Modified LEACH – Energy Efficient Wireless Networks Communication

Abuhelaleh, Mohammed Elleithy, Khaled Mismar, Thabet
School of Engineering, University of Bridgeport
Bridgeport, CT 06604
{mabuhela, elleithy, tmismar} @bridgeport.edu

Abstract-**Many algorithms and techniques were proposed to increase the efficiency of Sensor Networks. Due to high restrictions of this kind of networks, where the resources are limited, many factors may affect its work. Theses factors are: System throughput, system delay, and energy. Clustering protocols have been propose to decrease system throughput and system delay, and increase energy saving. In this paper, we propose a new technique that can be applied to sensor networks to produce high performance and stable Sensor Networks.**

Index Terms- **LEACH (Low Energy Adaptive Clustering Hierarchy), Sensor Networks, Network Performance, Routing.**

I. Introduction

There are many advantages of using sensor networks. They provide dynamic and wireless communication between nodes in a network, which provides more flexible communication. At the same time, sensor networks have some special characteristics compared to traditional networks, which makes it harder to deal with. The most important property that affects this type of networks is the limitation of the resources available, especially the energy.

Wireless Sensor Networks (WSNs) [2] are a special kind of Ad hoc networks that became one of the most interesting areas for researchers. Routing techniques are the most important issue for networks where resources are limited. Cluster-based organization has been proposed to provide an efficient way to save energy during communication [3]. In this kind of organization, nodes are organized into clusters. Cluster heads (CHs) pass messages between groups of nodes (group for each CH) and the base station (BS), (Figure1). This organization provides some energy saving which is the main advantage for proposing this organization. Depending on this organization, LEACH (Low Energy Adaptive Clustering Hierarchy) [3] enhanced security, where the CHs are rotating from node to node in the network making it harder for intruders to know the routing elements and attack them. [4]

In this paper, we discuss some existing work of LEACH and we focus on two important criteria; the performance and energy consumption. In section two, we discuss the original work of LEACH, and then in the third section we discuss one of the most interesting modifications proposed for LEACH to increase network performance (TCCA). In the fourth section, we discuss our proposal and we explain the main modifications that we applied on LEACH to improve network performance. In section 5, we discuss our experiment that we applied to show the improvements that may gain from applying our protocol comparing to the existing protocols.

II. Leach

Low Energy Adaptive Clustering Hierarchy has been presented by [1] to balance the draining of energy during communication between nodes in sensor networks. The BS assumed to be directly reachable by all nodes by transmitting with high enough power. Nodes send their sensor reports to their CHs, which then combine the reports in one aggregated report and send it to the BS. To avoid the energy draining of limited sets of CHs, LEACH rotates CHs randomly among all sensors in the network in order to distribute the energy consumption among all sensors. It works in rounds; in each round, LEACH elects CHs using a distributed algorithm and then dynamically clusters the remaining sensors around the CHs. Sensor-BS communication then uses this clustering result for the rest of the round. (See Fig.1)

A. LEACH Protocol

Routing in LEACH works in rounds and each round is divided into two phases, the Setup phase and the Steady State; each sensor knows when each round starts using a synchronized clock [1, 2].

Initially, each sensor decides if it will be a CH or not based on the desired percentage of the CHs for the network, and the number of times the sensor has been a CH (to control the energy consumption), this decision is made by the sensor (s) choosing a random number between Zero and One. Then it calculates the threshold for (s) T(s), then it compares the random number with resulting T(s); if the number is less than

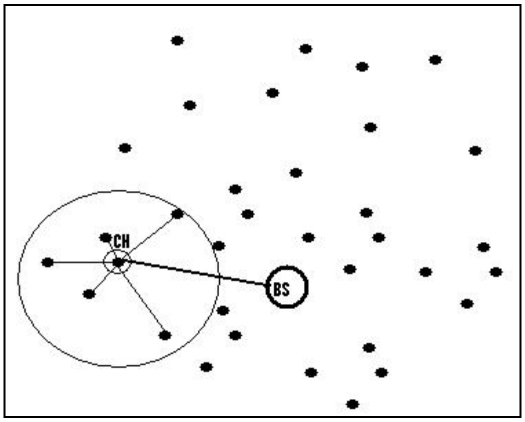

Figure1. Cluster organization for sensor networks

T(s), (s) becomes a CH for the current round. T(s) for x round with desired percentage of cluster heads P is calculated by (1):

$$\text{T(s)} = \left\{ \begin{array}{l} P \dfrac{P}{1 - P * (x \bmod \frac{1}{P})} \ldots\ldots. if n \in G \\ 0 \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots. otherwise \end{array} \right\} \ldots\ldots.. (1)$$

G is a set of nodes that have not been CHs in the last 1/p round.

Setup phase includes three steps. Step1 is the advertisement step, where each sensor decides its probability to become a CH, based on the desired percentage of CHs and its remaining energy, for the current round; Sensor who decides to become a CH broadcasts an advertising message to other nodes that it is ready to become a CH. Carrier sense multiple access protocol is used to avoid the collision. Clustering joining step is the second step, where the remaining sensors pick a cluster to join according to the highest signal received; then they send request messages to the desired CHs. Step three starts after the CHs receive all requests from other sensors, where CHs broadcast confirmation messages to their cluster members; these messages include the time slot schedule to be used during the steady state phase.

The Steady State phase (the actual communication) then starts. It consists of two steps; in the first step each nodes starts by send its sensor report to its CH based on the time provided by the time slot schedule. When CH receives all the reports, it aggregates them in one report and it sends this report to the BS (step 2). Next we show the details of each step by providing the content of each one; for this purpose we combine the two phases in one phase with five steps.

In step one, CH broadcasts to the rest of sensors, its ID and the Advertising message, then, in step two, each sensor sends its ID, CH ID, and the Join Request message to its desired CH. When CH received all requests, it broadcasts its ID, and the time slot schedule for sensors that includes each member with its time slot (step three). Each sensor then sends its ID, CH ID, and the sensing report to its CH (step four). Finally, each CH sends its ID, BS ID, and the aggregate report of its members to the BS.

The transmission of information between sensors, and between sensors and BSs, are performed using CSMA MAC protocol. On the other hand, they communicate using CDMA codes to reduce the interference that may occur from communication of nearby nodes.

B. Energy saving in LEACH

LEACH is a self-organization adaptive protocol, and it uses randomization to evenly distribute the energy load among the sensors in the network; this and the random way that CHs rotate around the various sensors reduce the possible draining of the battery for each sensor.

A local data compression to compress the amount of data being sent from clusters to BS is used to reduce the energy consumption and to enhance the system lifetime.

The time schedule that is being performed by CHs to their members, gives break time for sensors that have not reached their time yet, to be in sleeping mode which helps them save their energy for their scheduled time.

Finally, the nature of the way that LEACH changes CHs each round, and the way that each CH can be elected, provides high energy saving for whole network.

C. Security in LEACH

LEACH is more powerful against attacks than most other routing protocols [2, 4]. CHs in LEACH that directly communicate with BS can be anywhere in the network and they are changing from round to round, which makes it harder for intruders to identify the critical nodes in the network.

On the other hand, LEACH is vulnerable to a number of security attacks [2, 4], including spoofing, jamming, and replay attacks. Since LEACH is a cluster based protocol, it relies mainly on the CHs for routing and data aggregation, which makes the attacks involving CHs, the most harmful attacks.

Some kinds of attacks, such as sinkhole and selective forwarding, may occur if an intruder manages to become a CH, which results in disrupting the work of the network.

III. TCCA

Time-Controlled Clustering Algorithm (TCCA) allows multi-hop clusters using message time-to-live (TTL) and timestamp to control the way the clusters form. Residual energy is also considered before a sensor volunteers to become a CH, and a numerical model is provided to quantify its efficiency on energy usage.

A. TCCA Protocol

Similar to LEACH, TCCA's operation is divided into rounds with two phases concluded in each round (Setup phase and the Steady State phase). CHs are elected and the clusters are formed in Setup phase; then the complete cycle of data collection, aggregation and transfer to the BS occurs in the Steady State phase.

To determine the eligibility of sensor to be CH, TCCA adds some modifications to the LEACH technique. A sensor residual energy is considered and a random number between 0 and 1 (Tmin) is generated by each sensor to determine its eligibility to become CH. If this number is less than the variable threshold, the sensor becomes a CH for the current round. The threshold for sensor 's' in round r, with desired CH percentage *p*, residential energy RE and maximum energy MaxE is calculated by (2):

$$T(s) = \left\{ \begin{array}{ll} \max(P \dfrac{p}{1 - p(r \bmod \frac{1}{p})} \times \dfrac{RE}{MaxE} & , T\min) \ldots \forall s \in G \\ 0 & \forall s \notin G \end{array} \right. \quad \ldots\ldots (2)$$

G is a set of nodes that have not been CHs in the last 1/p round

When CH is elected, it advertises to other sensors to become its members; this advertisement message contains CH ID, initial TTL, timestamp and its residual energy. Sensors receive the message will forward it to their neighbors based on TTL value which may be based on the current energy level of CH; at the same time they join this CH with the rest of sensors who received the message. Once a sensor decides to join the cluster, it informs the corresponding CH by sending a join request message that carries sensor ID, CH ID, the original timestamp from advertising message and the remaining TTL value. The CH uses the timestamp to approximate the relative distance of its neighbors and to learn the best setup phase time for future rounds [3].

The time schedule that is to be advertised by the CH is based on the total number of its members and their relative distance, to avoid collision.

Timestamp and TTL are used in TCCA to give the CH the ability to produce multi-hops clusters in efficient way that has the same performance of the one-hop clusters.

B. Energy saving in TCCA

TCCA applies a new condition for electing CHs by considering the remaining energy of the sensors. At the same time, it guarantees that every sensor will become a CH at least one time per 1/P rounds, where P is the desired percentage of CHs. These modifications provide the network with high energy balance by distributing the energy among all sensors.

TCCA provides optimum cluster size (K) for K-hops in order to produce high performance similar to the performance in the one-hop network. Also it reduces the complexity of transmission schedule generation to O (1).

TCCA uses timestamp and Time to live (TTL) tags to control the cluster formation; this leads to gain more energy balance.

C. Security in TCCA

TCCA follows the main steps provided by LEACH with some modifications that do not affect the level of security that is provided by LEACH; this means that TCCA does not have enough protection against Spoofing, Jamming, Replay and some other kind of attacks.

IV. MODIFIED-LEACH

The operation of Modified-LEACH (Mod-LEACH) works in two rounds: a Full transmission round and a half transmission one.

Each sensor checks its ability to become a CH depending on the desired percentage of CHs, current round, and the remaining energy; we used the same formula used by TCCA to calculate the threshold.

The sensors that are able to become CH (ready sensors) for the current round start listening for any query that might be sent by other sensors; the other sensors start broadcasting their reports to their neighbors, the packets contain some other tags to determine the status of the packets; any ready sensor that receives the report saves it temporarily and sends a confirmation/request to the related sensor confirming that it is ready to send its report and providing its ability status to become a CH for next round. Sensors who receive the confirmation, reply back to the CH with another confirmation and save the CH id to use it for the next round (if the status of the CH shows that it is able to be a CH for two rounds; CHs will collect all the reports that have been confirmed in one compressed report and forwards it to the B.S. (This is considered as a full transmission round).

For the next round, the sensors with no CHs will repeat the same scenario, and the sensors with CHs will send the report only to their CHs; when the old CHs receive the reports, it will aggregate them in one report and forward it to the B.S. (this considered as a half transmission report).

Next we will explain in details the complete protocol.

A. Mod-LEACH Protocol

The operation of the Mod-LEACH occurs in rounds, and rounds are classified into two kinds, the full transmission round and the half transmission round. The main idea here is to skip the setup phase that is proposed by all other discussed protocols.

At the beginning of each round, CHs elect themselves. In order to determine the eligibility of sensor to be a CH, each sensor (S) generates a random number between 0 and 1; then this number is compared to a sensor variable threshold value T(S); if the value of the threshold is greater than the random number, the sensor becomes a CH for the current round (R). The Threshold value can be calculated using the same formula that is used by TCCA; first it calculates the threshold for two rounds as follows:

$$T(S)a = \left\{ \begin{array}{ll} \max(P\dfrac{P}{1-P(R \bmod \frac{1}{P})} \times \dfrac{\mathrm{Re}mEng}{MaxEng*2}, ...T\min)...if S \in G \\ 0 & otherwise \end{array} \right\} ...(3)$$

If formula (3) is approved, then it is ready to become a CH for two rounds. If formula (3) is not approved, the sensor will calculate the formula (4) to see if it is able to become a CH for only one round.

$$T(S) = \left\{ \begin{array}{ll} \max(P\dfrac{P}{1-P(R \bmod \frac{1}{P})} \times \dfrac{\mathrm{Re}mEng}{MaxEng}, ...T\min)...if S \in G \\ 0 & ,otherwise \end{array} \right\} ...(4)$$

Where P is the desired percentage of CHs, Tmin is a minimum threshold (to avoid the possibility of remaining energy shortage), and G is the set of sensors that have not became CHs in 1/P round, MaxEng is the maximum energy that the sensor could have, RemEng is the sensor remaining energy.

Each elected CH starts listing to the network; other sensors start broadcasting their reports to their neighbors (using Carrier sense multiple access protocol for transmission to avoid collisions); this message consists of Sensor ID, report, Requesting type tag (RT: 0 for request, 1 for approves), Time to live (TTL: set to 1, broadcast to only direct neighbors), packet request status tag (PR: 0 for the first packet, 1 for the second packet). When ready sensors (CHs) receive the messages, it saves each report with the node id temporarily in its memory, and then it sends requests with confirmation to those sensors indicating that it is ready to become their CH for the current round, when formula3 applies, it also indicates that it is also ready to be their CH for the next round; the message contains: CH id, pairs of Sensor id with its time (to prevent collisions and provide less delay), TTL (set to 1), RT (set to 1), PR (set to 0) and the ability tag (AT: 0 for one round ability, and 1 for two rounds ability). Sensors receive the message form CHS; if they receive more than one request then they will choose the one with the ability to become CH for two rounds, AT=1 (here it will save the CH id to use it in the next round); if they receive many requests with the same values, then they pick the CH randomly; Sensors then reply to CHs with confirmation; the message contains: Sensor id, CH id, and an Acknowledgment tag (ACK: set to 1). When a CH receives the confirmations it combines all the reports that it has in one compressed report and forwards it to the B.S.; the message contains: CH id, BS id and the aggregation report.

In the next round, sensors check first if they are group members of a CH with an ability to handle two rounds, if they are, then they use it for the current round (half transmission round is applied); the sensor sends its report to its CH; the message contains: Sensor id, CH id, PR (set to 1), TTL (set to 1), PR (set to 1). CH receives the reports, aggregate then in one report, and then send them to the B.S., the message contains: Ch id, B.S id, and the aggregation report; then the CH will send acknowledgments to its members and remove them from its memory; the acknowledgment message contains: Ch id, Sensor id, and ACK (set to 1); sensors who receive the acknowledgment then remove CH info from their memories.

In the case that the sensor does not have a CH from the previous round, it will repeat the first scenario for full round transmission.

B. *Energy saving in Mod-LEACH*

Mod-LEACH applies the same condition that has been applied by TCCA for electing CHs by considering the remaining energy of the sensors. At the same time, it guarantees that every sensor will become a CH at least one time per 1/P rounds, where P is the desired percentage of CHs. These modifications provide the network with high energy balance by distributing the energy among all sensors.

Mod-LEACH provides enhanced energy saving by dealing with double round technique, where it saves almost half of the energy used in one regular round; for the full transmission round, it will consume more energy than LEACH and TCCA, but it covers that gab in the next round, and even saves more total energy than other protocols may save.

Mod-LEACH uses Time to live (TTL) tags to control the cluster formation, where the broadcasting occurs only on the direct neighbors; this leads to a more energy balanced network.

C. *Security in Mod-LEACH*

Mod-LEACH provides the same level of security that has been provided by LEACH and TCCA, where it didn't affect the main idea of these protocols which is the dynamic rotation of CHs around the network.

V. Experimentation And Analysis

In this section, we discuss the numerical experimentation; here we describe the chosen parameters groups for each protocol, following the same scenario. The experiment is applied on LEACH, TCCA, and Mod-LEACH protocols; we applied them on three different network sizes (100, 1000, and 10000 sensors); for each size, 1000 rounds were processed with the following initial values of main parameters:

- The desired percentage of CHs (P) is set to 0.05.
- Each sensor starts with 0.5 j energy.
- The amplifier energy is assumed to be 100 pj.
- The electronic energy is assumed to be 50 nj.
- Each sensor data range is set to 30m.
- The message size of a sensor data is set to 50 bits.
- Each node has 2000-bit data packet to send to the BS.

Next, we analyze the results that appeared from applying our experiment on each of the protocols discussed before, using the same initial values and following the same scenario. We start with comparing the results based on energy saving results from each protocol, and then we discuss them based on data overload produced by each protocol, then we compare the results based on the number of the dead sensors at the end of the experiment for each protocol..

A. *Energy saving*

LEACH provides many techniques to save energy during network communication; where it is a self-organization, adaptive protocol and it uses randomization to evenly distribute the energy load among the sensors in the network, in addition to the random way that CHs rotate around the various sensors which is reducing the possible draining of the battery for each sensor. Also, performing a local data compression to compress the amount of data being sent from clusters to BS reduces the energy consumption and enhances the system lifetime. These factors, in addition to the way that CHs change every cycle provide LEACH with High energy saving. Mod-LEACH applies the same factors to Sensor networks which provide it with similar energy saving to the LEACH at this point.

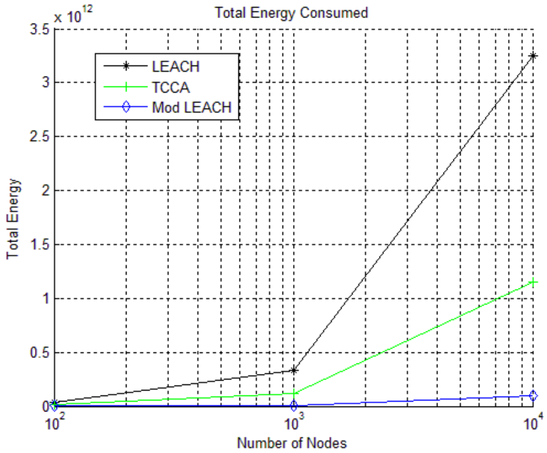

Fig.2. Total energy consumption in LEACH, TCCA, and Mod-LEACH after 1000 rounds for different network sizes (100, 1000, 10000).

TCCA adds additional factors to save energy; it uses Time to live (TTL) tag to control the cluster formation, which leads to gain more energy balance. It also uses the new condition provided by TCCA to elect CHs each round, which results in more energy control. [3] Shows that TCCA works almost three times better than LEACH in energy saving. Mod-LEACH applied TCCA factors, which means that it works three times better than LEACH in energy saving. Now by applying the new idea that we discussed before, we can notice that Mod-LEACH provides the network with almost four times more energy saving than what is provided by LEACH, and almost double of that in TCCA.

Our experiment shows that the variation of energy consumption is very small when network size is small (i.e. 100 sensors), but it varies more if we increase the network size. Fig.2 shows that, for network size of 10,000 sensors, total energy consumption is minimum in Mod-LEACH with almost $0.1X10^{12}$ nj, then TCCA comes with energy consumption of almost $1.2X10^{12}$ nj at second place, and last comes LEACH with $3.3X10^{12}$ nj. The variation comes from the nature of how Mod-LEACH works; using TTL, in addition to continuous checking of residual energy of each sensor, gives Mod-LEACH and TCCA protocols more energy balance for large network size; working and double round technique provides Mod-LEACH with more energy saving.

B. Data Overload

TCCA works with multi-hops clusters; this reduces the number of clusters, which reduces the total transactions required in network communications; this leads to highly reduce data overload compared with LEACH. Mod-LEACH has two different round types; in the full transmission round it will produce more data overload than that produced by TCCA and LEACH, but by applying the half transmission technique on the next round, we balance the increase in the data in the previous round and we provide less total data overload than that provided in double rounds with LEACH and TCCA.

Our experiment shows that, for a large network size (i.e. 10000 sensors); the total data overload is minimized using Mod-LEACH. Fig.3 shows that with Mod-LEACH data overload reaches almost $0.1X10^{12}$ bits, where in TCCA it reaches $2.2X10^{12}$ bits and in LEACH it reaches $9.3X10^{12}$; this shows that LEACH produces more data overload, almost nine times more than the data overload produced by Mod-LEACH. Moreover, TCCA produces more data overload, almost twice as much data overload produced by Mod-LEACH.

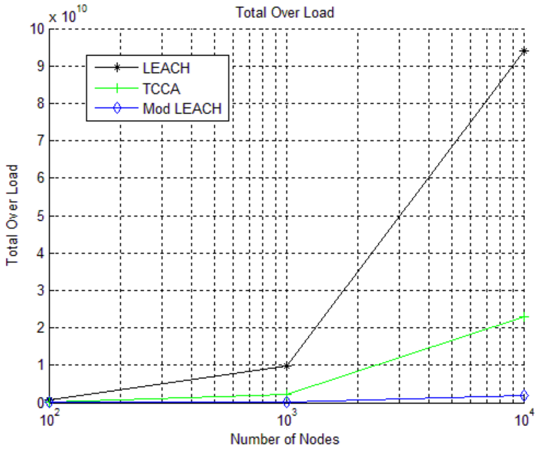

Fig.3. Total data overload in LEACH, TCCA, and Mod-LEACH after 1000 rounds for different network sizes (100, 1000, 10000).

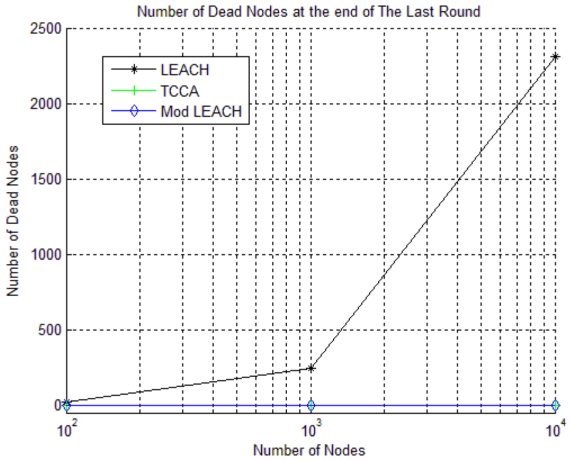

Fig.4. Dead nodes occur in LEACH, TCCA, and Mod-LEACH after 1000 rounds, for different network sizes (100, 1000, 10000).

C. Performance

Here, we analyze the performance based on the expected Dead Nodes that may result in each solution after the same number of rounds.

According to the energy saving analysis, we can figure out that the number of Dead Nodes that may appear in LEACH will be much higher than the number of dead nodes in Mod-LEACH, where the number of Dead Nodes depends on the energy consumption by the network.

Fig.4 shows that Mod-LEACH and TCCA remain completely alive (i.e. no dead sensors) after 1000 rounds. On the other hand, in the case of 10000 sensors network size LEACH results in almost 2300 dead sensors.

VI. CONCLUSIONS

Modified-LEACH provides large sensor networks with high energy saving, and high level of performance, more than nine times better than LEACH and twice better than TCCA. At the same time it produces a much higher level of network stability than offered by LEACH. These results show that our proposal provides an efficient solution for high performance sensor networks communication.

REFERENCES

[1] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In IEEE Hawaii Int. Conf. on System Sciences, pages 4–7, January 2000.

[2] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, A. A. F. Loureiro. SecLEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks. Fifth IEEE International Symposium on Network Computing and Applications (NCA'06)

[3] S. Selvakennedy, and S. Sinnappan. A Configurable Time-Controlled Clustering Algorithm for Wireless Networks. 2005 11th International Conference on Parallel and Distributed Systems (ICPADS'05).

[4] Chris Karlof, Naveen Sastry, and David Wagner. TinySec: A Link Layer Security Architecture for Wireless Sensor Networks. 2004 Conference on Embedded Networked Sensor Systems Proceedings of the 2nd international conference on Embedded networked sensor systems.

[5] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia. Establishing Pairwise Keys for Secure Communication in Ad Hoc Networks: A Probabilistic Approach. Proceedings of the 11th IEEE International Conference on Network Prot1ocols (ICNP'03)