ADV-MAC: Advertisement-based MAC Protocol for Wireless Sensor Networks

Surjya Ray, Ilker Demirkol and Wendi Heinzelman
Department of Electrical and Computer Engineering
University of Rochester, Rochester, New York
Email: {ray, demirkol, wheinzel}@ece.rochester.edu

Abstract—**Several Medium Access Control (MAC) protocols have been proposed for wireless sensor networks with the objective of minimizing energy consumption. For example, Sensor-MAC (S-MAC) was proposed to reduce energy consumption by introducing a duty cycle. Under variable traffic load, the fixed duty cycle of S-MAC results in energy loss due to idle listening along with higher latency and lower throughput compared to an adaptive duty cycle. Timeout-MAC (T-MAC) introduced such an adaptive duty cycle to handle variable traffic loads. However, nodes that do not take part in data exchange waste energy because of continuous renewal of their timeout values. In this paper we propose Advertisement MAC (ADV-MAC), a MAC protocol for wireless sensor networks that eliminates this energy waste by introducing the concept of advertising for data contention. ADV-MAC minimizes the energy lost in idle listening while maintaining an adaptive duty cycle to handle variable loads. Additionally, ADV-MAC enables energy efficient MAC-level multicasting. We provide detailed comparisons of the ADV-MAC protocol with S-MAC and T-MAC through extensive simulations. The simulation results show that ADV-MAC efficiently handles variable load situations and provides substantial gains over S-MAC and T-MAC in terms of energy (reduction of up to 45%) while faring as well as T-MAC in terms of throughput and latency.**

I. INTRODUCTION

Wireless sensor networks (WSNs) are used for a wide variety of purposes ranging from military applications such as border surveillance to civilian applications such as environment and habitat monitoring, health care applications and traffic control. WSNs consist of a large number of nodes that organize themselves into a multi-hop wireless network. These nodes are battery operated, have embedded memory and a processor and are fitted with one or more sensors for monitoring the environment. WSNs generally lack any fixed infrastructure, and nodes are often deployed in hostile or inaccessible environments, making it impractical to change exhausted batteries. As a result, conserving battery power is the main objective for such networks in order to maximize node lifetime. Latency, throughput and fairness are other important objectives.

Designing efficient MAC protocols for wireless sensor networks is challenging because the sensor nodes generally have limited battery life, memory capacity and processing capability. Additionally, the topology of a wireless sensor network often changes randomly as nodes die out and new nodes are added. The MAC protocol should easily accommodate such network dynamics.

The major cause of energy waste in conventional MAC protocols is *idle listening*. In MAC protocols such as IEEE 802.11 [1], the nodes must keep listening to the channel because they do not know when they might receive a message. Measurements have shown that energy consumed in idle listening is comparable to that consumed in receiving or transmitting. For example, the receive:send:idle power ratios on MicaZ motes are 1.13:1:1.13 [2] while the same ratios for Tmote Sky motes are 1.11:1:1.11 [3]. In many sensor networks, messages are sent in bursts when an event is sensed. At all other times, most nodes have no data to send. If a conventional MAC protocol is used in sensor networks, a very large part of the energy will be lost in idle listening.

This paper presents Advertisement-MAC (ADV-MAC), a MAC protocol designed to minimize energy wasted in idle listening. ADV-MAC reduces energy waste by utilizing a combined advertising and contention scheme. Simulations show that ADV-MAC performs very well in highly loaded and variable loaded networks, with substantial energy savings as high as 45% compared to the two well known MAC protocols, Sensor-MAC (S-MAC) [4] and Timeout-MAC (T-MAC) [5] without degrading the performance in terms of latency and throughput. Also, ADV-MAC introduces an energy efficient data-centric MAC-level multicasting scheme where a node can send data to a subset of its neighbors. Such MAC-level multicasting is not possible in S-MAC and T-MAC, which must resort to broadcasting.

The rest of the paper is organized as follows. In Section 2, we present a brief survey of existing energy-conserving MAC protocols. In Section 3, we describe the design of the ADV-MAC protocol and compare ADV-MAC to both S-MAC and T-MAC qualitatively. In Section 4, we describe our simulation setup, followed by a detailed discussion of the results. Finally in Section 5, we conclude the paper.

II. RELATED WORK

A wide range of MAC protocols have been proposed for wireless sensor networks, and they can be categorized as contention-based MAC protocols and TDMA-based MAC protocols. Contention-based protocols are widely employed

This work was supported in part by NSF CAREER grant #CNS-0448046 and in part by a Young Investigator grant from the Office of Naval Research, #N00014-05-1-0626.

because of their simplicity, robustness and flexibility. These protocols require little or no clock synchronization and no global topology information.

TDMA (Time Division Multiple Access) protocols are based on reservation [6] and scheduling [7], [8]. In this type of protocol, every node will eventually be able to transmit its data. This is accomplished by reserving slots for the nodes. Thus, TDMA protocols prevent collisions and have bounds on end-to-end delay. TDMA-based protocols are very energy efficient when the network is highly loaded and all slots are used. LEACH [9], TRAMA [10] and Bluetooth [11] are examples of TDMA-based MAC protocols. However, TDMA-based protocols suffer from synchronization issues. The nodes in TDMA-based protocols need to be synchronized, forming communication clusters. It is not easy to maintain clock synchronization among the nodes, especially in a large network. Also, when new nodes join the network or nodes leave the network, it is not easy to dynamically change the frame length and slot assignment in a cluster. Furthermore, developing an efficient schedule with a high degree of channel reuse is difficult, and the solution is NP-hard [12].

Certain MAC protocols introduce a sleep-listen schedule into contention-based (CSMA) protocols to save energy wasted in idle listening. The nodes in such protocols go to a low power sleep state whenever possible to save energy. Two well known protocols in this category are S-MAC [4] and T-MAC [5]. The basic idea of S-MAC protocol is to divide time into frames. Every frame has two parts: an active part and a sleeping part. During the active part, nodes can communicate with their neighbors. The method of communication is similar to that of IEEE 802.11, utilizing RTS, CTS, DATA and ACK packets. Only one packet can be sent in each frame in S-MAC. The node that sends out an RTS packet before any other nodes wins the medium and starts its data exchange. Other contending nodes must wait until the next frame to contend for the medium again. The duty cycle, i.e., the proportion of listen time within a frame, is set based on factors such as network density and message rate. Its value is decided before the network is set up and is static.

Although S-MAC reduces the idle listening time, a solution with a fixed duty cycle is not optimal. S-MAC basically trades energy for throughput and latency. While a low duty cycle reduces idle listening time, it results in high latency and low throughput in medium to high traffic conditions as only one data packet transmission can occur in each frame. On the other hand, if duty cycle is high, throughput and latency performance improves at the expense of reduced energy savings.

T-MAC [5] was proposed to improve the poor performance of S-MAC under variable loads. In T-MAC, the listen period ends when no activation event, such as sensing of any communication on the radio, has occurred in the channel for a time threshold TA. Nodes will keep on renewing their timeout values whenever an activation event occurs. When none of these events occur for a duration of a timeout period, the nodes go to sleep. Thus, TA determines the minimal amount of idle listening per frame. The lower limit on the length of the TA interval is $C + R + T$, where C is the length of the contention interval, R is the length of an RTS packet, and T is the turn-around time, i.e., the time between the end of reception of an RTS packet and the beginning of reception of the corresponding CTS packet. Receiving the start of the RTS or CTS packet from a neighbor is enough to trigger a renewed TA interval. In S-MAC, nodes go to sleep after overhearing an RTS or CTS destined for another node, which is called overhearing avoidance. If overhearing avoidance is used in T-MAC, collision overhead becomes higher since a node may miss RTS and CTS packets while sleeping and disturb communication when it wakes up. As a result, though overhearing avoidance saves energy, it is kept only as an option in the T-MAC protocol.

The TA timeouts make the active period in T-MAC adaptive to variable traffic loads. Thus, T-MAC can balance the energy-throughput-latency trade-off better than S-MAC under a variety of traffic conditions. However, whenever an activation event occurs, all nodes that hear the event renew their TA timer even if they are not a part of the transmission. As a result, nodes still end up wasting valuable energy. As we will show, ADV-MAC proposes a better way to conserve energy and at the same time to support variable traffic loads.

Another group of MAC protocols can be classified as low power listening (LPL) MAC protocols. These protocols are asynchronous and rely on preamble sampling for data transmission. Examples of such protocols are X-MAC [13] and B-MAC [14]. Although these protocols are very energy efficient, they are mostly suited for very low traffic loads. At high and variable traffic, because of long preambles, their throughput decreases and latency increases. Performance of such protocols reduces significantly when the actual neighborhood or traffic is different from the ideal model, especially when traffic rates vary greatly [15].

III. ADV-MAC DESIGN OVERVIEW

Reducing energy consumption is the main objective of ADV-MAC. The protocol aims to reduce energy wasted in idle listening as much as possible. In ADV-MAC, all nodes that are not part of any current data exchanges are put to sleep, saving valuable energy. Only nodes that are part of a data transmission stay awake for the contention.

A. Basic Operation of ADV-MAC

Fig. 1 shows the basic principle of ADV-MAC along with a comparison to S-MAC and T-MAC. The figure shows nodes A through G, all of which are in transmission range of each other. Node A and node C have data for node B and node D, respectively. Only the active times of the nodes are shown. All three protocols start their active times with a SYNC period which is used for synchronization and virtual clustering of the nodes as in [4]. Each frame in S-MAC consists of a fixed-length SYNC period, a fixed-length data period and a sleep period that depends on the duty cycle. T-MAC also has a fixed-length SYNC period, but the length of the data period and the length of the sleep period both depend on the local traffic

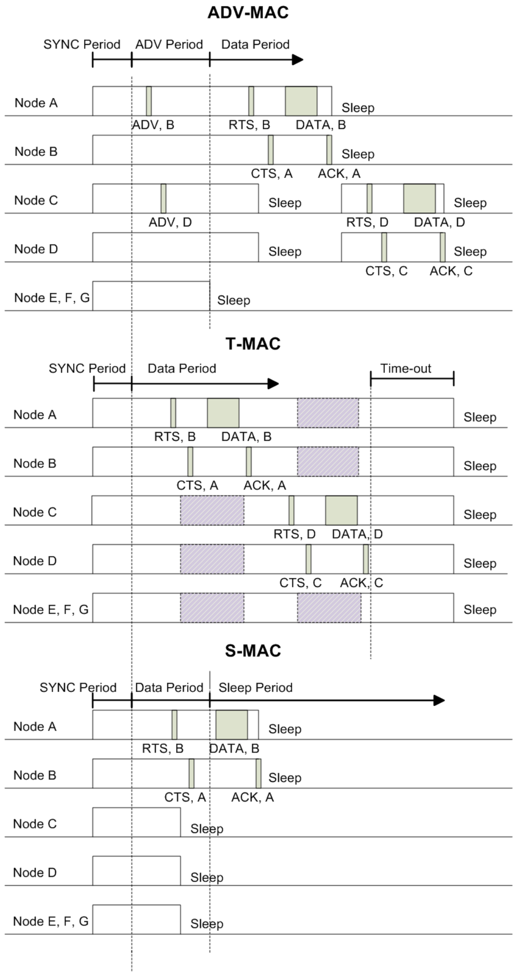


Fig. 1. Examples of ADV-MAC, T-MAC and S-MAC communication. Overhearing avoidance is optional for T-MAC, and it is shown by the hatched areas. The letters after the packets indicate the destination nodes.

conditions. While S-MAC and T-MAC begin their data period after the SYNC period, ADV-MAC has another small period called Advertisement period before the data period. The length of this advertisement period is just enough to transmit a few Advertisement (ADV) packets, which contain the ID of the intended receivers. ADV-MAC thus has a fixed-length SYNC period and a fixed-length Advertisement period, followed by a variable-length data period and a variable-length sleep period. It should be noted that while the data and sleep periods are variable in both ADV-MAC and T-MAC, the total frame time is fixed. Also, unlike S-MAC, ADV-MAC does not have a fixed duty cycle. Depending on the average traffic load, we can fix the total frame length as well as the length of the ADV period.

The advertisement time is divided into several slots. At the beginning of the advertisement time, if a node has any data to send, it randomly picks a slot and starts to listen to the channel until its slot time arrives. If there is no ADV transmission going on when its slot time arrives, it transmits its ADV packet. Note that other nodes may have completed their ADV transmission before this slot which enables multiple ADV transmissions in an ADV period. If the node senses a busy channel when its slot time arrives, it waits until the transmission is over, and then chooses a new random slot from the remaining slots and starts to listen to the channel again. The node will continue to do this until it is successful in transmitting the ADV packet or until the advertisement time has ended.

If the ADV packet is received by its intended receiver, that node will be aware that there is data pending for it. Thus, after the end of the ADV period, only the nodes that sent ADV packets and the intended receivers who successfully received the ADV packets will be awake for the data time. The remaining nodes go to sleep. ADV packets do not require any reply back as RTS packets do, and thus transmitters will not know if ADV collision occurred. In case of an ADV collision, the nodes whose packets collided will not know of their collision and will be awake while their intended receivers will be asleep.

After the ADV period, nodes that sent ADV packets will contend for the medium by listening to the medium for a random amount of time from the beginning of the data period and then sending an RTS packet. The node that wins the medium completes its data exchange. Nodes can send multiple packets. Once a node has won the medium, it need not send RTS packets for all the data packets, it just sends the data packets and the receiver replies back with an ACK. Since RTS packets contain the duration of the entire exchange time, the other remaining nodes having data to send will defer until the end of the data exchange as in IEEE 802.11 [1] and go to sleep for that duration. These nodes then wake up after the data exchange is over and begin contending for the medium. The nodes whose ADV packets collided will also try to send RTS packets. However, their intended receivers will be asleep, and these nodes will eventually go to sleep after their CTS timeout.

In S-MAC, only one packet can be sent in each frame. In the example shown in Fig. 1, node A wins the contention and completes the data exchange in the regular sleep period. Node C will have to wait for the next frame to send its data. On the contrary, in ADV-MAC, multiple packets can be sent within one frame, as illustrated in Fig. 1 where both node A and node C send out ADV packets within the same frame. After the end of the ADV period, only nodes A, B, C and D stay awake for the data exchange. The other nodes go to sleep. In T-MAC, each transmission or activity in the medium renews the timeout value of all nodes. Therefore, nodes E, F and G continue renewing their timers even if they are not participating in any data exchange. Finally, all nodes time out and go to sleep.

As Fig. 1 suggests, S-MAC should have the minimum energy consumption assuming the same frame sizes for all three protocols. However, this comes at the price of low throughput and high latency, as nodes in S-MAC can only transmit one packet in each frame. In order to improve the latency and throughput, the duty cycle must be increased for S-MAC. According to the S-MAC protocol, the active time is fixed. Thus, increasing the duty cycle means that the sleep period and hence the total frame time will be shorter. Nodes in S-MAC will wake up more frequently, leading to more frames in the same time duration. Hence, nodes end up using more energy to get better throughput and latency. For T-MAC, nodes that are not required for the data exchange stay awake and waste energy. As we shall see in the simulation results, ADV-MAC provides the lowest energy consumption while achieving high throughput and low latency.

Although ADV-MAC adds a new time period after the SYNC period, the energy consumption of ADV-MAC is not greater than that of S-MAC and T-MAC even in low traffic loads. The reason is as follows. Let us consider the case of no traffic with all three protocols having the same frame length. If the data period of S-MAC, the timeout period of T-MAC and the ADV period of ADV-MAC have the same duration, the energy consumption would be the same in all cases. This is because after the SYNC period, all nodes in S-MAC will be awake for the data period, all nodes will remain awake in T-MAC until they time out and all nodes in ADV-MAC will be awake for the ADV period. In our experiments, we set the ADV period to be equal to the Timeout period given in [5] and the experimental results show that ADV-MAC gives the lowest energy consumption for that throughput and latency as compared to S-MAC and T-MAC.

The basic T-MAC protocol suffers from the so called *early sleeping problem* [5]. Suppose node A has data for node B and node A loses contention because it hears an RTS or CTS from another data exchange. If node B is out of the range of this transmission, it will eventually time out and go to sleep before node A can send its data. This will result in an increase in latency and a decrease in throughput values. Early sleeping can also happen if a receiver cannot reply back with a CTS because it hears an RTS/CTS exchange from another data exchange. The ADV-MAC protocol, however, is inherently immune to the early sleeping problem. In ADV-MAC, only the nodes that are indicated as intended receivers in ADV packets remain awake in the data part of the active time. If they overhear the data exchange between other nodes (via RTS or CTS), they just go to sleep for the duration of the data exchange and wake up again to listen to the medium. If they do not hear anything, they will still stay awake for the RTS because they have prior knowledge of data waiting for them.

B. MAC Multicasting

In S-MAC and T-MAC, broadcasting takes place without any RTS/CTS mechanism, and data packets are sent directly. There may be situations where the sources broadcast different types of data and each receiving node is interested in a particular data type. For instance, there may be nodes equipped with different sensors broadcasting individual sensor measurements as separate packets, with nodes interested in only certain types of sensor data. In this type of application, a MAC level multicasting scheme can enable significant energy savings. Since nodes in S-MAC and T-MAC have no prior knowledge of which type of data is being broadcast, all nodes receive the data being broadcast even if they are not interested in that type of data, hence losing valuable energy. In ADV-MAC, ADV packets may have a field that contains the type of the data being sent. Only nodes that are interested in those types of data will stay awake in the data period. This enables efficient single-hop multicasting at the MAC level and saves a great deal of energy.

C. Energy Consumption

The energy consumed by the three protocols can be calculated approximately for simple cases. We assume that transmission, reception and idle energy consumption values are all approximately the same, as per the MicaZ and Tmote Sky energy dissipations [2][3]. Let us consider the case of *N* nodes in a virtual cluster, all of which are within transmission range of each other.

1) S-MAC: Let p be the duty cycle and t_{sim} be the simulation time. If w is the transmission, reception or idle listening power, then the total energy consumed per node in t_{sim} seconds is calculated as

$$E_{smac} = wpt_{sim}. \tag{1}$$

This equation does not consider any collisions or any data transmission continuing into the sleep part. In the original S-MAC protocol, nodes exchange data during the sleep time. This data exchange during the sleep time results in additional energy consumption which is not captured by (1). Also there are quite a few collisions in the SYNC period, which make the nodes go to sleep, hence saving energy. This is also not considered by (1). However, these two effects basically cancel each other, and (1) provides a reasonable approximation of the energy consumption. The equation remains the same for unicast and broadcast transmissions.

2) T-MAC: To calculate the total energy consumption in T-MAC, first let us calculate the total time spent awake by all nodes in the virtual cluster. We consider T-MAC with overhearing avoidance. Let N_s be the number of sources in the network each transmitting a packet every t_r seconds. The total time spent awake in the SYNC period and the final timeout period by all the N nodes during the simulation period is

$$NN_c(t_{sync} + t_{TA}), \tag{2}$$

where N_c is the total number of cycles in the simulation time t_{sim}. The duration of SYNC and time-out periods are denoted by t_{sync} and t_{TA}. The time spent in sending and receiving the CTS and data packets is

$$2N_p(t_{control} + t_{data}), \tag{3}$$

where $t_{control}$ is the duration of control packets (RTS, CTS or ACK), N_p is the total number of packets exchanged within the simulation time t_{sim} and calculated as $N_s t_{sim}/t_r$, and t_{data} is the duration of a data packet. Since we consider overhearing avoidance, the factor 2 is used in (3) to indicate that only two nodes are awake. Without overhearing avoidance, all nodes hearing the data would be awake, and the factor would be N instead of 2. The total time spent during the course of N_c

TABLE I
PARAMETER VALUES

Parameters	Values
Total Simulation Time (t_{sim})	200 *s*
Tx / Rx / Idle Listening Power	55.8 *mW*
Transmission Rate	250 *Kbps*
Transmission, Carrier sense Range	$100, 200$ *m*
Duration of frames (t_{frame})	238.4 *ms*
Dur. of sync (t_{sync}), contention period (t_{cw})	8.4, 13 *ms*
Dur. of time-out period (t_{TA}), ADV period (t_{ADV})	15 *ms*, 15 *ms*
Dur. of control ($t_{control}$)), data packet (t_{data})	0.9, 9.5 *ms*

cycles in transmitting and receiving the control packets and waiting in contention period is

$$NN_p(2t_{control} + \bar{t}_{cw}), \tag{4}$$

where $\bar{t}_{cw}$ is the average time spent in contention by a node.

Combining (2), (3) and (4), the total time spent awake by all N nodes in a unicast scenario is found to be

$$\begin{aligned} T_{tmac} = NN_c(t_{sync} + t_{TA}) + 2N_p(t_{control} + t_{data}) \\ + NN_p(2t_{control} + \bar{t}_{cw}). \end{aligned} \tag{5}$$

For a broadcast scenario, this equation becomes

$$T_{tmac} = NN_c(t_{sync} + t_{TA}) + NN_p(t_{data} + \bar{t}_{cw}). \tag{6}$$

If w is the transmission, reception or idle listening power, then the total energy spent per node is

$$E_{tmac} = \frac{T_{tmac}w}{N}. \tag{7}$$

3) ADV-MAC: As in T-MAC, let us calculate the total time spent awake by all nodes in the virtual cluster. Let n_a be the average number of packets transmitted in each cycle, given by N_p/N_c. The total time spent awake in the SYNC period and the advertisement period by all the N nodes during the duration of the simulation is

$$NN_c(t_{sync} + t_{ADV}). \tag{8}$$

where t_{ADV} is the duration of the advertisement period. The time spent in sending and receiving the data, CTS and ACK packet is given by

$$2N_p(t_{data} + 2t_{control}). \tag{9}$$

Again, the factor 2 is used since only two nodes are awake. The time spent during the course of N_c cycles in transmitting and receiving the control packets (RTS) and waiting in the contention period is given by

$$2N_c \sum_{i=0}^{[n_a]} (n_a - i)(t_{control} + \bar{t}_{cw}). \tag{10}$$

The summation term reflects the number of nodes that are waiting in contention and then receiving or transmitting RTS and CTS packets. For example, suppose there are 3 nodes waiting to transmit within a given cycle. Then, 6 nodes will wait for $(2t_{control} + \bar{t}_{cw})$ amount of time. The node that wins the contention will stay awake with its destination node, while the other 4 nodes will go to sleep. When this data transfer is complete, these 4 nodes will wake up and again spend $(2t_{control} + \bar{t}_{cw})$ amount of time. Then two nodes will stay awake and the remaining two will go to sleep and wake up when the data exchange is over. The summation is to add up all these $(2t_{control} + \bar{t}_{cw})$ periods spent awake by the nodes. Combining (8), (9) and (10), the total time spent awake by all N nodes together for a unicast scenario is

$$\begin{aligned} T_{advmac} = NN_c(t_{sync} + t_{ADV}) + 2N_p(t_{data} + 2t_{control}) \\ + 2N_c \sum_{i=0}^{[n_a]} (n_a - i)(t_{control} + \bar{t}_{cw}). \end{aligned} \tag{11}$$

For a broadcast scenario, the above equation becomes

$$\begin{aligned} T_{advmac} = NN_c(t_{sync} + t_{ADV}) + \\ (1 + \frac{N - N_s}{n_t})N_c \sum_{i=0}^{[n_a]} (n_a - i)(t_{data} + \bar{t}_{cw}). \end{aligned} \tag{12}$$

The summation term has a coefficient of $(1 + \frac{N-N_s}{n_t})$ instead of 2, where n_t is the number of different data types. This is because for each sender there is more than one receiver. Since each receiver selectively receives the broadcast packets, for each type of data packet, there will be $\frac{N-N_s}{n_t}$ receivers. As before, the total energy spent per node is

$$E_{advmac} = \frac{T_{advmac}w}{N}. \tag{13}$$

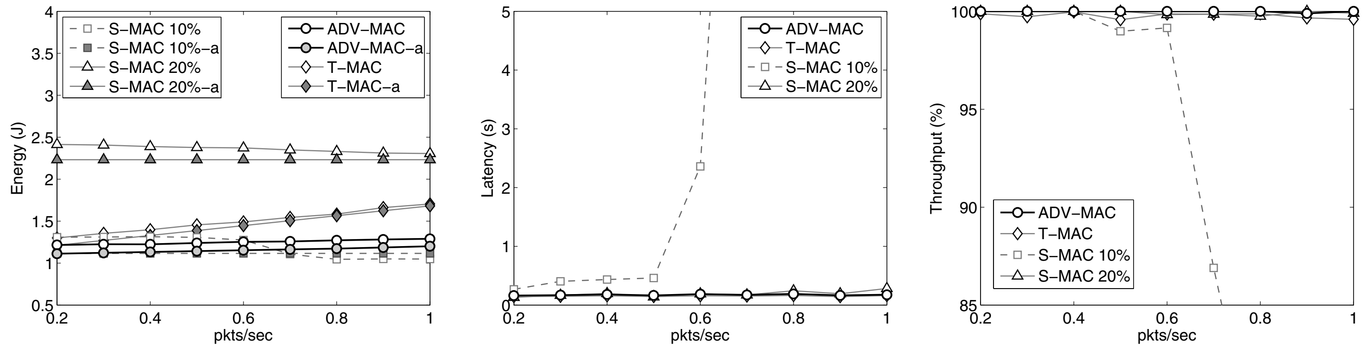
The energy consumption equations of S-MAC, T-MAC and ADV-MAC do not consider any collisions. However, if the network is not highly loaded, they provide a reasonable approximation.

IV. EXPERIMENTS

In our experiments, we compared the performance of the three protocols: S-MAC, T-MAC and ADV-MAC. We used energy consumption, throughput and latency as the three performance metrics for comparison.

A. Simulation Setup

We performed all simulations in ns 2.29 [16]. The S-MAC code is included in this version of ns. We coded T-MAC and ADV-MAC in ns-2 as well. We use three different duty cycle settings (10%, 20% and 30%) for S-MAC because one fixed duty cycle is not suitable for all traffic loads investigated. Also, we limit the data rate to 1 pkt/sec. Although ADV-MAC and T-MAC both can handle higher traffic loads, S-MAC cannot handle such high loads. The frame time for 10% duty cycle is 238.4 ms, and we set this frame time for T-MAC and ADV-MAC as well. We used a duration



(a) Energy Consumption vs. Data Rate.

(b) Latency vs. Data Rate.

(c) Throughput vs. Data Rate.

Fig. 2. Single hop, unicast vs. Data Rate: Performance comparison of ADV-MAC, T-MAC and S-MAC. The extension '-a' corresponds to analytical results.

of 15 ms for the time-out periods of T-MAC as in [5]. Since transmit, receive and idle listening have close energy consumption values [2][3], we set a common value of 55.8 mW for all three operations in accordance with [2] where the average current consumption value is 18.6 mA and a battery of 3 V is used. The transmission rate is 250 $kbps$, and the transmission range is 100 m, while the interference or carrier sense range is 200 m. The Advertisement period and the contention period are divided into slots of 0.1 ms each. All nodes in the simulations are placed randomly. We use T-MAC with over-hearing avoidance, as it is used in the simulations in [5]. Each data point shown in the results is the average of 10 simulation runs. The values used for the simulation parameters are summarized in Table I.

We used a duration of 15 ms for the Advertisement period, which is the same as the TA period TA period of T-MAC. However, since the Advertisement period is an important parameter for determining the efficiency of the ADV-MAC protocol, we performed an experiment to determine whether or not 15 ms would be acceptable. It was found that for the highest traffic load used in the simulations, the probability of ADV packet collision is 2% which is an acceptable value.

B. Results

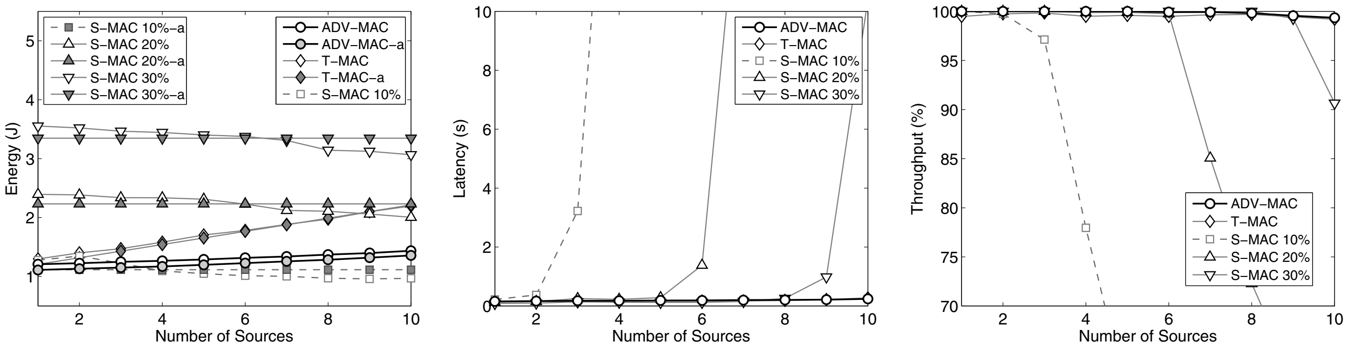
1) Effect of data rate on single hop, unicast scenario: In the first set of simulations, we investigate the effects of traffic load on energy consumption, latency and throughput. We consider an area of 50 m x 50 m with all nodes in transmission range of each other. There are 20 nodes in the area including 5 sources. The traffic load is varied by increasing the data rate from 0.2 pkt/sec to 1 pkt/sec.

Fig. 2(a) shows the energy consumptions obtained from the simulations as well as the results obtained from the energy equations (1), (7) and (13). Figs. 2(b) and 2(c) show the corresponding latency and throughput values, respectively. As seen from the figure, ADV-MAC and S-MAC with 10% duty cycle give the best energy consumption results, with ADV-MAC having slightly lower energy consumption for lower traffic loads and 10% S-MAC having slightly lower energy consumption for higher traffic loads. However, as seen from the corresponding latency and throughput values, which are shown in Figs. 4 and 5, respectively, S-MAC with such a low duty cycle actually cannot handle the high traffic loads and gives very poor throughput and very high latency results. However, ADV-MAC presents stable latency and throughput results for all traffic loads, showing its resiliency to variable data traffic loads and high traffic loads. As data rate increases beyond 0.5 pkt/sec, 10% S-MAC is no longer sufficient because of high latency and low throughput. At high data rates, 20% duty cycle gives acceptable values of latency and throughput for S-MAC. However, the energy consumption of ADV-MAC is 44% less than the energy consumption of S-MAC with 20% duty cycle at the highest data rate.

It is seen that as data rate increases, the energy consumption of ADV-MAC increases very little, but that of T-MAC increases much faster. This happens because nodes that are not a part of a data exchange can selectively go to sleep in ADV-MAC, but all nodes in the carrier sense range must be awake in T-MAC. From Fig. 2(a) we can see that the energy consumption of ADV-MAC at higher data rates is as much as 24% lower than that of T-MAC. Also, ADV-MAC has the least latency and the maximum throughput at all data rates. 20% S-MAC and T-MAC also have the least latency and maximum throughput, but their energy consumptions are much higher than ADV-MAC, as pointed out before. Thus ADV-MAC successfully adapts to traffic load, providing low energy consumption with high throughput and low latency over all traffic conditions.

2) Effect of number of sources on single hop, unicast scenario: In the second set of simulations, we investigate the effect of different numbers of sources on the performance of the three MAC protocols. The simulation setup is similar to the previous case, but we vary the number of sources from 1 to 10 and keep the data rate fixed at 1 pkt/sec for all sources.

Fig. 3(a) shows the energy consumptions for the three protocols obtained from simulations and analysis. Fig. 3(b) and Fig. 3(c) show the latency and throughput comparisons, respectively. 10% S-MAC is suitable only for 1-2 sources. 20% S-MAC gives acceptable values of latency and throughput up to 6 sources. Beyond that, 30% S-MAC is needed. T-MAC adapts successfully to increasing number of sources (i.e., increasing load) with the least latency and the maximum



(a) Energy Consumption vs. Number of Sources.

(b) Latency vs. Number of Sources.

(c) Throughput vs. Number of Sources.

Fig. 3. Single hop, unicast vs. Number of sources: Performance comparison of ADV-MAC, T-MAC and S-MAC. Extension '-a' represents analytical results.

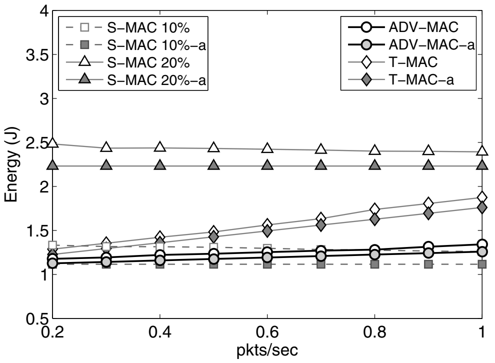


Fig. 4. Single hop, multicast : Energy Consumption vs. Data Rate. The extension '-a' corresponds to analytical results.

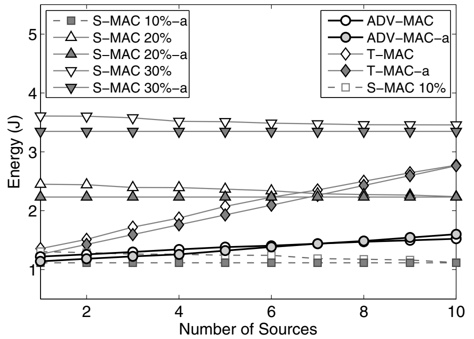
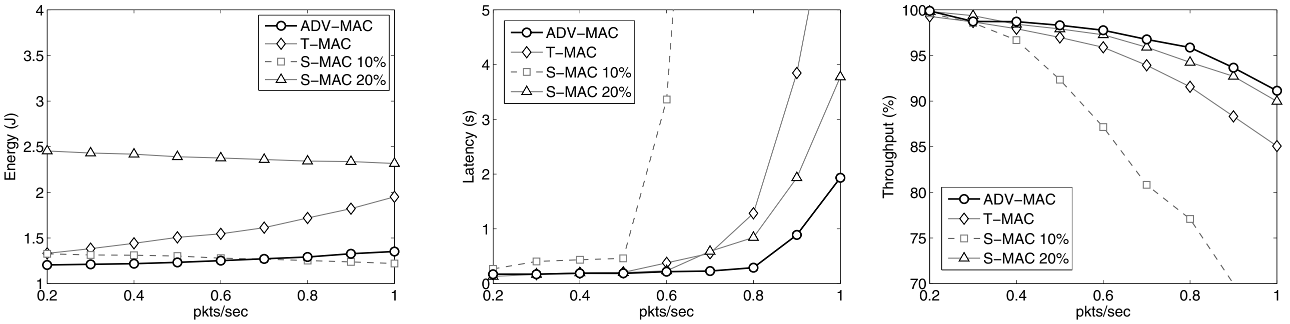


Fig. 5. Single hop, multicast : Energy Consumption vs. Number of Sources. The extension '-a' corresponds to analytical results.

throughput. However, the energy consumption of T-MAC increases with the number of sources. ADV-MAC, on the other hand, shows very little increase in energy consumption while maintaining similar latency and throughput as T-MAC. ADV-MAC provides up to 35% reduction in energy compared to T-MAC. It is to be noted that at 10 sources, both T-MAC and ADV-MAC initially have all the nodes awake for data transfer. However, as each pair of node complete their data exchange, they go to sleep in ADV-MAC. However, all nodes in T-MAC keep renewing their timers and stay awake until the last pair of nodes has completed their data exchange. Thus the energy consumption of T-MAC is considerably higher compared to ADV-MAC. Thus, ADV-MAC also adapts successfully to different numbers of sources with the least energy consumption and while still maintaining the minimum latency and the maximum throughput.

3) Effect of data rate on single hop, multicast scenario: In the third simulation setup, we consider the performance of the three protocols under multicasting as we vary the data rate. We consider an area of 50 m x 50 m with 20 nodes including 4 sources. All nodes are in transmission range of each other. There are 4 types of data, and each source broadcasts a specific type of data. Each receiving node is interested in only one of these four data types. The receiving node types are uniformly distributed. As in the first set of simulations, we vary the traffic load by increasing the data rate from 0.2 pkt/sec to 1 pkt/sec. Fig. 4 shows the energy consumption of the three protocols obtained from simulations and analysis under multicasting. As seen from the figure, ADV-MAC results in the least energy consumption values for all data rates. The energy savings of ADV-MAC is because ADV packets contain the type of the data to be broadcast. Hence, nodes that are not interested in that type of data go to sleep, saving energy. As a result, ADV-MAC provides up to 28% reduction in energy compared to T-MAC. The latency and throughput trends are similar to the first set of simulations and are not shown. Thus, ADV-MAC has the least energy consumption in a multicasting scenario with high throughput and low latency.

4) Effect of number of sources on single hop, multicast scenario: In this set of simulations, we consider the performance of the three protocols under multicasting as we increase the number of sources from 1 to 10. As in the previous simulations, there are 20 nodes including the sources in an area of 50 m x 50 m. The sources are transmitting at the rate of 1 pkt/sec. There are 4 data types, and the data types are uniformly distributed among the sources as well as the receivers. Fig. 5 shows the energy consumption results for the three protocols. The trend is similar to the unicast case in the second set of simulations. The energy consumption of ADV-MAC is 45% less compared to T-MAC for 10 sources. The latency and throughput (not shown) are similar to the



(a) Energy Consumption vs. Data Rate.

(b) Latency vs. Data Rate.

(c) Throughput vs. Data Rate.

Fig. 6. Multi hop, unicast vs. Data Rate: Performance comparison of ADV-MAC, T-MAC and S-MAC.

unicast case in the second set of simulations where ADV-MAC achieves the highest throughput and the least latency.

5) Effect of data rate on multi hop, unicast scenario: To investigate the performance of the three MAC protocols in a multi-hop communication environment, we define a new simulation set. We consider an area of 700 m x 700 m. There are 312 nodes with each node having an average of 20 neighbors in its transmission range. There are 20 source nodes, and we increase the data rate from 0.2 pkt/sec to 1 pkt/sec. On the average, each receiver will have 5 transmitting nodes in its carrier sense range. Fig. 6(a) shows the energy consumptions obtained from the simulations. Since the energy equations are valid only for the single hop case, we do not show analytical results. In the previous simulations, all nodes could hear each other. Therefore, T-MAC did not suffer from the early sleeping problem. However, since this is a multihop case, the early sleeping problem is present in T-MAC, and the effects are visible in the simulation results. It is seen that the energy consumption of ADV-MAC is as much as 30% less compared to T-MAC and as much as 41% less compared to S-MAC with 20% duty cycle.

Fig. 6(b) shows the latency comparison. The latency of T-MAC is more than that of ADV-MAC at higher data rates. This is because of the early sleeping problem. However ADV-MAC is immune to the problem, and its latency does not increase as much. The effect of the early sleeping problem is also visible in the throughput comparison as seen in Fig. 6(c). It is seen that the throughput of T-MAC drops faster than ADV-MAC as the data rate increases.

V. CONCLUSION AND FUTURE WORK

This paper presents ADV-MAC, a new MAC protocol for wireless sensor networks. ADV-MAC minimizes the energy lost due to idle listening by introducing the concept of advertising for contention. Simulations show that the protocol adapts nicely to low and high traffic loads as well as to variable loads. ADV-MAC provides further reductions in energy compared to S-MAC and T-MAC while not sacrificing throughput or latency. In fact, in multihop variable load cases, ADV-MAC not only has the least energy consumption, but also has better latency and throughput compared to T-MAC. Also, ADV-MAC introduces an energy efficient multicasting mechanism at the MAC level that is absent in S-MAC and T-MAC.

In the future, several other aspects of ADV-MAC will be investigated such as parameter analysis. We plan to optimize the length of the ADV period as well as the contention window. Also, we plan to develop a TDMA based version of this protocol. This may remove the RTS/CTS packet exchange required for data exchange, resulting in further energy savings.

REFERENCES

[1] "IEEE Standard 802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," LAN MAN Standards Committee of the IEEE Computer Society., June 1999.

[2] "Crossbow Technology Inc." [Online]. Available: http://www.xbow.com/

[3] "Sentilla Corporation." [Online]. Available: http://www.sentilla.com/

[4] W. Ye, J. Heidemann, and D. Estrin, "An energy-efficient mac protocol for wireless sensor networks," in *Proceedings of the IEEE Infocom*. New York, NY, USA: IEEE, June 2002, pp. 1567–1576.

[5] T. van Dam and K. Langendoen, "An adaptive energy-efficient mac protocol for wireless sensor networks," in *ACM SenSys '03*, 2003, pp. 171–180.

[6] C. Zhu and M. Corson, "A five-phase reservation protocol (fprp) for mobile ad hoc networks," *Proceedings of the IEEE Infocom '98*, vol. 1, pp. 322–331 vol.1, Mar 1998.

[7] Y. Wang and I. Henning, "A deterministic distributed tdma scheduling algorithm for wireless sensor networks," *Proceedings of WiCom Conference '07*, pp. 2759–2762, Sept. 2007.

[8] E. L. Lloyd, "Broadcast scheduling for tdma in wireless multihop networks," *Handbook of wireless networks and mobile computing*, pp. 347–370, 2002.

[9] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *Proceedings of the HICSS '00*, 2000, p. 8020.

[10] J. Mao, Z. Wu, and X. Wu, "A tdma scheduling scheme for many-to-one communications in wireless sensor networks," *Comput. Commun.*, vol. 30, no. 4, pp. 863–872, 2007.

[11] "Specification of the Bluetooth System: Core (2001)." [Online]. Available: http://www.bluetooth.org/

[12] S. Ramanathan, "A unified framework and algorithm for (t/f/c)dma channel assignment in wireless networks," *Proceedings of the IEEE Infocom '97*, vol. 2, pp. 900–907 vol.2, Apr 1997.

[13] M. Buettner, G. V. Yee, E. Anderson, and R. Han, "X-mac: a short preamble mac protocol for duty-cycled wireless sensor networks," in *ACM SenSys '06*. New York, NY, USA: ACM, 2006, pp. 307–320.

[14] J. Polastre, J. Hill, and D. Culler, "Versatile low power media access for wireless sensor networks," in *ACM SenSys '04*. New York, NY, USA: ACM, 2004, pp. 95–107.

[15] W. Ye, F. Silva, and J. Heidemann, "Ultra-low duty cycle mac with scheduled channel polling," in *ACM SenSys '06*. New York, NY, USA: ACM, 2006, pp. 321–334.

[16] "The Network Simulator NS-2," http://www.isi.edu/nsnam/ns/.