Egyptian Informatics Journal (2011) **12**, 61–72

Cairo University

Egyptian Informatics Journal

www.elsevier.com/locate/eij
www.sciencedirect.com

ORIGINAL ARTICLE

An Optimized Energy-aware Routing Protocol for Wireless Sensor Network

Basma M. Mohammad El-Basioni [a,*], **Sherine M. Abd El-kader** [b], **Hussein S. Eissa** [b], **Mohammed M. Zahra** [c]

[a] *Research Assistant at Electronics Research Institute, Computers and Systems Dept., Cairo, Egypt*
[b] *Associate Prof. at Electronics Research Institute, Computers and Systems Dept., Cairo, Egypt*
[c] *Associate Prof. at Communications Dept., Faculty of Engineering, Al-Azhar University, Cairo, Egypt*

Received 9 December 2010; accepted 31 March 2011
Available online 8 July 2011

KEYWORDS

Wireless Sensor Network;
Energy-efficiency;
Hierarchical routing;
Quality of service;
Aggregation

Abstract Because sensor nodes typically are battery-powered and in most cases it may not be possible to change or recharge batteries, the key challenge in Wireless Sensor Networks (WSNs) design is the energy-efficiency and how to deal with the trade-off between it and the QoS parameters required by some applications. This paper studies the QoS of an energy-efficient cluster-based routing protocol called Energy-Aware routing Protocol (EAP) in terms of lifetime, delay, loss percentage, and throughput, and proposes some modifications on it to enhance its performance. The modified protocol offers better characteristics in terms of packets loss, delay, and throughput, but slightly affects lifetime negatively. Simulation results showed that the modified protocol significantly outperforms EAP in terms of packet loss percentage by on average 93.4%.

1. Introduction

The key challenge a Wireless Sensor Networks (WSN) [1–5] routing has to deal with is the energy efficiency and prolonging network lifetime. Hierarchical communication among sensor nodes by clustering them is more scalable, energy-efficient, lower in latency, better in terms of network lifetime than flat communication. A hierarchical clustering routing protocol proposed in literature for data gathering applications in WSN called Energy-Aware routing Protocol (EAP) [6,7] meets several important requirements for a clustering algorithm. It is proved that EAP achieves a good performance in terms of lifetime. The advantages of EAP make it deserved to big interest and consideration, so this paper evaluates EAP by simulation in terms of some QoS parameters which are the lifetime, the end to end

---

* Corresponding author.
E-mail address: bbasioni@yahoo.com (B.M.M. El-Basioni).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.
doi:10.1016/j.eij.2011.03.001

Production and hosting by Elsevier

delay, the packet loss percentage, and the throughput, and proposes a modified protocol enhances its performance. Simulation showed that the modified protocol outperforms EAP in terms of packet loss percentage by on average 93.4%, and also its performance is better in terms of throughput and delay.

The remainder of this paper is organized as follows. Section 2 reviews related work, Section 3 discusses the EAP protocol, Section 4 explains the proposed modifications, Section 5 presents network model and assumptions, Section 6 evaluates the performance of the protocols, and finally Section 7 concludes the paper and discusses the directions for future work.

2. Related work

There are a lot of hierarchical clustering routing protocols proposed in literature; these protocols differ in many ways such as the basis on which the cluster heads (CHs) are selected, the used method for cluster formation, etc. These protocols such as:

Low-Energy Adaptive Clustering Hierarchy (LEACH) [8]: a predefined percentage of nodes are selected as CHs randomly and randomly rotated – with no probability of a CH to become again CH up to certain number of rounds. Each node selects the closest head as CH and sends data to it using Time Division Multiple Access (TDMA) schedule. LEACH may result in bad CH selection, bad CHs distribution, and instability in clusters number and size; this increases the load on CHs as well as on members, sometimes the distance between the CH and its member may be long, this leads to a long time for data to reach the CH which implies widening the time slot of the TDMA schedule, also sometimes the whole network formed in one cluster, this implies the lengthening of the TDMA schedule itself to be enough for all existing nodes minus one (the alone CH). This long TDMA schedule with its wide time slots increases the data latency.

LEACH-centralized (LEACH-C) [9]: is an improved scheme of LEACH in which a centralized algorithm at the base station (BS) makes cluster formation. Each node sends information about current location and energy level to BS, then the BS utilizes its global information of the network to produce better clusters that require less energy for data transmission. LEACH-C does not take into account a method to overcome the sensor node failure and it needs GPS or other location-tracking method.

Hybrid, Energy-Efficient, Distributed clustering approach (HEED) [10]: for prolonging network lifetime, CH selection in this protocol is primarily based on the residual energy of each node, and to increase energy efficiency and further prolong network lifetime, a secondary clustering parameter considers intra-cluster “communication cost” which can be a function of neighbor proximity or cluster density. The main objectives of HEED are to distribute energy consumption to prolong network lifetime, minimize energy during the CH selection phase, and minimize the control overhead of the network, but HEED needs multiple broadcasting for cluster formation and thus consumes more energy.

Two-Level hierarchy LEACH (TL-LEACH) [11]: is a proposed extension to the LEACH algorithm. It utilizes two levels of cluster heads (primary and secondary) in addition to the other simple sensing nodes. In this algorithm, the primary cluster head in each cluster communicates with the secondaries, and the corresponding secondaries communicate with the nodes in their sub-cluster. The two-level structure of TL-LEACH reduces the amount of nodes that need to transmit to the sink, effectively reducing the total energy usage.

Proxy-Enable Adaptive Clustering Hierarchy (PEACH) [12]: is a protocol that improves LEACH in terms of lifetime. This is achieved by selecting a proxy node which can assume the role of the current CH of weak power during one round. PEACH is based on the consensus of healthy nodes for the detection and manipulation of failure in any cluster head. It allows considerable improvement in the network lifetime by reducing the overhead of re-clustering.

Energy-Efficient Clustering Scheme (EECS) [13]: A constant number of candidate nodes for CH role are elected with a probability T and compete according to the residual energy within range $R_{compete}$. The candidate will be a head if it did not find another higher energy candidate, otherwise, it will give up competition with the first found higher energy candidate. The cluster size should be justified such that, the larger the distance between the CH and the BS is, the smaller cluster size the CH should accommodate. It is true that the CH selected is the candidate with larger residual energy in range $R_{compete}$, but the set of candidate nodes in the competition are selected randomly before the competition, this may result in non-optimal CH selection.

Power-Efficient and Adaptive Clustering Hierarchy (PEACH) [14]: The main goal of PEACH is to minimize the energy consumption of each node, and maximize network lifetime. By using overhearing characteristics of wireless communication, PEACH forms clusters without additional overhead and supports adaptive multi-level clustering; in addition, PEACH can be used for both location-unaware and location-aware Wireless Sensor Networks.

A Probability-Driven Unequal Clustering Mechanism for Wireless Sensor Networks (PRODUCE) [15]: organizes the network with unequal-sized clustering determined with localized probabilities and multi-hop routing based on stochastic geometry. Far clusters from the BS are made to have larger cluster sizes that allow focusing more on intra-cluster data processing rather than inter-cluster processing. It results in energy consumption balancing, increasing lifetime, and improving coverage.

3. Description of EAP

EAP works in rounds as LEACH and each round consists of two main phases, set-up phase and data phase. The set-up phase is subdivided into two phases, cluster formation phase and CHs tree construction phase. In the cluster formation phase, each node takes one of three states (roles), candidate, plain, or head, while in the tree construction phase, each CH takes additional role(s) to form the CHs tree, these roles are child, parent, and root, so that the CH may be a Child Cluster Head only (CCH), Parent Cluster Head (PCH) which indoors is a child, Root Cluster Head which may be a parent but not a child (RCH/RPCH). The flow charts represent the phases of EAP operations are depicted in Figs. 1–3.

In the flow charts, E_a is the average residual energy of all neighbors in the cluster range, and it is computed from the neighborhood table information by using Eq. (1) as in [6,7].

$$E_a = \frac{\sum_{j=1}^{m} v_j \cdot E_{residual}}{m} \tag{1}$$

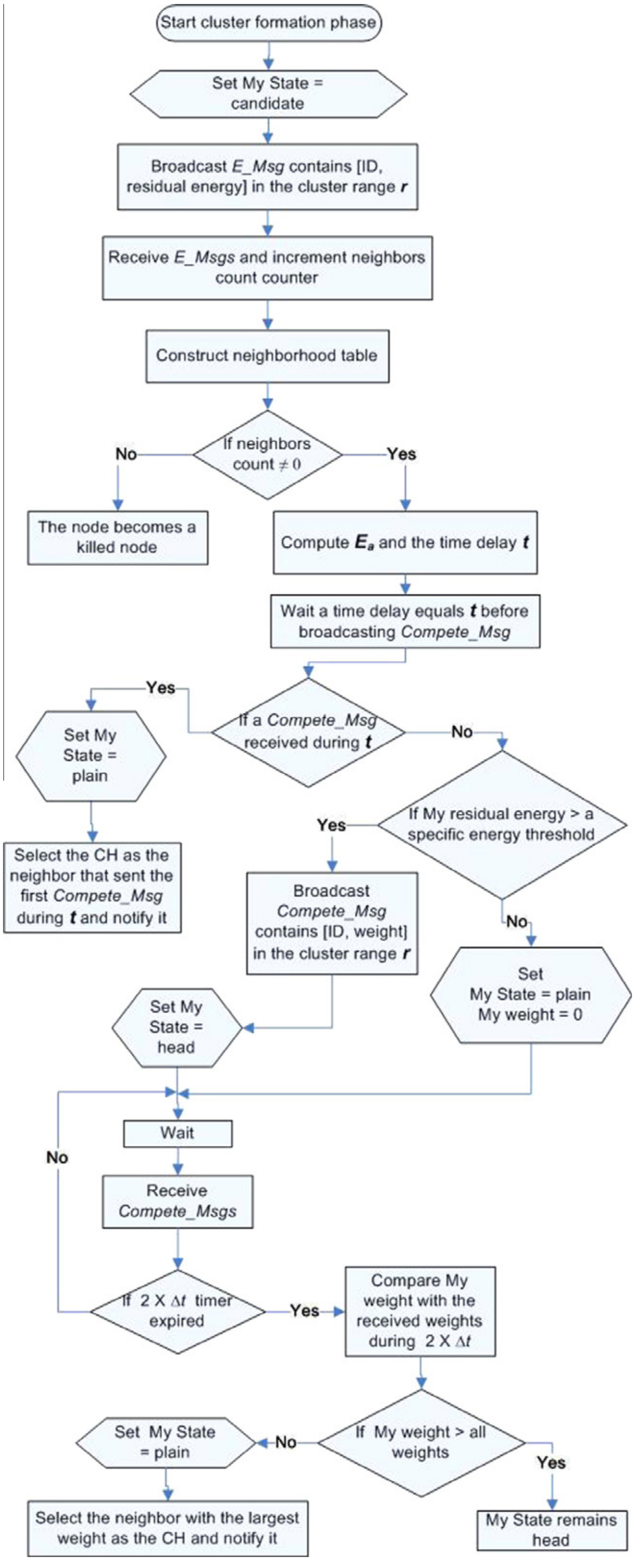

Figure 1 Cluster formation phase flow chart.

where m is the number of nodes within cluster range and v_j represents any node in this cluster range. The time delay (t) is calculated according to Eq. (2) as in [6,7].

$$t = k \times (T_{clustering}) \times \left(\frac{E_a}{E_{residual}}\right) \tag{2}$$

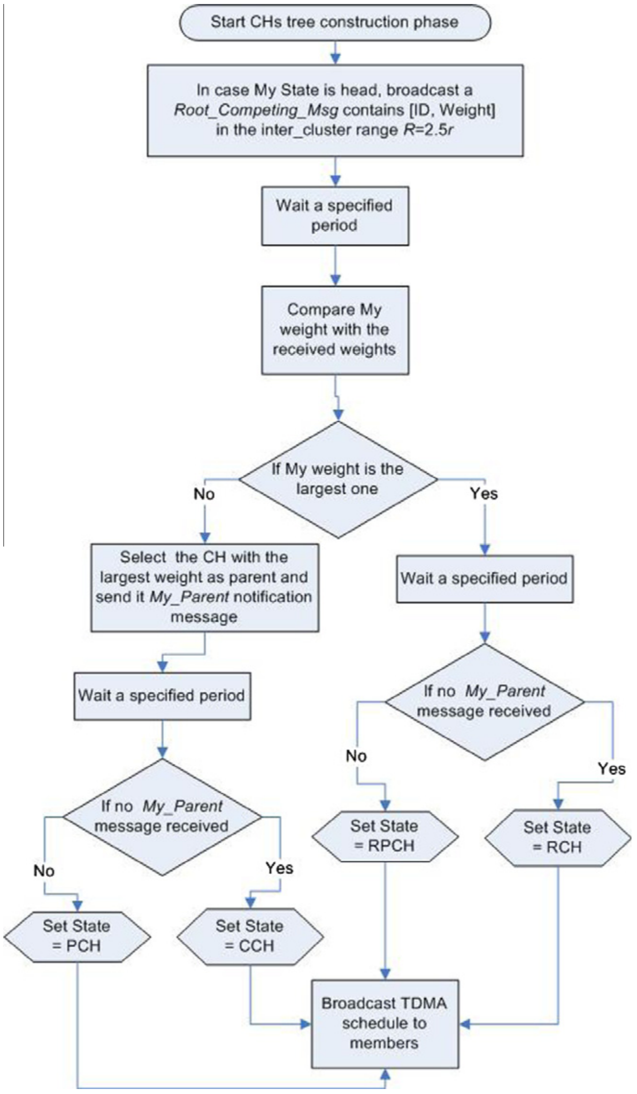

Figure 2 Tree construction phase flow chart.

where k is a real value uniformly distributed between 0 and 1, and $T_{clustering}$ is the time duration for CHs election. The weight of a node i is computed by Eq. (3) as in [6,7].

$$\text{weight}_i = \left(\frac{D(RSS_i) \times E_a}{D(RSS_{\max}) \times E_{residual}}\right) \tag{3}$$

where RSS_i denotes node i's received signal strength of the signal broadcasted by the BS, RSS_{max} is a constant which is determined by the location of the BS, and D is a function used for estimating the distance between node i and the BS.

4. The new proposed algorithm

EAP achieves a good performance in terms of lifetime, but it lacks a mechanism that informs member nodes about their CH death and informs CHs about their parents' death during the round. So, EAP leads to energy loss that is consumed in sending packets to a dead node and loss in the sent packets, and these losses continue until the end of the round. EAP authors concerned only in their evaluation of the protocol on the network lifetime and they did not consider other

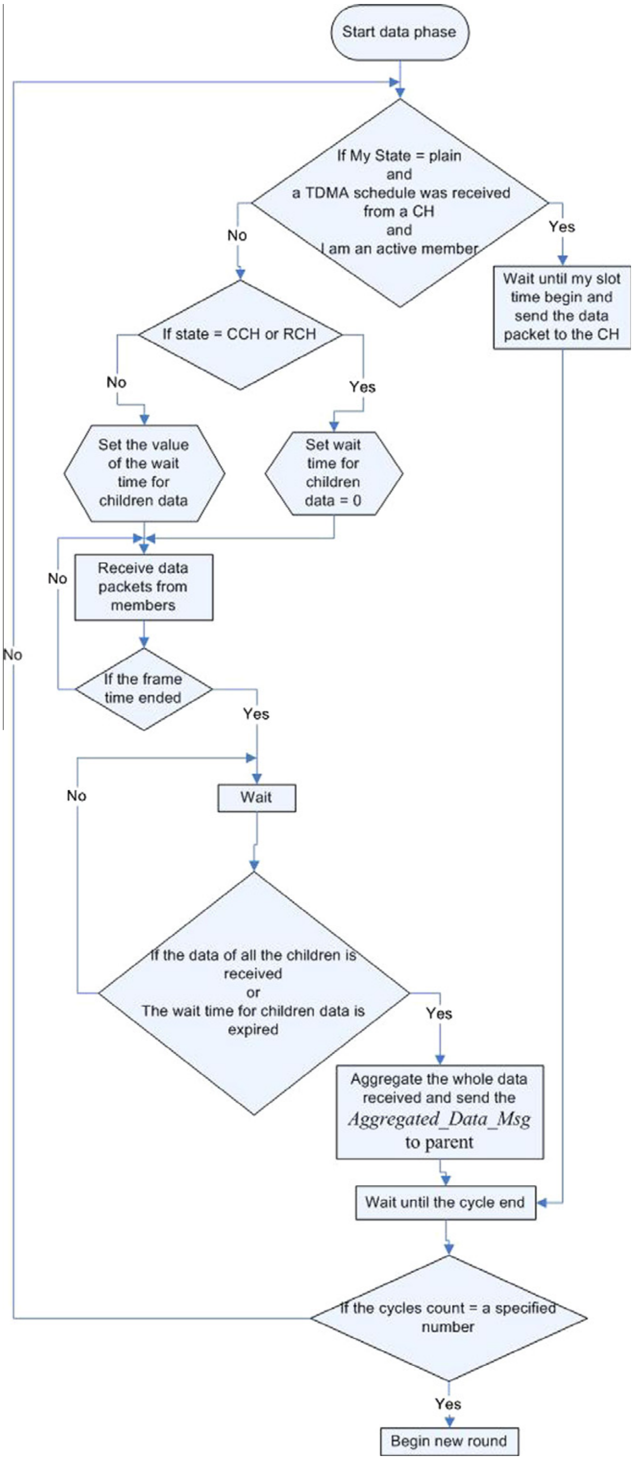

Figure 3 Data phase flow chart.

important QoS parameters such as delay, packet loss, and throughput. It was found that the common cause of EAP losses is the death of the root during the round when it is overloaded. Fig. 4 demonstrates the two common cases of root overload. In Fig. 4a, a CH is located in the range of all the remaining CHs and it has the largest weight among them, so all of them select it as a parent and it becomes a root constituting a star topology not a tree topology which increases the aggregation load on it. In Fig. 4b, a CH is far from the other CHs so it is obliged to be a root although it is very far from the sink which increases the transmission load on it. Fig. 5 shows the constructed tree in a round of EAP.

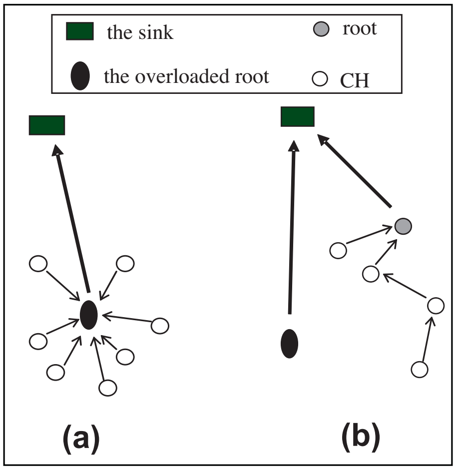

Figure 4 The two common root overload cases in EAP: (a) there is one root and all of the CHs are its children directly and (b) the CH is isolated from the other CHs.

It should be mentioned that the most common overload case is the first one, this means that the constructed tree is not ideal, it does not aid in balancing the load of relaying data among the CHs. To eliminate or decrease EAP losses, we can directly use a recovery method from CH (child, parent, or root) death or failure, but this method will exhaust a lot of energy and reduce lifetime and may also affect the other good performance metrics of EAP. So, the protection from loss cause is better than the cure from it. The protection manner used in the modified protocol Low Loss Energy-Aware routing Protocol (LLEAP) consists of two techniques, the first technique is used to increase the lifetime, so that if a loss occurred, the recovery method does not significantly affect the characteristic of LLEAP with respect to the lifetime, and after the

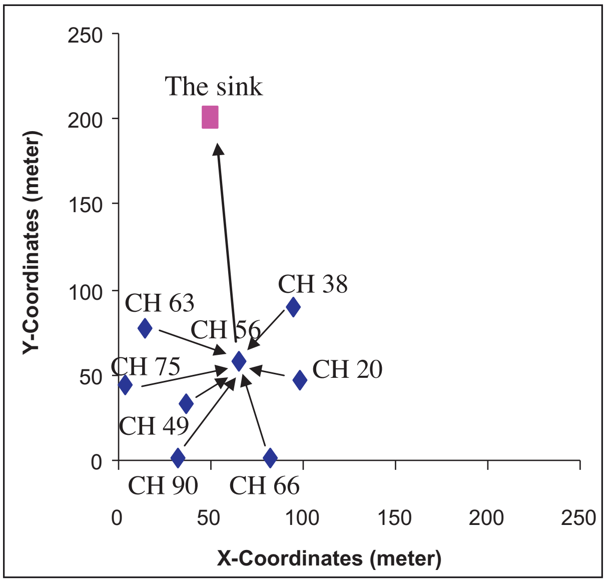

Figure 5 An example of the first case of root overload.

recovery, LLEAP lifetime remains on average as EAP lifetime. The lifetime is increased by developing a schedule for nodes to sleep and wake up to save their energy. The second technique is used to reduce the number of occurrence times of the previously mentioned common cause for loss in EAP to reduce losses and the energy consumed in the recovery from it.

4.1. Losses reduction method

The shape of the constructed tree should be controlled to reduce the occurrence of loss cause; to be more precise, the formation of star topology among CHs should be avoided, the maximum limit or the average value of the ‘‘branching factor’’ which can be defined as the variable number represents the number of children of each parent node should be decreased as much as possible especially the branching factor value of the root node, the branching factor values of all CHs should be convergent to distribute the tree aggregation load among them, and this done by modifying the weight used in the tree construction phase.

EAP uses the same weight for head selection and tree construction. To study this weight well, first it should be studied as a head selection weight, the equation of head selection weight should satisfy that relation among its parameters: the less ratio of the average residual energy of node neighbors to its residual energy $\left(\frac{E_a}{E_{residual}}\right)$, the greater the node weight, the greater the likelihood of that node to become a CH (so that the selected CH will collect in its cluster the maximum number of small residual energy nodes decreasing the load on them and giving the other nodes which have relatively higher energy the chance to become CHs), this requires reversing the ratio $\frac{E_a}{E_{residual}}$ which used in the equation of head selection weight in EAP. The weight used in LLEAP for head selection uses the reversed ratio as in Eq. (4)

$$\text{LLEAP CH selection weight} = \left(\frac{D(RSS_i) \times E_{residual}}{D(RSS_{\max}) \times E_a}\right) \tag{4}$$

For tree construction phase, the weight in Eq. (4) is not applicable because the tree construction is performed in level 1 of the hierarchy where the network is summarized to the graph composed of CHs only, so, in tree construction phase, E_a has no meaning and no effect, rather, it may have a negative impact on the selection of inappropriate CH as a root; and above E_a at the time of sending *Root_Compting_Msg* is no longer a correct estimation of the average residual energy at this time, because it is calculated at the start of the round before nodes send and/or receive different numbers of other messages. So, the tree construction weight in LLEAP is calculated as in Eq. (5).

$$\text{LLEAP tree construction weight} = \left(\frac{D(RSS_i)}{D(RSS_{\max})} \times E_{residual}\right) \tag{5}$$

By applying this weight in the example of Fig. 5, two trees will constructed as shown in Fig. 6. As shown in Fig. 6, this weight decreased the branching factors of all CHs and made them convergent, reduced the load on the root node with respect to the number of children (i.e., with respect to the energy consumed in aggregation), select the appropriate CH for the root role which has the largest residual energy and smallest distance to the sink, and reduced the load of the root with respect to the energy consumed in the transmission to the sink. But it should be noted that the number of roots increased which resulted in a reduction in lifetime while the losses were decreased slightly.

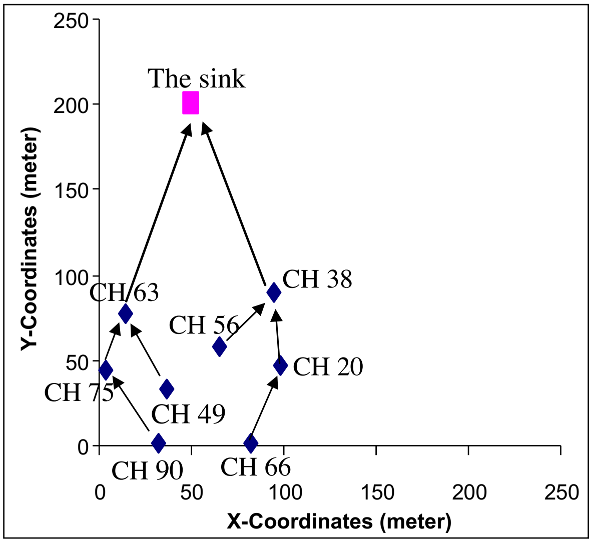

Figure 6 The network of Figure 5 after applying the new weight.

This problem was solved by making a second iteration for tree construction in which the root nodes formed in the first iteration broadcast *Root_Compting_Msg* message in larger transmission range *RR*. This message contains the weight of the root calculated as in the first iteration and also contains a list of the children of the sending root. After a specified period, each root compares the weights it received during this period with its own, if it has the largest weight, it remains a root; if not, before it chooses the root with the largest weight as parent it considers the children of this root, if one of them or more are located in its transmission range *R* used in the first iteration, it chooses the child with the largest weight as parent, otherwise it chooses the root itself. This decreases the aggregation load on the selected root in iteration2, and decreases the transmission load on the root that joins with its tree the tree of the selected root in iteration2. By this method the two constructed trees in the previous example will be integrated into one tree as shown in Fig. 7. It should be noted that the algorithm still has the advantages of using the modified weight; in addition, the number of roots was reduced and the probability of the existence of an isolated root was also reduced (the second common case of packets loss cause in EAP).

But, if the number of roots generated in the first iteration equals one, the second iteration becomes useless, but it wastes time and energy, so to deal with this shortcoming without sending or receiving any additional control messages, all nodes take the decision to use the second iteration for a specified constant number of rounds and invert this decision for another specified constant number of rounds, taking into consideration that the error resulting from the first decision is better than the resulting error from the second decision-making.

The used method for aggregation in the protocol cannot be ignored, because it may be a cause of losses. Up to now in EAP and LLEAP implementations, each parent waits after the frame time a period for its children to aggregate their aggregated data with its cluster members’ data and send the total

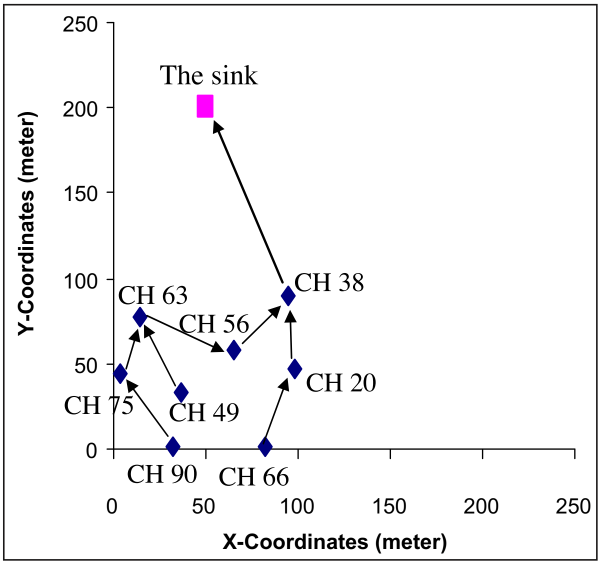

Figure 7 The constructed tree after adding the second tree construction iteration.

aggregated data to its parent once. This wait period has been set equal to $(2 \times \text{expected number of CHs} - 2) \times T_{child\_aggre}$, but, the root waits a period equals $(2 \times \text{expected number of CHs} - 1) \times T_{child\_aggre}$, where T_{child_aggre} is the maximum time needed for the packet to propagate from children to parent. After frame time during this wait period, if a parent received the data messages from all its children, it will cut this period, aggregate data, and send it to its parent. This aggregation method cannot deal with some situations, such as the situation when a parent waits for one of its children, while this child dead. The parent of this waiting parent will wait it and so on. Because parents stay the same time waiting their children, each child will send its aggregated message to its parent after its parent sends its aggregated message to its parent, so its data will be lost. The solution of this problem is to differentiate the wait time for each parent according to its level in the tree. This done by maintaining a variable for each CH represents its level in the tree, and initializing it to value "one" at the beginning of each cycle. Each head sends its value of this variable with the aggregated data message to its parent, and each parent updates its level variable value at the time for aggregation, and also updates its wait time (see Fig. 8) according to Eqs. (6) and (7) respectively:

$$\begin{aligned}\text{The new value (or new level)} = {} & \text{previous level} \\ & + \text{the largest level value} \\ & \text{among levels in all the} \\ & \text{aggregated data messages} \\ & \text{received from its children}\end{aligned} \tag{6}$$

$$\text{wait time} = (\text{level} - 1) \times T_{child\_aggre} \tag{7}$$

4.2. The proposed schedule technique

A schedule technique is used in LLEAP to save nodes energy wasted in overhearing. The timeline of LLEAP operation is shown in Fig. 9 and the details of the schedule are described as follows: During Startup Time (which is the period required for nodes after deployment to receive the *OK* message from the sink to acquire the *RSS*), all nodes wake up receiving *OK* message. During Table Filling period all nodes are awake for sending and receiving *E_Msg* and filling the neighborhood table. During (t) all nodes are awake until a *Compete_Msg* received, after that they send *Join_Msg* message to their selected CH and sleep. During $(2 \times \Delta t)$, plain nodes sleep while head nodes awake receiving *Compete_Msg* or *Join_Msg*. In the remaining time in Clustering phase, plain nodes formed after time $(2 \times \Delta t)$ are awake for sending *Join_Msg* to their selected CH, and head nodes are still awake for receiving *Join_Msg*. Through Root Role Compete period in iteration1, plain nodes sleep, head nodes awake sending and receiving *Root_Compting_Msg*, and nodes which are still candidate sleep. During Notify Parent period in iteration1, plain nodes sleep, head nodes awake sending and receiving *My_Parent* messages, and candidates sleep.

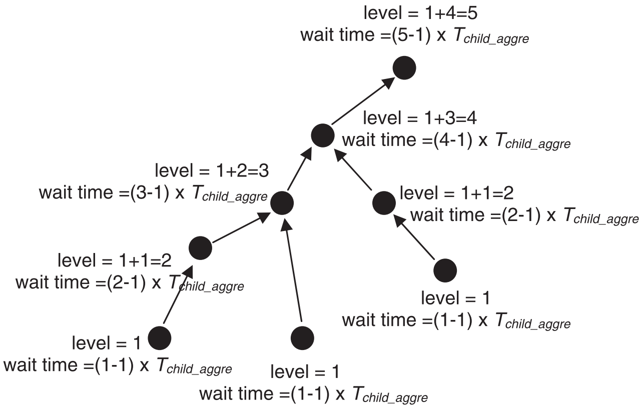

Figure 8 Level and wait time for CHs.

Over Root Role Compete period in iteration2, plain nodes sleep, head nodes (children and parents) sleep, root nodes awake sending and receiving *Root_Compting_Msg*, and candidate nodes sleep. Over Notify Parent period in iteration2, plain nodes sleep, head nodes awake sending and receiving *My_Parent* messages, and candidate nodes sleep. During TDMA Send/Receive period, plain nodes awake receiving *TDMA_Schedule* message, head nodes wake up sending *TDMA_Schedule* message, and candidate nodes sleep. During each Slot period, one of the plain nodes in each cluster awake sending *Data_Msg*, head nodes awake receiving *Data_Msg*, and candidate nodes sleep. Finally, during Data Relay period (which is the period required for the aggregated messages from all the constructed trees to reach the sink), plain nodes sleep, head nodes awake relaying the aggregated data to the sink, and candidate nodes sleep.

5. The network model and scenario assumptions

The simulation assumed that there are N static sensor nodes are randomly and densely scattered in a two-dimensional square field A, and the sensor network has the following properties:

- There is only one sink in the field, which is deployed at a fixed place outside A.

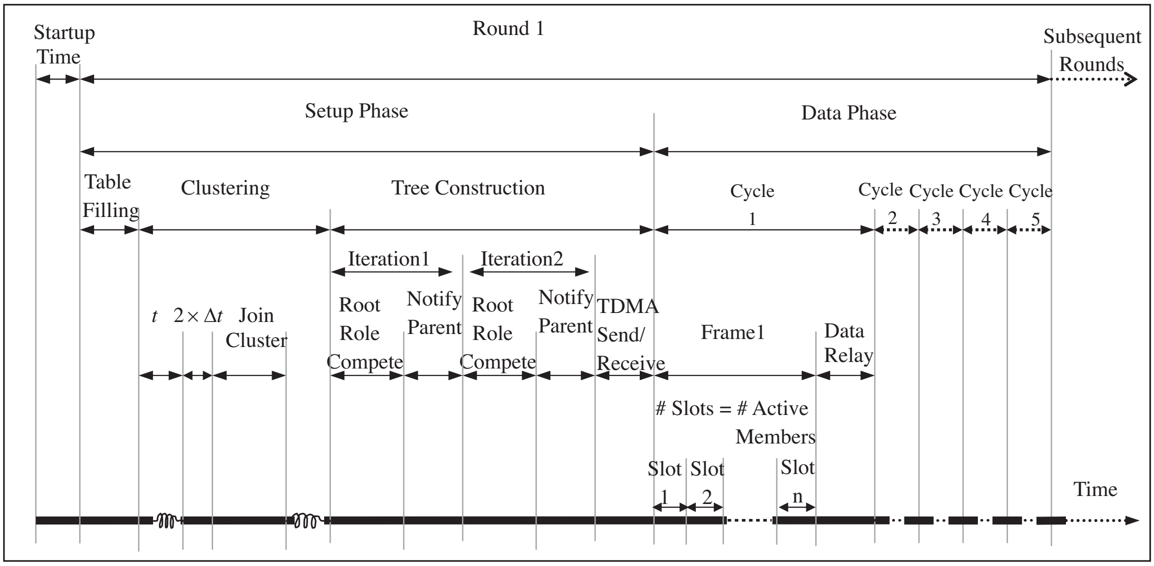

Figure 9 The timeline of LLEAP.

- Sensor nodes are location-unaware, non-rechargeable, and always have data to send.
- The node can vary its transmission power depending on the distance to the receiver.
- For simplicity, it is assumed that the probability of signal collision and interference in the wireless channel is ignorable and the radio transmitter, radio amplifier and data fusion unit are the main energy consumers of a sensor node.
- Packets loss due to factors other than node death does not exist or is ignorable.
- The Radio H.W. energy dissipation model used is as in [9].
- The consumed energy in aggregating L k-bit signals into a single k-bit signal = $L \times E_{DA} \times k$, where E_{DA} denotes the energy consumed by data fusion.
- As assumed in EAP paper, there are five cycles in each round.
- A node considered to be dead or killed when it becomes not capable of transmitting data to the sink, and this occurs in three cases:

(1) The node residual energy becomes below a threshold (E_{death}) equals to the energy required for a member node to participate in a round, so that it transmits at least one data packet.
(2) There is no head in its cluster range.
(3) There is no any node in its cluster range for any reason and it cannot be a head.

6. Performance evaluation for protocols

To assess the performance of the protocols, a set of simulation runs were carried out. The simulation runs were conducted using the discrete event simulator OMNeT++ [16] as the simulation platform to generate a network in 100×100 m^2 area in which sensor nodes are distributed statically and uniformly. The sink node is located at point (50,200) and it is assumed that it has infinite power and other resources. The signal propagation model used is the free space propagation model when the propagation distance is less than the threshold distance d_0, and the ground reflection (two-ray) propagation model when the propagation distance is greater than the threshold distance d_0 [17,18]. A simple temperature sensing application is used in simulation; the protocols also can be suitable for any other WSN application that requires data gathering such as precision agriculture, product quality and habitat monitoring. Protocols performance is compared with respect to the metrics defined in Section 6.2.1 using the parameters' values stated in Section 6.1; the results and analysis are in Section 6.2.

6.1. Simulation parameters

This section discusses the chosen values of the parameters used in simulation. Most of the parameters' values are selected as the same as those in EAP paper for comparing the protocols under the same conditions which EAP authors used, for example, the monitored area size, the position of the sink, the initial energy, $E_{threshold}$, the threshold distance, cluster radius, sensing radius, data packet size, and broadcast packet size. The inter-cluster range used in EAP is $2.5 \times r$, but in LLEAP there are two iterations for tree construction in the first iteration the inter-cluster range used (R) is $2 \times r$ this decreases the broadcasting load on CHs and also contributes on reducing the branching factor, in the second iteration the inter-cluster range used (RR) is $3 \times r$, although this range is greater than the range used in EAP, but this increases the broadcasting load on only a small number of nodes which are the previously formed roots, and also it decreases the probability of the existence of more than one root at the end of the tree construction phase. The parameters used in the simulation are reported in Table 1.

6.2. Results and analysis

In this section, the simulations which were carried out to compare the performance of the protocols are reported. In simulation, the number of the initial deployed nodes is increased from 100 to 500 node with step 100, and for each value the average value of one of the metrics defined in Section 6.2.1 is calculated for each protocol. Most of the results obtained in this paper were obtained by averaging multiple independent simulation runs with a 95% confidence interval, where each run uses a different randomly-generated topology of sensor nodes.

6.2.1. Performance analysis for EAP

To evaluate EAP and study its limitations, EAP was implemented and its implementation was validated by obtaining approximately the same results for some curves presented in EAP paper, under the same mentioned conditions, assumptions, network model, and parameters' values, such as the curve represents EAP lifetime shown in Fig. 13, then EAP QoS parameters such as delay, throughput, and packet losses were studied and compared with LEACH same parameters; these QoS parameters are defined as follows: in computation, raw data packets are used instead of the aggregated data packets as noticed from these definitions.

(1) *Packet loss percentage*: the ratio of the number of raw data packets lost due to node death to the total number of raw data packets transmitted in the network until its lifetime end.

$$\text{Packet loss percentage} = \frac{\text{number of transmitted raw data packets} - \text{number of raw data packets received at the sink}}{\text{number of transmitted raw data packets}} \times 100 \qquad (8)$$

(2) *Lifetime*: the count of rounds until the Last Node Dies (LND).

(3) *Throughput*: is the number of sent data bits by plain nodes which the sink benefit by per second and it can be considered as a measure to extent of sink benefit from the network lifetime because it determines whether the amount of data received to the sink is suitable with respect to the network lifetime or there is a lot of time lost in setup and data relaying.

$$\text{Throughput} = \frac{\text{number of bits received at the sink during the lifetime regardless of the form in which they are received}}{\text{network lifetime in seconds}} \qquad (9)$$

(4) *End to end delay*: the average time it takes for raw data packets to traverse from the plain nodes to the sink regardless of the form in which they are received, in other words it is the average delay of the sink in benefiting from every raw data packet sent until lifetime end.

$$\text{End to end delay} = \frac{\sum_{\text{all received raw datapackets at the sink}}(\text{receiving time} - \text{sending time})}{\text{total number of raw data packets received at the sink}} = \frac{\sum_{j=1}^{n}\sum_{i=1}^{h_j}(T_j + P_{ji})}{n} \qquad (10)$$

where n is the total number of raw data packets received at the sink, h_j is the total number of hops traversed by the raw data packet number j, T_j is the transmission delay of the raw data packet number j, P_{ji} is the propagation delay of the raw data packet number j through the hop number i. The following discussion demonstrates the results of the evaluation.

6.2.1.1. End to end delay. EAP average delay is not constant and it approximately increased slightly with the increases in the number of deployed nodes as shown in Fig. 10. In EAP implementation, if the number of a member's neighbors in its sensing radius is greater than a threshold value related to the quality of coverage determined by the application, it goes to be asleep with probability $(1 - 1/\mathit{threshold})$ [19], which means the more neighbors it has, the more probability to be asleep. This causes a reduction in the average number of active members per cluster when the number of deployed nodes increases and reaches 300, then it continues to decrease as shown in Fig. 11.

The average end to end delay is inversely proportional to the count of active members; the smaller the active members count, the larger the slot time, the more time most cluster members wait after sending the data message before the end of the frame and the beginning of data aggregation and relaying. The average count of CHs formed in EAP network is approximately constant with the increase on deployed nodes number (i.e. the average number of hops through the path to the sink is approximately constant). The delay in each hop equals to the transmission delay plus the propagation delay, while the transmission delay is the same for all nodes because the length of the data packet is constant, and the propagation delay has a small value that will not greatly affect the value of the end to end delay. Because the average end to end delay is inversely proportional to the count of active members and because the average number of hops through the path to the sink is approximately constant with the increase on deployed nodes number, the curve represents the average end to end delay of EAP takes the inverse behavior of the curve represents the average number of active members per cluster.

The average end to end delay of LEACH is approximately constant by changing nodes number and this is due to that the average number of cluster members in LEACH network is constant and every packet sent by member nodes in the network passes an equal number of hops (two hops) to reach the sink. Fig. 12 demonstrates the average end to end delay for both EAP and LEACH. From Fig. 12, it could be concluded that the average delay of EAP is less than that of LEACH by on average 99.9% although in EAP, the packets from member nodes may go across more than two hops, this is mainly due to the previously mentioned requirement of LEACH for a wide TDMA schedule to accommodate all nodes expected to be deployed in the network minus one as members.

Table 1 Simulation parameters.

Parameters	Value
Network filed	(100,100)
Nodes number	100–500
Cluster radius r	30 m
Sensing radius r_s	10 m
Inter-cluster range R	$2 \times r$
Inter-cluster range RR	$3 \times r$
Sink position	(50,200)
Initial energy	2 J
Data packet size	525 Bytes
Broadcast packet size	25 Bytes
$E_{threshold}$	0.01 J
E_{elec}	50 nJ/bit
e_{fs}	10 pJ/bit/m^2
e_{amp}	0.0013 pJ/bit/m^4
E_{DA}	5 nJ/bit/signal
Threshold distance d_0	75 m
$RSS_{\max}$	−98 dBm
Coverage expectation	95%

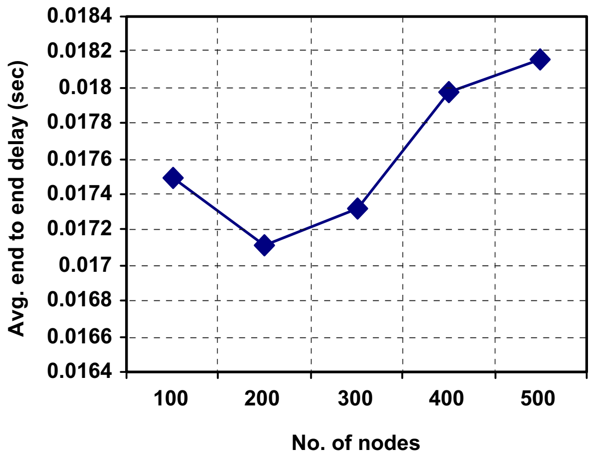

Figure 10 The avg. end to end delay for EAP vs. number of nodes.

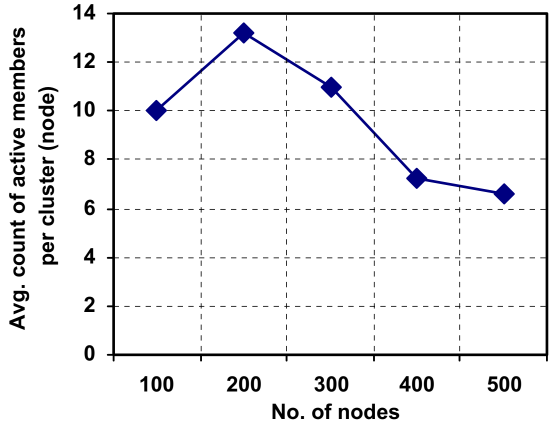

Figure 11 The avg. count of members per cluster for EAP vs. number of nodes.

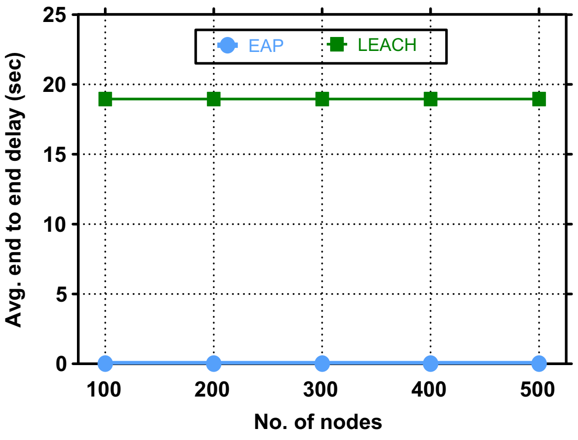

Figure 12 Average end to end delay vs. number of nodes.

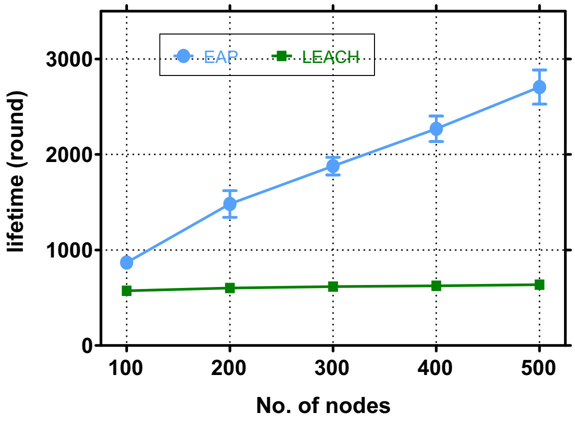

Figure 13 Network lifetime vs. number of nodes.

6.2.1.2. Lifetime. Fig. 13 demonstrates the lifetime of both EAP and LEACH. From Fig. 13, it is obvious that the lifetime of EAP increased with the number of deployed nodes due to the intra-cluster coverage method used. LEACH lifetime increases slightly with the number of deployed nodes as shown in Fig. 14 which represents a magnification of the part of Fig. 13 which represents LEACH lifetime curve. LEACH lifetime is less than that of EAP by on average 61.5% because LEACH bear with load unbalance and high energy consumption from both CHs and members.

6.2.1.3. Packet loss percentage. Fig. 15 demonstrates the average packet loss percentage for both EAP and LEACH. As mentioned the reason of packet loss in EAP is the lack of a mechanism that informs about CH death. Although LEACH has the same limitation and also in LEACH the probability of CH death is higher than in EAP because of higher CH energy consumption, the packet loss percentage of EAP is higher than that of LEACH by 35% as shown in Fig. 15. The count of LEACH lost packets is smaller than EAP because in EAP the network has only a single point of failure, i.e. if in a round the dead CH is the root, all the generated data in the whole network through this round will be lost. Also the fact that the count of the generated packets in LEACH is redundant and higher than that of EAP specifically when the number of

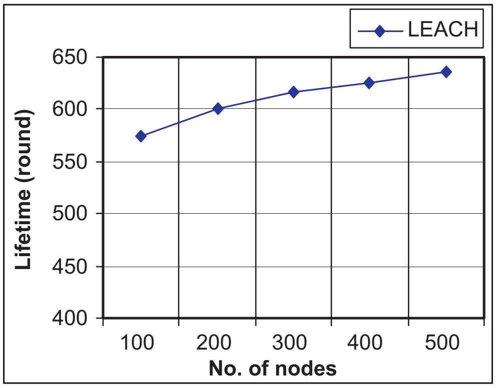

Figure 14 LEACH network lifetime vs. number of nodes.

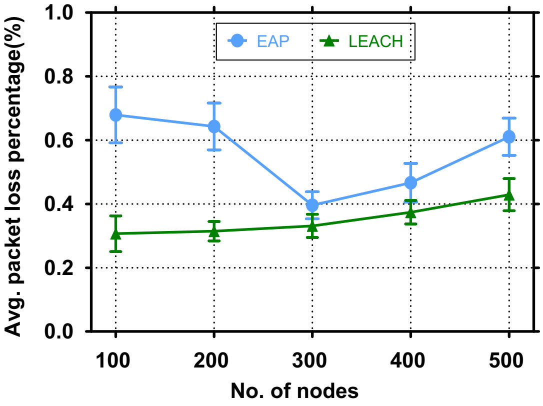

Figure 15 Average packet loss percentage vs. number of nodes.

deployed nodes increases cannot be ignored, so the percentage of the lost packets to the generated packets in LEACH network is smaller. LEACH generated packets are redundant and more than that of EAP when the number of nodes increases because LEACH does not use coverage method and all nodes are active in the same time that the coverage method used in EAP implementation in addition to it causes the increase of inactive members when the number of nodes increases, it causes the existence of a number of successive inactive rounds which do not used in sending data, rather these rounds consume nodes energy in sending control packets in the interval precedes activity test, so some nodes die quickly after or during these rounds without benefiting from them in sending data. Due to these reasons at 400 and 500 deployed nodes where the count of inactive members and successive inactive rounds increase, the average count of generated raw data packets decreases as shown in Fig. 16. From Fig. 15, it could be noticed that the curve of LEACH average packet loss percentage takes a stable behavior, while the curve represents EAP average packet loss percentage changes its behavior after the 300 node point taking the inverse behavior of generated raw data packets curve as it is inversely proportional to it.

6.2.1.4. Throughput. Fig. 17 demonstrates the average throughput of both EAP and LEACH. From Fig. 17, it could be noticed that the throughput of EAP is higher than that of LEACH by on average 99.6%. This is also because LEACH

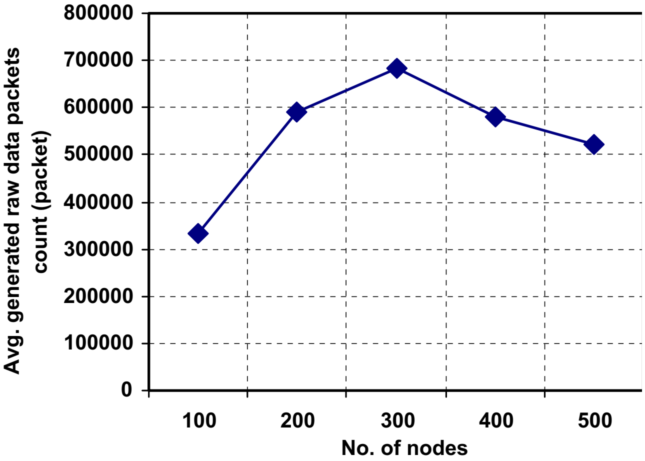

Figure 16 Average generated raw data packets count for EAP vs. number of nodes.

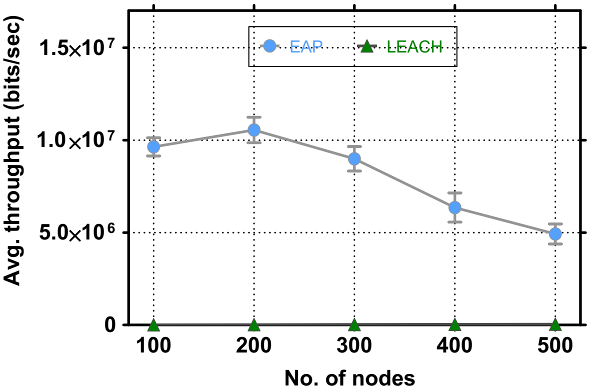

Figure 17 Average throughput vs. number of nodes.

design requires a wide TDMA schedule to accommodate the circumstances when only one CH formed and the remaining large number of nodes becomes its members, while the average number of cluster members in LEACH network is not large and constant, i.e., in most cases the long TDMA frame divided of a small number of nodes, as a result the slot time dedicated to each node becomes longer than its need, it sends its data packet in a small percent of this slot time and the remaining of the slot represents wasted time. This increases the wasted time through network lifetime, thus the throughput decreased.

It is obvious from Fig. 17 that the throughput of EAP decreased dramatically after the number deployed nodes increased over 200 nodes, but the difference between the lower point in EAP throughput curve and the higher point of LEACH throughput curve is still large. The throughput of EAP decreased mainly due to the appearance of a number of successive inactive rounds as mentioned before when the number of deployed nodes equals 300 and this number increased at 400 and 500 node, also because the throughput is directly proportional to the count of active members per cluster, it takes the same behavior as the curve represents the average count of active members per cluster shown in Fig. 11. Also it is expected that the decrease in the average count of active

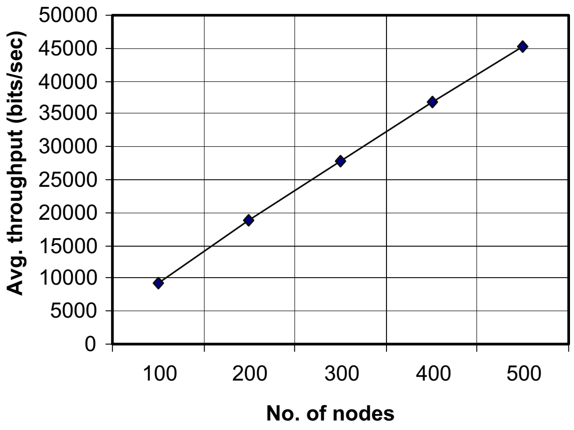

Figure 18 Average throughput for LEACH vs. number of nodes.

members in EAP and accordingly the decrease in the average throughput will not continue but they will approximately become steady at a certain node density and it appears from Figs. 11 and 17 that the change in their curves begins to decrease at 400 deployed nodes. The throughput of LEACH increases linearly with the number of deployed nodes; this is not clear in Fig. 17 due to the scale of the vertical axis and the big difference between LEACH and EAP curves, the curve represents LEACH throughput is redrawn in Fig. 18 with an appropriate scale.

6.2.2. *Performance analysis for LLEAP*

This section shows the results obtained from LLEAP evaluation and compares these results with the results obtained from EAP and LEACH evaluation. The results obtained are shown in Figs. 19–22, and illustrated in Sections 6.2.2.1–6.2.2.4.

6.2.2.1. Packet loss percentage. LLEAP overcomes the disadvantage of EAP which is the single point of failure by reducing this failure, i.e., reducing root death by reducing the aggregation and transmission loads on it. In LLEAP generally the number of CH death is reduced and in case of CH death, the death is at the end of the round after the CH sends the last aggregated data message. Therefore, LLEAP losses become less than EAP, and also less than LEACH. As shown in Fig. 19 the average loss percentage of LLEAP is below the average loss percentage of EAP by on average 93.4%, and below the average loss percentage of LEACH by on average 89.3%.

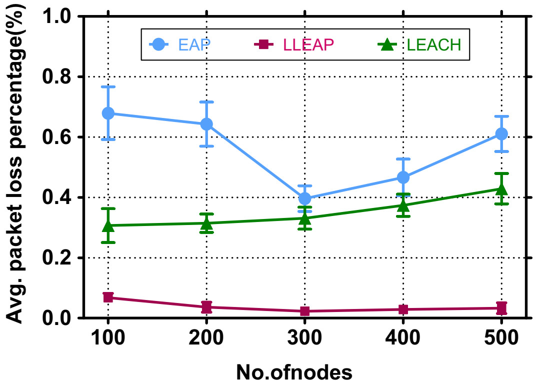

Figure 19 Average packet loss percentage vs. number of nodes.

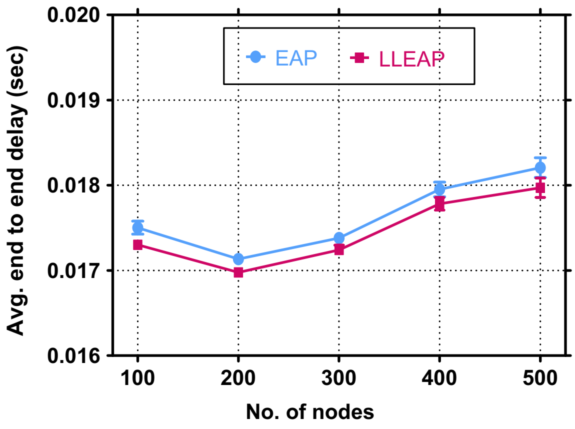

Figure 20 Average end to end delay vs. number of nodes.

6.2.2.2. End to end delay. Although the number of hops is increased and the path to the sink is elongated in LLEAP, the delay is decreased as shown in Fig. 20 by on average 1%, because the used method of aggregation made the waiting period for children data proportional to the level of the parent in the tree. From Fig. 12, it is known that LEACH delay is higher than EAP delay with high percentage, so from both Figs. 12 and 20 it is self-evident that LEACH delay is also higher than LLEAP with high percentage, so that in Fig. 20, it was sufficient to zoom in EAP and LLEAP curves only to clarify the difference between them.

6.2.2.3. Throughput. As shown in Fig. 21, LLEAP throughput is increased over EAP throughput by approximately 11.8% although the setup time increased. This is due to the decrease in the wasted time from LLEAP network, which untapped in sending data, as a result of the reduction of the successive inactive rounds count and at the same time the average number of generated data packets does not remain as it is in EAP, but it increased over it slightly, and packets loss decreased.

6.2.2.4. Lifetime. Without simulation it was difficult to expect whether the lifetime of LLEAP network will be less or more

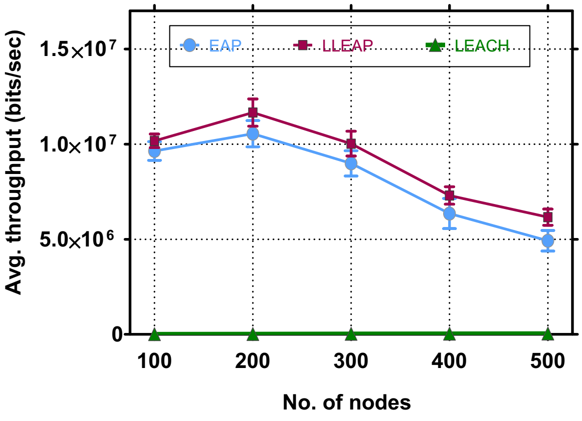

Figure 21 Average throughput vs. number of nodes.

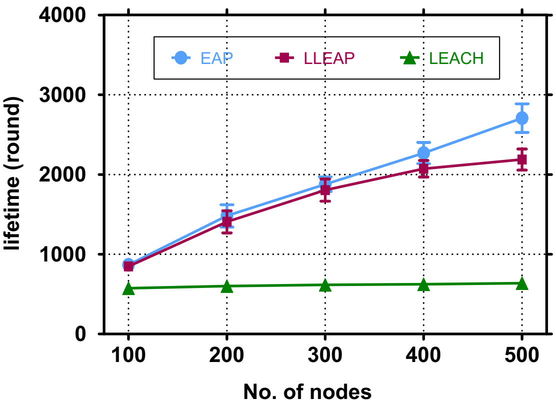

Figure 22 Network lifetime vs. number of nodes.

than EAP, because LLEAP uses a time schedule to save nodes energy and increase lifetime, in the same time, some modifications which used in LLEAP exhaust nodes energy. By using the simulation as shown in Fig. 22, the LND lifetime of the LLEAP is below the LND lifetime of EAP by approximately 7.8%. The percentage of LLEAP lifetime reduction is small when the number of deployed nodes is 100, 200, or 300, but this percentage increased notably when the number of deployed nodes increased to 400 and 500 nodes. This may not be considered as a disadvantage of LLEAP, because the increase in the percentage of LLEAP lifetime reduction is accompanied by an approximately equal percent increase in LLEAP throughput over EAP; this means that most of the large increase in EAP lifetime over LLEAP when the number of deployed nodes is 400 and 500 are not exploited in data transmission.

7. Conclusions and future work

This paper proposed a modified version of EAP which is called LLEAP. LLEAP has the same three phases of EAP except some modifications in each phase. LLEAP modifies EAP in terms of some QoS parameters by modifying the weights equations, adding a second iteration for tree construction, using schedule technique for nodes sleep and awakening to save nodes energy, and using an aggregation method decreases delay and packet losses. Simulation results showed that LLEAP offers improvements over EAP in loss percentage, throughput, and delay by on average 93.4%, 11.8%, and 1% respectively, while LND lifetime reduced by 7.8%.

There are several future works we would like to focus on such as: first, how to increase the network lifetime without affecting the other performance metrics of LLEAP. Second, using a best method for canceling the second tree construction iteration when only one root formed in the first iteration. Finally, trying to get over the constraints imposed upon the protocol by some of the suggested assumptions and making the protocol satisfies other applications' requirements such as mobility, node addition, and event-driven or query-driven data delivery methods.

References

[1] Boukerche A. Algorithms and protocols for wireless sensor networks. John Willey & Sons Inc.; 2009.
[2] Misra S, Woungang I, Misra SC. Guide to wireless sensor networks. Springer-Verlag London Limited; 2009.
[3] Li Y, Thai M, Wu W. Wireless sensor networks and applications. Springer Science & Business Media LLC; 2008.
[4] Kutyowski M, Cichoń J, Kubiak P. Algorithmic aspects of wireless sensor networks. Springer-Verlag London Limited; 2008.
[5] Swami A, Zhao Q, Hong Y-W, Tong L. Wireless sensor networks signal processing and communications perspectives. John Wiley & Sons; 2007.
[6] Liu M, Cao J, Chen G, Wang X. An energy-aware routing protocol in wireless sensor networks. J Sens 2009;9(1):445–62.
[7] Liu M, Zheng Y, Cao J, Chen G, Chen L, Gong H. An energy-aware protocol for data gathering applications in wireless sensor networks. In: Proc of the IEEE international conference on communications (ICC 2007), Glasgow; 2007. p. 3629–35.
[8] Heinzelman W, Chandrakasan A, Balakrishnan H. Energy-efficient communication protocol for wireless microsensor networks. In: Proc of the 33rd annual Hawaii international conference on system sciences; 2000. p. 1–10.
[9] Heinzelman W, Chandrakasan A, Balakrishnan H. An application specific protocol architecture for wireless microsensor networks. J IEEE Trans Wire Commun 2002;1(4):660–70.
[10] Younis O, Fahmy S. HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans Mobile Comput 2004;3(4):366–79.
[11] Loscri V, Morabito G, Marano S. A two-levels hierarchy for lowenergy adaptive clustering hierarchy (TL-LEACH). In: Proc of IEEE 62nd vehicular technology conference; 2005. p. 1809–13.
[12] Kim KT, Youn HY. PEACH: proxy-enable adaptive clustering hierarchy for wireless sensor network. In: Proc of the international conference on wireless network; 2005. p. 52–7.
[13] Ye M, Li C, Chen G, Wu J. EECS: an energy efficient clustering scheme in wireless sensor networks. In: Proc of IEEE intl performance computing and communications conference (IPCCC). Los Alamitos: IEEE Computer Society Press; 2005. p. 535–40.
[14] Sangho Y, Junyoung H, Yookun C, Jiman H. PEACH: power-efficient and adaptive clustering hierarchy protocol for wireless sensor networks. Comput Commun 2007;30:2842–52.
[15] Kim J-H, Chauhdary S, Yang W-C, Kim D-S, Park M-S. PRODUCE: a probability-driven unequal clustering mechanism for wireless sensor networks. In: Proc of 22nd international conference on advanced information networking and applications—workshops, Okinawa, Japan; 2008. p. 928–33.
[16] The omnet++ homepage. <http://www.omnetpp.org>.
[17] Santi P. Topology control in wireless ad hoc and sensor networks. John Willey & Sons Ltd.; 2005.
[18] Nassef L. On the effects of fading and mobility in on-demand routing protocols. Egypt Inf J 2010;11(2):67–74.
[19] Gong H, Liu M, Mao Y, Chen L, Xie L. Distributed energy efficient data gathering with intra-cluster coverage in wireless sensor networks. APWeb, LNCS 3841; 2006. p. 109–20.