

# An Adaptive Coordinated Medium Access Control for Wireless Sensor Networks

Jing Ai, Jingfei Kong, Damla Turgut  
Electrical and Computer Engineering  
University of Central Florida  
Orlando, FL 32816  
Email: {jingai, jfkong, turgut}@cs.ucf.edu

**Abstract**—In this paper, we have developed Adaptive Coordinated Medium Access Control (AC-MAC), a contention-based Medium Access Control protocol for wireless sensor networks. To handle the load variations in some real-time sensor applications, ACMAC introduces the adaptive duty cycle scheme within the framework of sensor-MAC (S-MAC). The novelty of our protocol is that it improves latency and throughput under a wide range of traffic loads while remaining as energy-efficient as S-MAC. We illustrate such optimized trade-offs of AC-MAC via extensive simulations performed over wireless sensor networks. Our simulation results show that AC-MAC is as energy-efficient as S-MAC while its latency and throughput are always trying to follow the classic IEEE 802.11 MAC (no duty cycle), which outperform the S-MAC (fixed duty cycle), specially under the heavy load.

## I. INTRODUCTION

Recently the convergence of micro-electro-mechanical system (MEMS) technology, wireless communications and digital electronics has enabled the development of small size, low cost, low power sensor nodes that are capable of sensing, computing and communicating untethered in short-distances [1]. Compared with traditional sensors without communication capability, these nodes facilitate random deployment inside or close to the target with collaboration among them to fulfill tasks in a remote environment or even inaccessible terrain. These features ensure a wide range of promising applications for wireless sensor networks based on collaborative efforts of a large number of these nodes: battlefield surveillance, environment monitoring, vehicle tracking, smart environment and so on. There are some important constraints for realization of wireless sensor network applications. One of the most important considerations is the requirement for low power consumption since sensor nodes carry limited power sources and they are often expected to work in a remote or inhospitable environment without attendance during their lifetime. Therefore, protocols for wireless sensor networks must focus on energy-efficiency.

The primary consumers of energy in wireless networks are the RF communications [2], which are very costly compared to other electrical hardware functions including instruction execution and so on, the efforts of energy-minimization techniques are focused on the following approaches:

- *Coordinating the transmission range dynamically to reduce the per packet energy consumption:* This tech-

nique is termed as “power control” [3], [4]. Reduced transmission power also promotes the spatial reuse by maintaining connectivity at the lowest level; however, it lengthens the routes which can affect the end-to-end reliability of flows.

- *Reducing the volume of data that needs to be transmitted:* Instead of end-to-end traffic model in other networks, it is possible to intelligently combine or aggregate data into small set of information and make contribution towards energy savings [5], [6]. Furthermore, by combining these unreliable readings, it can produce more accurate signal by improving the common signal and reducing the noise.
- *Powering down the nodes' antennas during times of inactivity:* Such inactive states can be explored in different ways: from the space perspective, there are topology control techniques [7], [8], which trade network density for energy savings while preserving the data forwarding capacity of the network. Furthermore, by alleviating the restriction of network capacity preservation, [9] trades off the extensive energy savings for an increased latency to set up a multi-hop path; from the time perspective, there are TDMA-based schemes, which allow nodes to go to sleep or standby during particular time-slots of a predetermined schedule [10]. Moreover, some variant techniques ([11], [12], [13], [14]), combined with TDMA-based scheduling, enable nodes to shut down their antennas when they are not involved in any traffic in the following slotted time periods.

The above up-to-date designs for energy-efficient protocols in wireless sensor networks reveal that to be *adaptive* is the key to achieve efficient usage of energy, which usually has two forms:

- 1) *Topology adaptive:* It employs the minimum number of nodes and power levels to construct the communication system which consumes the lowest energy.
- 2) *Traffic adaptive:* It requires the communication system to be available only when there is traffic.

Furthermore, these two different mechanisms are orthogonal and can be combined to achieve the energy savings to the largest extent.

In this paper, focusing on the MAC layer, we only explore

the latter principle to present AC-MAC protocol for wireless sensor networks. Following the basic periodic listen and sleep scheduling proposed by S-MAC [13], AC-MAC introduces adaptive duty cycle technique to adapt its behavior to the traffic loads. Compared to its predecessor S-MAC, which achieves energy-efficiency by sacrificing other important attributes, such as throughput and latency and so on, AC-MAC is designed to provide more optimized trade-offs among energy, throughput and latency. Our simulation results also validate that it is at least as energy-efficient as S-MAC while offering the changing latency and throughput under various traffic loads. Such a traffic-adaptive attribution is very useful to some real-time sensor applications, such as target-tracking, in which the amount of instantaneous traffic load depends on the speed of the moving target. It is better to meet the energy-efficiency, latency as well as throughput simultaneously under the requirement of acquiring such dynamic tracking information accurately and timely in some scenarios.

The remainder of the paper is organized as follows. Section II describes the detailed design of AC-MAC protocol. Section III presents the simulation study which contains simulation metrics and simulation results. Section IV reviews related work in the literature with reference to energy-efficiency schemes at the MAC layer for wireless sensor networks. Finally, section V concludes the paper and outlines directions for ongoing research.

## II. AC-MAC PROTOCOL DESIGN

### A. Motivation

Energy-efficiency is the first design consideration for wireless sensor networks. Narrowed down into MAC layer, the efforts for efficient usage of energy are mainly focused on turning down the radios of nodes as much as possible to reduce the energy waste for their idle states. The reasons are as follows:

- 1) An idle receive circuit can consume almost as much energy as an active transmitter under current wireless interface design.
- 2) The traffic model of a wireless sensor network largely depends on its application. However, as to a quite representative application in wireless sensor networks, event tracking, which has widespread use in applications such as security surveillance and wildlife habitat monitoring, nodes will remain largely inactive for long period of time but then suddenly become active when something is detected.

S-MAC [13] is one of the early works in this area. As a signal-frequency contention-based protocol, it inherits the good scalability from the family of CSMA MAC protocols. The synchronization between the sender and the receiver, viewed as the first level, is achieved by the regular RTS/CTS mechanism; to reduce the energy wasted by idle listening, S-MAC introduces the periodic listen and sleep scheduling, the second level synchronization, by virtual clustering technique during the startup phase. Therefore, nodes will only be active



Fig. 1. The periodic listen and sleep cycle for S-MAC.

in relatively short listen intervals for communication while be inactive in relatively long sleep intervals for energy saving. However, the fixed parameters for S-MAC, such as listen interval, sleep interval and so on, provide a limited adaptability to a range of traffic loads, since throughput is limited by the fraction of the listen interval and latency may be increased if a message arrives during the sleep interval. As a result, these parameters must be carefully tuned for a specific application, otherwise it will incur large latency and low throughput which may not be tolerated by the application despite of efficient energy consumption.

As an optimization option for S-MAC, *Adaptive Listening* [14] is also proposed by the same authors to reduce latency at the price of some energy consumption. Although one half reduced latency is achieved theoretically by providing one more communication chance within a frame which is composed by a listen interval and the following sleep interval, it is still limited by its inflexibility infrastructure.

Thus, take S-MAC as a basis, the design goal of AC-MAC is to provide a traffic-adaptive framework for energy-efficient MAC protocol in wireless sensor networks, in which a third-level synchronization depending on traffic conditions will be introduced.

### B. Synchronization and Virtual Clustering

Adopting the framework of periodic listen and sleep scheduling from S-MAC, AC-MAC inherits its scalability and energy-efficiency. In the following, synchronization and the scheme to achieve such synchronization during the startup phase are described respectively.

Once synchronization is achieved among nodes, time is divided into a fairly large number of frames. Every frame has two intervals: a listen interval and a sleep interval. During the sleep interval, most of the nodes turn off their radios to preserve energy, while few nodes keep awake to continue the data transmission negotiated in the previous listen interval until they finish. During the listen interval, all nodes wake up and contend for the shared media. Once seized the channel, the nodes start the transmissions which usually span to the next coming sleep interval while most of the other nodes turn off their radios due to the collision avoidance mechanism by receiving the RTS or CTS packet not targeted for them. The scheme of such periodic listen and sleep is shown in Figure 1.

In addition, the listen interval can be further divided into SYNC interval and RTS/CTS intervals. SYNC interval is used for maintaining the synchronization among nodes by exchanging SYNC packets, and RTS/CTS interval is used for medium sensing and collision avoidance. In summary, as seen

from the above Figure 1, the timing relation among various intervals can be formalized as follows:

$$T_{listen} = T_{SYNC} + T_{RTS/CTS} \quad (1)$$

$$T_{frame} = T_{listen} + T_{sleep} \quad (2)$$

and the duty cycle which can indicate the percentage of energy saving is defined as:

$$D = \frac{T_{listen}}{T_{frame}} \quad (3)$$

However, it must be noted that before entering into the periodic listen and sleep cycles, it needs time for the startup phase. Such a synchronization scheme can be termed as *virtual clustering technique*, which urges nodes to form clusters with the same schedule without enforcing this schedule to all nodes in the network. The algorithm is executed in the following way:

When a node comes to life, it starts listening while waiting. If it does not hear anything for a certain amount of time, it chooses a frame schedule and transmits a SYNC packet, which contains the time until the next frame starts. If the node hears a SYNC packet from another node during the startup period, it follows the schedule in that SYNC packet and transmits its own SYNC accordingly.

Nodes retransmit their SYNC packets after some time durations. Nodes must also listen for a complete frame sporadically, so that they can detect the existence of different schedules. This allows new and mobile nodes to adapt to an existing group. If a node has a schedule and hears a SYNC with a different schedule from another node, it must adopt both schedules. It must also transmit a SYNC with its own schedule to the other node and let the other node know about the presence of another schedule. Adopting both schedules means that the node will have an activation event at the start of both frames.

Nodes must start a data transmission only at the start of their own active time. During that time, both neighbors with the same schedule, and neighbors that have adopted the schedule as extra are awake. If a node would start transmission at the start of a neighbor's frame, it might be transmitting to another sleeping neighbor. Note that this scheme makes it possible that broadcasts only need to be transmitted once.

### C. Adaptive Duty Cycle

Dynamically coordinating the duty cycles is the key scheme of AC-MAC in order to be adaptive to a wide range of traffic loads. The premise is not to violate the established listen and sleep scheduling among nodes since the realization of synchronization among nodes needs a very high cost during the initial startup phase. Currently, AC-MAC uses the number of packets queued at the MAC layer as an indication of the traffic load. Based on the traffic information, AC-MAC can provide an adaptive number of chances for communications

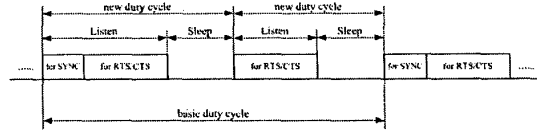


Fig. 2. The periodic listen and sleep cycle for AC-MAC. For reason of possible high traffic, AC-MAC provides 2 chances of communication within corresponding one cycle interval of S-MAC.

within one basic cycle time for S-MAC. The scheme of adaptive duty cycle in AC-MAC is shown in Figure 2.

The decisions of how many reduced duty cycles needed to fill the interval of the former basic cycle are first made by individual node distributed in the following way: the number of packets queued at MAC layer for node  $i$ , say  $N_i$ , can be mapped into a right value,  $R_i$ . It can be expressed as:

$$R_i = f(N_i) \quad (4)$$

where  $f(\cdot)$  is an application-specific function which expresses the desire of a node for transmission. Each new reduced duty cycle composes a fixed RTS/CTS interval and a varying sleep interval depending on  $R_i$ . Thus, if we know  $R_i$ , the reduced duty cycle can be computed as:

$$\begin{aligned} \hat{T}_{frame} &= T_{RTS/CTS} + \hat{T}_{sleep} \\ &= \frac{T_{RTS/CTS} + T_{sleep}}{R_i} \end{aligned} \quad (5)$$

The range of the value of  $R_i$  is limited by the timing relation within one cycle time. Obviously, the lower limit of  $R_i$  should be 1 to enable AC-MAC to keep the same behavior as S-MAC. As for the upper limit of  $R_i$ , with the increase of the value  $R_i$ , the sleep interval in a reduced cycle time is decreasing since the RTS/CTS interval is fixed. Without collision between the data transmission and the next following RTS/CTS interval, the shortest sleep interval should cover the maximum time for data transmission. If the time for data transmission during one cycle time is no more than  $T_{data}$ , the upper limit of  $R_i$  can be derived as follows:

$$\begin{aligned} T_{frame} &= (T_{RTS/CTS} + \hat{T}_{sleep})R_{i,max} + T_{SYNC} \\ &= (T_{RTS/CTS} + T_{data})R_{i,max} + T_{SYNC} \end{aligned} \quad (6)$$

Therefore, the upper limit of  $R_{i,max}$  is:

$$R_{i,max} = \frac{T_{RTS/CTS} + T_{sleep}}{T_{RTS/CTS} + T_{data}} \quad (7)$$

In order to keep the basic schedule framework of S-MAC, the  $R_i$  value is determined only at every beginning of the basic cycle time by every node  $i$ . Next, node will try to announce its  $R$  values via RTS/CTS packet. It is natural to let the node with the biggest  $R$  value occupy the transmission media for the optimized latency and throughput metrics. This can be done

TABLE I  
PARAMETERS OF ACMAC IMPLEMENTATION ON NS-2

Radio bandwidth	20kbps
Control packet length	10 bytes
Data packet length	up to 250 bytes
MAC header length	10 bytes
Contention window for SYNC (SYNC.CW)	31 slots
Contention window for data (DATA.CW)	63 slots
Slot Time	1ms
DIFS	10ms
SIFS	5ms
Transmission power	36mW
Receiving power	14.4mW
Sleep power	15μW
$f(N_i)$ for node $i$	$\min(N_i, R_{i,max})$
$g(R_i)$ for node $i$	$DATA\_CW - 3 * R_i$

by setting a smaller contention window ( $cw$ ) for the node with the bigger  $R$  value:

$$cw_i = g(R_i) \quad (8)$$

where  $g(\cdot)$  is monotonously decreasing function and is also application-specific. However, such a mechanism will increase the probabilities of collisions when there is a heavy load of traffic among the local nodes. As a result, there should be a trade-off by choosing an appropriate  $g(\cdot)$  depending on the network topology information.

In the following, when one node  $i$  has seized the chance to send out its RTS packet, it will append the  $R_i$  value to the RTS packet, so each node within its one-hop away may learn the  $R_i$  value by equation (5). As for the intended receiver, upon receiving the RTS packet, it will append the  $R_i$  value in its CTS packet to propagate such new duty cycle to the nodes within its one-hop away. It should be noted that one node will accept only one  $R$  value within one basic cycle time. Thus, all the nodes, either within one-hop away of the transmitter or the receiver, may follow the same new duty cycle within one basic cycle time for S-MAC. Therefore, the adaptive duty cycle algorithm is performed on per basic cycle time basis.

### III. SIMULATION STUDY

The purpose of our simulation is to demonstrate the effectiveness of the proposed AC-MAC protocol, measure its performance, and compare it with its predecessor S-MAC as well as the IEEE 802.11 MAC protocol which obtains the best latency and throughput performance metrics; however, without energy-efficient design goals in mind.

The implementations of both S-MAC and IEEE 802.11 have already been included in the newest version of ns-2 [15] distribution. For convenience of comparing the three protocols, we develop the AC-MAC in the same simulation platform. The important simulation parameters used are presented in the table I. Moreover, there is no appropriate routing protocol implemented in current ns-2 version to cooperate S-MAC and our AC-MAC for a wireless sensor network. Thus, we also develop an "IdeaAgent" for routing, which can compute the optimized routing based on the global information. Thus, our

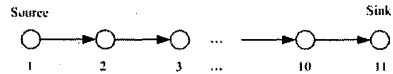


Fig. 3. Topology: ten-hop linear network with one source and one sink.

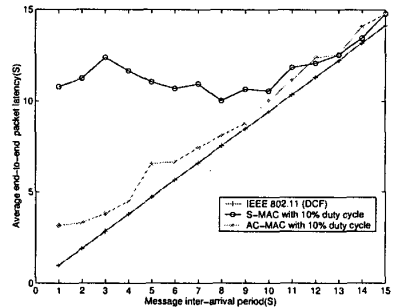


Fig. 4. Average end-to-end message latency over 10-hop network.

simulation results directly reflect the performance metrics of the MAC layer.

Our simulation model is described as follows. For comparison purpose, we use the same topology which was used to evaluate the performance of S-MAC in [14]. It is a linear network with eleven nodes, as shown in Figure 3. The nodes are configured in the way that it can be reached only by its neighbor nodes. The first node is the source and the last node is the sink.

We vary the traffic load by changing the packet inter-arrival time on the source node. The packet inter-arrival time changes from 1s to 15s. Under each traffic condition, the simulation is independently carried out for 10 times. In each simulation, the source node sends 50 messages each with 50 bytes and there is no fragmentation on any messages.

In our simulations, three different MAC protocols, the IEEE 802.11, the S-MAC with 10% duty cycle, and the AC-MAC with 10% duty cycle are compared. Since the periodic listen time is 160ms and 10% duty cycle corresponds to a frame length of 1.60s.

#### A. Measurement of End-to-End Latency

Since S-MAC trades off latency for energy-savings, it is expected to have very long latency in such multihop network scenario due to the periodic sleep on each node. By introducing the adaptive duty cycle scheme, the sleep delay for a sender waiting for the wakeup of a receiver will be largely reduced at each hop in AC-MAC, so that the end-to-end latency is decreased. As for IEEE 802.11 MAC protocol, without periodic sleep, the end-to-end latency is mainly contributed by carrier sense delay, backoff delay and transmission delay which are inherent in contention-based MAC protocols. Figure 4 shows the measurement of average end-to-end message latency.

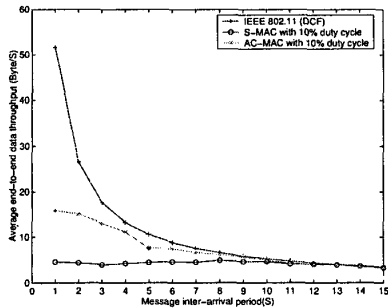


Fig. 5. Average end-to-end throughput over 10-hop network.

It can be seen that the end-to-end latency of IEEE 802.11 MAC protocol increases linearly with the decrease of traffic load, since the message delivery delay always dominates the end-to-end latency in these scenarios; as to S-MAC, in high traffic load where the packet inter-arrival time is less than 10s, its end-to-end latency does not change obviously and remains large compared to the other two protocols, whereas in low traffic loads where the packet inter-arrival time is larger than 10s, the latency smoothly increases and approaches that of IEEE 802.11 MAC protocol with the decrease of traffic load; as to AC-MAC, due to the adaptive duty cycle scheme, its end-to-end latency always follows the IEEE 802.11 MAC protocol and a bit larger than it under a wide range of traffic loads.

#### B. Measurement of End-to-End Throughput

Since S-MAC trades off throughput for energy-savings, it is expected to have very low throughput in such multihop network scenario due to very limited chances of data transmission for nodes. By introducing the adaptive duty cycle scheme, AC-MAC can provide multiple chances for data transmission within the basic cycle time for S-MAC depending on the load of traffic, so that the end-to-end throughput is increased. As for IEEE 802.11 MAC protocol without periodic sleep, data transmissions are performed on contention-based. Figure 5 shows the measurement of the average end-to-end throughput.

It can be seen that the throughput does not change much within a wide range of traffic loads since the throughput is restricted by its fixed duty cycle whereas the throughput of AC-MAC and IEEE 802.11 MAC protocol keeps the track of the message delivery rate.

To further compare AC-MAC and S-MAC, we find that in high traffic loads where the packet inter-arrival time is less than 10s, the messages are queued in the MAC layer. Due to the adaptive duty cycle scheme, AC-MAC achieves much higher end-to-end throughput than S-MAC. In low traffic loads where the packet inter-arrival time is larger than 10s, the messages are not queued and the adaptive duty cycle scheme will not be performed. In this case, AC-MAC obtains almost the same throughput as S-MAC.

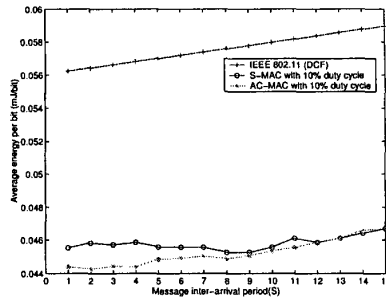


Fig. 6. Average energy consumed per bits over 10-hop network.

#### C. Measurement of Energy Consumption

We use EPB (energy per useful bit) to measure the energy consumption. The actual time is counted from the transmission of the first useful bit.

Due to the periodic listen and sleep scheduling adopted by both S-MAC and AC-MAC, EPB of these two MAC protocols are obviously less than that of IEEE 802.11 MAC protocol which is designed without energy-efficiency consideration. From Figure 6, it can be seen that EPB for all MAC protocols is increasing with the decreasing of traffic load since they spend more time for idle listening when the arrival of the messages are less frequent.

Furthermore, AC-MAC is almost as energy-efficient as S-MAC in terms of EPB. In high traffic load where the packet inter-arrival time is less than 10s, EPB for AC-MAC is even slightly lower than S-MAC. It can be explained as multiple data transmissions in a basic cycle time in AC-MAC save more overhead of SYNC packets than only one data transmission in one cycle time in S-MAC. In low traffic loads where the packet inter-arrival time is larger than 10s, EPB for AC-MAC is almost the same as that of S-MAC since the  $R$  value in AC-MAC will always be 0.

#### IV. RELATED WORK

MAC protocols for wireless sensor networks are designed to be energy-efficient which is determined by the restricted resources of sensor nodes. The basic scheme is to enable nodes which are not involved in the communication to keep their radios off as long as possible thereby to meet the need of energy constrained environments. Therefore, a highly efficient *synchronization* scheme between the communication pairs becomes critical. It assures the nodes to be synchronized as soon as possible once they wake up for the communication purpose.

Bluetooth [16] is a TDMA-based MAC protocol. Its TDMA mechanism has a natural advantage of energy conservation compared to contention-based protocols. Since the duty cycle of the radio is reduced and there is no contention-introduced overhead and collisions. Synchronization is easily achieved based on reservation and scheduling. However, using TDMA

protocol usually requires the nodes to form real communication clusters. Moreover, when the number of nodes within a cluster changes, it is not trivial for a TDMA protocol to dynamically change its frame length and time slot assignment. Thus normally its scalability is not as good as that of a contention-based protocol.

TRMAC [12] is a MAC protocol combined with scheduling and contention in which time is organized as sections of random and scheduled access periods. Time synchronization has been achieved in random access periods. During these periods, nodes also obtain the consistent two-hop topology information by Neighbor Protocol (NP). Upon that, in the following scheduled access periods, transmission schedules for a node are pre-computed and announced by Schedule Exchange Protocol (SEP) with the aid of Adaptive Election Algorithm (AEA), including its intended receivers and the associated time slots. Thus, transmissions will be scheduled accordingly.

S-MAC [13], the most relevant protocol to our approach, also adopts a hybrid scheme, however the medium reservation is different from TRMAC. Synchronization scheme, termed as virtual clustering, forms common schedules to enable nodes to go to sleep periodically after the initial startup phase.

T-MAC [11] adopts the same scheme synchronization as S-MAC. To further decrease energy wasted on idle listening, in which nodes wait for potentially incoming messages, it also introduces the adaptive duty cycle in another way in which it dynamically ends the listen interval to handle load variations in time and location. This scheme achieves a better energy-efficiency than S-MAC while still maintaining a reasonable throughput. However, it introduces another "early sleeping" problem which limits its latency and throughput.

#### V. CONCLUSIONS AND FUTURE WORK

To improve the trade-offs among energy, latency and throughput of S-MAC, an energy-efficient MAC protocol, AC-MAC, is explicitly designed for wireless sensor networks. AC-MAC introduces the adaptive duty cycle technique which depends on the different loads of traffic. Simulation results show that AC-MAC achieves considerably better latency and throughput in a wide range of traffic loads compared to S-MAC while maintaining the same level of energy-efficiency as S-MAC.

In addition to conducting a more comprehensive quantitative study of the proposed approach, we plan to address the following issues as part of our future work:

- 1) The virtual clustering technique, proposed in the S-MAC protocol and reused in the AC-MAC protocol, has not been researched thoroughly. Since synchronization of schedules can have a great impact on the energy consumption, clustering is essential.
- 2) With the increase of the node density, the probability of collision in RTS/CTS interval will increase due to the fixed and small contention windows. Thus, all the performance metrics will be worsened in both S-MAC and AC-MAC. It is expected that suitable topology

adaptive mechanisms can be incorporated to lessen the above problem.

- 3) Differentiated services for different types of messages in terms of latency and/or throughput, are possible to be developed according to the application-specific function  $f(\cdot)$  mentioned in equation (4).

#### REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey, Computer Networks, Vol. 38, No. 4, pp. 393-422, March 2002
- [2] L. Feeney and M. Nilsson, "Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment", In Proceedings of the IEEE INFOCOM, Vol. 5, No. 8, Anchorage, AK, Arp. 2001
- [3] A. Sheth, R. Han, "Adaptive Power Control and Selective Radio Activation For Low-Power Infrastructure-Mode 802.11 LANs", IEEE Workshop on Mobile and Wireless Networks (MWN), 2003, pp. 812-818. Held in conjunction with ICDCS 2003, Providence, Rhode Island
- [4] R. R. Kompella, A. C. Snoeren, "Practical Lazy scheduling in Sensor Networks", In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, Nov. 2003
- [5] C. Intanagonwivat, D. Estrin, R. Govindan, and J. Heidemann, "Impact of network density on data aggregation in wireless sensor networks", In Proceedings of International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria, July 2002
- [6] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong, "TAG: a Tiny AGgregation Service for Ad-Hoc Sensor Networks", In Proceedings of the ACM Symposium on Operating System Design and Implementation (OSDI), Dec. 2002
- [7] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris, "Span: An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks", In Proceedings of the 6th ACM MOBI-COM, Rome, Italy, July 2001
- [8] Y. Xu, J. Heidemann, and D. Estrin, "Geography-informed Energy Conservation for Ad Hoc Routing", In Proceedings of the 6th ACM MOBI-COM, Rome, Italy, July 2001
- [9] C. Schurgers, V. Tsatsis, and M. Srivastava, "STEM: Topology management for energy efficient sensor networks", In IEEE Aerospace Conference, pp. 78-89, March, 2002
- [10] B. Tavli, W. B. Heinzelman, "MH-TRACE: Multi-Hop Time Reservation Using Adaptive Control for Energy Efficiency", IEEE Military Communications Conference (MILCOM 2003), Boston, MA, Oct. 2003
- [11] T. V. Dam, and K. Langendoen, "An Adaptive Energy-Efficient MAC Protocol for Wireless Sensor Networks", In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, Nov. 2003
- [12] V. Rajendran, K. Obraczka, and J.J. Garcia-Luna-Aceves, "Energy-Efficient, Collision-Free Medium Access Control for Wireless Sensor Networks", In Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys), Los Angeles, CA, Nov. 2003
- [13] W. Ye, J. Heidemann and D. Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks", In Proceedings of the 21st International Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '02), New York, June 2002
- [14] W. Ye, J. Heidemann and D. Estrin, "Medium Access Control with Coordinated, Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Transaction on Networking, 2003
- [15] The Network Simulator - ns-2, <http://www.isi.edu/nsnam/ns/>
- [16] J. C. Haartsen, "The Bluetooth radio system", IEEE Personal Communications Magazine, pp. 28-36, Feb. 2000