocol has a number of drawbacks which make it undesirable for certain sensor applications and/or network configurations. Some of these drawbacks are: tasks of slot allocation and schedule maintenance could be complex and it is difficult to change frame size and slot assignment when introducing new nodes. In addition, following a rigid transmission schedule does not allow for effective avoidance of idle listening.

In this paper we present a centralized yet an adaptive multihop scheduling TDMA-based MAC protocol, which supports convergecast applications in event-driven WSNs with the objective of energy efficiency and delay guarantee. The proposed protocol adopts a novel slot requests aggregation mechanism to determine with high confidence the active nodes within the detected event area. Based on the collected topology information and the aggregated slot requests, time slots are assigned to nodes.

The rest of the paper is organized as follows: Section 2 reviews related MAC protocols in WSNs while focusing on TDMA and scheduling-based protocols. Section 3 presents the proposed adaptive TDMA scheduling protocol for in WSNs. Performance evaluation of the proposed protocol is discussed in Section 4. Finally, Section 5 concludes the paper.

2. Related Work

Duty cycle protocols, such as S-MAC [6] , T-MAC [7] and DMAC [8] rely mainly on CSMA in their operation. They allow nodes to alternate between two periods: *active period* and *sleep period*. During the listen period, nodes listen to the channel for possible traffic from neighbors and go back into sleep mode when that period expires. A potential transmitting node must acquire knowledge about listen periods of targeted receiving nodes and send its packet at the right time. An entire cycle consisting of a sleep and listen period is called a *wakeup period*. The ratio of the listen period length to the wakeup period length is called *duty cycle*. It is important to note that a short duty cycle means that a node sleeps most of the time to avoid idle listening and conserve energy. However, as the duty cycle gets shorter, the listen period also gets shorter, which might increase contention and subsequently collisions among nodes wanting to transmit to the same target node, especially in heavy load situations. In addition a long sleep period increases latency.

TDMA MAC protocols [9, 10, 11, 12, 13] assign time slots during which nodes either send/receive or sleep. The scheduling process can be centralized or distributed. For example, the Interleaved Spatial Temporal (IST) Scheduling [9] proposes a centralized slot assignment based on nodes' relative distance (hop count) from the sink. The objective is to facilitates the flow of traffic from nodes to the sink in convergecast applications. Two scheduling schemes are proposed: Top-Down (TD) scheduling and Bottom-Up (BU) scheduling. In TD scheduling all possible transmission slot assignments, i.e. send, receive or sleep, for nodes one hop from the sink are performed then for those in the second hop, and so on. While in BU, nodes with the maximum hop count are the first to be assigned transmission slots, then nodes in the next hop closer to the sink and so on. Unlike, IST scheduling, Data Aggregation TDMA Protocol (DATP) [12] is a distributed protocol and does not require nodes to keep track of their neighbors schedules. It consists of three phases of operations: a quiet phase in which time synchronization and a routing tree are performed; an event-triggered scheduling phase in which DATP is executed; and a data transmission phase in which the event is reported.

Hybrid TDMA MAC protocols [14, 15, 16, 17] attempt to integrate the controllability of TDMA protocols with the flexibility of distributed protocols. Therefore, adapt to current network and traffic conditions. For example, Z-MAC [16] proposes a novel way of combining CSMA and TDMA to adapt to the level of contention in the network. During Z-MAC steady state operation, a node can be in one of two modes: low contention (LCL) or high contention level (HCL). In the LCL mode, any node can compete to transmit in any slot (Z-MAC behaves like CSMA) while in the HCL mode, only the owners of the current slot and their one-hop neighbors are allowed to compete for channel access (Z-MAC enforces the use of TDMA). To conserve energy, Z-MAC runs on top of B-MAC [18], a preamble sampling LPL protocol. Each node maintains a listening duty cycle separated by a *check period* and each transmission is preceded by a preamble as large as the check period.

3. OCS

In this paper, we propose an adaptive and energy efficient MAC layer protocol that allows nodes to sleep most of the time then wake-up at specific time slots to send and/or relay requests for slot assignment to the sink. Based on current topology and network activity information, the sink creates a multihop schedule that carries a new slot assignment for the requesting nodes as well as the relaying ones. Unlike many proposed TDMA-based protocols, time slots are only assigned to nodes that are sources or relays of traffic. In addition, when there is no traffic is received from these sources for a specific time period, the sink will remove those nodes from the last slot assignment.

The proposed protocol goes through two main phases during its operation: a setup phase and a steady state phase. In the following we discuss the details of each phase.

3.1. Setup phase

The setup phase is the initial step during which sensors determine the presence of neighboring nodes in their vicinity. This information is relayed to the sink which will use it during the slot assignment phase. For the setup phase we use PROGRESSIVE, [19] which is an energy and time efficient topology construction and setup protocol. It consists of three phases: *neighbor discovery*, *topology collection and construction*, and *progressive adjustment*. The advantage of PROGRESSIVE is that is conserves energy by controlling the use of CSMA while collecting network connectivity information. Undiscovered nodes allowed to enter the sleep mode and only those, which have one or more scheduled neighbor will wake up at specific time interval to send their connectivity information to the sink. Doing so, will progressively collect topology information at the sink using less energy consumption.

3.2. Frame Structure and Size

During the course of the steady state phase, nodes operate in one of two types of frames: a *sleep frame* or an *active frame*. Each frame has S *data exchange slots*. During the sleep frame (see Figure 1) all nodes are assigned Sleep in each data slot. In the active frame, however, nodes can be assigned Send, Receive, or Sleep depending on the outcome of the scheduling algorithm, which is discussed in more detail later Section 3.3. During the steady state , the data exchange slot duration T_{slot} can be represented by:

$$T_{slot} = T_{CCA} + T_{DATA} + T_{ACK} \tag{1}$$

where T_{CCA} is the Clear Channel Assessment time while T_{DATA} and T_{ACK} are the transmission times of data packet and the acknowledgement, respectively.

The two other types of slots are ScheduleReq slot and Resp/Sync slot. These slots are key elements in achieving energy- and delay-efficient design of OCS. There is a single ScheduleReq slot within each sleep frame. This slot is used as a low power listening mechanism by nodes to transmit and/or relay *schedule requests* (when sampled data is ready) to the sink. Based on the received node requests, the sink schedules nodes and transmits the new slot assignment in what we call Schdl packet during the Resp/Sync slot. The time (number of data slots) between the ScheduleReq and the next Resp/Sync should be long enough for the sink to compute a new schedule. If requests are not received then, the Schdl packet is not transmitted and the next frame remains as a sleep frame. This mechanism allows nodes to sleep during the sleep frame and only wake up for a short period of time during the ScheduleReq slot to sense any traffic request in case there is some or to continue sleeping in case there is none. The other function of the Resp/Sync slot is to synchronize nodes by using the Schdl packet or Beacons when there is no network activity, i.e. when the Schdl packet is not transmitted.

Setting the sleep frame size is important for achieving certain application requirements, for instance energy and delay. With respect to energy, nodes sleep time is proportional to frame size, i.e. the bigger the frame size, the longer nodes are able to sleep and hence, conserve energy. However, for event-driven applications with short delay requirements, the longer sleep periods translate into longer end-to-end delays. Therefore, the application sampling rate - $r_{sampling}$ and maximum end-to-end delay - $delay_{max}$ requirements must be considered in deciding the optimal frame length.

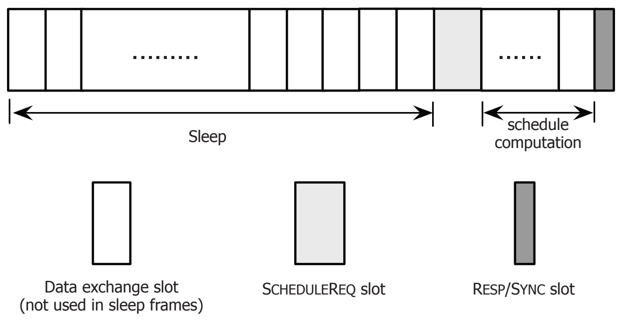

Fig. 1. Sleep frame structure.

3.3. *Steady State Phase with Adaptive Slot Assignment*

The underlying scheduling schemes provide a highly conflict-free multihop TDMA access for sensors to transmit their sensing data back to the sink. In this paper we use the IST scheduling schemes [9]. However, unlike our previous work, to achieve energy efficiency we assume that the default mode of operation uses sleep frames. Except during the ScheduleReq, nodes sleep for the whole frame. Nodes stay active in the ScheduleReq slot to transmit (if a slot is needed) and/or relay schedule requests transmitted by neighbors. When an event is detected or data is sampled as a result of a query, the involved nodes utilize the upcoming ScheduleReq slot to demand a new schedule during which these nodes are being assigned transmission slots. This way the protocol assign active time slots to nodes adaptively and on-demand only to active sources and relays. A frame that has Send or Receive slots is called an active frame (Figure 2).

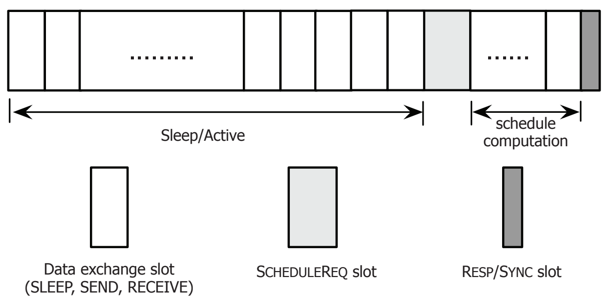

Fig. 2. The Active frame structure.

A special care must be taken in developing the mechanism to transmit and forward schedule requests to the sink. First, successfully delivering schedule requests to the sink is the only way to guarantee that nodes are able to get slot assignment and, subsequently, communicate their sensing data. Second, schedule requests are overhead that eventually contributes to energy consumption. Third, the duration of the ScheduleReq slot affects the delay as well as energy consumption. Having a long ScheduleReq slot means longer periods of idle listening, especially when data is not sampled which translates into higher energy consumption. In addition, nodes would experience more delay before being able to transit their data.

To address these issues, a schedule request aggregation mechanism during which multiple schedule requests are combined into a single one is proposed. This mechanism requires dividing the ScheduleReq slot into fixed size *request sub-slots*. Each sub-slot is long enough to transmit a single schedule request (see Figure 3). Moreover, a *schedule request aggregation tree*, computed by the sink is used to assign a *request listen window* (a number of request sub-slots) for each node within the ScheduleReq slot. Listen window assignment information is included in the Schdl frame transmitted by the sink. During each window a node must listen for schedule request transmissions made by its child nodes. Once the listen window expires, the aggregated schedule requests (including that of the receiving node, if available) are broadcasted in a new packet. It has to be noted that there could be more than one parent (upstream node) listening to the same child. Hence, the same schedule request might be routed through out multiple paths before reaching the sink.

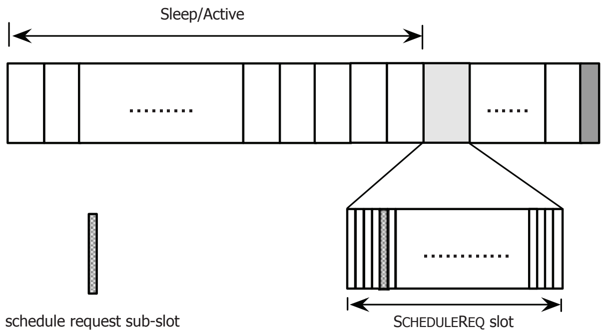

Fig. 3. Schedule request sub-slots.

The proposed request aggregation mechanism is simple and yet energy-efficient in transmitting many schedule requests in a short period of time. The objective is to control the duration of SCHEDULEREQ slot by using very small fixed-size request packets. The design of these packets is based on the assumption that nodes in close proximity can detect the same event, therefore, generate schedule requests at very close time intervals. The variation of the time instant when the event is detected and reported may depend on the application and the observed phenomenon. In this work, we assume that this variation is very small such that nodes are able to detect and start reporting the event at a negligible time difference. Therefore, it is sufficient for the sink to identify, with some degree of confidence, the relative perimeter of the active area, hence schedule nodes within that active area.

To obtain a slot assignment nodes broadcast their schedule requests during the next SCHEDULEREQ slot. Each schedule request packet is assumed to have two main fields: *activity rootID* and *activity depth*. After receiving a schedule request, the node sets itself as the *activity root* in two cases. First, if it is a source node and second, if it has received schedule requests from different branches of a tree rooted at itself. Otherwise, it would forward the request packet as is. The activity depth is the difference in hop count between the activity root and first initiator of the schedule request. Received schedule requests are stored and aggregated until the schedule listen window expires. Then they are forwarded to the next hop. Once a schedule request is received at the sink, the activity root and all nodes within the activity depth from that root are set as active sources. Figure 4 provides an example of a network with two detected. The schedule request aggregation trees show the activity root and activity depth for each event.

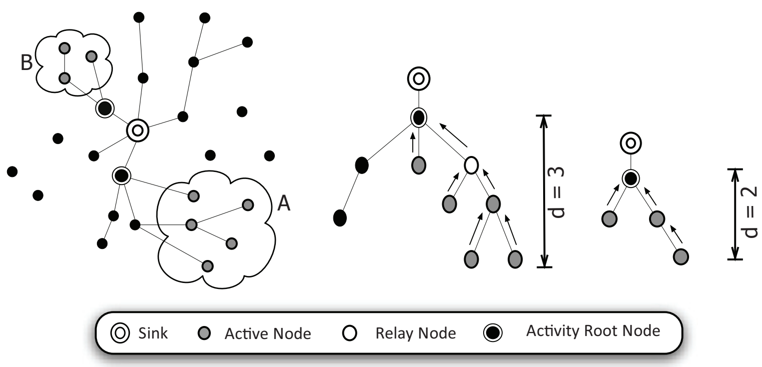

Fig. 4. Schedule request. (a) Events A and B are detected in two different geographical areas of a network (b) Request transmission and request aggregation tree for event A (*depth* = 3) (c) Request transmission and request aggregation tree for event B (*depth* = 2).

Despite the advantages this mechanism offers, it has some drawbacks:

- Inactive sources are unnecessarily selected as members of the aggregation tree. Figure 4 shows two inactive nodes which are in event *A*'s aggregation tree.
- Large number of inactive nodes are redundantly scheduled if sources are scattered throughout the network and the activity root is set on the bases of different schedule requests generated by nodes, which are very far, in terms of hop count, from the activity root.
- Schedule requests are unacknowledged, end-to-end delays might increase if request packets are lost due to transmission errors.

A possible solution to the first problem is to associate a timer - $t_{active-source}(n)$ for each source node n. This timer is activated whenever a new data packet is received from n. If data is not received, the timer expires and this particular source is removed from the list of active sources and eventually, would not be removed from future slots assignments. As for the second problems, any node receiving schedule requests from two different nodes would not set itself as the activity root. Instead, it would forward the schedule request with the larger activity depth and eliminate the other.

4. Performance Evaluation

To assess the performance of OCS MAC, various simulations are conducted using ns-2 [20] *v2.31*. Simulations results are averages of 20 independent (different seed) replications, plotted with 95% confidence intervals. The performance OCS is compared against that of ALWAYS-ON, S-MAC, DMA and Z-MAC in terms of delay, total energy consumption and energy efficiency. For all protocols, it is assume that the routing paths are predetermined. This is necessary to have a fair comparison among protocols and to isolate delays and energy consumption due to acquiring these routing paths.

Table 2 shows the time and current drawn while performing radio and microcontroller operations in MICA2 [18, 21, 22, 23] sensor nodes. The choice of MICA2 as a platform for evaluating OCS makes it possible to validate and compare our results with that of other protocols. With respect to the sink, it is assumed to have an unlimited source of energy.

Nodes' Placement	Randomly uniform	Avg. Node Degree	5
Sink Placement	Center of the field	Number of Sink(s)	One
Transmission Range	20m	Interference Range	45m

Table 1. Default Simulation Parameters.

Turn on radio	1.5 ms	Switch to RX/TX	250 μs
Time to sample radio	350 μs	Evaluate radio sample	100 μs
Radio Tx Current	16.5mA	Radio Rx Current	9.6mA
Radio Idle Current	7.68mA	Radio Sleep Current	0.2μA
MCU Active Current	8mA	MCU Sleep Current	14μA

Table 2. MICA2 Sensor Timings and Power Consumption.

4.1. Network Size

This section discusses protocols' performance when the network size is varied in [40 - 160] nodes and the sampling rate is 0.1 pkt/sec. Figure 5 shows the delay of all protocols when the optimal energy efficiency is the primary requirement. It can be observed that OCS, DMAC and ALWAYS-ON protocols experience a continuous increase in the delay as the network size increases. OCS shows a slight increase in delay because the value of S is the same for all network sizes. The increase can be looked at as an increase in the scheduling delay. On the other hand, the effect of choosing different parameter values is reflected on the performance of Z-MAC and S-MAC as a fluctuation in delay. While S-MAC maintain a delay within 1s, Z-MAC's delay increases sharply once the network size is over 80 nodes. DMAC is the most severely affected protocol when the network size increases while ALWAYS-ON has the best delay among all. The reason for DMAC's performance is its spatial contention problem in which nodes in the same hop wake up at the same time and compete to transmit to the same neighbor, but only one successfully do. Nodes with unsuccessful transmissions have to wait for the next wakeup interval to go through the same process.

In terms of throughput, Figure 6, ALWAYS-ON maintains high throughput for all network sizes. Likewise, both OCS and S-MAC achieve high throughput. The reasons are that S-MAC is based on a simple CSMA duty-cycle based operation and OCS because of its adaptivity and near-conflict free scheduling. Among all, DMAC has the worst throughput when the network size increases.

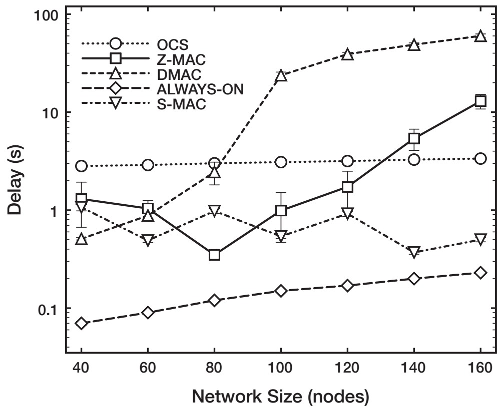

Fig. 5. Delay of all protocols under variable number of nodes.

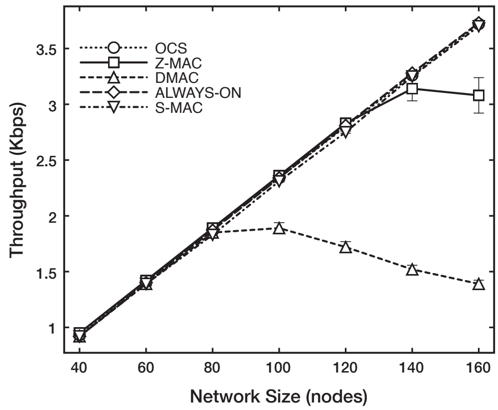

Fig. 6. Throughput of all protocols under variable number of nodes.

Similarly, an increase in network size causes an increase in nodes energy consumption, Figure 7. OCS has a superior performance compared to other protocols. It consumes the least energy per node, less than 0.5 *Joule* hence, the best energy efficiency. OCS low energy consumption is a result of its ability to put nodes to sleep for the longest possible period and when nodes wake up they experience no contention and the least interference during data transmission. ALWAYS-ON is the highest energy consuming protocol because nodes remain in the idle state most of the time. DMAC consumes less energy than both Z-MAC and S-MAC and it is also more energy efficient.

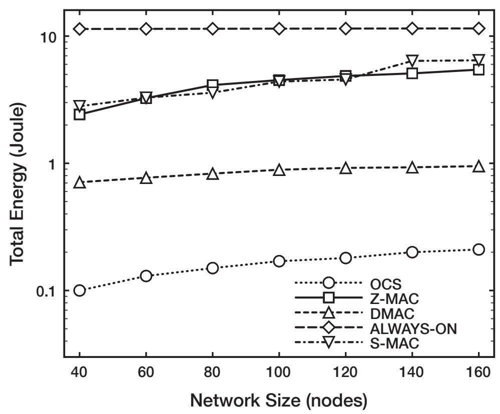

Fig. 7. Energy consumption of all protocols under variable number of nodes.

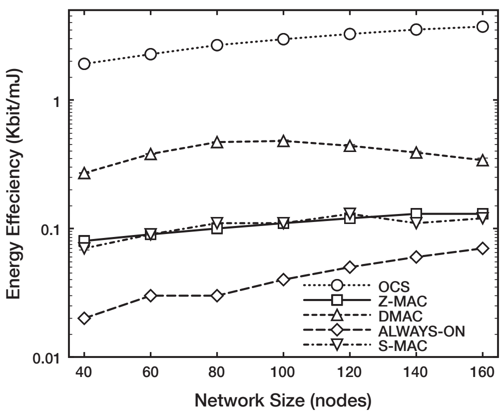

Fig. 8. Energy efficiency of all protocols under variable number of nodes.

5. Conclusion

This paper presents OCS, an adaptive TDMA MAC protocol for wireless sensor networks . OCS utilizes requests aggregation to dynamically allocate time slots for active sources and the required relays. The protocol utilizes Interleaved Spatial Temporal scheduling, which facilitates the flow of the collected data from sources to the sink in convergecast applications. Simulation results show that OCS outperform a number of other WSN MAC protocols in terms of delay and energy efficiency.

6. Acknowledgment

This work was supported in part by King Fahd University of Petroleum and Minerals under Grant No. JP111003, Energy-efficient MAC Protocols for Low-power Wireless Sensor Networks.

References

[1] F. Wang, J. Liu, Networked Wireless Sensor Data Collection: Issues, Challenges, and Approaches, IEEE Communications Surveys Tutorials 13 (4) (2011) 673 –687.
[2] P. Suriyachai, U. Roedig, A. Scott, A Survey of MAC Protocols for Mission-Critical Applications in Wireless Sensor Networks, IEEE Communications Surveys Tutorials PP (99) (2011) 1 –25.
[3] W. Wang, N. Wang, E. Jafer, M. Hayes, B. O'Flynn, C. O'Mathuna, Autonomous Wireless Sensor Network Based Building Energy and Environment Monitoring System Design, in: 2010 International Conference on Environmental Science and Information Application Technology (ESIAT), Vol. 3, 2010, pp. 367 –372.
[4] A. Nasipuri, R. Cox, J. Conrad, L. Van der Zel, B. Rodriguez, R. McKosky, Design Considerations for a Large-scale Wireless Sensor Network for Substation Monitoring, in: Local Computer Networks (LCN), 2010 IEEE 35th Conference on, 2010, pp. 866 –873.
[5] A. Bachir, M. Dohler, T. Watteyne, K. Leung, MAC Essentials for Wireless Sensor Networks, IEEE Communications Surveys Tutorials 12 (2) (2010) 222 –248.
[6] W. Ye, J. Heidemann, D. Estrin, Medium access control with coordinated adaptive sleeping for wireless sensor networks, IEEE/ACM Transactions on Networking 12 (3) (2004) 493 – 506.
[7] T. van Dam., K. Langendoen, An adaptive energy-efficient MAC protocol for wireless sensor networks, in: Proc. 1st International Conference on Embedded Networked Sensor Systems, 2003, pp. 171 – 180.
[8] G. Lu, B. Krishnamachari, C. S. Raghavendr, An adaptive energy-efficient and low-latency MAC for data gathering in wireless sensor networks, in: 18th International Parallel and Distributed Processing Symposium, 2004, pp. 224–.
[9] A. Barnawi, R. Hafez, Performance of TDMA scheduling algorithms in the presence of data correlation in sensor networks, in: Proc. IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Niagara Falls, ON, 2008, pp. 675–680.
[10] S. Ray, I. Demirkol, W. Heinzelman, ATMA: Advertisement-Based TDMA Protocol for Bursty Traffic in Wireless Sensor Networks, in: Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE, 2010, pp. 1 –5.
[11] D. Wu, G.-Y. Wang, X.-L. Li, Distributed TDMA Scheduling Protocol Based on Conflict-free for Wireless Sensor Networks, in: 2010 International Conference on Intelligent Computing and Integrated Systems (ICISS), 2010, pp. 876 –879.
[12] M. O. Díaz-Anadón, K. K. Leung, TDMA Scheduling for Event-triggered Data Aggregation in Irregular Wireless Sensor Networks, Computer Communications 34 (17) (2011) 2072–2081.
[13] Y. Wang, P. Shi, K. Li, Z. Chen, D-TDMA: An Approach of Dynamic TDMA Scheduling for Target Tracking in Wireless Sensor Networks, in: 2010 IEEE/ACM Int'l Conference on Int'l Conference on Cyber, Physical and Social Computing (CPSCom), Green Computing and Communications (GreenCom), 2010, pp. 546 –553.
[14] L. Sitanayah, C. Sreenan, K. Brown, ER-MAC: A Hybrid MAC Protocol for Emergency Response Wireless Sensor Networks, in: 2010 Fourth International Conference on Sensor Technologies and Applications (SENSORCOMM), 2010, pp. 244 –249.
[15] W. Lee, A. Datta, R. Cardell-Oliver, FlexiTP: A Flexible-Schedule-Based TDMA Protocol for Fault-Tolerant and Energy-Efficient Wireless Sensor Networks, Parallel and Distributed Systems, IEEE Transactions on 19 (6) (2008) 851 –864.
[16] I. Rhee, A. Warrier, M. Aia, J. Min, M. Sichitiu, Z-MAC: A Hybrid MAC for Wireless Sensor Networks, IEEE/ACM Transactions on Networking 16 (3) (2008) 511 – 524.
[17] G. P. Halkes, K. G. Langendoen, Crankshaft: an Energy-efficient MAC-protocol for Dense Wireless Sensor Networks, in: Proc. of the 4th European Conference on Wireless Sensor Networks, 2007, pp. 228–244.
[18] J. Polastre, J. Hill, D. Culler, Versatile low power media access for wireless sensor networks, in: Proc. 2nd International Conference on Embedded Networked Sensor Systems, 2004, pp. 95 – 107.
[19] A. Barnawi, R. Hafez, Progressive: a topology discovery and scheduling protocol for wireless sensor networks, in: Proc. of the 12th ACM international Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '09, 2009, pp. 325–328.
[20] ns-2 - The Network Simulator, http://www.isi.edu/nsnam/ns/.
[21] Texas Instruments. CC1000 - Single Chip Very Low Power RF Transceiver, http://www.ti.com.
[22] Atmel - Atmega128, http://www.atmel.com.
[23] Crossbow Technology Inc. MICA2, http://www.xbow.com.