Building an Energy-Efficient Prediction S-MAC Protocol for Wireless Networks

Mahmoud Abdel-Aziz El-Sakhawy Othman
Information Technology Sector,
Egyptian Financial Supervisory Authority (*EFSA*)
Cairo, Egypt
mahmoud.elsakhawy@efsa.gov.eg

Prof. Dr. Imane Aly Saroit Ismail
Vice Dean for Education & Student Affairs,
Faculty of Computers & Information, Cairo University
Giza, Egypt
iasi63@hotmail.com

***Abstract*—With the rapid development of wireless networking and micro-electro-mechanical systems (*MEMS*), wireless sensor networks (*WSNs*) have been immerged. WSNs consist of large amount of small, low-end, resource constrained devices, called sensors. Since sensor nodes are usually intended to be deployed in unattended or even hostile environments, it is almost impossible to recharge or replace their batteries. One of the most important research issues in the wireless sensor networks is to extend the network lifetime by energy efficient battery management. So, there are a lot of approaches that are designed to reduce the power consumption of the wireless sensor nodes. In this paper; a new protocol named "*prediction S-MAC protocol*" is proposed to reduce the power consumption of the wireless sensor nodes and to improve their performance compared to the previous S-MAC protocols.**

Keywords - Wireless sensor network; Sensor medium access control (S-MAC) protocol; periodic listen and sleep; adaptive listen, prolong listen, prediction S-MAC protocol.

I. INTRODUCTION

The term wireless networking refers to technology that enables two or more computers to communicate using standard network protocols and utilizing radio waves to maintain communication channels between computers.

Wireless sensor networking is an emerging technology that has a wide range of potential applications including environment monitoring, smart spaces, medical systems and robotic exploration. Such networks consist of large numbers of distributed nodes that organize themselves into a multi-hop wireless network [1 - 4].

Since wireless sensors are usually intended to be deployed in unattended or even hostile environments, it is almost impossible to recharge or replace their batteries [5]. The lifetime of a sensor node is much dependent on its power consumption. Hence, energy efficiency is of highly concern to the wireless sensor network design. So, there are a lot of approaches designed to reduce energy consumption in the wireless sensor networks [6 - 9]. Periodic listen and sleep protocol [6, 7], adaptive listen protocol [8] and prolong listen protocol [9] are examples of these protocols.

In the periodic listen and sleep protocol [6, 7], the time of each node is divided into two successive intervals; listen intervals and sleep intervals. In the sleep interval; a node sleeps completely to preserve its power consumption. It turns off its radio and sets a timer to awake itself later to see if any other node wants to talk to it during listen time. This method decreases the average nodes power consumption, but increases average packets delay.

Adaptive listen protocol [8] is a modification of the periodic listen and sleep protocol; it reduces the packets' delay (*resulted in the periodic listen & sleep protocol*) by reducing the time spent in idle listen. Its basic idea is to let the node that is going to enter its sleep mode and overhears its neighbor's transmissions (ideally only RTS or CTS) wakes up for a short time at the end of the transmission. In this way, if the node is the destination node, its neighbour is able to immediately pass the data to it instead of waiting for its scheduled listen time.

Prolong listening protocol is proposed to improve the performance of the two previous protocols [9]. It also reduces the time spent in idle listen but by a greater value. It uses the concepts of both the periodic listen & sleep protocol and the adaptive listen protocol. In addition, if no RTS or CTS are heard before the node goes to its sleep mode, it sends a Ready To Receive (*RTR*) message to all its neighbours asking them if they are going to send in a short period of time (*prolong listen time*). If the node gets an answer, it exceeds its listen interval by a prolong listen time, on which it can send and receive instead of waiting for its scheduled listen time, so its neighbour is able to pass the data to it immediately instead of waiting for its scheduled listen time. If the node doesn't receive any answer, it will go to sleep until its next scheduled listen time. Results showed that prolong listen protocol increases both throughput and nodes life while decreases both delay and power consumption compared to periodic listen & sleep and adaptive listen protocols [9].

In this paper, a new S-MAC protocol named; "*prediction S-MAC protocol*" is proposed to improve the performance of the previous S-MAC protocols. Its basic idea is to divide the whole time of the node into two successive intervals; working interval (*listen interval*), in which the node is expected to send or receive packets and non-working interval (*sleep interval*), in which the node is not expected to send or receive packets.

The remainder of this paper is organized as follow: in the second section, medium access control for wireless sensor networks (*S-MAC*) and sources of energy waste in wireless

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

networks are illustrated. In addition, three existing S-MAC protocols; periodic listen & sleep protocol, adaptive listen protocol and prolong listen protocol are explained. The proposed prediction S-MAC protocol is explained and illustrated by detailed example in part three. Protocols implementations, parameters evaluation and results of the compared algorithms are discussed in part four. Finally, conclusion and future trends are given in section five.

II. MEDIUM ACCESS CONTROL FOR WIRELESS SENSOR NETWORKS (*S-MAC*)

Medium Access Control (*MAC*) is a sub layer of the Data Link Layer of the seven layer Open Systems Interconnection (*OSI*) model. This layer is responsible for controlling the access of nodes to the medium to transmit or receive data. Sensor medium-access control protocols (*S-MAC*) are *MACs* designed for wireless sensor networks. The main task of the *S-MAC* protocol is to organize how the nodes in the *WSN* access the radio between nodes that are in radio range of each other. The most important attributes of S-MAC protocols to meet the challenges of the sensor network and its applications are; *collision avoidance, energy efficiency, scalability, channel utilization, latency, throughput and fairness* [6 - 8].

A. Energy Efficiency in MAC Protocols

Energy efficiency is one of the most important issues in wireless sensor networks. To design an energy-efficient *MAC* protocol, the following question must be considered: what causes energy waste from the *MAC* perspective?. The following sources are major causes of energy waste [8]:

- ***Collision*** is the first source of energy waste. When two packets are transmitted at the same time and collide, they become corrupted and must be discarded. Follow-on retransmissions consume energy too. All *S-MAC* protocols try to avoid collisions one way or another.
- ***Idle listening*** happens when the radio is listening to the channel to receive possible data. The cost is especially high in many sensor network applications where there is no data to send during the period when nothing is sensed.
- ***Overhearing*** occurs when a node receives packets that are destined to other nodes. Overhearing unnecessary traffic can be a dominant factor of energy waste when traffic load is heavy and node density is high.
- ***Control packet overhead*** represents transmission and reception of control packets consume energy.
- ***Overmitting*** is the last source of energy waste, which is caused by the transmission of a message when the destination node is not ready. Given the facts above, a correctly-designed MAC protocol should prevent these energy wastes.

B. Studied three existing S-MACprotocols

Sensor MAC protocols achieve an energy saving by controlling the radio to avoid or reduce energy waste from the above sources of energy waste. Turning off the radio when it is not needed is an important strategy for energy conservation. In this part, three existing S-MAC protocols are explained; periodic listen and sleep, adaptive listen and prolong listen. These protocols have techniques in order to reduce the nodes' power consumption.

1) Periodic Listen and Sleep Protocol: This method was first proposed in [6] to reduce the power consumption of each node. It uses the fact that some nodes are idle for long time; means that the data rate is very low, so it is not necessary to keep nodes listening all the time. Periodic listen and sleep protocol reduces the listen time by putting nodes into periodic sleep state.

The basic scheme is shown in figure 1. Each node sleeps for some time, and then wakes up and listens to see if any other node wants to talk to it. During sleeping, the node turns off its radio, and sets a timer to awake it self later. A complete cycle of listen and sleep is called a *frame*. The listen interval is normally fixed according to physical-layer and *MAC-layer* parameters. The duty cycle is defined as the ratio of the listen interval to the frame length [7].

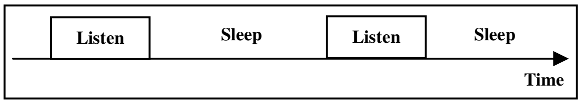

Figure 1. Periodic listen and sleep protocol.

All nodes are free to choose their own listen/sleep schedules, meaning that neighboring nodes may have different schedules. It should be noticed that not all neighboring nodes can synchronize together in a multi-hop network. Nodes exchange their schedules by periodically broadcasting a *SYNC* packet to their immediate neighbors, thus ensuring that all neighboring nodes can communicate even if they have different schedules. A node talks to its neighbors at their scheduled listen time, for example, if node *A* wants to talk to node *B*, it must wait until *B* is listening.

***Advantage*:** The scheme of periodic listen and sleep is able to significantly reduce the time spent on idle listening when traffic load is light, so the power consumption is reduced.

***Disadvantage*:** The downside of the scheme is the increased delay due to the periodic sleeping, which can accumulate on each hop.

2) Adaptive listen Protocol: The adaptive listen protocol was proposed in [7] to improve the delay caused by the periodic sleep of each node in a multi-hop network. It is modification of the periodic listen and sleep protocol, the basic idea is to let the node whose sleep interval is about to start and overhears its neighbor's transmissions (*ideally only RTS or CTS*) wakes up for a short period of time at the end of the transmission. In this way, if the node is the destination node, its neighbor will be able to immediately pass the data to it instead of waiting for its scheduled listen time, other nodes will go back to sleep until its next scheduled listen time. SYNC packets are sent at scheduled listen time to ensure all neighbors can receive it.

For example in figure 2, nodes 2 and 7 are about to enter their sleep interval, but node 1 has a packet to send to node 2, so it sends a *RTS*. All nodes in node 1's range; 2, 7 and 9 hear the transmission so nodes 2 and 7 will extend their listen interval to receive the RTS (*node 9 is already in the listen interval*). After receiving the *RTS*, node 2 will extend its listen interval to serve the packet (*sends CTS, receives data and sends an ACK*), while node 7 doesn't have to extend its listen interval any more, so it enters its sleep interval.

***Advantage*:**

Adaptive listen protocol improves throughput and decreases delay & power consumption compared to periodic listen and sleep protocol.

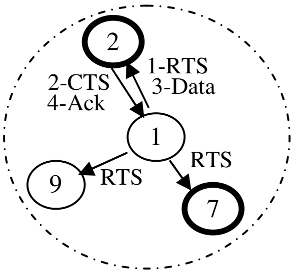

Figure 2. Adaptive listen protocol

***Disadvantage*:**

Since any packet transmitted by a node is received by all its neighbours even though only one of them is the intended receiver, it is clear that all nodes that overhear their neighbour's transmissions (*RTS* or *CTS*) wake up until they discover that the transmission is not for them although only one node is intended.

3) Prolong Listen Protocol: Prolong listening protocol is proposed in [9], which is a modification of both the periodic listen and sleep and adaptive listening protocols to improve their performance. This method takes the benefits of the two previous methods: first, it uses periodic listen and sleep concept and second, nodes that overhear RTS or CTS from its neighbors extend its listen interval to be able to receive packets instead of letting them wait for its scheduled listen time. The new part is; if no RTS and CTS are heard before the node goes to its sleep mode, it sends a ready to receive (*RTR*) message to all its neighbors asking them if they are going to send in a short period of time (*prolong listen time*). If the node gets an answer, it will exceeds its listening interval by a prolong listen time, on which it can send and receive, so its neighbor is able to immediately pass the data to it instead of waiting for its scheduled listen time. If the node doesn't receive any answer, it will go to sleep until its next scheduled listen time [9].

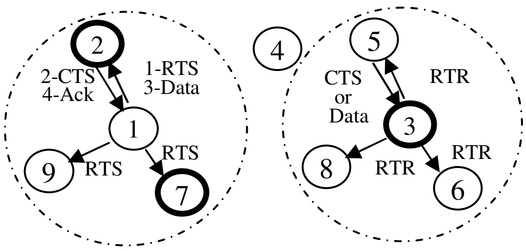

Figure 3. Prolong listen protocol.

For example, in figure 3, nodes 2, 3 and 7 are about to enter their sleep interval. But since node 2 hears a *RTS* from node 1, so it extends its listen interval to serve the packet as in the adaptive listen protocol. While node 3 hears nothing, so it sends a *RTR* message to all its neighbors. All nodes in node 3's range; 5, 6 and 8 hear the transmission. Node 5 responds (*by sending a RTR reply or by just sending the data*), so node 3 prolong its listen interval to serve the packet. Note that, node 4 does nothing because it is out of range.

***Advantage*:** Since prolong listen protocol services a lot of packets during prolong listen time instead of letting them wait for their next scheduled listen time, so it improves throughput and decreases delay and power consumption compared to periodic listen and sleep protocol and adaptive listening protocol.

***Disadvantage*:** It is clear that all nodes that overhear their neighbour's transmissions (*RTS or CTS*) wake up until they discover that the transmission is not for them although only one node is intended.

It should be noted that not all next-hop nodes can overhear a *RTR* message from the transmitting node because they are not at the scheduled listen time or they do not have data packets to send. So if a node starts a transmission by sending out an *RTR* message during prolong listen time, it might not get a reply. In this case, it just goes back to sleep and will try again at the next normal listen time and a *RTR* message consume energy too.

III. PROPOSED PREDICTION S-MAC PROTOCOL

The previous S-MAC protocols are based on initial listen & sleep intervals, where listen time in each frame is fixed usually about 300 msec, while the sleep time can be changed to reflect different duty cycles. The downside of the previous schemes is the increased delay due to the periodic sleeping which is accumulated on each hop. In addition, during listen intervals, nodes may have no data to transmit / receive (*idle*) or service their data in a partial time of the listen intervals. These techniques imply to minimize the sensor node lifetime.

In this section; we propose a new protocol named *"prediction S-MAC protocol"* to handle the problems of the previous S-MAC protocols. It does not depend on fixed listen and sleep intervals. Instead the node transmits only (*send/receive*) according to the prediction of its listen intervals, otherwise it goes to sleep mode and turns off its radio until expectation of its next listen interval. The basic idea of the proposed protocol is to divide the whole time of the node into two successive intervals; working interval (*listen interval*), in which the node is expected to send or receive packets and non-working interval (*sleep interval*), in which the node is not expected to send or receive packets.

Confidence interval method is used to predict the working and non-working intervals based on the last previous *N* listen (*working*) intervals. It is expected that the proposed prediction S-MAC protocol will increase both throughput and nodes' life, while it will decrease both delay and power consumption compared to the prolong listen protocol which was considered as the best protocol of the existing S-MAC protocols.

A. Parts of the proposed protocol

Proposed prediction S-MAC protocol consists of the following parts; non-sleep periods, prediction S-MAC intervals, packets arrival and adaptive listen / sleep as shown in figure 4. Prediction S-MAC intervals part consists of two steps; confidence interval calculation and expected listen & sleep intervals. In the following steps, parts of the proposed prediction S-MAC protocol are explained.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

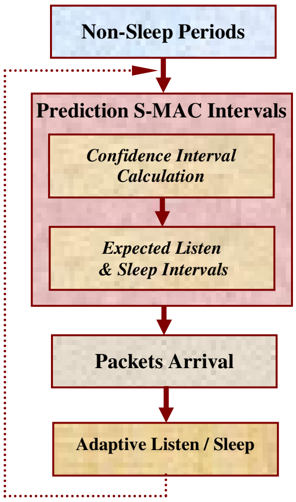

Figure 4. Parts of the prediction S-MAC algorithm

- *Non-sleep periods*
  In the non-sleep periods, nodes are always in active mode (*transmit, receive or idle state*) without sleep time, take the first (*N*) listen (*send / receive*) intervals in order to predict the next listen intervals.
- *Prediction S-MAC intervals*
  In this part, listen & sleep intervals are expected based on the last previous (*N*) listen intervals. Confidence interval method is used to predict the listen intervals by calculating both the mean and variance of the last previous (*N*) listen intervals. Then law of large numbers of the ratios 90 %, 95 %, 99 % (*or whatever*) is used to predict the start and end calculated confidence listen intervals. Then lower and upper bounds of the expected listen intervals are determined by adding both the *start & end* calculated confidence listen intervals to the upper bounds of the last previous intervals. Sleep intervals can be also expected. This part is divided into two steps:

1) Confidence interval calculation:

In this step, confidence interval (*C.I*) method is used to expect the listen intervals based on the last previous (*N*) listen intervals as following:

- As known, confidence interval method gives an estimated range of values which is likely to include an unknown population parameter, the estimated range being calculated from a given set of sample data [10]. So, by using the confidence interval method the next listen interval (*N+1*) can be expected based on the last previous (*N*) listen intervals by the chosen ratios of 90%, 95%, 99% (*or whatever*) using the *mean* and *variance* of these (*N*) listen intervals.
- To expect the listen interval (*N+1*) based on the last previous (*N*) listen intervals (*Non-sleep periods*) do the following;
  - Compute both the mean and variance of these (*N*) listen intervals where;

$$- \overline{L}_{N_0} = \frac{\sum_{j=1}^{N} listen_j}{N} \quad (1)$$

where, N: is the previous listen intervals used,
$listen_j$: is the listen period of interval j.

  - *Variance* (S_0^2) of these (*N*) listen intervals is;

$$- S_0^2 = \frac{Y_{N_0}}{N} - (\overline{L}_{N_0})^2 \quad (2)$$

$$where, \; Y_{N_0} = \sum_{j=1}^{N} (Listen_j)^2 \quad (3)$$

  - By using law of large numbers and substituting into the following equation to compute both the *start & end* calculated confidence listen interval (*N + 1*) [10];

  - *Start and end* calculated confidence listen interval (*N+1*)

are $$\overline{L}_{N_0} \mp m * \frac{S_0}{\sqrt{N}} \quad (4)$$

where, m = 1.65 (using C.I of 90 %)
= 1.96 (using C.I of 95 %)
= 2.58 (using C.I of 99 %),
S is the standard deviations = $\sqrt{S^2}$.

- Similarly to expect the listen intervals (*i*) based on the last previous (*N*) listen intervals, where *i = N+2, N+3,* ……, and so on.
  - Update both the last value of *mean & variance* by adding the last previous expected listen interval (*i-1*) and excluding the previous listen interval (*i-N-1*) where;

$$- \overline{L}_{N_i} = \overline{L}_{N_{(i-1)}} - \frac{listen_{(i-N-1)}}{N} + \frac{listen_{(i-1)}}{N} \quad (5)$$

$$- S_i^2 = \frac{Y_{N_i}}{N} - (\overline{L}_{N_i})^2 \quad \text{Where,} \quad (6)$$

$$Y_{N_i} = Y_{N_{(i-1)}} - (Listen_{(i-N-1)})^2 + (Listen_{(i-1)})^2 \quad (7)$$

  - Calculate both the start and end calculated confidence interval (*i*) where;
    - *Start & end calculated confidence listen interval (i)*

are $$\overline{L}_{N_i} \mp m * \frac{S_i}{\sqrt{N}} \quad (8)$$

2) Expected listen & sleep intervals:

In this step, both the *lower & upper* pounds of the expected listen intervals are computed as following:

- After determining both the *start & end* calculated confidence listen interval (*i*), the *lower & upper* bounds of the expected listen interval (*i*) are expected by adding both the *start & end* calculated confidence listen interval (*i*) to the upper bound of the last previous interval (*i-1*).
- Lower & upper bounds of the sleep intervals can be also expected.

- *Packets arrival*
  a. If the arrival packets are in the expected listen interval (*expected by 95 % or 99 %*);
  - Send the packets.
  - Extend listen time, if transmission time is more than the expected listen interval.

  b. If the arrival packets are in the expected sleep interval (*expected by 5% or 1 %*);
  - Do not send the packets.
  - Reschedule the packets start time to the next predicted listen time.
- *Adaptive listen / sleep*
  Since transmit time, receive time, idle time and sleep time of each node in the prediction S-MAC protocol are needed to evaluate the proposed protocol. Therefore, these times are assigned by adaptation listen (*send / receive*) intervals of the non-sleep periods according to the expected listen & sleep intervals of the prediction S-MAC intervals.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

B. Example of the proposed protocol

In the following example, steps of working the proposed prediction S-MAC protocol are illustrated.

1) Non-sleep Periods: In the *non-sleep periods*, nodes are always in active mode (*transmit, receive or idle* state) without sleep time. Suppose a network includes node (*N1*) which has the following transmissions with the other nodes (*N2*, *N3* and *N4*). Figure 5.a shows a sample from node (*N1*) transmissions while figure 5.b illustrates send, receive and idle periods of node (*N1*).

Source	Destination	Status	Start time	Packets length
1	2	send	5	15
3	1	recv	25	10
1	4	send	45	13
1	3	send	65	25
2	1	recv	95	10
4	1	recv	115	5
1	3	send	127	8
3	1	recv	140	10
1	2	send	163	12

Figure 5.a. A sample from transmissions concerning node *N1*.

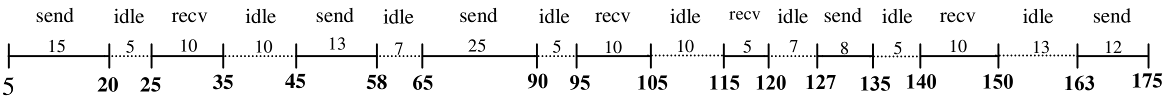

Figure 5.b. Send, receive and idle periods of node *N1*.

Figure 5. Non-sleep periods.

2) Prediction S-MAC Intervals: In this part, listen and sleep intervals are expected based on the last previous (*N*) listen intervals. It is divided into two steps; *confidence interval* (*C.I*) *calculation* and *expected listen & sleep intervals*.

a) Confidence interval (C.I) calculation: Confidence interval method is used to calculate both the *start & end* calculated confidence listen intervals by calculating both the *mean* & *variance* of the last previous (*N*) listen intervals. Then, law of large numbers of the ratio 95% is used in order to calculate both the *start & end* calculated confidence listen intervals. For example, fifth listen interval can be calculated based on the first four listen (*send, receive*) intervals by calculating both the *mean & variance* of these first four *send/receive* intervals of *the non-sleep periods*. Then, substituting in law of large numbers of the ratio 95% to get the *start & end* calculated confidence fifth listen interval as follows;

- The listen intervals of the first four listen (*send, receive*) intervals of the *non-sleep* periods are; L_0 = 15, 10, 13, 25 ms.
- both the mean & variance of these listen intervals are;

$$\overline{L}_0 = \frac{(15+10+13+25)}{4} = 15.75, \qquad (\overline{L}_0)^2 = 248.1$$

$$- \ Y_0 = (15)^2 + (10)^2 + (13)^2 + (25)^2$$
$$= 225 + 100 + 169 + 625 = 1119,$$

$$- \ S_0^2 = \frac{Y_0}{N} - (\overline{L}_0)^2 = \frac{1119}{4} - 248.1 = 31.65, \quad S_0 = 5.6$$

- Start calculated confidence listen interval =

$$\overline{L}_0 - m*\frac{S_0}{\sqrt{N}} = 15.75 - 1.96 * \frac{5.6}{2} \approx 10$$

- End calculated confidence listen interval =

$$\overline{L}_0 + m*\frac{S_0}{\sqrt{N}} = 15.75 + 1.96 * \frac{5.6}{2} \approx 21$$

By updating both the *mean & variance* of the last expected listen interval (*fifth listen interval*), the *start & end* calculated confidence of six listen interval can be calculated based on the last previous four listen intervals as following;

- The last four previous listen intervals are;

L_6 = 10, 13, 25, 11 msec. (*where 11 is the fifth listen interval calculated from the first four send/receive intervals of the non-sleep periods).*

- Calculate both the *mean & variance* of the last four previous listen intervals where;

$$- \ \overline{L}_6 = 15.75 - \frac{15}{4} + \frac{11}{4} = 14.75, \qquad (\overline{L}_6)^2 = 217.6$$

$$- \ Y_6 = 1119 - (15)^2 + (11)^2 = 1119 - 225 + 121 = 1015$$

$$- \ S_6^2 = \frac{Y_6}{N} - (\overline{L}_6)^2 = \frac{1015}{4} - 217.6 = 36.15,$$

$$S_6 = \sqrt{36.15} = 6.01$$

- *Start* calculated confidence listen interval $= \overline{L}_6 - m*\frac{S_6}{\sqrt{N}}$

$$= 14.75 - 1.96 * \frac{6.01}{2} \approx 9$$

- *End* calculated confidence listen interval $= \overline{L}_6 + m*\frac{S_6}{\sqrt{N}}$

$$= 14.75 + 1.96 * \frac{6.01}{2} \approx 21$$

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

b) Expected listen & sleep intervals: In the expected listen and sleep intervals, *lower* & *upper* bounds of the expected listen intervals are obtained by adding the *start & end* calculated confidence listen intervals to the upper bounds of the last previous intervals. In the previous example, after expecting both the *start & end* calculated confidence listen intervals, both the *lower & upper* bounds of the listen & sleep intervals are predicted as following;

- *Lower bound* of the expected fifth listen interval = 90 + 10 = 100 msec. (*where 90 is the upper bound of the fourth listen interval as appearing in the figure 5*).
- *Upper* bound of the expected fifth listen interval = 90 + 21 = 111 msec.
- Therefore, the *lower & upper* bounds of the expected fifth listen interval are 100 msec and 111 ms respectively, Δt = 11 msec.
- Also, *lower & upper* bounds of the expected sleep interval are 90 msec and 100 msec respectively as shown in figure 6.

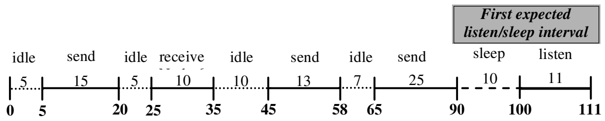

Figure 6. First expected Listen /sleep interval

- Also, *lower* bound of the expected six listen interval = 111 + 9 = 120 msec. (*where 111 is the upper bound of the fifth listen interval expected as shown in figure 6*).
- *Upper bound* of the expected six listen interval = 111 + 21 = 132 msec.
- Therefore, *lower* & *upper* bounds of the expected six listen interval are 120 msec and 132 msec respectively, Δt = 12 msec.
- Also, *lower* & *upper* bounds of the expected sleep interval are 111 msec and 120 msec respectively as shown in figure 7.

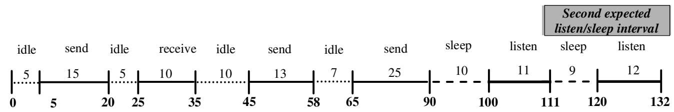

Figure 7. Second expected listen /sleep interval

By applying the previous steps, the seventh and eighth listen / sleep intervals can be expected where;

- Third expected listen/sleep interval (*seventh listen /sleep interval*) is:
  - Start & end calculated confidence listen interval = (10, 21).
  - *Lower bound* of the expected seventh listen interval = 132 + 10 = 142 msec.
  - *Upper bound* of the expected seventh listen interval = 132 + 21 = 153 msec.
  - Therefore, *lower & upper* bounds of the expected seventh listen interval are 142 ms and 153 ms respectively, Δt = 11 ms.
  - Also, *lower & upper* bounds of the expected sleep interval are 132 ms and 142 ms respectively as shown in figure 7.
- Fourth expected listen/sleep interval (*eighth listen/sleep interval*) is:
  - *Start & end* calculated confidence listen interval = (9, 21)
  - *Lower bound* of the expected eighth listen interval = 153 + 9 = 162 msec.
  - *Upper bound* of the expected eighth listen interval = 153 + 21 = 174 msec.
  - Therefore, *lower* & *upper* bounds of the expected eighth listen interval are 162 msec and 174 msec respectively, Δt = 12 msec.
  - Also, *lower & upper* bounds of the expected sleep interval are 153 msec and 162 msec respectively as shown in figure 8.
  - Therefore, listen/sleep intervals of the prediction S-MAC intervals are shown in figure 8.

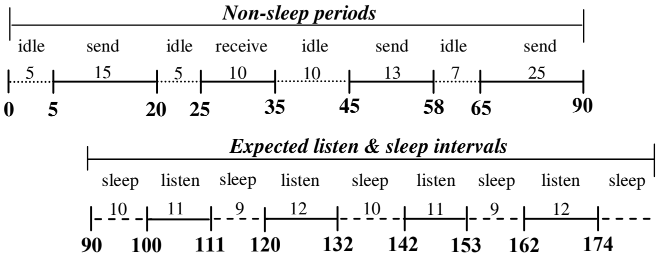

Figure 8. Expected listen & sleep intervals

Note that, nodes in the *prediction S-MAC intervals* are almost active during the expected listen time otherwise; nodes are almost in a sleep mode to serve the radio power consumption. Also, the arrival packets are almost in the expected listen interval by the ratio 95% to be served. We expect *the proposed prediction S-MAC protocol* will improve the performance parameters evaluation; average packet delay, throughput, average node power consumption and sensor node life compared to *the prolong listen protocol*.

3) Packets arrival:

a. If the arrival packets are in expected listen interval;
  - Send the packets.
  - Extend listen time, if transmission time is more than the expected listen interval as shown in figure 9.

b. If the arrival packets are in expected sleep interval;
  - Do not send the packets.
  - Reschedule the packets start time to the next expected listen time as shown in figure 9.

4) Adaptive listen / sleep periods:

To measure the performance parameters of the prediction S-MAC protocol; *average packet delay*, *throughput*, *average node power consumption* and *average node life*, we need to calculate both *transmit time*, *receive time*, *idle time* and *sleep time* for each node in the proposed protocol. These times are assigned by adaptation the listen (*send / receive*) intervals of the non-sleep periods according to the expected listen & sleep intervals of the prediction S-MAC intervals. Therefore, *transmit time, receive time, idle time* and *sleep time* of each node in the proposed prediction S-MAC protocol are assigned accurately as shown in figure 9.

As shown in figure 9, transmit time, receive time, sleep time and idle time of the node (*N1*) in the used example can be measured as follows;

- *Transmitting time* = 15 + 13 + 25 + 8 + 12 = 73 msec.
- *Receiving time = 10 + 11 + 5 + 10 = 36 msec.*
- *Sleep time = 10 + 9 + 7 + 9 = 35 msec.*
- *Idle time = 5 + 5 + 10 + 7+ 1+ 2+ 1+ 1= 32 msec.*

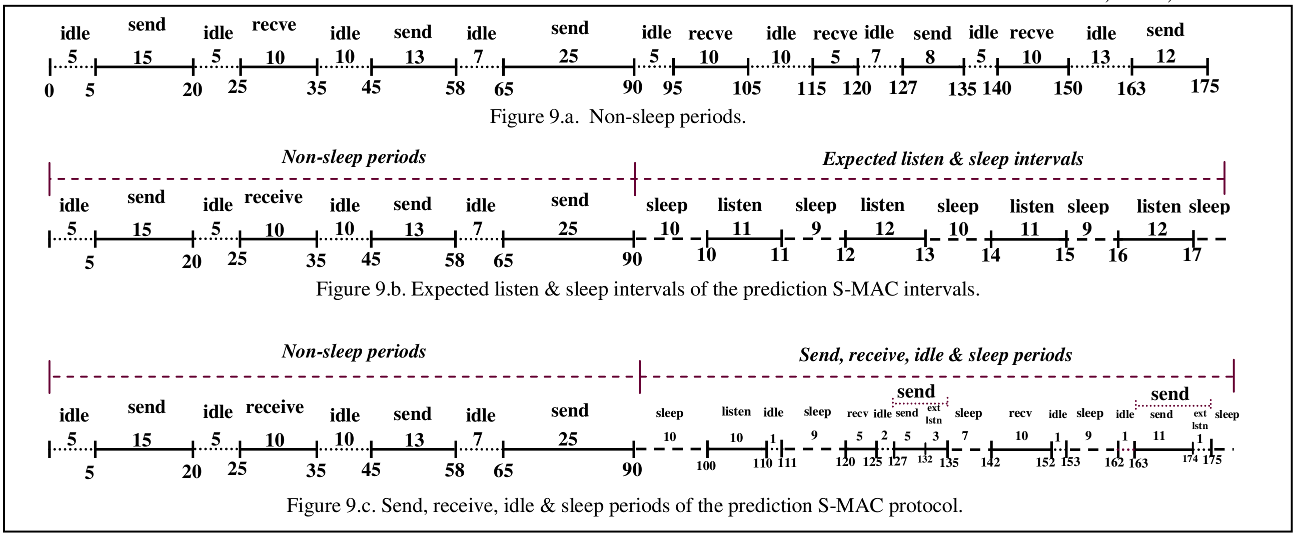

Figure 9.a. Non-sleep periods.

Figure 9.b. Expected listen & sleep intervals of the prediction S-MAC intervals.

Figure 9.c. Send, receive, idle & sleep periods of the prediction S-MAC protocol.

Figure 9. Adaptive listen / sleep concerning node (*N1*).

IV. PROTOCOLS IMPLEMENTATION

A simulation program is built using "*Visual C++ programming language*" in order to build and compare both the proposed prediction S-MAC protocol and the prolong listen protocol. The simulation program is divided into seven main parts; *packets creation*, *non-sleep periods*, *prediction S-MAC intervals*, *packets arrival*, *adaptive listen/sleep*, *prolong listen protocol* and *performance parameters evaluation* for each protocol. Prediction S-MAC intervals part consists of two steps; confidence interval calculation and expected listen & sleep intervals as shown in figure 10.

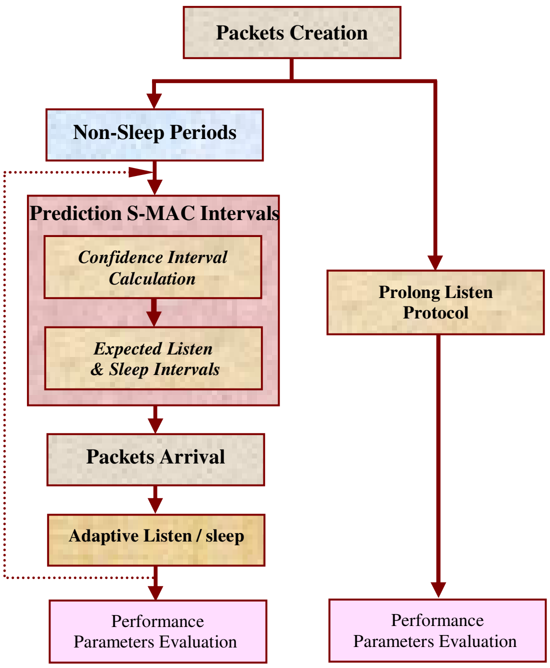

Figure10. Main parts of the simulation program

In order to obtain accurate results (*similar to real cases*), the packets' information is created randomly. However, the same packets' information must be used for the compared protocols in order to obtain accurate results. Therefore, the packets' creation part is separated from the parts of the compared S-MAC protocols, in fact its outputs is considered as common inputs for the compared S-MAC protocols.

Packets creation intervals are considered a common part for both the compared algorithms with generated values and the inputs for the S-MAC protocols. Of course; using the same input values for the compared protocols allows real comparison between them.

A. *Performance parameters evaluation*

The following parameters are used to evaluate both the *proposed prediction S-MAC* protocol and the *prolong listen* protocol; average packet delay, throughput, average node power consumption and average node life. A proposed protocol's objective is to increase both throughput and average node life while decreasing both delay and average node power consumption.

1) *Average packet's delay:*

Packet delay refers to the delay from when a sender has a packet to send until the packet is successfully received by the receiver. In sensor networks, the importance of delay depends on the application. Of course, the previous S-MAC protocols have longer delay due to the periodic sleeping on each hop. The objective of the proposed *prediction S-MAC* protocol is minimizing average packet delay compared to the *prolong listen* protocol. Average packet delay is calculated as follows:

$$\text{Average packet Delay} = \left(\frac{\sum_{\forall \text{packets}} (\text{Arrival time at destination} - \text{Initial time at source})}{\text{Total number of packets}} \right) \text{ in ms}$$

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

2) Throughput: Throughput (*often measured in bits or bytes or packets per second*) refers to the amount of data successfully transferred from a sender to a receiver in a given time. Many factors affect throughput, including efficiency of collision avoidance, channel utilization, delay and control packet overhead. As with delay, the importance of throughput depends on the application. The proposed prediction S-MAC protocol's objective is to increase throughput compared to other protocols. Throughput is calculated as follows:

$$Throughput = \frac{Total\ number\ of\ packets\,(X)}{Largest\ arrival\ time - Smallest\ initial\ time}\quad in\ pkts/sec.$$

3) Average node power consumption:

With large numbers of battery-powered nodes, it is very difficult to change or recharge batteries for these nodes. On many hardware platforms, the radio is a major energy consumer. The energy consumption of the node is measured by multiplying the amount of time that the radio on each node has spent in different modes: sleep, idle, transmitting and receiving by the required power to operate the radio in that mode. The objective of the proposed prediction S-MAC protocol is to minimize the power consumption of each node compared to the prolong listen protocol. Average node power consumption is calculated as follows:

$$Average\ node\ power\ consumption = \frac{\sum_{\forall node}\left(\sum_{\forall state}\left(\begin{array}{l}Time\ spent\ by\ the\ node\ in\ a\ state\\ \quad * Power\ consumed\ in\ this\ state\end{array}\right)\right)}{Number\ of\ nodes}$$

$$Where\ state \in \{idle, transmitting, receiving, sleep\}$$

4) Average node life: The lifetime T_i of node i is defined as the expected time for the energy E_i to be exhausted, where each node i has the limited energy E_i of node i to be exhausted. The network lifetime T of the system is defined as the time when the first sensor i is drained of its energy, that is to say, the system lifetime T of a sensor network is the minimum lifetime of all nodes of the network,

$T = min\{T_1, T_2, ..., T_n\}$ [11].

Because the compared protocols have different algorithms, where *prolong listen protocol* has fixed listen periods and the sleep periods are very long. Therefore, calculating node lifetime using real time (*in sec*) may increase in case of using more sleep time. So it will not a good parameter, therefore instead of using real time to calculate the node lifetime, we will use number of served packets. That is mean, the node lifetime will not be calculated as the time in second the node will go down after, instead it will be calculated as the number of packets the node can serve before going down. So, average node life is calculated as follows:

1. For each node of both the compared protocols, calculate the following:
   - Total number of transmitted packets (K).
   - Total power consumption of transmitted packets (P).
   - Divide the total power consumption of transmitted packets (P) by the total number of transmitted packets (K) to get average packet power consumption (P_K).

   Average packet power consumption (P_k) $= \frac{p}{k}$ *mw/pkt.*
2. Using standard maximum battery power consumption of the sensor node (P_S) = 2850 mAh * 3 V = 8610 mwh. (*Each node has two AA alkaline batteries*) [12].
3. Divide maximum battery power consumption of the sensor node (P_S) by the average packet power consumption (P_K) to get the average number of packets that each node should transmit before running out of energy (T_K).

   Average number of served packets $T_k = \frac{P_s}{P_k}$ *pkts*
4. Therefore, the average nod life in packets is calculated from the following general equation;

   $$Average\ node\ life\ in\ packets = \sum_{\forall node}\left(\frac{\begin{array}{l}Maximum\ battery\ power\ consumption\ of\ the\ node\\ \quad * Total\ number\ of\ transmitted\ packets\ of\ that\ node\end{array}}{Total\ power\ consumption\ of\ transmitted\ packets\ of\ that\ node}\right)$$
5. Of course, a protocol that transmits a big number of packets before the nodes running out of energy is considered as longer life.

B. Simulation Parameters

The simulation program used the following values to build and compare the two protocols:

- Number of nodes (N) takes the values 10, 20, 30 and 40 nodes consequently.
- Node's range (R) is taken as 100 m * 100 m.
- Number of packets generated at each message is taken as a random number from 1 to 10 packets/node.
- Message length (M) is considered as multiple of a unit packet in the number of packets generated at each node.
- History interval count (H) is considered as the first 10 listen intervals.
- Minimum number of messages (*MSG*) that is created at each node is equal to 30 messages.
- The radio power consumption taken in receiving, transmitting and sleeping is *45 mw*, *60 mw* and *90 μw* respectively. There is no difference between listening and receiving mode [17].
- Average number of packets/node/sec (*Data rate step values* (A)) takes the values 20, 40, 60, 80,100 and 120 pkts/node/sec.
- Time increasing at the source nodes (Δd) is a random number.
- Value resulted at each data rate step point (A) is the average of running the simulated program five times.
- Confidence interval taken is considered as 95%.

  ($\bar{L}_{N_i} \pm m * \frac{S_i}{\sqrt{N}}$, m = *1.96*).
- Total battery power consumption of the sensor node is calculated by multiplying its volt (*1.5 V*) by capacity (*2870 mAh*) (*each sensor node has two AA alkaline batteries*).

- For the *prolong listen protocol*, the following values are taken:
  - Listen interval (*L*) is fixed and equal to 300 msec.
  - Sleep interval (*S*) is fixed and equal to 1000 msec.
  - Prolong listening time (*P*) is equal to 20 msec.
  - Start listen time of nodes (*ST*) is a random number from 1 to 25 msec.

C. *Results*

Both *the proposed prediction S-MAC protocol* and *the prolong listen protocol* are simulated using "*visual C++ programming language*". Performance parameters evaluation resulted from the simulation program; average packet delay, throughput, average node power consumption and average node life are computed to evaluate the compared protocols. For each parameter, four figures (*from a to d*) are used to compare the two protocols changing number of nodes (*N*) from ten to forty nodes by a step of ten. Average number of packets/node/sec (*data rate step values*) used are; 20, 40, 60, 80, 100 and 120 packets/node/sec.

For the prolong listen protocol; listen time (*L*) is fixed at 300 ms. while sleep time (*S*) is fixed at 1000 ms. Also, prolong listen time (*P*) used is 20 ms. while start listen time of each node is random time varying from 1 to 25 ms. To simulate reality, parameters used in the simulation program are generated randomly and in order to obtain accurate results, each point in the data rate step values is the average of running the simulated program five times.

1) *Average packet delay*

It is known that real packet delay is the sum of waiting time and transmission time. As shown in figure 11, It is clear that in case of the prolong listen protocol, increasing sleep time leads to higher average packet delay since the time that a packet needs to wait for the node to enter listen mode increases. Also, note that for the two protocols, increasing both average number of packets per node per second (*data rate step values*) and number of nodes lead to increasing average packet delay as shown in figure 11.

Note that, using *the prolong listen protocol* leads to a higher average packet delay than using *prediction S-MAC protocol*. The results are expected since in case of *the prolong listen protocol*, packets always wait a long time (*long sleep intervals*) for the node to enter the listen mode. While in case of *the prediction S-MAC protocol*, there is no fixed listen and sleep intervals. However, the nodes transmit only (*send/receive*) according to the prediction of their listen time, otherwise the nodes go to sleep mode and turn off their radio power until next prediction listen time. This implies to decreasing waiting time compared to the prolong listen protocol. So, the proposed prediction S-MAC protocol decreases average packet delay compared to the prolong listen protocol as shown in figure 11.

2) *Throughput*

Throughput refers to the amount of data successfully transferred from a sender to a receiver in a given time (*often measured in bits or bytes or packets per second*). As shown in figure 12, it is clear that in case of the prolong listen protocol; long sleep intervals lead to lower throughput than in the prediction S-MAC protocol. Since packets in the prolong listen protocol have to wait for a longer time for the nodes to enter their listen interval, so fewer packets are delivered per second (*throughput*). Of course for the two protocols, in the highest traffic load (*increasing both data rates and number of nodes*), contention happens at each hop, which can significantly reduce throughput. This leads to semi-straight lines appearing when the number of nodes is equal to forty nodes.

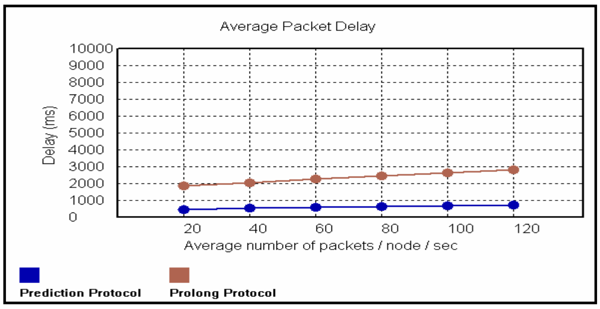

Figure 11.a. Average packet delay at *N* = 10 nodes.

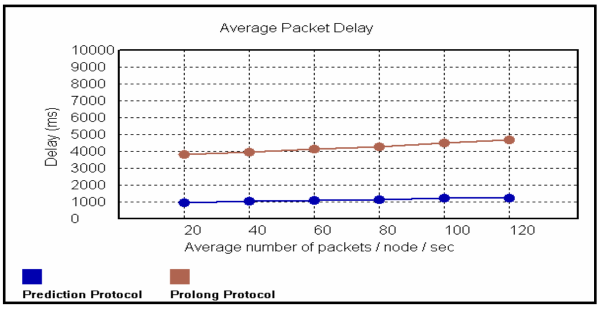

Figure 11.b. Average packet delay at *N* = 20 nodes.

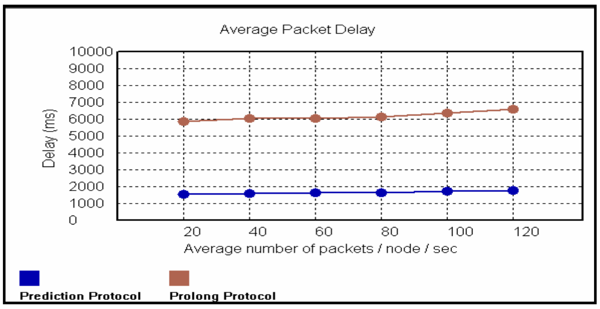

Figure 11.c. Average packet delay at *N* = 30 nodes.

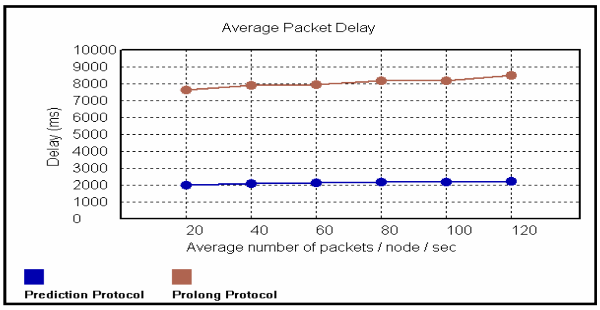

Figure 11.d. Average packet delay at *N* = 40 nodes.

Figure 11. Average packet delay.

As shown in figure12 using prolong listen protocol leads to lower throughput than using prediction S-MAC protocol. These results are expected since in case of the prolong listen protocol, packets always have to wait for the nodes entering their sleep mode, so number of packets delivered per second (*throughput*) decreases. While using prediction S-MAC protocol improves throughput by a great value since there is no fixed listen & sleep intervals, so packets do not wait long sleep time to be transmitted. Therefore, packets are served in the same listen interval instead of waiting for next listen intervals. Thus, number of packets delivered per second (*throughput*) increases and this leads to increasing throughput compared to the prolong listen protocol as shown in the figures.

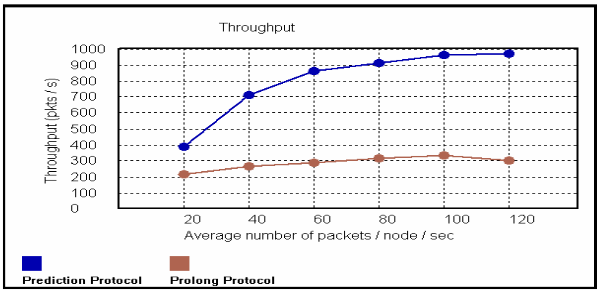

Figure 12.a. Throughput at $N = 10$ nodes.

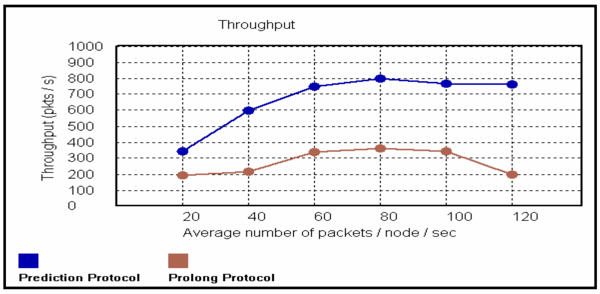

Figure 12.b. Throughput at $N = 20$ nodes.

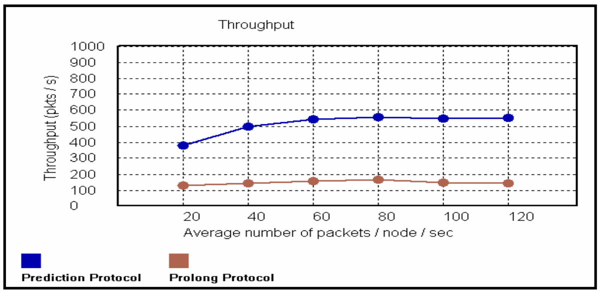

Figure 12.c. Throughput at $N = 30$ nodes.

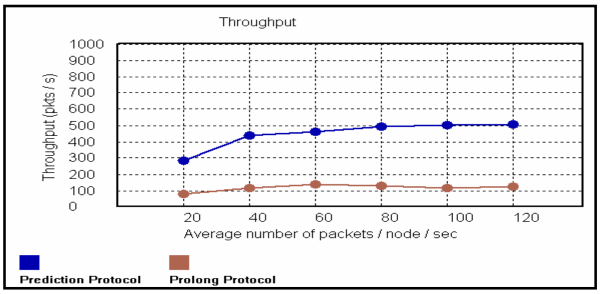

Figure 12.d. Throughput at $N = 40$ nodes.

Figure12. Throughput

3) Average node power consumption

Energy consumption for each node is calculated by multiplying the energy consumed at each mode (*sleep, idle, transmitting and receiving*) by the time that the node has spent in that mode. As shown in figure 13 note that, using the *prolong listen protocol* leads to higher average node power consumption than using the *prediction S-MAC protocol.* Since in case of *the prolong listen protocol* remind that, any node does not send a *RTS* message unless the destination is in listen mode and packets have to wait a long time for the destination node to enter the listen state. In fact, some queued packets may have to wait more than one period if their nodes are serving others, this implies to increasing sleep time and idle time. In addition, some *RTS* messages were sent by the source nodes and may be not answered. These *RTS* messages have to be resent and increase the transmission time. Although not all the next-hop nodes could overhear the *RTR* messages from the transmitting nodes, since they are not at the scheduled listen time or they do not have data packets to send. Therefore, if a node starts a transmission by sending out a *RTR* message during prolong listen time, it might not get a reply. In this case, it just goes back to sleep mode and will try again at the next normal listen time, which increases transmission time and *RTR* messages consume energy. Thus, as sleep time increases, average node power consumption increases.

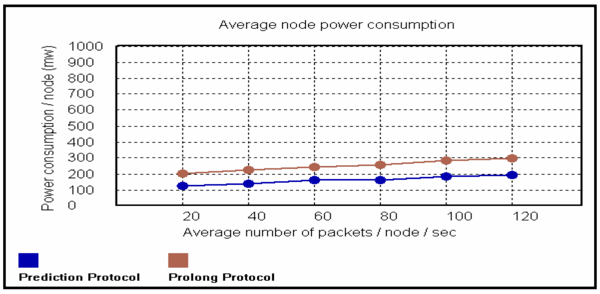

Figure 13.a. Av. node pw. consumption at $N = 10$ nodes.

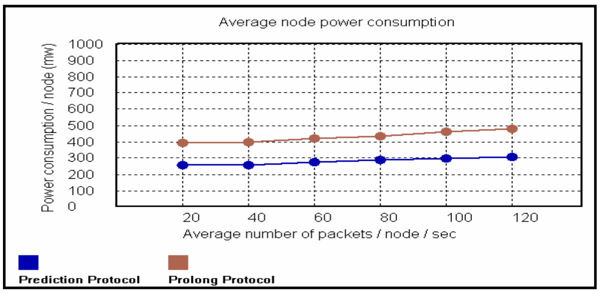

Figure 13.b. Av. node pw. consumption at $N = 20$ nodes.

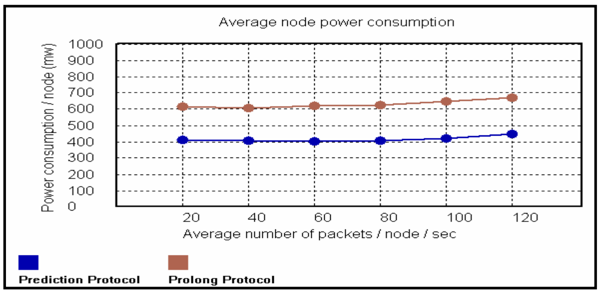

Figure 13.c. Av. node pw. consumption at $N = 30$ nodes.

http://sites.google.com/site/ijcsis/
ISSN 1947-5500

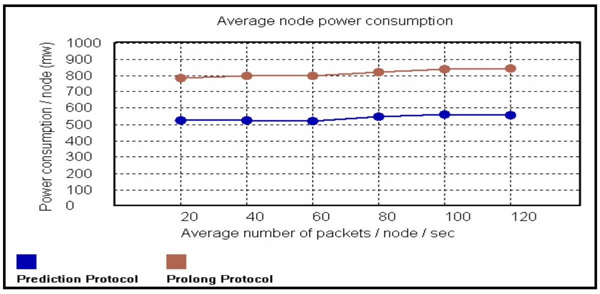

Figure 13.d. Av. node pw. consumption at $N = 40$ nodes.

While in case of *the prediction S-MAC protocol*, there is no fixed listen and sleep intervals and nodes become active only when transmitting (*send or receive*), otherwise nodes turn off radio power until their next prediction active time. Therefore, packets do not have to wait a long time for the node to enter the active state. Thus idle time is decreased. In addition, there are no *RTR* messages and just small numbers of *RTS* messages is repeated. This leads to lower transmission time than *prolong listen protocol*. As a result, *the prediction S-MAC protocol* decreases the average node power consumption compared to *the prolong listen protocol*. In addition, it is clear that for the two protocols, increasing both average number of packets per node per second (*data rate step values*) and number of nodes imply to increasing average node power consumption as shown in figure 13.

4) Average node life

The lifetime T_i of node i is usually defined as the expected time for the energy E_i to be exhausted, where each node i has the limited energy E_i of node i to be exhausted [11]. Considering the amount of time until the sensor node runs out of energy to refer to the average node life is not fair. So as a good idea instead of using the amount of time to calculate the average node life, the number of packets each node can serve before the node runs out of energy is used to refer to the average node life. That means; a protocol that serves a larger number of packets before the first sensor node die has longer life than the other.

As shown in figure 14, it is clear that the proposed prediction S-MAC protocol serves a bigger number of packets before the nodes exhaust their power compared to the prolong listen protocol. Therefore, the proposed prediction S-MAC protocol has a longer node life than the prolong listen protocol. These results are logic since packets in the prolong listen protocol have to wait for a longer time for the nodes to enter their listen mode, so fewer packets are delivered per second and nodes' battery are exhausted during long waiting time.

While in case of the *prediction S-MAC* protocol, there is no fixed listen and sleep intervals where, nodes active only when transmitting (*send or receive*) and turn off their radio power until next expected working time. In addition, packets do not have to wait a long time for the node to enter the active state and almost wait only if the destination node is busy. Therefore, idle time is decreased and number of served packets are increased before nodes run out of energy and consequently increasing average node life compared to the *prolong listen* protocol. Of course for the two protocols, increasing data rates and increasing number of nodes lead to decreasing average node life.

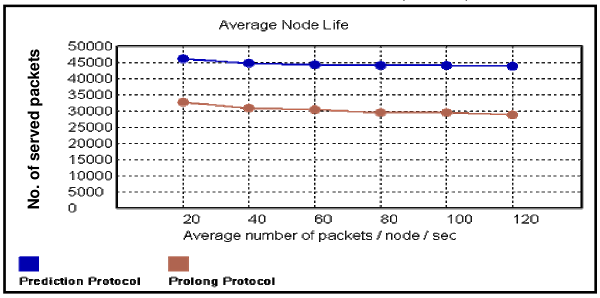

Figure 14.a. Average node life at $N = 10$ nodes.

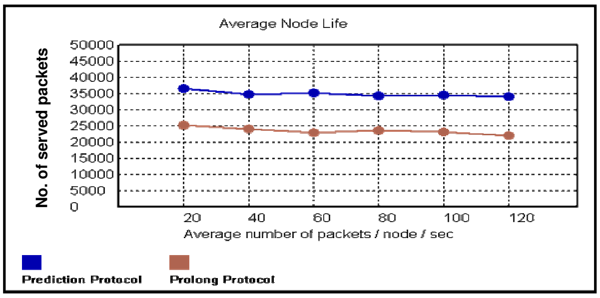

Figure 14.b. Average node life at $N = 20$ nodes.

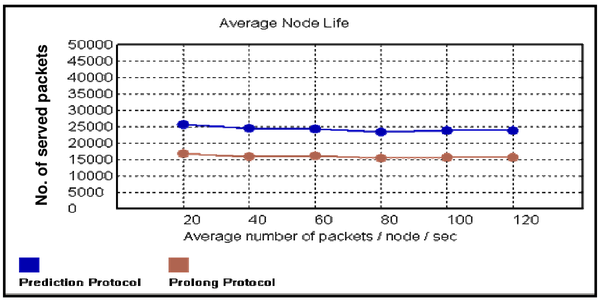

Figure 14.c. Average node life at $N = 30$ nodes.

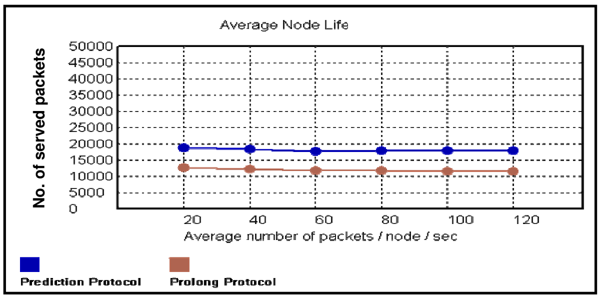

Figure 14.d. Average node life at $N = 40$ nodes.

Figure 14. Average node life.

As final words; both the proposed prediction S-MAC protocol and the prolong listen protocol are simulated by using visual C++ programming language. Performance parameters evaluation resulted from the simulation program; average packet delay, throughput, average node power consumption and average node life are evaluated for both the compared protocols. Results illustrate that the proposed prediction S-MAC protocol improves the performance of the network compared to the prolong listen protocol; it leads to lower average packet delay, higher throughput, lower average node power

consumption and longer average node life. In addition, for the two protocols; increasing both the average number of packets per node per second (data rate step values) and number of nodes lead to increase both the average packet delay and the average node power consumption while decrease both the throughput and average node life.

V. Conclusion and Future Trends

In this paper, a new S-MAC protocol named; *"prediction S-MAC protocol*" is proposed to improve the performance of the previous S-MAC protocols. The basic idea of the proposed protocol is to divide the whole time of the node into two successive intervals; working interval (*listen interval*), in which the node is expected to send or receive packets and non-working interval (*sleep interval*), in which the node is not expected to send or receive packets.

Confidence interval method and law of large numbers are used to predict the *working* and *non-working* (*listen/sleep*) intervals based on the last previous *N* listen intervals by the ratios 90%, 95%, 99% (*or whatever*). The proposed prediction S-MAC protocol was compared with the prolong listen protocol which was considered as the best protocol of the existing S-MAC protocols. Results proved that the proposed prediction S-MAC protocol increased both throughput and average node life while decreased both delay and average node power consumption compared to the prolong listen protocol.

As a future work, we will tray to get a model which gives both estimation and prediction of the future energy consumption in sensor nodes. This model is based on the statistics methods such as Markov chains. If the sensor node can predict its power consumption then it would be better to transmit the predicted energy in the batteries for the path discovery, this will allow also a priori reaction and a possible optimization of the mechanism applied for the minimization of the energy consumption, which depends essentially on the remaining energy in sensors batteries.

VI. References

1. T.S. Rappaport, "*Wireless Communications, Principles and Practice*", Prentice Hall, 2edition, Dec 31, 2001.
2. Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci, "*A Survey on Sensor Networks*", IEEE Communications Magazine, vol. 50, no.8, pp. 102-114, August 2002.
3. J. Heidemann, F. Silva, C. Intanagonwiwat, R. Govindan, D. Estrin, and D. Ganesan, "*Building efficient wireless sensor networks with low-level naming*", Proceedings of the eighteenth ACM symposium on Operating systems principles, Banff, Alberta, Canada, pp. 146-159, October 2004.
4. Deepak Ganesan, Alberto Cerpa, Wei Ye, Yan Yu, Jerry Zhao and Deborah Estrin, "*Networking Issues in Wireless Sensor Networks*", Journal of parallel and distributed computing, vol.64, no.7, pp.799-814, July 2004.
5. O. Ocakoglu and O. Ercetin, "*Energy Efficient Random Sleep-Awake Schedule Design*", IEEE communication letters, Vol. 10, No.7, JULY 2006.
6. Wei Ye, John Heidemann and Deborah Estrin, "*An energy-efficient MAC protocol for wireless sensor networks*", Proceedings of the IEEE INFOCOM, New York, USA, pp.1567-1576, June 2002.
7. Wei Ye, John Heidemann and Deborah Estrin,"*Medium Access Control with Coordinated, Adaptive Sleeping for wireless sensor networks*", IEEE/ACM transactions on networking, vol. 12, no 3, pp. 493-506, June 2004.
8. Wei Ye and John Heidemann, "*Medium Access Control in Wireless Sensor Networks*", Proceedings of the USC/ISI TECHNICAL REPORT ISI-TR580, October 2003.
9. Imane A. Saroit and Mahmoud A. El-Sakhawy, "*Propose Medium Access Control for Wireless Sensor Networks*", published in the third annual conference (*INFOS 2005*), faculty of computers and information, Cairo university, 19-22 March 2005. www.fci-cu.edu.eg/infos 2005.
10. Douglas G Altman, David Machin, Trevor N Bryant and Martin J Gardener, "*Statistics with confidence*", 2nd edition.
11. Yunxia Chen and Qing Zhao, "*Maximizing the Lifetime of Sensor Network Using Local Information on Channel State and Residual Energy*", Conference on Information Sciences and Systems, The Johns Hopkins University, 2008.
12. Michael Day, "*Using power solutions to extend battery life in applications*", Analog Applications Journal, 2009.
13. Yuan Li, Wei Ye, John Heidemann, and Rohit Kulkarni, "*Design and Evaluation of Network Reconfiguration Protocols for Mostly-Off Sensor Networks*" Information Sciences Institute, University of Southern California, 2007.