International Journal of Latest Engineering Research and Applications (IJLERA) ISSN: 2455-7137

Volume – 02, Issue – 04, April – 2017, PP – 130-136

Energy Efficient Modified S-MAC Protocol for Wireless Sensor Network

Rohit Pandhram[1], Priya Sundararaj[2], Ashwini Patil[3], Pranay Mokal[4], S.V.Patil[5]

[1,2,3,4,5]*Electronics and Telecommunication Department, Bharati Vidyapeeth College of Engineering,* NM

Abstract: The wireless medium being inherently broadcast in nature and hence prone to interferences requires highly optimized medium access control (MAC) protocols. This holds particularly true for wireless sensor networks (WSNs) consisting of a large amount of miniaturized battery-powered wireless networked sensors required to operate for years with no human intervention. There has hence been a growing interest on understanding and optimizing WSN MAC protocols in recent years, where the limited and constrained resources have driven research towards primarily reducing energy consumption of MAC functionalities.

In this paper, we have modified S-MAC protocol using novels techniques to reduce energy consumption namely coordinated adaptive sleeping, message passing and idle listening. We evaluate our modified S-MAC protocol using NS2 software and compare the results with simpler sensor medium access control protocol and also IEEE 802.11.

I. Introduction

A Wireless sensor network consists of a number sensors spread across a geographical area. These sensors are distributed either randomly or in patter. Wireless sensor networks have been subject of interest due to their wide variety of potential applications. Initially these networks were primarily researched for their military applications, and now the network have a wide industrial and consumer applicability through their remote interaction with the physical world. In many applications, wireless sensor network face some drawbacks, such as limited power, limited communication abilities, limited computation capabilities, huge deployment area, and the node count in a network. Because of all these drawbacks with wireless sensor networks, there are many challenging problems.

Among all, a very important issue regarding the design of senor networks is power consumption. Efficiency of energy consumption is an important challenge for these wireless sensor networks. All the sensor nodes have a limited battery supplies and a limited sensing capabilities. Network lifetime has a strong dependence on the sensor's battery power. Because these senor networks have a large number of nodes, allowing some nodes to sleep for a particular time intervals can result in an increased network lifetime. This paper presents sensor-MAC protocol with updated sleeping schedules and various energy conversing techniques

The major sources of energy waste are as following. The first one is collision. When a transmitted packet is corrupted it has to be discarded, and the follow-on retransmissions increase energy consumption. Collision increases latency as we ll. The second source is overhearing, meaning that a node pick up packets that are destined to other nodes. The third source is control packet overhead. Sending and receiving control packets consumes energy too, and less useful data packets can be transmitted. The last major source of inefficiency is idle listening, i.e., listening to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, nodes are in idle mode for most of the time. However, in many MAC protocols such as IEEE 802.11 or CDMA nodes must listen to the channel to receive possible traffic. Many measurements have shown that idle listening consumes 50–100% of the energy required for receiving. Simpler sensor medium access control tries to reduce the waste of energy from all the above sources. In exchange we accept some reduction in both per-hop fairness and latency. Although per-hop fairness and latency are reduced, we will argue that the reduction does not necessarily result in lower end-to-end fairness and latency.

II. Literature Survey

A sensor network is defined as being composed of a large number of nodes which are deployed densely in close proximity to the phenomenon to be monitored. Each of these nodes collects data and its purpose is to route this information back to a sink. The network must possess self-organizing capabilities since the positions of individual nodes are not predetermined. Co-operation among nodes is the dominant feature of this type of network, where groups of nodes co-operate to discriminate the information gathered in their vicinity to the user. The authors point out that none of the studies surveyed has a fully integrated view of all the factors driving the design of sensor networks and proceeds to present its own communication architecture and design factors to be used as a guideline and as a tool to compare various protocols. The design factors listed are as follows:

- **Scalability:-**Number in the order of hundreds or thousands. Protocols should be able to scale to such high degree and take advantage of the high density of such networks.
- **Sensor Network Topology:-**Must be maintained even with very high node densities.
- **Environment:-** Nodes are operating in inaccessible locations either because of hostile environment or because they are embedded in a structure.
- **Transmission Media:-** RF, Infrared and Optical.
- **Power Consumption:-** Power conservation and power management are primary design factors.

In et.al [2] varroosis" A SURVEY OF MAC PROTOCOLS FOR WIRELESS SENSOR NETWORKS**",** The medium access control protocols for the wireless sensor network have to achieve two objectives. The first objective is the creation of the sensor network infrastructure. A large number of sensor nodes are deployed and the MAC scheme must establish the communication link between the sensor nodes. The second objective is to share the communication medium fairly and efficiently.

- **Attributes of a good MAC Protocol:**

To design a good MAC protocol for the wireless sensor networks, the following attributes are to be considered:

(i)Energy Efficiency: The first is the energy efficiency. The sensor nodes are battery powered and it is often very difficult to change or recharge batteries for these sensor nodes. Sometimes it is beneficial to replace the sensor node rather than recharging them.

(ii) Latency: The second is latency. Latency requirement basically depends on the application. In the sensor network applications, the detected events must be reported to the sink node in real time so that the appropriate action could be taken immediately.

(iii)Throughput: Throughput requirement also varies with different applications. Some of the sensor network application requires sampling the information with fine temporal resolution. In such sensor applications it is better that sink node receives more data.

(iv)Fairness: In many sensor network applications when bandwidth is limited, it is necessary to ensure that the sink node receives information from all sensor nodes fairly. However among all of the above aspects the energy efficiency and throughput are the major aspects. Energy efficiency can be increased by minimizing the energy wastage.

❖ **Major Sources of Energy Wastes:**

Major sources of energy waste in wireless sensor network are basically of four types:

(i)Collision: The first one is the collision. When a transmitted packet is corrupted due to interference, it has to be discarded and the follow on retransmissions increase energy consumption. Collision increases latency also.

(ii)Overhearing: The second is overhearing, meaning that a node picks up packets that are destined to other nodes.

(iii)Packet Overhead: The third source is control packet overhead. Sending and receiving control packets consumes energy too and less useful data packets can be transmitted.

(iv)Idle Listening: The last major source of inefficiency is idle listening i.e., listening to receive possible traffic that is not sent. This is especially true in many sensor network applications. If nothing is sensed, the sensor node will be in idle state for most of the time. The main goal of any MAC protocol for sensor network is to minimize the energy waste due to idle listening, overhearing and collision.

❖ **MAC Performance Matrices:**

In order to evaluate and compare the performance of energy conscious MAC protocols, the following matrices are being used by the research community.

(i)Energy Consumption per bit: - The energy efficiency of the sensor nodes can be defined as the total energy consumed / total bits transmitted. The unit of energy efficiency is joules/bit. The lesser the number, the better is the efficiency of a protocol in transmitting the information in the network. This performance matrices gets affected by all the major sources of energy waste in wireless sensor network such as idle listening, collisions, control packet overhead and overhearing.

(ii)Average Delivery Ratio: - The average packet delivery ratio is the number of packets received to the number of packets sent averaged over all the nodes.

(iii)Average Packet Latency: - The average packet latency is the average time taken by the packets to reach to the sink node.

(iv)Network Throughput:-The network throughput is defined as the total number of packets delivered at the sink node per time unit.

III. Methodology and Tecniques

The medium access control protocols for the wireless sensor networks can be classified broadly into two categories: Contention based and Schedule based. The schedule based protocol can avoid collisions, overhearing and idle listening by scheduling transmit & listen periods but have strict time synchronization requirements. The contention based protocols on the other hand relax time synchronization requirements and can easily adjust to the topology changes as some new nodes may join and others may die few years after deployment. These protocols are based on Carrier Sense Multiple Access (CSMA) technique.

- **IEEE 802.11**

The IEEE 802.11 is a well-known contention based medium access control protocol which uses carrier sensing and randomized back-offs to avoid collisions of the data packets. The Power Save Mode (PSM) of the IEEE 802.11 protocol reduces the idle listening by periodically entering into the sleep state. This PSM mode is for the single-hop network where the time synchronization is simple and may not be suitable for multi-hop networks.

- **PAMAS: Power Aware Multi-Access Signaling**

It is one of the earliest contention based MAC protocol designed with energy efficiency as the main objective. In this protocol nodes which are not transmitting or receiving are turned "OFF" in order to conserve energy. This protocol uses two separate channels for the data and control packets. It requires the use of two radios in the different frequency bands at each sensor node leading to the increase in the sensors cost, size and design complexity.

- **TDMA Based MAC protocol**

TDMA Based MAC protocol assumes the formation of clusters in the network. Each of the cluster sensor nodes is managed by the Gateway. The Gateways collects the information from the other sensor nodes within its cluster, performs the data fusion, communicates with the other gateways and finally sends the data to the control center. The assignment of the time slots to the sensor nodes within its cluster is performed by Gateways. The main disadvantage of this protocol is that it requires real clusters and it does not support scalability.

IV. Sensor Mac Protocol

The main goal in our MAC protocol design is to reduce energy consumption, while supporting good scalability and collision avoidance. Our protocol tries to reduce energy consumption from all the sources that we have identified to cause energy waste, *i.e.,* idle listening, collision, overhearing and control overhead. To achieve the design goal, we have developed the SMAC that consists of three major components: periodic listen and sleep, collision and overhearing avoidance, and message passing.

❖ **Network and Application Assumptions**

Since sensor networks are somewhat different than traditional IP networks or ad hoc networks of laptop computers, we summarize our assumptions about sensor networks and applications. We expect sensor networks to be composed of many small nodes deployed in an ad hoc fashion. Sensor networks will be composed of many small nodes to take advantage of physical proximity to the target to simplify signal processing. The large number of nodes can also take advantage of short-range, multi-hop communication (instead of long-range communication) to conserve energy. Most communication will be between nodes as peers, rather than to a single base-station. Because there are many nodes, they will be deployed casually in an ad hoc fashion, rather than carefully positioned. Nodes must therefore self-configure.

We expect most sensor networks to be dedicated to a single application or a few collaborative applications, thus rather than node-level fairness (like in the Internet), we focus on maximizing system-wide application performance. In-network processing is critical to sensor network lifetime. Since sensor networks are committed to one or a few applications, application-specific code can be distributed through the network and activated when necessary or distributed on-demand.

Techniques such as data aggregation can reduce Listen Sleep Listen Sleep time traffic, while collaborative signal processing can reduce traffic and improve sensing quality. In-network processing implies that data will be processed as whole messages at a time in store-and forward fashion, so packet or fragment-level interleaving from multiple sources only increases overall latency.

Finally, we expect that applications will have long idle periods and can tolerate some latency. In sensor networks, the application such as surveillance or monitoring will be vigilant for long periods of time, but largely inactive until something is detected. For such applications, network lifetime is critical. These classes of applications can often also tolerate some additional latency. For example, the speed of the sensed object places a bound on how rapidly the network must detect an object. These assumptions about the network and application strongly influence our MAC design and motivate its differences from existing protocols such as IEEE 802.11.

- **Periodic Listen and Sleep**

In many sensor network applications, nodes are in idle for a long time if no sensing event happens. Given the fact that the data rate during this period is very low, it is not necessary to keep nodes listening all the time. Our protocol reduces the listen time by letting node go into periodic sleep mode. For example, if in each second a node sleeps for half second and listens for the other half; its duty cycle is reduced to 50%. So we can achieve close to 50% energy savings.

Each node 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. The duration of time for listening and sleeping can be selected according to different application scenarios. For simplicity these values are the same for all the nodes.

Our scheme requires periodic synchronization among neighboring nodes to remedy their clock drift. We use two techniques to make it robust to synchronization errors. First, all timestamps that are exchanged are relative rather than absolute. Second, the listen period is significantly longer than clock error or drift.

- **Choosing and Maintaining Schedules**

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. It follows the steps below to choose its schedule and establish its schedule table.

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 seconds. 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 choosing 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 td and rebroadcasts this schedule, indicating that it will sleep in t-td seconds. The random delay is for collision avoidance, so that multiple followers triggered from the same synchronizer do not systematically collide when re-broadcasting the schedule.
3. If a node receives a different schedule after it selects and broadcasts its own schedule, it adopts both schedules (i.e., it schedules itself to wake up at the times of both is neighbor and itself). It broadcasts its own schedule before going to sleep. We expect that nodes only rarely adopt multiple schedules, since every node tries to follow existing schedules before choosing an independent one. On the other hand, it is possible that some neighboring nodes fail to discover each other at beginning due to collisions when broadcasting schedules. They may still find each other later in their subsequent periodic listening.

- **Maintaining 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 periodically update each other their schedules to prevent long-time clock drift. The updating period can be quite long. Updating schedules is accomplished by sending a SYNC packet. The SYNC packet is very short, and includes the address of the sender and the time of its next sleep. The next-sleep time is relative to the moment that the sender finishes transmitting 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 further 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 randomly 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.

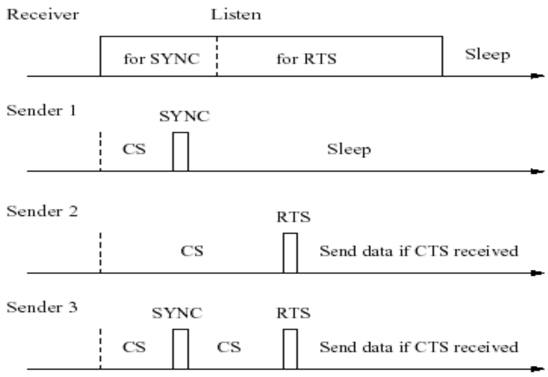


Fig. 3. Timing relationship between a receiver and different senders. CS stands for carrier sense.

❖ **Collision and Overhearing Avoidance**

Collision avoidance is a basic task of MAC protocols. SMAC 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.

Since multiple senders may want to send to a receiver at the same time, they need to contend for the medium to avoid collisions. Among contention based protocols, the IEEE 802.11 does a very good job of collision avoidance. Our protocol follows similar procedures, including both virtual and physical carrier sense and RTS/CTS exchange. We adopt the RTS/CTS mechanism to address the hidden terminal problem. There is a duration field in each transmitted packet that indicates 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 a variable called the network allocation vector (NAV) and sets a timer for it. Every time when the NAV timer fires, the node decrements the NAV value until it reaches zero.

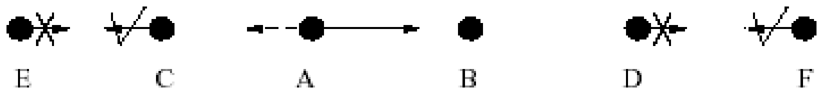


Fig. 4. Who should sleep when node A is transmitting to B?

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. Physical carrier sense is performed at the physical layer by listening to the channel for possible transmissions. The randomized carrier sense time is very important for collision avoidance. The medium is determined as free if both virtual and physical carrier sense indicates that it is free. All senders perform carrier sense before initiating a transmission. If a node fails to get the medium, it goes to sleep and wakes up when the receiver is free and listening again. Broadcast packets are sent without using RTS/CTS. Unicast packets follow the sequence of RTS/CTS/DATA/ACK between the sender and the receiver

❖ **Message Passing**

A message is the collection of meaningful, inter-related units of data. It can be a long series of packets or a short packet, and usually the receiver needs to obtain all the data units before it can perform in-network data processing or aggregation. The disadvantages of transmitting a long message as a single packet are the high cost of re-transmitting the long packet if only a few bits have been corrupted in the first transmission. However, if we fragment the long message into many independent small packets, we have to pay the penalty of large control overhead and longer delay. It is so because the RTS and CTS packets are used in contention for each independent packet.

Our approach is to fragment the long message into many small fragments, and transmit them in burst. Only one RTS packet and one CTS packet are used. They reserve the medium or transmitting all the fragments. Every time a data fragment is transmitted, the sender waits for an ACK from the receiver. If it fails to receive the ACK, it will extend the reserved transmission time for one more fragment, and re-transmit the current fragment immediately. As before, all packets have the duration field, which is now the time needed for transmitting all the remaining data fragments and ACK packets. If a neighboring node hears a RTS or CTS packet, it will go to sleep for the time that is needed to transmit all the fragments. Switching the radio from sleep to active does not occur instantaneously. Therefore, it is desirable to reduce the frequency of switching modes. The message passing scheme tries to put nodes into sleep state as long as possible, and hence reduces switching overhead. The purpose of using ACK after each data fragment is to prevent the hidden terminal problem.

It is possible that a neighboring node wakes up or a new node joins in the middle of a transmission. If the node is only the neighbor of the receiver but not the sender, it will not hear the data fragments being sent by the sender. If the receiver does not send ACK frequently, the new node may mistakenly infer from its carrier sense that the medium is clear. If it starts transmitting, the current transmission will be corrupted at the receiver. Each data fragment and ACK packet also has the duration field. In this way, if a node wakes up or a new node joins in the middle, it can properly go to sleep no matter if it is the neighbor of the sender or the receiver. For example, suppose a neighboring node receives an RTS from the sender or CTS from the receiver, it goes to sleep for the entire message time.

V. Implementation

NS stands for Network Simulator and it is extensively used by network research community. Users can define arbitrary network topologies composed of nodes, routers, links and shared medium. A rich set of protocol objects can then be attached to nodes, usually as agents. NS2 is written in C++ with an Otcl interpreter as a frontend. NS2 is an open-source software and free to use for all users. Due to its open source nature, a user can modify parameters at different layers, create his or her own applications, and develop new protocol. After running the TCL script in the NS2 we get the output in two forms:

1. We can visualize output in NAM editor.
2. We can analyze output using trace file.(For trace file analysis we need a awk programming support)

SIMULATION PARAMETERS USED IN THE NETWORK

Simulator	NS2
Type of channel	Wireless channel
Propagation model	Two Ray Ground
MAC Protocol	S-MAC
Interface Queue Type	Drop Tail Queue
Antenna Type	Omni Direction
Traffic Source	TCP
Routing Protocol	AODV/DSDV
Max. Simulation Time	150 sec
Number Of Nodes	10,20,50,100,150

- **Simulation**

Following are the steps to write TCL file to in NS which is necessary to run the simulation.

1. Define goal and expected result.
2. Create network topology.
   - Nodes
   - Link (simplex/duplex, BW, delay, etc.)
3. Specify agents (TCP, UDP, etc.)
4. Traffic sources (CBR)
5. Simulation scenario (wired, wireless, wired-cum-wireless)

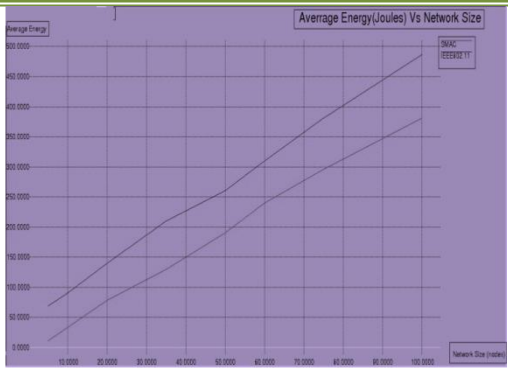
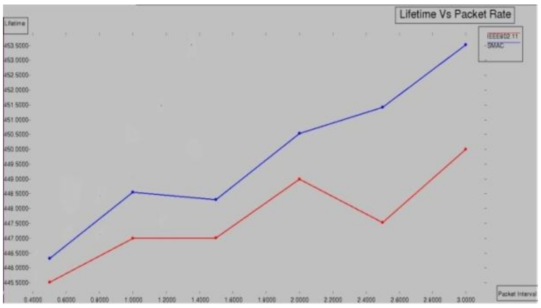
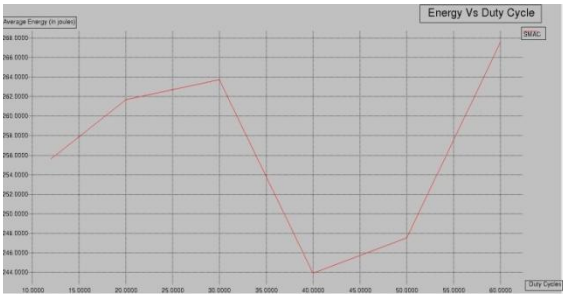


Fig – Average Energy (IEEE802.11 vs SMAC)





VI. Conclusion

Energy is an essential and important factor in lifetime of a sensor network.
The main purpose of wireless network establishment is, sharing of information through communication and energy is the key required for this communication.
The energy efficient protocols contribute to save energy and helps to have better communication network and system.

VII. Acknowledgement

We extend our sincere thanks to our Principal Dr. M. Z. Shaikh for providing us all the resources. We would like to thank Head of Department Prof. P. A. Kharade and project convener Prof. M. M. Bulhe for their constant motivation. We extend our heartfelt thanks to our guide Prof. S.V.Patil for their guidance throughout our project. Lastly we would thank each and every.

VIII. Reference

[1]. Kazem sohraby,Daniel minoli,Taieb znati," Wireless sensor networks" book
[2]. Wei Ye,John heidemann,Deborah estrin,"An energy efficient MAC protocol for wireless sensor networks"
[3]. Koen langendoen,"Medium access control in wireless sensor networks"
[4]. Abdelmalik bachir,Mischa dohler,senior member, IEEE, Thomas watteyne,member,IEEE,and kin k. leung, fellow, IEEE,"MAC essentials for wireless sensor networks".