International Journal of Multimedia and Ubiquitous Engineering
Vol. 4, No. 2, April, 2009

Performance of a Wireless Sensor Network MAC Protocol with a Global Sleep Schedule

Somnath Ghosh, Prakash Veeraraghavan, Samar Singh, and Lei Zhang
Department of Computer Science and Computer Engineering
Latrobe University
Melbourne, VIC 3086
S.Ghosh@latrobe.edu.au

Abstract

Medium access control protocols for wireless sensor networks are almost always designed to be energy efficient. One mechanism used to reduce energy expenditure is to periodically turn off the radio receivers of the sensor nodes in a coordinated manner. The nodes with radio receivers turned off are said to be in the sleep mode. Nodes form virtual clusters based on common sleep schedules. However, protocols like the widely used S-MAC may require some nodes to follow multiple sleep schedules causing them to wake up more often than the other nodes. This paper demonstrates in some wireless sensor networks using S-MAC, a significant proportion of the nodes may have to stay awake much longer than envisaged. A modification of the protocol is then proposed to eliminate the need for some nodes to stay awake longer than the other nodes. The modified version improves the energy efficiency and increases the life span of a wireless sensor network. The paper concludes with the result of simulation studies which indicate that the use of the proposed protocol is expected to increase the life time of wireless sensor networks significantly.

1. Introduction

Wireless sensor networks (WSNs) constitute a special class of wireless data communication networks. A node (called a *sensor node*) in a wireless sensor network is a low cost, resource constrained device. Sensor nodes are typically deployed in large number (hence the requirement to be low cost), and are often positioned randomly. Sensor nodes are generally battery powered. In many applications they are placed in inaccessible locations, making battery replacement impractical. As a consequence, energy efficiency is an important requirement in a medium access control protocol for most wireless sensor networks. The WSN protocols for accessing the shared communications medium have design objectives which are quite different from those used in other types of computer networks. There are many other significant differences between WSNs and other types of computer networks which influence the design requirements of the medium access protocols for WSNs, e.g., wireless sensor networks are usually required to be self-configuring and expandable.

Many medium access control (MAC) protocols for wireless sensor networks have been proposed in the recent years. Most of these protocols have energy conservation as an objective. The pattern of energy use in the sensor nodes, however, depends on the nature of the application. As the range of applications which use WSNs is large and diverse, the proposed protocols display much diversity. Most of these protocols use either a contention based mechanism or a time schedule or a combination of the two for accessing the shared medium.

A protocol proposed by Ye *et al*, named S-MAC [1,2], is a robust medium access control (MAC) protocol for wireless sensor networks. Owing to its success in significant reduction in energy consumption and its robustness, S-MAC has been used in many wireless sensor networks (WSNs). It is one of the networking protocols included in TinyOS, a popular operating system for a number of platforms available as WSN nodes [3]. Many other MAC protocols have been proposed recently which are based on, or inspired by, S-MAC [4,5,6].

S-MAC reduces energy consumption by allowing the nodes to periodically turn off their radio receivers (and any other resources that have no work to do) and enter a low power *sleep* state. The *duty cycle* of a node is the ratio of the time it is *awake* (i.e. not in the sleep state) to the total time. The lower the duty cycle, the lower is the power consumption of a sensor node.

In S-MAC the channel access is contention based, using a scheme similar to the IEEE802.11 distributed coordination function. However, unlike the IEEE 802.11 MAC protocol, the intervals when contention can occur are scheduled. S-MAC, therefore, combines the features of both contention based as well as time scheduled protocols. Even though the contention interval in S-MAC is scheduled, S-MAC requires much looser time synchronisation than TDMA based protocols. This allows the S-MAC nodes to use inexpensive timing hardware and simpler synchronisation algorithms. Furthermore, S-MAC does not suffer from the limited scalability generally associated with TDMA schemes. An outline of S-MAC protocol is given in Section 2.

S-MAC saves energy by requiring the nodes to periodically listen for any communication for a short interval and then allowing them to sleep, if the node is not involved in data communication, for the rest of a pre-determined duration (a *frame*) to conserve energy. The listen interval and the sleep interval for a node occur according to a schedule which it follows. S-MAC has a mechanism which nodes use to learn the sleep schedules of the nodes in their neighbourhood, and using this knowledge they communicate with their neighbours. For successful communication, a transmitting node must transmit when the intended receiver is not sleeping.

The duty cycle of the sleep schedule of a node determines its base rate of energy depletion. The base rate is the energy depletion rate when the node neither transmits nor receives data frames. The actual power consumption depends on the data traffic at the node. If all nodes have the same duty cycle, they all have the same base power consumption. In a wireless sensor network using S-MAC, some nodes may, however, have to wake up and listen more often than once per frame. In Section 3, we show that a significant proportion of the nodes in a WSN may have to wake up more than once per frame. Since all nodes may not have the same duty cycle, some nodes may deplete their energy faster and die earlier. This reduces the average life of the WSN nodes, and hence reduces the average useful life of the sensor network itself. Furthermore, the non-uniform life span of the nodes (nodes which sleep less die earlier) eventually creates areas within the WSN which do not have sensor coverage and adversely affects the network connectivity.

We then propose, in Section 4, a modification in S-MAC which ensures that all nodes keep awake for the same fraction of the time (i.e. have the same duty-cycle.) This improves the energy efficiency of the protocol. The modification does not increase the complexity of the protocol, and retains its robustness.

In Section 5, we present a set of simulation experiments to estimate the improvement in energy efficiency realised through the modifications proposed in this paper.

2. Overview of S-MAC

This section provides a summary of the S-MAC protocol based on a complete description in [2]. Many details of the protocol, which are not relevant to the present work, have not been included in this summary.

2.1 Avoidance of collisions and overhearing

In S-MAC the nodes conserve energy by sleeping (i.e. turning off their receivers and any other hardware resources which are not required while the node is in the sleep state), and wake up periodically to listen and check if any of its neighbours want to communicate with it. The duration of the listen interval and the sleep interval are fixed in a system according to the application requirements. A listen interval followed by a sleep interval comprises a *frame* [Figure 1]. The node which needs to send data initiates the communication process during the listen interval of the intended receiver, and then continue the data transfer during the time that normally is the sleep interval. Other nodes go back to sleep turning their radios off, and thus avoid overhearing and idle listening.

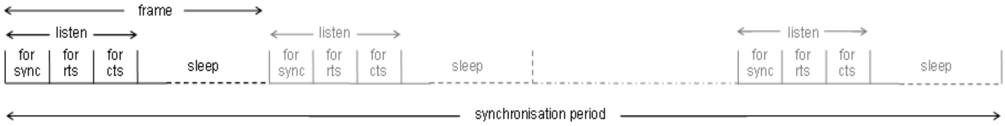


Figure1. S-MAC Frame

As Figure 1 shows, the listen interval is divided into three sub-intervals. Two of these intervals are used for an exchange of *RTS* and *CTS* control frames to avoid the hidden terminal problem in the way the RTS-CTS exchange is used in the IEEE 802.11 MAC protocol [7]. The function of the remaining sub-interval is explained later. The protocol uses both physical and virtual carrier sensing for collision avoidance. To enable virtual carrier sensing the RTS ans CTS frames carry a field indicating the remaining duration of the transmission of the message. A node which overhears a RTS or a CTS frame intended for another node uses this value to determine how long the node must refrain from transmission.

The sub-intervals in the listen interval are slotted. Control frame transmission follows the CSMA/CA protocol. A node with a control frame to transmit randomly selects a slot in the appropriate sub-interval. If it has not detected any transmission by the end of the randomly selected slot, it starts sending the control frame. Once a successful exchange of RTS-CTS control frames has taken place, the transmitter and the receiver stay awake and communicate. The other nodes in the neighbourhood enter the sleep state.

2.2. Choosing and Maintaining Schedules

Since the nodes periodically sleep with their radios turned off, a node must know the listening and sleep schedule of a neighbour with which it wishes to communicate. Nodes exchange their schedules by periodically broadcasting a special control frame, the *sync frame*. There is a sub-interval reserved for *sync* frames in the listen interval. The *sync* sub-interval is slotted like the other sun-interval of the listen interval; and, as is the case with the other control frames, the *sync* control frames are broadcast using the CSMA/CA protocol. The protocol requires each node to broadcast a *sync* frame at least once in a predetermined synchronisation period (Figure 1). A node builds a table of the schedules of its neighbours by listening to the *sync* frames.

Sync frames also enable a node to choose its schedule when it starts. The procedure outlined below is followed by the S-MAC nodes for choosing its schedule and for constructing the table of schedules of its neighbours:

A node first listens to the broadcasts in its neighbourhood for a pre-determined duration which is at least as long as the synchronisation period. If it does not hear a valid *sync* frame (containing a schedule from a neighbour), it arbitrarily chooses a schedule for itself and starts following it. The node also broadcasts a *sync* frame to announce its schedule.

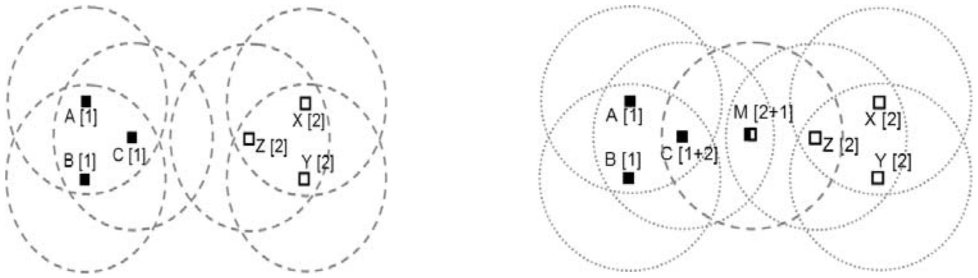
1. If the node receives a schedule from a neighbour before it has chosen or announced a schedule on its own, it adopts the received schedule and starts following it. The node announces this schedule as its own schedule by broadcasting a *sync* frame during its next scheduled listen time.

2. If the node receives a different schedule after it has chosen and announced a schedule, there are two cases to consider:

   a) If the node has no other neighbour, it discards the previously chosen schedule, and starts following the new schedule.If the node has no other neighbour, it discards the previously chosen schedule, and starts following the new schedule.

   b) If the node already follows a schedule with one or more neighbours, it starts waking up at the listen interval of the newly received schedule, in addition to following the earlier schedule (which is the schedule it continue to periodically broadcast as its own in the *sync* frames.)

Once a node has chosen a schedule using the procedure described above, it continues to broadcast this schedule in a *sync* frame at least once in a sychronisation period.



(a) Before (b) After

Figure 2. Sleep schedule before and after node M join the network

2.3 Problem with multiple schedules - longer listen time

As the outline of the procedure for choosing and maintaining the listen and sleep schedule in the previous section indicates, some nodes may have to keep wake during the listen time of more than one schedules. This happens, for example, if a node, when it starts up, finds some of its neighbours following one schedules and the rest following another. The nodes following a shared schedule are said to form a *virtual cluster*. Figure 2 shows an example of this situation. Before node M starts up, two isolated virtual clusters of nodes exist. Nodes A, B and C follow one schedule (schedule 1); and nodes X, Y and Z follow another schedule (schedule 2). The circle around a node indicates the communication range of the node. When M starts, during its initial listening spanning a synchronisation period, it receives *sync* frames corresponding to both the schedules. M will then adopt one of the schedules (e.g. schedule 2) as its own, and announce this schedule in its *sync* frames. However, it will also have to wake up during the listen time of the other schedule. Thus M has higher duty cycle, and consumes more energy.

Our investigation reported in the next section shows that a significant number of nodes in a wireless sensor network using the S-MAC protocol may have to following multiple schedules.

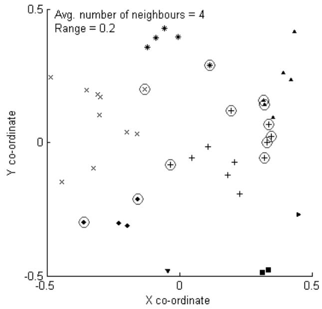


Figure 3. A WSN showing schedules followed by nodes

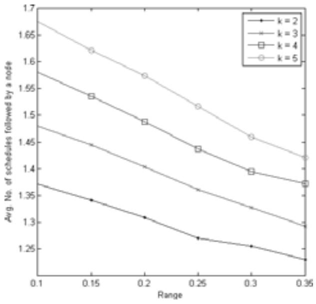


Figure 4. Average number of schedules

3. **Monte Carlo simulation to estimate the occurrence of nodes following multiple schedules**

We simulated a wireless sensor network system in which sensor nodes were randomly placed, with uniform probability density, in a area of the size $1km \times 1km$. The nodes started up in a randomly selected order. The number of nodes in the network was chosen so that the nodes had a specified average number of neighbours, k. The objective of the study was to estimate the fraction of the nodes which are required to wake up and listen during more than one schedule. Figure 3 shows an example of the networks used in the simulation experiments carried out in this investigation. In the wireless sensor network shown in the figure, a node has, on average, four neighbours ($k = 4$). The communication range of a node is 0.2 km (i.e. a node can directly send frames to nodes up to 0.2 km away, and can directly receive frames from nodes up to 0.2 km away.) The nodes are shown by symbols like +, e, ×, W etc. The nodes which follow a common schedule share a symbol. The nodes which are required to wake up during more than one schedule have a small circle drawn around them. In this system there are 40 nodes. Twelve of these nodes follow two schedules. In other words, the average number of schedules followed by a node in this wireless sensor network is 1.3. Contrary to the assumption stated in [2], the occurrence of nodes which have to follow multiple schedules is not rare. Nodes which have to follow multiple schedules will have higher energy consumption, and hence shorter life span. The results of comprehensive simulations reported next show that some nodes in our simulation need to follow as many as four schedules, making these nodes very short lived.

Monte Carlo simulations of the type described earlier in this section were conducted for sensor network systems with nodes with different radio ranges and with different value of k, the average number of neighbours for a node. For each combination of the two parameters, 200 wireless sensor networks were generated by randomly placing the required number of nodes in a $1km \times 1km$ area. For each system the average value of the number of schedules followed by the nodes was computed. This value was then averaged over the 200 randomly generated systems with the same parameters.

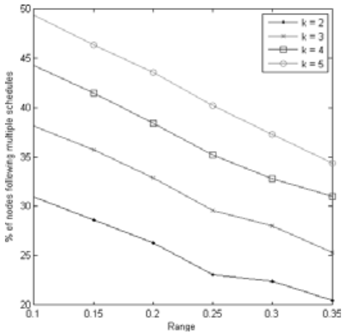


Figure 5. Nodes following multiple schedules

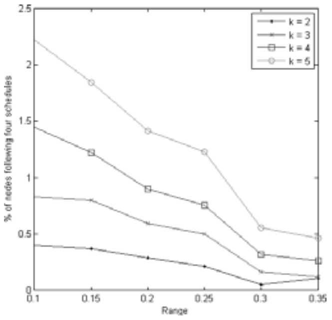


Figure 6. Nodes following 4 schedules

The results are shown in Figures 4, 5 and 6. The average number of schedules followed by a node in the simulated WSNs is significantly higher than 1, specially in a multi-hop system with low energy budget nodes where the radio range of the nodes is small. On average a node keeps awake longer (has a higher duty cycle) than would be the case if it followed only one schedule. What is more serious is the fact that all nodes do not follow the same number of schedules. The nodes which follow a larger number of schedules deplete their battery much earlier than others, causing holes in the sensor coverage and routing difficulties. Figure 5 shows that in some of the systems in our investigation as many as 50% of the nodes had to follow multiple schedules. Figure 6 shows that a few of the nodes followed as many as four schedules.

4. Proposed modification

The previous section establishes that a WSN using the S-MAC protocol may have a high proportion of its node waking up during more than one schedules. This has a significant adverse impact on the life span and connectivity of a WSN with energy constrained nodes. In this section we propose a modification of the S-MAC protocol. The modified protocol requires that when connectivity is established between two (or more) isolated virtual clusters (each following an independently chosen schedule) due to the introduction of a new node in the common neighbourhood of the clusters, all nodes of these clusters form a single cluster by adopting the schedule of one of the clusters. This process of merger of clusters ensures that, except for the short period when clusters are merging, nodes follow exactly one cycle, avoiding the problems associated with multiple schedules.

When clusters merge, the schedule of one of the merging clusters is chosen to be the common schedule of the newly formed larger cluster. The process of merger of clusters requires the schedules followed by the individual clusters to be identified. In the modified version of S-MAC, we propose that the schedules be identified by the unique node identifier of the node which created the schedule. The schedule identifiers chosen this way are unique and linearly ordered. When previously isolated clusters merge, the common schedule chosen by the merged cluster is the schedule with the highest identifier.

4.1. Frame Format

The *sync* frame in S-MAC has a field which holds the node identifier of the sender, and another field which indicates the end of the current sleep time. These fields are used by the receivers to discover the identity of the sender (neighbour discovery) and to learn its sleep schedule. The modified version of S-MAC presented here requires an additional field in the *sync* frame to hold the identifier of the schedule followed by the sender. Every *sync* frame not only identifies the sender, it also identifies the schedule followed by the sender. This is the only modification required in the format of the S-MAC frames.

4.2. Procedure for choosing and maintaining schedules

With this change in the *sync* frame, the procedure for choosing and maintaining sleep schedules in the modified version of S-MAC is as follows:

A node first listens to the broadcasts in its neighbourhood for a pre-determined duration which is at least as long as the synchronisation period. If it does not hear a valid *sync* frame (containing a schedule from a neighbour), it arbitrarily chooses a schedule for itself and starts following it. *This schedule is assigned the node identifier of the node as the schedule identifier*. The node also broadcasts a *sync* frame to announce its schedule. The *sync* frame contains the schedule identifier in one of its fields.

1. If the node receives schedules from its neighbour before it has chosen or announced a schedule on its own, it chooses as its own schedule the received schedule with the highest identifier, and tries to announce this schedule as its schedule using *sync* frames during the scheduled listen times of each of the received schedules. After transmitting *sync* frames advertising its chosen schedule during the listen time of all received schedules, the node starts following and advertising the chosen schedule only.

2. If the node receives a different schedule after it has chosen and announced a schedule, there are two cases to consider:

    a) If the schedule identifier of the received schedule is higher than the schedule identifier of the currently chosen schedule, it advertises the new schedule as its schedule using *sync* frames during the listen time of both schedules (the new one as well as the previous one) and then discards the previously chosen schedule, and starts following and advertising only the new schedule.

    b) If the schedule identifier of the received schedule is no higher than the schedule identifier of its current schedule, it transmits a *sync* advertising its current schedule during the listen time of the received schedule. After this, the received schedule is ignored, and the node continues to follow and advertise the schedule it has been following. (This case is not normal. It can occur only as a consequence of loss of an earlier *sync* frame due to collision.)

Having determined its listen and sleep schedule, a node continues to broadcast this schedule in a *sync* frame at least once in a sychronisation period.

Since control frame transmission is contention based, it is possible for a node to miss a *sync* frame transmitted by one of its neighbours. For instance, a *sync* frame may get corrupted by a collision or interference. In order that such missed *sync* frames do not cause two neighbours to fail to discover each other for ever, the initial neighbour discovery (which every node performs at start up by listening to the medium for a whole synchronisation period) has to be carried out by each node periodically. This process requires the node to stay awake and listen for *sync* frames for at least a whole synchronisation period, and consequently, consumes a large amount of energy. The process should not be carried out very frequently. The appropriate frequency for this process depends on the application and the operating environment of the WSN. The frequency should be determined on the basis of the application characteristics and the nature of the operating environment. This periodic neighbour discovery process exists in the original S-MAC.

To distinguish the modified version of the protocol presented in this section from the original S-MAC protocol, we refer to the modified version in the rest of the paper by the acronym S-MACL (for *S-MAC* Globa*L*).

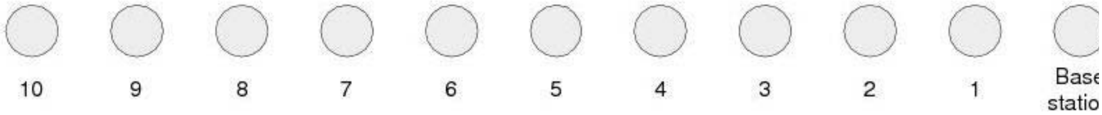


Figure 7. Scenario 1–Linear topology

5. Simulation and performance analysis

The modified protocol, S-MACL, eliminates the need for any node to follow multiple schedules. The nodes which would have depleted their energy because of having to follow multiple schedules under S-MAC, are expected to last longer under S-MACL. As wireless sensor network using S-MACL is expected to have a longer life time. This section describes the simulation studies conducted to estimate the extension gained in life time of a senson network when S-MACL is used.

5.1 Performance estimation using simulation

Network Simulator version 2 (*NS-2*) has been used as experiment platform. *NS-2* provides extensive support for queuing algorithms, routing protocols, multi-cast protocols and IP protocols over both wired network and wireless network. For the present work, S-MACL module for *NS-2* was implemented. The S-MACL module for *NS-2* is based on the built-in S-MAC module in *NS-2*.

We have simulated wireless sensor networks with regular topology as well as randomly generated topology. In our simulations, each node in the sensor networks start with a fixed amount of energy. We simulated each sensor network once with S-MAC and then with S-MACL. In these studies we recorded the remaining energy of each node in the trace file generated by *NS-2*. From the trace files the following values of interest were computed:

1) the life span of the node which depleted its energy (died) first,
2) the average life span of the nodes, and
3) the total energy consumption of the network.

5.2 Simulation parameters

1) **S-MAC/S-MACL parameters**: Sleep wake up cycles is enabled with a duty cycle 10. Every node can handle up to 20 neighbours and up to 10 different schedules. DropTail is used as queuing algorithm.

2) **Radio parameters**: Using a omni-directional antenna which is 1.5 meters about the node. Radio module is a 914MHz Lucent WaveLAN DSSS radio interface. Prorogation module is TwoRayGround module. In this case, the transmission range is 250 meters and carrier sensing range is 550 meters.

3) **Network topology**: The distance between two one-hop away neighbour nodes is up to 250 meters which is the maximum transmission range of nodes, which means a nodes can only talk with its one-hop away neighbours. There is one base station in a network.The following topologies were used in the simulations:

    a. nodes deployed in a straight line,
    b. nodes deployed in a matrix, and
    c. randomly generated topology.

4. **Traffic model**: Every non-base station nodes keep sending a 500 bytes packet to the base station every 200 seconds.

5. **Energy model**: The base station used in simulation has infinite energy. The default energy model of wireless node in *NS-2* is used in simulations. The different levels of energy consumption for different operations are defined in the following:

    a. idle state: 0.05 watt
    b. sleep state: 0.001 watt
    c. state transition from sleep to idle: 0.1 watt
    d. transition time (sleep to idle): 0.005 sec
    e. receive power: 0.5 watt
    f. transmit power: 0.5 watt

6. **Routing**: Static routing was used in the simulations. Using static routing allowed us to focus on MAC layer energy efficiency.

5.3 Results

This section reports the results of the simulation studies conducted to estimate the improvement realised by S-MACL in energy efficiency and the consequent extension in the life time of the network.

5.3.1. Network life

The energy depletion time (time at which the energy available at a node becomes zero) of each node is extracted from trace files generated by the simulation runs. Then the average life time of nodes are calculated. Also the life time of the first node to deplete its energy is also recorded.

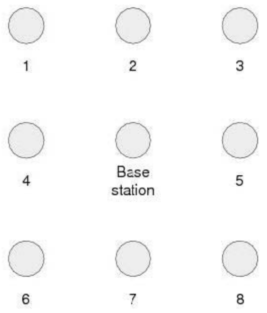


Figure 8. Matrix with 8 Nodes

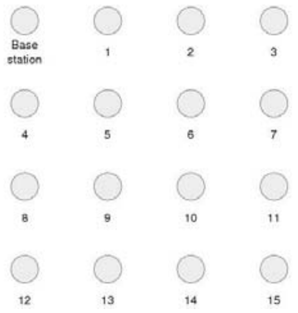


Figure 9. Matrix with 15 Nodes

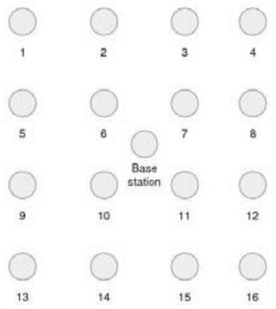


Figure 10. Matrix with 16 Nodes

Table 1. Performance in Linear Topology

No. of nodes	MAC protocol	No. of collisions	T_a	T_f
10	S-MAC	247	5256	3691
10	S-MACL	6	5615	3657

Table 2. Performance in matrix topology

No. of nodes	MAC protocol	No. of collisions	T_a	T_f
8	S-MAC	816	5413	4231
8	S-MACL	250	7212	6834
15	S-MAC	2066	3771	2428
15	S-MACL	1674	5303	3527
16	S-MAC	1815	3838	2395
16	S-MACL	881	5352	4323

In the simplest network topology investigated, all the nodes are placed in a straight line. The network topology is shown in Figure 7. The average lifetime of nodes T_a and the time at which the earliest node to exhaust its energy does so, T_f , are shown in Table 1. In this case, the performance of S-MACL, in terms of the life time of the nodes, is not significantly different from the performance of SMAC. The reason is that in this simple network topology, every node has at most two neighbours, limiting the possible multiplicity of sleep schedules to 2. However, S-MACL is still better than S-MAC if the number of collisions happened in network is considered (collisions increase the latency in message delivery.) In the simulation reported here, the number collisions in S-MACL is 6 compares to 247 in S-MAC.

In the next topology investigated the nodes were placed on a square matrix.. The network topologies used are shown in Figures 8 to 10. The sensor network in Figure 8 has eight sensor nodes and the base station forming a $3x3$ matrix with the base station at one of the corners. The network in Figure 9 has 15 sensor nodes and the base station forming a $4x4$ matrix with the base station at one of the corners. Figures 10 shows a sensor network of 16 nodes placed in a $4x4$ matrix with the base station located at the centre.

The simulation result for the matrix topology in Table 2 shows that S-MACL achieves a considerable improvement in performance over S-MACL. For example, the average lifetime of nodes in the network in Figure 10 is 5352 seconds in S-MACL and 3838 seconds in S-MAC. The average lifetime increase nearly 40 percent in S-MACL.

Figure 12 shows another important attribute T_f , which is the time at which the node which exhausts its energy first does so.

In the simulation conducted in the network in Figure 10, T_f is 4323 seconds in S-MACL which is nearly two times the corresponding value of 2395 seconds in S-MAC. The early dying nodes in S-MAC are usually the border nodes which relay packets between clusters of nodes following different sleep schedules. These nodes have to follow multiple sleep schedules, and consequently, have to stay awake longer. This is the reason why border nodes die earlier in S-MAC. After some border node near base station dies, there may be some nodes can not connect to base station at all. Although these nodes are still alive, they are actually in a isolated island. In this situation, the area earlier covered by these nodes is no longer not covered. This simulation indicates that the global energy efficiency achieved by S-MACL using global sleeping schedule can greatly increase the life span of a wireless sensor network.

5.3.2 Network energy consumption

The energy consumption in sensor networks using S-MAC and S-MACL protocols is investigated in following simulations. In these simulations the initial energy of each node was set to 1000 joules. Then after 10000 seconds simulation time, the energy left of each node was calculated. The same network topologies are used as in last section. The average energy consumption of the nodes in the network E_a and the maximum energy consumption by a node E_m were calculated from trace files. The results are listed in Table 3. The results in this table are consistent with the results of the simulations reported earlier in this paper. As is to be expected, S-MACL consumes less energy than S-MAC. However, the magnitude of difference depends on the topology. As seen in the earlier simulations as well, the difference between S-MAC and S-MACL in the network with linear topology (10 nodes) is not significant. In the networks with the matrix topology, the difference is significant. In these simulations, it can be observed that up to 35 percent energy can be save by S-MACL compares to S-MAC. The results in Table 3 suggest that a border node in some sensor networks using S-MACL may consume half the energy consumed in an S-MAC network with the same topology. Thus a border node's life time may be doubled in S-MACL. Extending the life of border nodes can keep network functional (connected) longer. The results shown in Figure 14 support the same inference.

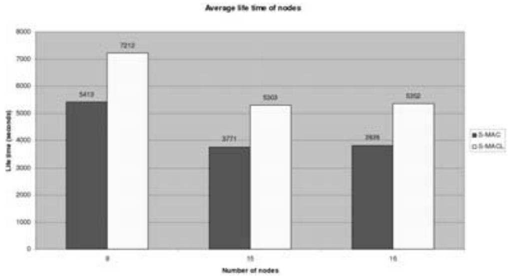


Figure 11. Average lifetime

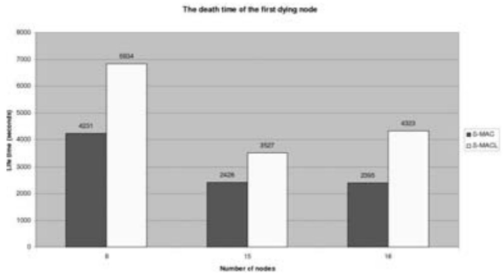


Figure 12. Time to first energy depletion

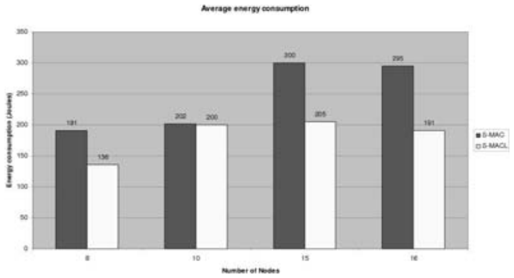


Figure 13. Average energy consumption

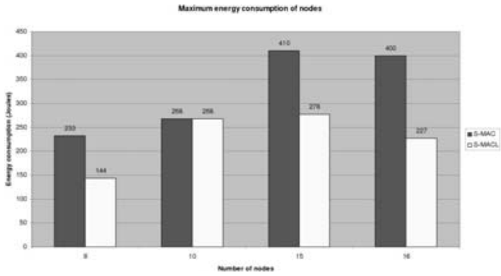


Figure 14. maximum energy consumption

Table 3. Energy consumption in the 8, 10, 15, and 16 node networks

No. of Nodes	MAC protocol	No. of Collisions	E_a	E_m
8	S-MAC	1907	191	233
8	S-MACL	391	136	144
10	S-MAC	531	202	268
10	S-MACL	6	200	268
15	S-MAC	6803	300	410
15	S-MACL	3629	205	278
16	S-MAC	6712	295	400
16	S-MACL	1840	191	227

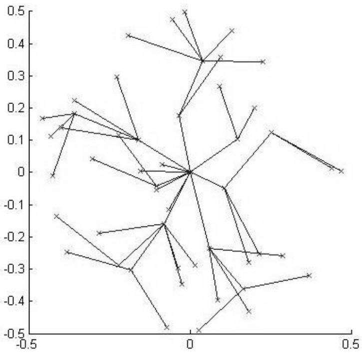


Figure 15. Random topology with 50 nodes

5.3.3. Simulations with randomly generated topology

The simulations reported so far in this paper were conducted on selected regular topologies. We also conducted simulation experiments with sensor networks with randomly places sensor nodes. Random network topologies were generated by placing 50 nodes randomly (with uniform probability density) in a circular area with a 0.5 km radius. Static routing along the shortest paths was used in these simulation experiments. A sample network topology is shown in Figure 15. The crosses represent sensor nodes. The edges between nodes are their static routing path. A base station is located at the center of the circle.

Simulations were repeated 10 times using different randomly generated sensor networks of 50 nodes as described in the previous paragraph. The average values of the performance figures are shown in Table 4. The results in the table clearly indicate the performance improvement realised by S-MACL by eliminating multiple sleep schedules.

Table 4. Performance in randomly generated network

No.of Nodes	MAC Protocol	No. of collisions	T_a	T_f
50	S-MAC	6065	2248	1534
50	S-MACL	3697	3697	2347

The simulation results presented above clearly indicate that S-MACL can greatly prolong life time of a wireless sensor network. In S-MAC, the short life time of border nodes is due to multiple sleep schedules used in the network. The early death of these border nodes reduces the network's coverage badly. In S-MACL, all nodes consume less energy, especially for the borders who have to act as intermediate routers. The global sleeping schedule used in S-MACL can greatly increase the life time of border nodes. Under very simple network topology, even S-MACL's performance is as S-MAC, S-MACL reduce the number of collisions in a network. The aim of simulation is achieved. S-MACL saves more energy than S-MAC in most of the scenarios.

6. Conclusion

This paper proposes a modified version of the S-MAC protocol which is more energy efficient than the original version by Ye *et al* [1,2]. The modification does not change the mechanism for transmission and reception of the data frames. It only changes the procedure for determining the sleep schedules of a nodes at the time a node starts up. In the original S-MAC, the nodes form virtual clusters around shared schedules. As the simulation results reported in this paper demonstrates, a sensor network using the original S-MAC is likely to have many virtual clusters, and a significant proportion of nodes belong to multiple virtual clusters. The nodes which belong to multiple clusters have to follow multiple sleep schedules. These nodes, therefore, have a higher duty cycle than the nodes which belong to only one virtual cluster. In the modified version of the protocol presented in this paper, a node which discovers the existence of two (or more) isolated virtual clusters, triggers a change of sleep schedules in some of the nodes so that the isolated clusters merge into one cluster which follows a single sleep schedule. In this version a node follows only one schedule except for the short intervals when isolated clusters merge. In the proposed version of S-MAC all nodes have the same, low, duty cycle. As a consequence, not only is the base power consumption of all nodes low, all nodes have the same life span, giving the wireless sensor network a longer useful life.

The paper concludes with simulation studies carried out on a set of wireless sensor networks which included networks with regular topologies as well as networks with randomly placed nodes. Results of these simulations indicate that the use of S-MACL, the proposed protocol, is expected to increase the life time of wireless sensor networks significantly.

References

[1] W.Yee, J. Heidemann and D. Estrin, “An energy efficient MAC protocol for sensor networks,” in *INFOCOMM 2002, Twenty-first Annual joint conference of the IEEE computer and Communication Socities proceedings, IEEE,* vol.3, pp. 1567-1576 (June 2002)

[2] W.Ye, J. Heidemann and D. Estrin, “Medium access control with coordinated adaptive sleeping for wireless sensor networks”, *IEEE/ACM Transactions on Networking*, vol. 12, pp. 493-506 (June 2004)

[3] P. Levis, S. Madden, D.G. Joe Polastre, R. Szewczyk, A. Woo, E. Brewer and D. Culler, “ The emergence of networking attractions and techniques in TinyOS”, *Proceedings of the First USENIX/ACM symposium* (2004)

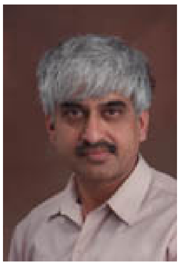
[4] T. Van Dam and K. Langendoen, *An adaptice energy efficient MAC protocol for wireless sensor networks, Proceedings of the 1st International conference on Embedded Network sensor systems,* (2003)

[5] P. Lin, C. Qiao and X. Wang, *Medium access control with a dynamic duty cycle for sensor networks, IEEE Wireless Communications and Networking conference 2004, WCNC,* vol. 3, pp. 1534-1539 (March 2004)

[6] C. Suh and Y-B. Ko, *A traffic aware energy efficient MAC protocol for wireless sensor networks, IEEE International symposium on Circuits and Systems,* vol. 3, pp. 2975-2978 (May 2005)

[7] *Wireless LAN medium access control and physical layer specifications,* IEEE 802.11 std. (1999)

Authors



Somnath Ghosh obtained his B.Tech. (Electrical Engg.) degree from Indian Institute of Technology, Kanpur, M.Tech. degree from Indian Institute of Technology, Bombay and PhD from Strathclyde University, Glasgow. He has worked for the Department of Atomic Energy of India, and taught at Indian Institute of Technology, Delhi. He currently works as a Senior Lecturer in the Department of Computer Science and Computer Engineering, La Trobe University.

Prakash Veeraraghavan obtained his PhD from Indian Institute of Technology, Chennai, India. His research interest includes Algorithms, optimization, protocols and security. He is currently working as a senior lecturer in the Department of Computer Science and Computer Engineering, La Trobe University.

Samar Singh obtained his B.Sc (Hons) in Physics and M.Sc in Operational Research from the University of Delhi and his PhD from IIT, Delhi, India. From 1975 to 1977 he was a Humboldt Research Fellow at the University of Erlangen-Nuremberg. From 1977 to 1986 he worked in Government and Industry in India, from 1986 to 1992 he was with the department of Computer science and engineering, IIT, Delhi, India and from 1992 to 2007 with the department of computer science and computer engineering, La Trobe University. Since 2007 he is an adjunct associate professor with La Trobe University.