Efficient MAC Protocol for Wireless Sensor Networks

Shiv Singh
SGT Institute of Engineering and Technology

Pratika Saini
(Asst. Prof.), SGT Institute of Engineering and Technology

Abstract—A wireless sensor network (WSN) is a collection of a large number of sensor nodes which are deployed over an area to perform some computations based on information collected from the surroundings. Each node in the network have a battery, but it is very difficult to replace or recharge batteries; therefore, the main task is: "how can we reduce the WSN energy utilization?" Hence, increasing the lifetime of the network through decreasing the energy is an main challenge in WSN; sensors cannot be easily replaced or recharged due to their ad-hoc deployment in various kinds of environment. This paper proposes S-MAC, a protocol designed for WSN. Wireless sensor networks (WSN) use battery-operated computing and sensing devices. To collect information. A network of these devices will be useful for a common application such as environmental monitoring or forecasting. We expect sensor networks to be deployed in an ad hoc fashion, with individual nodes remaining inactive for long periods of time, but then becoming active when something is detected. These types of sensor networks and applications can be implemented by MAC that is different from other wireless MACs such as IEEE 802.11 in almost every way. Low energy utilization and self-configuration are main goals, while every node to be fairness and latency are less main. S-MAC uses three techniques to reduce energy utilization and support self-configuration. To reduce energy utilization in listening to an idle channel these nodes gets periodically sleep.

I. INTRODUCTI ON

WIRELESS sensor networking is an emerging technology that has a wide range of potential applications including environment monitoring, smart spaces, medical systems, agriculture and robotics. Such a network generally consists of a large number of nodes that organize themselves into a multi-hop wireless network. Each node has sensors, embedded processors with in and low-power radios, and is battery operated. These nodes collectively perform a common task.

A wireless sensor network is made up of three components: Sensors Nodes, Task Manager Node (User) and Interconnect Backbone

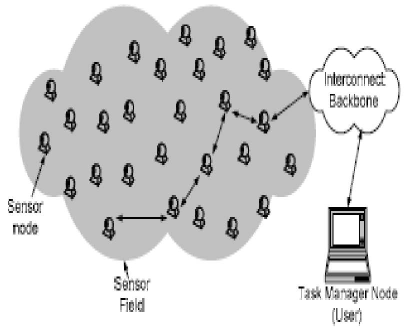

Fig 1: Wireless Sensor Network (WSN

Like in all shared-medium networks, medium access control (MAC) is an main technique that enables the successful op- eration of the network. One fundamental task of the MAC pro- tocol is to avoid collisions so that two interfering nodes do not transmit signals at the same time. There are many MAC protocols that have been developed for wireless voice and data communication networks. Some of them include the code division multiple access (CDMA), time division multiple access (TDMA) and protocols like IEEE 802.11 [1].

A good MAC protocol for the wireless sensor networks has the following attributes. The first is the energy conservation.

As stated above, sensor nodes are likely to be battery powered, and it is often very difficult to change or recharge batteries for these nodes. In fact, we can expect in near future that some of these nodes to be cheap enough that they are replaced rather than recharged. Increasing network lifetime for these nodes is a main issue. Another main attribute is the scalability to the different changes in the network size, number of nodes and various topologies. Some nodes may wear out over time; some new nodes may join later; some nodes may move to different locations. The network topology can also change over time as well due to many reasons. A good MAC protocol should easily work with such network changes. Other main attributes include fairness, latency, throughput and bandwidth utilization. These attributes are generally the main concerns in traditional wireless voice and data networks, but in sensor networks they are less important.

This paper presents sensor-MAC (S-MAC), a new MAC pro- tocol explicitly made for wireless sensor networks. While reducing energy utilization is the primary goal in our design, our protocol also has good compatibility and collision avoidance and detection ability. It produces good scalability and collision avoidance by using a combined scheduling and contention scheme. We have identified the following major sources of energy leakage. The first one is collision. When a sent packet is corrupted it has to be discarded, and the packet needs to be retransmitted then it will increase energy utilization. Collision increases latency as well. The second source is overhearing meaning that a node hears packets that are destined to another node. The third source is control packet overhead. Sending and receiving control packets consumes more energy, and less useful data packets can be sent. The last major source is idle listening, *i.e.*, listening to traffic that is not sent. This is especially true in many sensor networks. If nothing is sensed, nodes can be in idle mode for most of the time.

But, in many MAC protocols such as IEEE 802.11 or TDMA nodes must listen to the channel to receive possible traffic. Many measurements have shown that idle listening consumes less energy required for receiving. S-MAC tries to reduce the energy utilization from all the above sources. Due to this we accept some reduction in both per-hop fairness and latency. Although per-hop fairness and latency is being reduced, we can argue that the reduction does not necessarily result in lower peer to peer fairness and latency.

In normal wireless voice or data networks, every user wants equal opportunity and time to use the medium, i.e sending or receiving data for their applications. Per-hop MAC level fairness is thus a main issue. But, in sensor networks, all nodes work for a single common task. Generally there is only one application which is to be performed by network. Sometimes, a node may have more data to send than some other nodes. In this kind of situation fairness is not as main as long as performance at application-level is not degraded. In our protocol, we are using the concept of message passing to transmit a very long message efficiently. The basic idea is to divide the long message into small parts and transmit them in a burst. Due to this method t a node that has more data to send gets more time to access the medium. This is not fair for those nodes that only have some short packets to send, since their short packets have to wait a long time for very long signals or packets. But, message passing can achieve energy savings by reducing control overhead and avoiding overhearing. Importance of latency depends on what application is running and the node state. During a period when there is no sensing event, there is generally very little or no data flow in the network. Most of the time nodes remain in the idle state. S-MAC protocol therefore lets nodes periodically sleep then in the idle listening mode. When in the sleep mode, a node will switched off its radio. This design reduces the energy utilization due to idle listening. But it will increase the latency, since a sender must wait for the receiver to wake up before sending dataAnother main task of wireless sensor networks is the in- network data processing. It can greatly reduce energy utilization compared to transmitting all the raw data to the end node [4], [5], [6]. In-network processing requires store-and- forward processing of messages. A message is a meaningful unit of data that a node can process (difference or filter, *etc.*). A message can be long and consists of many small mades. In contrast, message passing reduces message-level latency by trading off the made-level fairness

1.1 WSN Requirements and Challenges

It must support the following requirements in deployment: scalability, reliability, responsiveness, mobility, and power efficiency. The description of these:

Reliability- The ability of the network for reliable data transmission in a state of continuous change of network structure.

Scalability- It is the ability of the network to grow without excessive overhead
Responsiveness -The ability of the network to quickly adapt itself to changes in topology.
Mobility- It is the ability of the network to handle mobile nodes and changeable data paths.

1.2 Applications: Area monitoring

In area monitoring, the WSN is deployed over a region where some phenomenon is to be monitored. In military, it is the used for detects enemy intrusion; a civilian example is the geo-fencing of gas or oil pipelines.

1.3 Environmental/Earth monitoring

The term Environmental Sensor Networks has evolved to cover many applications of WSNs to earth science research including sensing volcanoes, oceans, glaciers, forests etc. Some examples of major areas listed below.

- Air quality monitoring

In dangerous surroundings, real time monitoring of harm- ful gases is a concerning process. Wireless sensor networks have been launched to produce specific solutions for people.

- Interior monitoring

Wireless internal monitoring solutions provide tabs for large areas as well as ensure the precise gas concentration degree.

- Exterior monitoring

External aerial quality monitoring needs the use of precise wireless sensors, rain & wind resistant solutions as well as energy reaping methods to assure extensive liberty to machine that will likely have tough access.

II. RELATED WORK

The medium access control is a broad research area, and many researchers have done research work in the new area of low power and wireless sensor networks [11], [12], [13], [14].

Current MAC design for wireless sensor networks can be broadly divided into contention-based and TDMA protocols. The standardized IEEE 802.11 distributed coordination function (DCF) [1] is an example of the contention-based protocol, and is mainly built on the research protocol MACAW [15]. It is widely used in ad hoc wireless networks because of its simplic- ity and robustness to the hidden terminal problem. But, re- cent work [2] has shown that the energy utilization using this MAC is very high when nodes are in idle mode. This is mainly due to the idle listening. PAMAS [10] made an improvement by trying to avoid the overhearings among neighboring nodes. Our paper also exploits similar method for energy savings. The main difference of our work with PAMAS is that we do not use any out-of-channel signaling. Whereas in PAMAS, it requires two independent radio channels, which in most cases indicates two independent radio systems on each node. PAMAS does not address the issue of reduce idle listening The other class of MAC protocols is based on scheduling and reservation, for example TDMA-based protocols. TDMA protocols are useful for energy conservation compared to contention protocols because the duty cycle of the radio is reduced and there is no contention-introduced overhead and collisions. But, TDMA protocol requires the nodes to form actual communication clusters, like Bluetooth [16], [17] and LEACH [13]. Managing inter cluster communication is not an easy task. Moreover, when the number of nodes within a cluster changes, it is not easy for a TDMA protocol to dynamically change its frame length and time slot assignment. So its scalability is generally not as good as that of a contention-based protocol. For example, Bluetooth may have at most 8 active nodes in a cluster.

Sohrabi and Pottie [12] proposed a self-organization protocol for wireless sensor networks. Each node maintains a TDMA- like frame, called super frame, in which the node schedules dif- ferent time slots to communicate with its known neighbors. At each time slot, it only talks to one neighbor. To avoid inter- ference between adjacent links, the protocol assigns different channels, *i.e.*, frequency (FDMA) or spreading code (CDMA), to potentially interfering links. Although the super frame struc- ture is similar to a TDMA frame, it does not prevent two interfering nodes from accessing the medium at the same time. The actual multiple access is accomplished by FDMA or CDMA. A disadvantage of the scheme is its low bandwidth utilization. For example, if a node only has packets to be sent to one neighbor, it cannot reuse the time slots scheduled to other neighbors. Piconet [11] is an architecture designed for low-power ad hoc wireless networks. One main feature of piconet is that it a puts nodes into periodic sleep for energy conservation. The scheme that piconet uses to coordinate neighboring nodes is to let a node broadcast its address before it can actually starts listening. If a node wants to signal to a neighboring node, it must wait until it receives the neighbor’s broadcast Woo and Culler [14] examined different configurations of carrier sense multiple access (CSMA) and proposed an adaptive rate control mechanism, whose main goal is to achieve fair bandwidth allocation to all nodes in a multi-hop network. They have used the motes and TinyOS platform to test and measure different MAC schemes. In comparison, our approach does not promote per-node fairness, and even trade it off for further energy savings.

III. SENSOR-MAC PROTOCOL DESIGN

The main goal in our MAC protocol design is to reduce en- ergy utilization, while supporting good scalability and colli- sion avoidance. Our protocol reduces energy utilization from all the sources that we have identified to cause energy waste, *i.e.*, idle listening, collision, overhearing and control overhead. To achieve our goal, we have designed the S-MAC protocol that consists of three main components: periodic listen and sleep, collision and overhearing avoidance, and message passing. Before describing all of them we first discuss our assumptions about the wireless sensor network and it applications.

3.1. Network Design

We suppose sensor networks to be made up of many small nodes deployed in an ad hoc fashion. The large number of nodes can

also be useful for short-range, multi-hop communication (instead of long range communication) to conserve energy. We expect most sensor networks to be dedicated to a single application or a few collaborative applications, thus rather than node-level fairness, we focus on maximizing system-wide application performance. Techniques such as data aggregation can reduce traffic, while collaborative signal processing can reduce traffic and improve sensing quality. In-network processing, data will be processed as whole messages at a time in store-and- forward fashion, so packet or made-level interleaving from multiple sources only increases overall latency.

3.2. Periodic Listen and Sleep

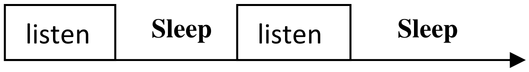

Fig. 2. Periodic listen and sleep.

Basic working of this scheme is shown in the figure above. Every node in the network goes to sleep for some time, and then wakes up and listens to see if any other node wants to talk to it. During sleep, the node turns off its radio, and sets a timer to awake itself later. Two techniques are used here as:

Time Stamp

Larger listen period then clock error

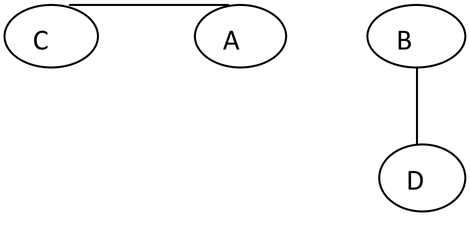

Fig 3 . Neighbouring nodes in the network with different schedule coordinate with each other

It should be noticed that not all neigh-boring nodes can synchronize together in a multi-hop network. Two neighboring nodes A and B may have different schedules if they each in turn must synchronize with different nodes, C and D, respectively, as shown in Figure 3. Nodes exchange their schedules by broadcasting it to all its immediate neighbors. This ensures that all neighboring nodes can talk to each other even if they have different schedules. For example, in Figure 2 if node A wants to talk to node B, it just wait until B is listening. If multiple neighbors want to talk to a node, they need to contend for the medium when the node is listening. The downside of the scheme is that the latency is increased due to the periodic sleep of each node. Moreover, the delay can accumulate on each hop. So the latency requirement of the application places a fundamental limit on the sleep time After they start data transmission, they do not follow their sleep schedules until they finish transmission.

3.3 Scheduling

Before each node starts its periodic listen and sleep, it needs to choose a schedule and exchange it with its neighbors. Each node maintains a *schedule table* that stores the schedules of all its known neighbors. Working of schedule table is given below.

1. The node first listens for a certain amount of time. If it does not hear a schedule from another node, it randomly chooses a time to go to sleep and immediately broadcasts its schedule in a SYNC message, indicating that it will go to sleep after t sec- onds. We call such a node a *synchronizer*, since it chooses its schedule independently and other nodes will synchronize with it.
2. If the node receives a schedule from a neighbor before choos- ing its own schedule, it follows that schedule by setting its schedule to be the same. We call such a node a *follower*. It then waits for a random delay t_d and rebroadcasts this schedule, indicating that it will sleep in $t - t_d$ seconds. The random delay is for collision avoidance, so that multiple followers triggered from the same synchronizer do not systematically collide when rebroadcasting the schedule

3. If a node receives a different schedule after it selects and broadcasts its own schedule, it adopts both schedules. It broadcasts it own schedule before going to sleep..

3.4. Synchronization

The listen/sleep scheme requires synchronization among neighboring nodes. Although the long listen time can tolerate fairly large clock drift, neighboring nodes still need to period- ically update each other their schedules to prevent long-time clock drift. The updating period can be quite long. The mea- surements on our testbed nodes show that it can be on the order of tens of seconds.

Updating schedules is accomplished by sending a SYNC packet. The SYNC packet is very short, and includes the ad- dress of the sender and the time of its next sleep. The next-sleep time is relative to the moment that the sender finishes transmit- ting the SYNC packet, which is approximately when receivers get the packet (since propagation delays are short). Receivers will adjust their timers immediately after they receive the SYNC packet. A node will go to sleep when the timer fires.

In order for a node to receive both SYNC packets and data packets, we divide its listen interval into two parts. The first part is for receiving SYNC packets, and the second one is for receiving RTS packets, as shown in Figure 3. Each part is fur- ther divided into many time slots for senders to perform carrier sense. For example, if a sender wants to send a SYNC packet, it starts carrier sense when the receiver begins listening. It ran- domly selects a time slot to finish its carrier sense. If it has not detected any transmission by the end of the time slot, it wins the medium and starts sending its SYNC packet at that time. The same procedure is followed when sending data packets

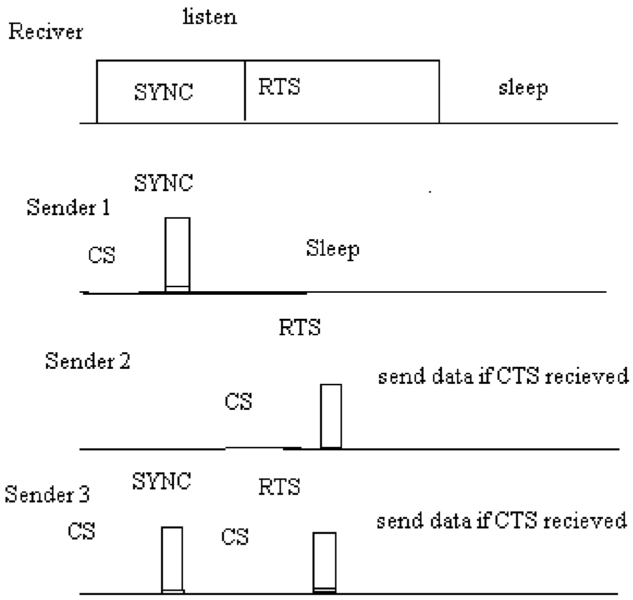

Fig. 4. Timing relationship between a receiver and different senders.

Fig. 4 also shows the timing relationship of three possi- ble situations that a sender transmits to a receiver. CS stands for carrier sense. In the figure, sender 1 only sends a SYNC packet. Sender 2 only wants to send data. Sender 3 sends a SYNC packet and a RTS packet.

3.5. Collision and Overhearing Avoidance

Collision avoidance is a basic task of MAC protocols. S- MAC adopts a contention-based scheme. It is common that any packet transmitted by a node is received by all its neighbors even though only one of them is the intended receiver. Overhearing makes contention-based protocols less efficient in energy than TDMA protocols. So it needs to be avoided.

3.5.1 Avoidance of Collision

Since multiple senders may want to send to a receiver at the same time, they need to contend for the medium to avoid col- lisions. Among contention based protocols, the 802.11 does a very good job of collision avoidance. Our protocol follows sim- ilar procedures, including both virtual and physical carrier sense and RTS/CTS exchange. We adopt the RTS/CTS mechanism to address the hidden terminal problem [15].

There is a duration field in each transmitted packet that indi- cates how long the remaining transmission will be. So if a node receives a packet destined to another node, it knows how long it has to keep silent. The node records this value in an variable called the network allocation vector (NAV) [1] and sets a timer for it. Every time when the NAV timer fires, the node decrements the NAV value until it reaches zero. When a node has data to send, it first looks at the NAV. If its value is not zero, the node determines that the medium is busy. This is called virtual carrier sense.

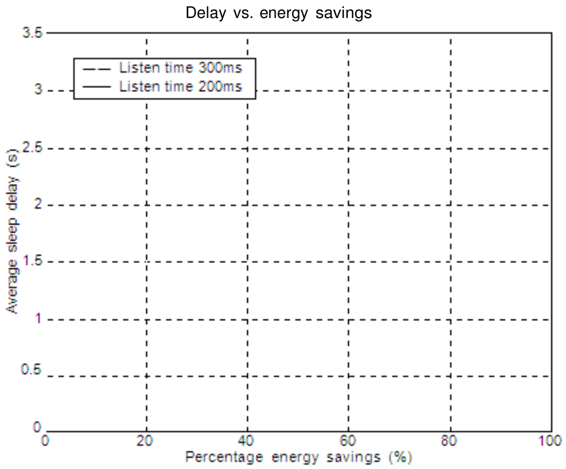

Fig. 5. Energy savings vs. average sleep delay for the listen time of 30ms

up. We call it as a Delay in Sleep since it is caused by the sleep of the receiver.
We call a complete cycle of the listen and sleep a part. As- sume a packet arrives at the sender with equal probability in time within a frame. So the average sleep delay on the sender is

$$D_s = \frac{T_{frame}}{2} \qquad (1)$$

where

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

Comparing with protocols without periodic sleep, the relative energy savings in S-MAC is

$$E_s = \frac{T_{sleep}}{T_{frame}} = 1 \quad \frac{T_{listen}}{T_{frame}} \qquad (3)$$

The last item in the above equation is the duty cycle of the node. It is desirable to have the listen time as short as possible so that for a certain duty cycle, the average sleep delay is short. In our implementation we set the listen time as 300ms. Figure 5 shows the percentage of energy savings E_s vs. average sleep delay D_s on each node for the listen time of 300ms and 200ms. We can see that even if the sleep time is zero (no sleeping) there is still a delay. This effect is because contention only starts at the beginning of each listen interval.

IV. PROTOCOL IMPLEMENTATION

The purpose of our implementation is to demonstrate the ef- fectiveness of our protocol and to compare our protocol with 802.11 through some basic experiments.

4.1. Testbed

We use Rene Motes, developed at UCB [7], as our develop- ment platform and testbed (see Figure 6). A mote is slightly larger than a quarter. The heart of the node is the Atmel

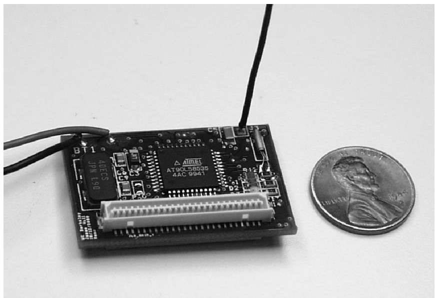

Fig. 6. The UCB Rene Mote.

AT90LS8535 microcontroller [18], which has 8K bytes of pro- grammable flash and 512 bytes of data memory. The radio transceiver on the mote is the model TR1000 from RF Monolithics, Inc [8]. When using the OOK(on-off keyed) modulation, it provides a transmission rate of 19.2 Kbps. It has three working modes, *i.e.*, receiving, transmitting and sleep, each drawing the input current of 4.5mA, 12mA (peak) and 5µA respectively.

Our motes use TinyOS, a useful event-driven operating system [9], [19]. It provides the basic mechanism for packet transmitting, receiving and processing. TinyOS promotes mod- ularity, data sharing and reuse. the standard release of TinyOS has only one type of packet, which consists of a header, the payload and a cyclic redundancy check (CRC). The length of the header or the payload can be changed to different values. But, once they are defined, all packets have the same length and format.

Generally the control packets, such as RTS, CTS and ACK, are very short and without payload. So we have created an other. packet type in TinyOS, the control packet, which only has the 6-byte header and the 2-byte CRC. We have modified several TinyOS components to accommodate the new packet. This enables us to efficiently implement MAC protocols and accurately measure their performance.

4.2. Implementation of MAC Protocols

We have implemented three MAC modules on the mote and TinyOS platform, as listed below.

1. Simplified IEEE 802.11 DCF
2. Message passing with overhearing avoidance
3. The complete S-MAC

For the purpose of performance comparison, we first imple- mented a simplified version of IEEE 802.11 DCF. It has the fol- lowing major pieces: physical and virtual carrier sense, back- off and retry, RTS/CTS/DATA/ACK packet exchange, and frag- mentation support.

The time spent of each carrier sense is a random time within the contention window. The randomization is very inportant to avoid collisions at the first step. For ease, the contention window does not exponentially gets big when backoff happens. With 802.11 the radio of each node does not go into sleep.

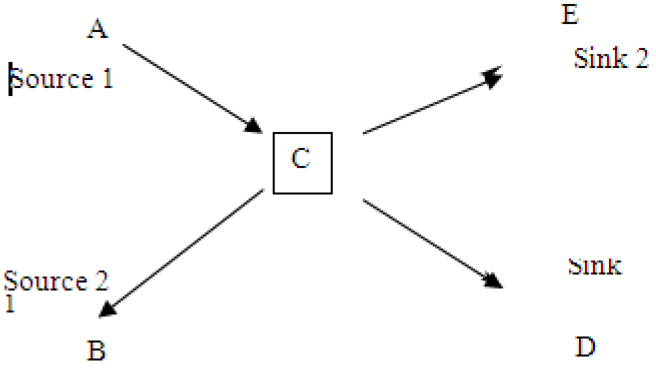

Fig. 7. Topology used in experiments: two-hop network with two sources and two sinks.

mode. It is either in listen/receiving mode or transmitting mode. The other module is the message passing with avoidance of overhearing. It achieves energy savings by avoiding overhearing, reducing control overhead and contention times. It does not include the period listen and sleep. So there is no additional delay comparing with the simplified IEEE 802.11. The radio of each node goes into the sleep mode only when its neighbors are in transmission.

With the message passing module we have incorporated peri- odic listen and sleep, and completed most basic functionalities in S-MAC. Currently, the listen time for each node is 300ms, and sleep time can be changed to different values, such as 300ms,

500ms, 1s, *etc.*, which makes different duty cycles of the radio. We can also specify the frequency that the SYNC packet is sent for schedule update between neighboring nodes. In our follow- ing experiments, we have chosen the sleep time as 1 second and the frequency for schedule update is 10 listen/sleep period, *i.e.*,

13 seconds.

It should be noted that the energy savings in the current imple- mentation is only due to the sleep of the radio. In other words, the microcontroller does not go to sleep. It actually has a sleep mode, which consumes much less energy and can be waked up by a low-frequency watchdog timer. If we put the microcon- troller into the sleep mode as well when the radio is sleeping, we are able to save more energy.

V. EXPERI MENTATI ON

The main goal of the experimentation described here is to measure the energy utilization of the radio for using each of the MAC modules we have implemented.

5.1. Experiment Setup

Figure 7 is the topology we used in our experiments. This is a two-hop network with two sources and two sinks. Packets from source A flow through node C and end at sink D, while those from B also pass through C but end at E. The topology is simple, but it is sufficient to show the basic characteristics of the MAC protocols.

We will look at the energy utilization of each node when utilizing different MAC protocols and under different traffic loads.

The two sources periodically generate a sensing message, which is divided into some mades. In the simplified IEEE 802.11 MAC, these mades are sent in a burst, *i.e.*, RTS/CTS is not used for each made. We did not measure the 802.11 MAC without madeation, which treats each made as an independent packet and uses RTS/CTS for each of them. message passing is used, and parts of a message are always Signaled in a burst.

We change the traffic load by varying the inter-arrival period of the messages. If the message inter-arrival period is 5 seconds, a message is generated every 5 seconds by each source node. In our following experiments, the message inter-arrival period varies from 1s to 10s.

For each traffic pattern, we have done 10 independent tests to measure the energy utilization of each node when using different MAC protocols. In each test, each source periodi- cally generates 10 messages, which in turn is madeed into 10 small data packets supported by the TinyOS. Thus in each ex- periment, there are 200 TinyOS data packets to be passed from their sources to their sinks. For the highest rate with a 1s inter- arrival time, the wireless channel is nearly fully utilized due to its low bandwidth.

We measure the amount of time that each node has used to pass these packets as well as the percentage time its radio has spent in each mode (transmitting, receiving, listening or sleep). The energy utilization in each node is then calculated by mul- tiplying the time with the required power to operate the radio in that mode. We found the power utilization from the data sheet of the radio transceiver, which is 13.5mW, 24.75mW and 15µW, in receiving, transmitting and sleep respectively. There is no dif- ference between listening and receiving in this radio transceiver model.

5.2. Results and Analysis

The experiments are carried out on the three MAC modules we have implemented on our testbed nodes. In the result graphs, the simplified IEEE 802.11 DCF is denoted as ‘IEEE 802.11’. The message passing with overhearing avoidance is identified as ‘Overhearing avoidance’. The complete S-MAC protocol, which includes all pieces of our new protocol, is denoted as ‘S- MAC’.

We first look at the experiment results on the source nodes A and B. Figure 8 is the measured average energy utilization from these two nodes. The traffic is heavy when the message inter-arrival time is less than 4s. In this case, 802.11 MAC uses more than twice the energy used by S-MAC. Since idle listening rarely happens, energy savings from periodic sleeping is very limited. S-MAC achieves energy savings mainly by avoiding overhearing and efficiently transmitting a long message.

When the message inter-arrival period is larger than 4s, traffic load becomes light. In this case, the complete S-MAC protocol has the best energy property, and far outperforms 802.11 MAC. Message passing with overhearing avoidance also performs bet- ter than 802.11 MAC. But, as shown in the figure, when idle listening dominates the total energy utilization, the pe- riodic sleep plays a key role for energy savings. The energy utilization of S-MAC is relatively independent of the traffic pattern.

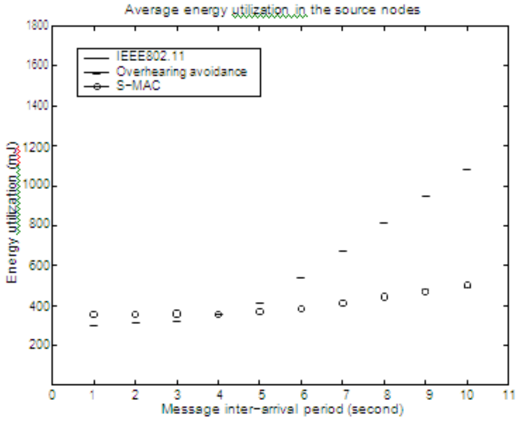

Fig. 8. Measured energy utilization in the source nodes.

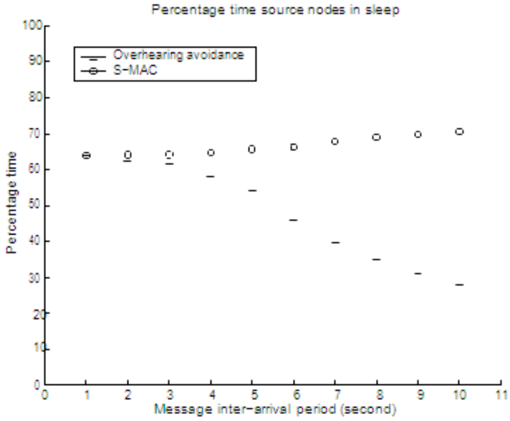

Fig. 9. Measured percentage of time that the source nodes in the sleep mode

Compared with 802.11, message passing with overhearing avoidance saves almost the same amount of energy under all traffic conditions. This result is due to overhearing avoidance among neighboring nodes A, B and C. The number of packets to be sent by each of them are the same in all traffic conditions. Figure 9 shows the percentage of time that the source nodes are in the sleep mode. It is interesting that the S-MAC protocol adjusts the sleep time according to traffic patterns. When there is little traffic, the node has more sleep time (although there is a limit by the duty cycle of the node). When traffic increasesnodes have fewer chances to go to periodic sleep and thus spend more time in transmission.

This is a useful feature for sensor network applications, since the traffic load indeed changes over time. When there is no sens- ing event, the traffic is very light. When some nodes detects an event, it may trigger a big sensor like a camera, which will gen- erate heavy traffic. The S-MAC protocol is able to adapt to the traffic changes. In comparison, the module of message passing with overhearing avoidance does not have periodic sleep, and nodes spend more and more time in idle listening when traffic load decreases.

Figure 10 shows the measured energy utilization in the in- termediate node C. We can see in the light traffic case, it still outperforms 802.11 MAC. In heave traffic case, it consumes slightly more energy than 802.11. One reason is that S-MAC has synchronization overhead of sending and receiving SYNC packets. Another reason is that S-MAC introduces more latency and actually uses more time to pass the same amount of data.

In fact, if the traffic is extremely heavy and a node does not have any chance to follow its sleep schedule, the scheme of pe- riodic listen and sleep does not benefit at all. But, message passing and overhearing avoidance are still effective means of saving energy. This has been illustrated in the results of the

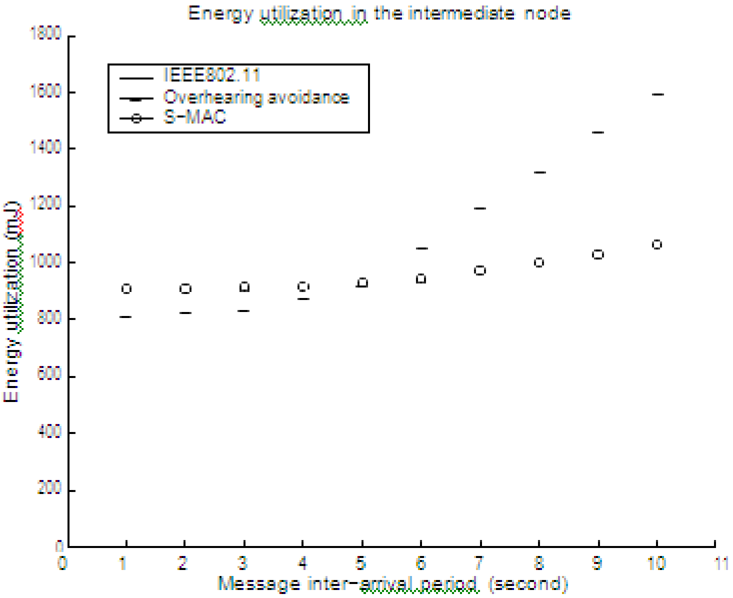

Fig. 10. Measured energy utilization in the intermediate node.

source nodes (Figure 8). But we cannot see similar results on the intermediate node C, since all packet transmissions involve this node. In this case, its energy utilization is about the same as that of using the 802.11 MAC.

VI. CONCLUSIONS AND FUTURE WORK

Wireless sensor network is a network of sensor nodes. Sensors have their limited energy and limited range to communicate with other nodes.So main concern is to optimize energy utilization. In this work first task is cluster head formation then clusters head selection. In this work energy utilization, total power, cluster throughput, and packet delivery ration is also implemented. Clusters head selection depends on maximum energy remaining after every transmission and receiving. The amount of transmission data via the cluster heads and sends data to the BS. It is also proved that energy utilization is reduced and also increase the lifetime of the WSN. It also increases the total power and packet data ratio.

REFERENCES

[1] F.Akyildiz,W.Su,Y.Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: a Survey", Computer Networks ,Vol 38, 2002.

[2] M. Ilyas and I. Mahgoub "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems," Proceedings of the CRC Press, 2005.

[3] J.Chang, L.Tassiulas, "Energy conserving routing in wireless ad hoc networks", Nineteenth International Annual Joint Conference of the IEEE Computer and Communications Societies, 2000.

[4] Y. Xu, J. Heidemann, D. Estrin, "Geography-informed energy conservation for ad hoc", ACM MobiCom, 2001.

[5] M.Zorzi,R.R. Rao, "Geographic Random Forwarding (GeRaF) for Ad Hoc and Sensor Networks: Energy and Latency Performance",IEEE Transactions Mobile Computing,2003.

[6] B.Chen,K.Jamieson,H.Balakrishnan, R. Morris,"Span: an energyefficient coordination algorithm for topology maintenance in ad hoc wireless networks", ACM Wireless Networks, 2002.

[7] C. Schurgers,V.Tsiatsis,M.B.Srivastava,"STEM: topology management for energy efficient sensor networks", IEEE Aerospace Conference", 2002.

[8] G.Anastasi,M.Coti,M.Frrancesco,A.Passarella, "Energy conservation in wireless sensor networks: A survey", Elsever, Ad Hoc Network ,2009.

[9] I.Demirkol,C.Ersoy,F.Alagöz, "MAC Protocols for Wireless Sensor Networks: A Survey", IEEE Communications Magazine,April,2006.

[10] A.Bachir, Mischa Dohler,T.Watteyne,K.Leung, "MAC Essentials for Wireless Sensor Networks", IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 12, 2010.

[11] M.C.Vuran,O.B.Akan,I.F.Akyildiz,"Spatio-temporal correlation: theory and applications for wireless sensor networks", Computer Networks Journal,2004 .

[12] D.Chu,A.Deshpande,J.M.Hellerstein,W.Hong,"Approximate data collection in sensor networks using probabilistic models",22nd

International Conference on Data Engineering ,2007.

[13] D.Tulone, S. Madden, "PAQ: time series forecasting for approximate query answering in sensor networks", 3rd European Conference on Wireless Sensor Networks ,2008.

[14] E.Fasolo,M.Rossi, J.Widmer, M. Zorzi,"In-network aggregation techniques for wireless sensor networks: a survey", IEEE Wireless Communications, 2007.

[15] F.Bouabdallah,N.Bouabdallah,R.Boutaba,"On Balancing Energy Utilization in Wireless Sensor Networks",IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY, VOL.58, 2009.

[16] M.Ilyas, I.Mahgoub, "Handbook of Sensor Networks: Compact Wireless and Wired Sensing Systems", in: Proceedings of the CRC Press, London, Washington, D.C, 2005.

[17] M.Dong, K.Yung,W. Kaiser,"Low Power Signal Processing Architectures for Network Microsensors", International Symposium on Low Power Electronics and Design,pp.173–177, 1997.

[18] M. Medidi,Y.Zhou,"Extending lifetime with differential duty cycles in wireless sensor networks" ,*Globecom07*, 2007.

[19] P.Casari,A. Marcucci, M. Nati, C. Petrioli, M. Zorzi,"A detailed simulation study of geographic random forwarding (GeRaF) in wireless sensor networks", Proc. of Military Communications Conference 2005,Vol. 1,pp. 59-68, 17-20 , 2005.