QUID 2017, pp. 1073-1084, Special Issue N°1- ISSN: 1692-343X, Medellín-Colombia

REDUCING ENERGY CONSUMPTION FOR CHAIN-BASED ROUTING PROTOCOLS IN LARGE-SCALE WIRELESS SENSOR NETWORKS

(Recibido el 13-06-2017. Aprobado el 08-09-2017)

Reza Mohamaddoust
Payame Noor University (PNU),
Department of IT Engineering,
IRAN.
pnu1392@yahoo.com

Madjid Khalilian
Islamic Azad University,
Department of Computer Engineering Karaj Branch,
IRAN.
khalilian@kiau.ac.ir

RESUMEN: Las redes de sensores que consisten en nodos con energía de batería limitada y comunicaciones inalámbricas se despliegan para recopilar información útil del campo. La recopilación de información detectada en un método de eficiencia energética es fundamental para mejorar la vida útil de la red en las redes de sensores inalámbricos basadas en cadenas. En esta investigación, presentamos tres métodos para la corrección de cadenas en el protocolo PEGASIS. Uno de los principales puntos de debilidad del protocolo PEGASIS es la probabilidad de muerte temprana de algunos nodos en la red, su razón es la existencia de los bordes largos en la cadena. Los métodos propuestos en esta investigación previenen la transmisión larga entre sensores usando algoritmos distribuidos y corrigiendo los bordes largos de la cadena. Asimismo, presentamos un método de corrección de cadenas en las redes en las que los nodos realizan sus operaciones sin tener conocimiento de sus propias ubicaciones o de otras redes. Estas correcciones conducen a la reducción de la energía consumida por los sensores. Esta reducción de la energía consumida lleva a mejorar la vida útil de los sensores. La mejora de la vida útil de los nodos aumenta el tiempo de cobertura total de la red. El aumento del tiempo de cobertura total de la red conduce a mejorar la disponibilidad, fiabilidad y seguridad en las redes a gran escala. Finalmente, los métodos propuestos se comparan con el protocolo PEGASIS y el método PEGASIS intragráfico. Los resultados de la simulación muestran que en los métodos propuestos hay mejoras significativas en la vida útil de la red y en el tiempo de cobertura total de la red.

Palabras clave: red de sensores inalámbricos, vida útil de la red, protocolos de recopilación de datos en cadena, protocolo PEGASIS.

ABSTRACT: Sensor networks consisting of nodes with limited battery power and wireless communications are deployed to collect useful information from the field. Gathering sensed information in an energy efficient method is critical to enhancing the network efficient lifetime in the chain-based wireless sensor networks. In this research, we present three methods for the chain correction in the PEGASIS protocol. One of the main points of weakness of the PEGASIS protocol is the probability of early death of some nodes in the network, its reason is the existence of the long edges in the chain. The proposed methods in this research prevent the long transmission between sensors by using distributed algorithms and correcting the long edges of the chain. Also, we presented a chain correction method in the networks in which nodes perform their operations without awareness of their own locations or that of other network.

Citar, estilo APA: Mohamaddoust, R., & Khalilian, M. (2017). Reducing energy consumption for chain-based routing protocols in large-scale wireless sensor networks. *Revista QUID (Special Issue)*, 1073-1084.

These corrections lead to reduction of the energy consumed by sensors. This reduction of the energy consumed lead to enhancing the sensors lifetime. The improvement of nodes lifetime increases the full coverage time of the network. The increase of full coverage time of the network leads to improve availability, reliability and safety in the large scale networks. Finally, the proposed methods are compared with the PEGASIS protocol and intra-grid PEGASIS method. Simulation results show that in the proposed methods there are significant improvements in the network efficient lifetime and the full coverage time of the network.

Keywords: wireless sensor network, network efficient lifetime, chain-based data gathering protocols, PEGASIS protocol.

1. INTRODUCTION

Advancements in integrated circuits, micro electro-mechanical systems (MEMS) and communication theory have resulted in the development of wireless sensor networks (WSN). A sensor node is a low cost, low power and multi-functional electronic device consisting of sensing, processing and communicating components. Each sensor node is reliant solely on its limited battery power. The sensor nodes are randomly distributed in the monitored region. When the network is established and activated, it is assumed that there will be no physical access to the nodes. Lack of access to the nodes results in the node being removed from the networks once its battery power is exhausted. Therefore, the network lifetime is dependent on the lifetime of its distributed sensor nodes. WSNs are used in various fields such as wildlife observation (Mainwaring et al. 2002), disaster relief (Cayirci et al. 2007), monitoring of hazardous environments (Werner et al. 2006), etc. Communications and calculations are the most important operations which are performed in a WSN. The power requirements of communication operations are significantly greater than those required for calculations. Therefore most energy is consumed through sending and receiving activities.

Considering high energy consumption in communications, the limitation of sensor energy resources and an increase of the WSN lifetime; designers are looking for approaches and architectures which have efficient energy consumption in the all network operations such as data detecting, collecting, transmitting and receiving. Additionally, designers are considering methods to increase the network lifetime. This involves equity of energy consumption in all sensor nodes to prevent energy depletion drop out of sensors in a special part of the network.

In an effort to increase network lifetime, the authors in (Zhu et al. 2010) tried to optimize network lifetime by using a genetic algorithm. Other research such as (Guo et al. 2010; Acharya et al. 2009) imitated ants' behavior in routing operation. Authors in (Lindsey et al. 2002; Ding et al. 2003; Rohankar et al. 2015; Tan et al. 2003; Meghanathan et al. 2010; Rana et al. 2014; Rana et al. 2015; Gengsheng et al. 2009; Pal et al. 2010; Shin et al. 2008; Min et al. 2008) tried to reduce the energy consumed by sensor nodes through corrections in the architecture of WSNs; chain-based architectures are samples of this architectures. In a chain-based architecture, all sensor nodes are organized into a chain where its overall length is minimized using a greedy algorithm method (Lindsey et al. 2002). In this chain, each node receives the data from the previous neighbor, fuses it with its own data and transmits to the next neighbor in the chain. Finally, the leader node in the chain fuses the received data from its neighbors with its own data and sends it to the sink. In this type of architecture, all of the sensors as the leader can transmit the data to the sink, alternatively which this act can help distribute energy load uniformly between all of the sensors throughout the network.

Security has been introduced as a combined concept which includes; availability, reliability, safety, confidentiality and integrity (Avizienis et al. 2001). The amount of emphasis on each of these factors varies according to the type of network application. The improvement of nodes lifetime extend the availability, reliability and safety duration of the network; therefore the improvement of each of these factors improves network security. For instance if the network monitors a border area between two countries, the lifetime improvement of all sensor nodes increases the full coverage time of the network. This increase results in an economic improvement when surveillance time, data collection and access to network information are considered. As a result network reliability can be said to increase.

This study uses three methods which pay attention to the chain correction in power-efficient gathering in sensor information systems (PEGASIS), a chain-based protocol. Chain correction in PEGASIS improves the network lifetime remarkably, resulting in improved security in the monitored field. The proposed methods have been compared with PEGASIS, intra-grid PEGASIS and ideal PEGASIS. Simulations results confirm the above claims. The related works are presented in section 2. In section 3, the radio model for energy calculations used throughout this paper is discussed. Section 4 describes proposed methods. Simulation results are presented in Section 5. Finally, some concluding remarks are given in Section 6.

2. RELATED WORKS

PEGASIS (Lindsey et al. 2002; Ding et al. 2003) is a chain-based protocol in which each sensor nodes are randomly distributed in the environment under supervision. It is assumed that each sensor performs some of the functions; data detection, wireless communication, data fusion and each achieves global knowledge of its own location and that of other sensor nodes in the network. The chain construction is started before the first round of communication from the farthest node to the sink (if there are some nodes with equal distances to the sink, one of them is chosen at random to be the 1st node in the chain). Then the nearest neighbor to this selected node is chosen as the next node of the chain. The next neighbors in the chain are chosen accordingly by a greedy algorithm of unvisited nodes. After the chain formation, one node is chosen as a leader to transmit to the sink. The chain leader in the round *i* is a node of the chain where its index is equal to (*i mod N*), *N* represents the number of live nodes in the present round. Therefore, in a network with *N* nodes, each node can be a leader approximately once every *N* rounds. Consequently, the leader in each round of communication will be at a random position on the chain. This is important as nodes to die at random locations. Node death at random places is a result of the fair energy distribution in the network and improves the sensor network robustness against failures. The outcome of the protocol operation, energy distribution and random node death is an increase in the network lifetime.

In a given round, we can use a simple control token passing approach initiated by the chain leader to initiate the data transmission from the ends of the chain (token cost is very small because token size is very small). In the process of data collection, each node in the chain receives the data of its previous neighbor in the chain. Each node in the chain fuses its sensed data with the received data and generates a data package that is similar in length to package it had just received. It then transmits this new package and the token to the next neighbor in the chain (in case of having two neighbors). If the leader has two neighbors, it will pass the token along the chain after receiving the data and the token from its previous neighbor to the other end of the chain and wait to receive it with collected data of that side of the chain. Finally, the leader fuses the collected data from each of its two neighbors (if available) and its own sensed data all together and sends the resulted data to the sink as Figure. 1. Each round finishes when the leader transmits the data to the sink. Ideal PEGASIS (Chen et al. 2012) is an ideal situation of the PEGASIS protocol. Ideal PEGASIS, DCBRP (Abdulameer et al. 2016) and CCM (Tang et al. 2012) sensor nodes lie in specific places with equal distances. The chain is constructed line by line as Figure. 2. In (Chen et al. 2012)all of the operations are similar to PEGASIS.

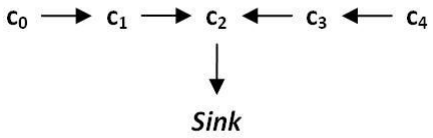

Figure. 1. Collecting data by the leader node and sending it to the sink (Lindsey et al. 2002).

CCM has presented a mixed chain-cluster based routing algorithm CCM for data gathering. It organizes the sensor network as a set of horizontal chains and a vertical cluster and routes data in two phases. In the first phase, sensor nodes in each chain send data to the chain head, using chain based routing. In the second phase, all the chain heads form a cluster and send the data, which are fused from their own chains, to a voted cluster head. DCBRP focuses on the Chain-Head Selection mechanism (CHS).

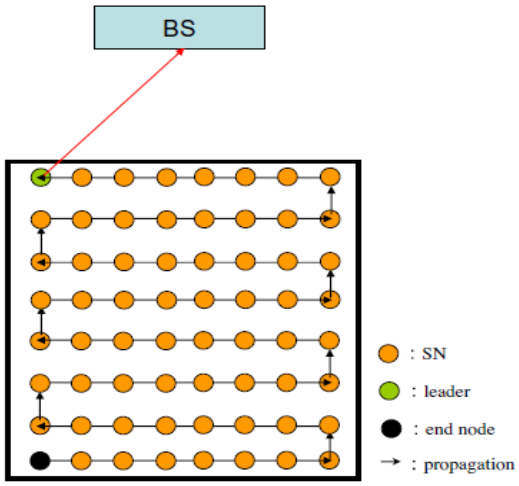

Figure 2. Transmission path in ideal PEGASIS architecture (Min et al. 2008).

The intra-grid PEGASIS method (Chen et al. 2012) is the combination of PEGASIS and a grid-based data gathering method in which the dimensions of the grid is specified and the nodes are uniformly distributed. If the number of nodes is not in proportion to a uniform distribution, then there are two possible cases: (a) The grid-cell that is closest to the sink will have one more node. This is called a near intra-grid chain. (b) The grid-cell that is farthest from the sink will have one more node. This is called a far intra-grid chain.

Then, the chain is constructed in three stages: (a) Using a greedy algorithm, a chain is constructed within each grid-cell. (b) A fully connected chain is established through the network when the chains inside of all grid-cells are joined. (c) Finally, a node is chosen as the leader. Then, like PEGASIS, the collected data are

gathered by leader from the other nodes of the chain and sent to sink. Figure 3 shows these stages.

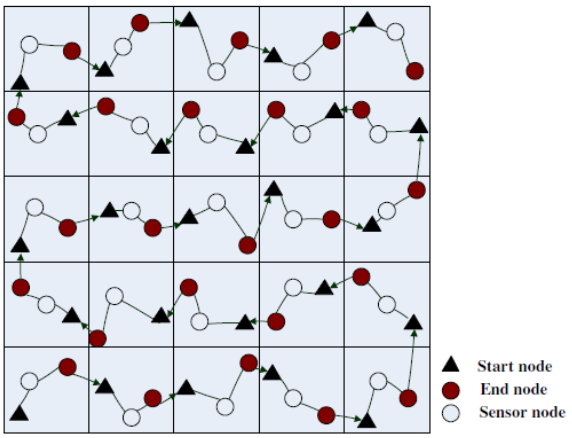

Figure 3. Intra-grid PEGASIS method with a 5x5 grid of 1 m2 grid-cells (Chen et al. 2012).

3. RADIO MODEL FOR ENERGY CALCULATIONS

The radio model used in the proposed methods is similar to that in the PEGASIS protocol (Lindsey et al. 2002; Ding et al. 2003). It is assumed that the radio channel is symmetric so that the energy required to transmit a message from node A to node B is the same as the energy required to transmit a message from node B to node A for a given signal-to-noise ratio $\left(SNR = \left(P_{signal}/P_{noise}\right) = \left(E_b/N_0\right)\right)$ (It is also assumed that all sensors always have some data to send. For the comparative evaluation purposes of this research we assume that there are no packet losses in the network. The radios have power control and can expend the minimum required energy to send the data to the target node. The radios can be turned off to prevent them from receiving unintended transmissions. A radio dissipates E_{elec}E to run the transmitter or receiver circuitry and ε_{amp} for the transmitter amplifier. This amount of energy for transmitter amplifier is the required energy to amplify the signals containing data while keeping the signal energy at the acceptable level of SNR. This level enables a sensor node to send its data to target sensor, despite a level of basis noise. The amount of energy consumed for data transmission depends on transmission distance (d). This dependency is proportional to the square of the transmission distance (d^2). Therefore, the equations that are used to calculate; transmission costs (equation 3) and receiving costs (equation 4) for a *k-bit* message and a distance *d* in this radio model are shown below:

$$E_{elec} = 50nJ/bit \quad (1)$$

$$\varepsilon_{amp} = 100pJ/bit/m^2 \quad (2)$$

$$E_{Tx}(k,d) = E_{elec} * k + \varepsilon_{amp} * k * d^2 \quad (3)$$

$$E_{Rx}(k) = E_{elec} * k \quad (4)$$

Receiving data is also a high cost operation, therefore, the number of reception and transmission operations should be minimized to reduce the energy cost of an application.

4. THE PROPOSED METHODS TO ENHANCE THE NETWORK EFFICIENT LIFETIME

Before mentioning the proposed methods, we express some specific features of the PEGASIS protocol, which constitute the basis of proposed methods. It is worth mentioning that by network efficient lifetime we mean the first node death time and by the network lifetime we mean the last node death time.

4.1. Studying factors in influencing the enhancing of network efficient lifetime in PEGASIS protocol

The number of nodes and their deployments, the amount of their energy, sink location, the size of network area and other environmental parameters are influential on the network efficient lifetime in the PEGASIS protocol. Improvement of the network efficient lifetime for scenarios requiring great or full coverage of the considered environment is of great importance. In general, PEGASIS can create different chains based on node deployment (if all parameters of the network are assumed the same). The length of these chains and distances between the nodes in a chain play an important role in the network efficient lifetime because all transmissions are performed sequentially through these chains. Figure 4 shows the difference between the network efficient lifetime and the network lifetime in four different deployment scenarios (all networks have the features of scenario 1 shown in table 1). Our research shows that the first node death (network efficient lifetime) in these networks occurs between rounds 272 and 909, a range of 639, whereas the death time of the last node (network lifetime) is separated by less than 100 rounds across the scenarios.

In these deployment scenarios, the main reason for the difference (~ 600 rounds) in the network efficient lifetime is the existence of long edges in the first chain. In this respect, the nodes connected to the longer edges lose their energy faster than the nodes connected to the shorter edges because they consume more energy while transmitting data to their neighbor nodes (the energy consumed is proportional to the square of transmission distance). When there are one or more long edges in the chain, one of the connected nodes to the long edges usually loses its energy soonest.

Therefore with regard to the points raised; with the transmutation of long edges to shorter edges, the death time of their connected nodes is postponed. Consequently, chain correction and especially correcting the first chain can increase the network efficient lifetime. This is the basis of proposed methods in this research and is named "chain correction".

Like the PEGASIS protocol, it is assumed in the first and second proposed methods that the nodes know their location and that of the sink. In the third proposed method, in contrast with PEGASIS, nodes have no information about the environment and only become aware of their own distance to the sink at the start of network configuration (like LEACH protocol (Heinzelman et al. 2000; Mahapatra et al. 2015; Manikandan et al. 2015)). This proposed method has favorite results due to its few assumptions compared with other methods.

In Figure 4, the first and second nodes which lose their energy are marked by an oval sign and a rectangle sign, respectively.

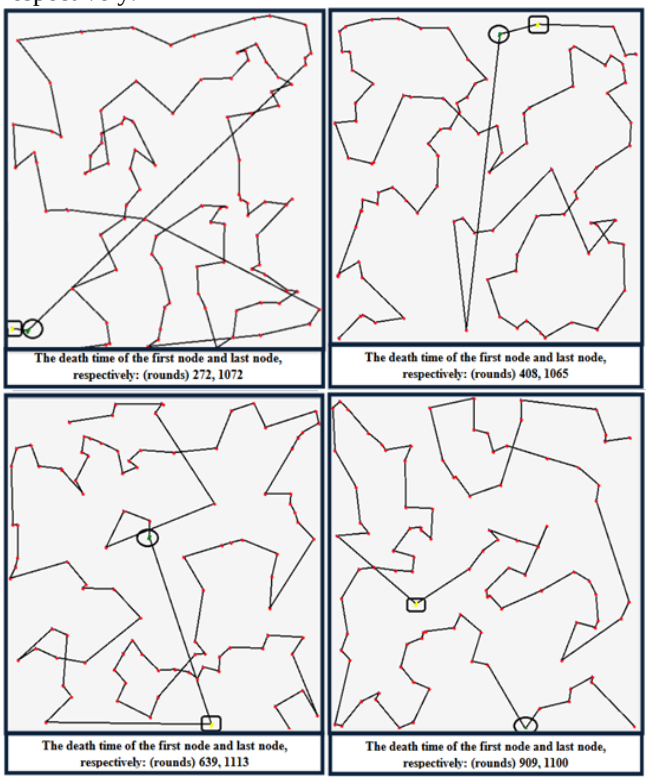

Figure 4. The chains created by PEGASIS in four different deployments of the same network

4.2. Single-stage chain correction algorithm

This section explains the algorithm for correcting the longest edge from the first chain of the network. This algorithm can be repeated to correct subsequent long edges in the chain.

Like PEGASIS, it is assumed that each node is aware of its own location and the location of its neighbors. The chain is created by a greedy algorithm. The farthest node to the sink is the first node of the chain. This initiator node considers its own distance to the next node of the chain as the longest edge. In the continuation of the chain construction, if the distance between the second and third nodes of chain is greater than the length of the first created edge then the new edge is considered as the longest edge. Afterwards, each new edge is compared with the current longest edge and replaces it as the current longest edge if the new edge is longer.

When the longest edge of the chain is found, it must be corrected. Figure 5 shows the correction stages of the longest edge of the chain. In this figure, there are 10 nodes, S_1 to S_{10}. S_1 is the farthest node from the sink. S_LS_R is the longest edge on the chain. Step (a) shows the initial chain and the longest edge of the chain. The pink area in the step (b) is a part of the imaginary circle in the network that its center point is S_R and its radius is equal to S_RS_L. Steps e, f, g and h show the replacement edges for the chain correction, respectively. The new created edges are shown with a red line and the removed edges are shown with dotted blue line.

4.2.1. Chain correction algorithm

We assume that the chain has N nodes. We show each node by S_i where i=1,2,3,…,N. Fig. 6 shows the initial chain created by the greedy algorithm in PEGASIS. S_1 is the farthest node from the sink and is from w where the chain construction begins. S_N is the last node in the chain. Neighboring nodes S_L and S_R with (L<R) in node sequence are connected to the longest edge of the chain. S_LS_R is the edge on which the correcting operation is first performed. If S_LS_R is deleted then there appear two separate chains in which nodes S_1 and S_L are at the beginning and the end of the first chain with S_R and S_N are at the beginning and the end of the second chain. An auxiliary node is used in the proposed methods to correct the main chain. This auxiliary node is S_C in the first chain or S_D in the second chain. S_C and S_D are close to S_R and they are in the spatial amplitude of S_LS_R.

To start the correction operations of the chain, S_R finds the auxiliary node. If the auxiliary node is before node S_L in the first chain, the chain changes similar to fig. 7 and otherwise it changes like fig. 8. The figures 7 and 8 show there will exist no ring in the result chain because no node is used in the chain more than once.

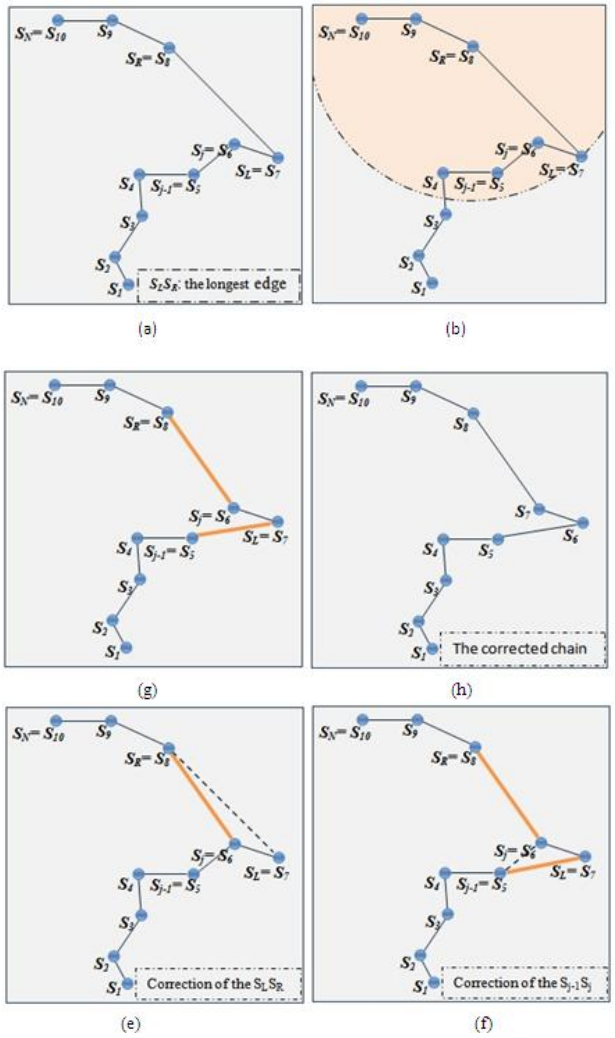

Figure 5. Stages of the chain correction in the proposed methods

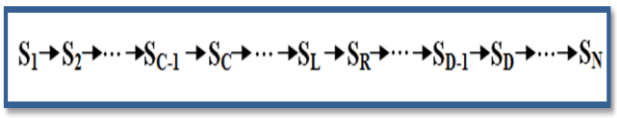

Figure 6. This chain is the primary chain which has the longest edge (SLSR) on it.

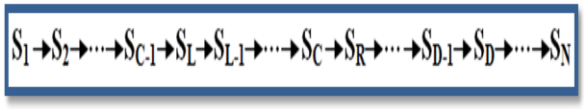

Figure 7. The corrected chain (The chain is corrected by node SC located on the first chain).

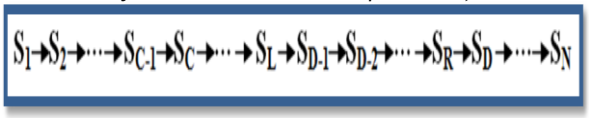

Figure 8. The corrected chain (The chain is corrected by node SD located on the second chain).

Figure 9 shows the pseudo-code of the chain correction as a function. The chain correction starts after the initial chain is constructed and the messages are sent. At this stage the last node of the chain SN, which is aware of the longest edge, sends a message to the node located in the end of the longest edge SR to inform that it is responsible for starting the correction operations. Then node SR sends a message to the nodes within the spatial amplitude (or special domain) of SLSR containing the location of nodes SL and SR. Each node which received this message informs SR of its location and its previous node location within the chain. SR chooses the most suitable auxiliary node Sj for connection among all of the messages it receives. It must be noted, after all located nodes in that amplitude, they receive the message from SR. First, each of them calculates the following inequality and if the inequality is true, it sends a message to SR. This will reduce the number of communications and the sent messages between nodes. Then, each one of them that would satisfy in the inequality:

$$\left(S_jS_R\right)^2+\left(S_{j-1}S_L\right)^2<\left(S_RS_L\right)^2 \qquad (5)$$

When the node SR chooses the suitable auxiliary node from the received messages, it informs the relevant nodes of the required corrections (the connections of nodes SjSR and SLSj). This is performed by sending the message to node Sj. Then node Sj relays this message to Sj-1 which then informs SL. When the above corrections have been completed, then the actions related to changing of the direction of signal flow in the relevant edges in the chain are executed. See fig 5(b), initially the signal flows from S7 to S8 but after correction S7 and S8 swap positions in the chain so the signal direction between them is reversed. Therefore, each node corrects its sequence number in the chain.

4.3 Multi-stage chain correction algorithm

By repeating the single-stage correction method, which corrected the longest edge of the chain, we can sequentially correct the remaining long edges in the chain. In this method when making chain, each node is added to the chain then it investigates 'm' long current edges ('m' is equal to the number of the longest edges in the chain that we want to correct them). This act causes that when the last node is connected to the chain, 'm' long edges in the chain are specified. In the multi-stage algorithm, after the single-stage correction method has been used to correct the initial longest edge, a new sub-set of edges are created that their lengths have not been assessed. Now, the connected nodes to these new edges compare their edges length with the length of the next known longest edge. If one of the new edges is longer, it becomes the next longest edge. After the list of the next m-1 long edges has been corrected, node S_j sends a message to the second connected node to the next long edge to announce it that it continues the chain correction. The above stages continue while the next m-1 long edges

of the chain are corrected. This method increases the number of calculations by the nodes. However, the energy consumption of the calculations is negligible due to the value of 'm' is small

```
void CorrectionChain(int[] Chain, int SL, int SR)
{
   double MinDis = DisBetweenNodes(SR,SL);
   int Sj = SL;
   foreach(Si inside selected area)
   {
       if((DisBetweenNodes(SR,Si)^2 + DisBetweenNodes(SL,Si-1)^2) < MinDis^2)
       {
           MinDis = DisBetweenNodes(SR,Si) + DisBetweenNodes(SL,Si-1);
           Sj = Si;
       }
   }
   /* Correcting chain Links */
   if (Sj < SL)
   {
      foreach (Si<SL, i=1,2,3,...,L)
          CorrectedChain[i] = Chain[i];
       foreach(Si>=Sj, i=L,L-1,...,j)    //Here Sj same SC
          CorrectedChain[i] = Chain[Si];
       foreach(Si<SN, i=R,R+1,...,N)
          CorrectedChain[i] = Chain[i];
   }
   if (Sj > SL)
   {
       foreach (Si<=SL, i=1,2,3,...,L)
          CorrectedChain[i] = Chain[i];
       foreach(Si>=Sj-1, i=R,R-1,...,j) //Here Sj same SD
          CorrectedChain[i] = Chain[Si];
       foreach(Si<SN ,i=j,j+1,...,N)
          CorrectedChain[i] = Chain[i];
   }
}
```

Figure 9. The pseudo-code for the chain correction

With a suitable definition of a threshold, we can prevent the additional corrections in the network. Some networks have the suitable length of edges. One way to determine the value of this threshold is to use a coefficient of the length of edges in the ideal PEGASIS protocol.

4.4. Chain correction algorithm with nodes free from location knowledge of the network

This proposed method is suitable for the networks in which nodes are distributed randomly and do not have information about their own location and other nodes in the network.

This method requires a presetting phase. In the presetting phase, the sink sends a message throughout the network. All nodes estimate their distance to the sink based on the intensity of the received message signal. Then, the formation of the chain begins from a random node in the network (similar to PEGASIS which starts the chain buildup from the farthest node to the sink). The initiator node of the chain sends a message to its neighbors (the nodes have the ability to control power), and waits to receive their response. When all the responses have been received, the initiator node calculates its distance to the neighbors, again based on the intensity of the received signal. Then the initiator node uses the greedy algorithm and makes contact with its nearest neighbor. This is sequentially repeated for the other nodes so that the chain is formed.

This method is somewhat similar to the methods presented in sections 4.2 and 4.3. In this method each node can consider 'm' long edges of the chain, too.
In the previous methods, the location of nodes was used to correct the chain which is impossible in this method. Instead, this method implements corrections by using nodes distances to each other. After completion of the chain creation and determination of the longest edge, the last node in the chain sends a message to S_R which starts the correction operations. S_R sends a message to the spatial amplitude of S_LS_R (S_R sends a message to the range, containing the size of edge S_LS_R). This spatial amplitude is equal to the length of the longest edge. First, S_L receives this message then sends the received message to the same amplitude. Hereby, when all the available nodes in the considered amplitude get both messages of nodes S_L and S_R, they estimate their distance to nodes S_L and S_R. Then, they send their distance to S_L and S_R and their previous node address for node S_R. For performing correction operations, S_R can choose the most appropriate of the two auxiliary nodes (nodes S_j and S_{j-1}) after receiving all of the messages. S_R chooses one node among the auxiliary nodes for which the inequality 5 is true about them. The selected node has the minimum value of this inequality.

When S_R chooses the most suitable auxiliary node (S_j) from amongst the received messages, it sends a message which instructs S_j to perform the required corrections (the connection of nodes S_R to S_j and S_{j-1} to S_L). Then, S_j relays this message to S_{j-1} and also, S_{j-1} relays it to S_L. Finally, after these corrections, the actions related to the alteration of signal direction in the relevant edges in the chain are applied so that each node corrects its sequence number in the chain.

5. RADIO MODEL FOR ENERGY CALCULATIONS

As explained in section 4, the nodes exchange some messages in the chain correction operations. In this section, we introduce a suitable module for WSN and explain the format of the messages.

In different applications of WSN, we can use the communication protocol based on standard IEEE 802.15.4 (ZigBee Document 2008; MRF24J40MA Data Sheet 2008) This module adds unique features to the network by supporting ZigBee protocol. Fig. 10 shows the general format of Medium Access Control (MAC) in this module. Due to the use of chain-based methods in the network, just a single-level addressing method is required. That is to say each node has a unique address. Part of the address field can include the address of the

source device, the target device and their Personal Area Network (PAN) address. The length of the PAN address field in the proposed methods is zero. Also the address of each node is specified by two bytes. In this module the network can have at most 65536 nodes.

The payload plays an important role in the node communications. The types of messages that can be exchanged between nodes are limited. An identification code is allocated to each type of messages and also, each message can have value. The type of the messages and its values are located in part of the frame payload. Each node can interpret the message by processing this field and it returns a suitable response if needed. Table 1 shows some messages related to chain correction operations.

Octets: 2	1	0/2	0/2/8	0/2	0/2/8	0/5/6/10/14	variable	2
Frame Control	Sequence Number	Destination PAN Identifier	Destination Address	Source PAN Identifier	Source Address	Auxiliary Security Header	Frame Payload	FCS
		Addressing fields						
MHR							MAC Payload	MFR

Figure 10. The general format of a MAC frame in the IEEE 802.15.4. standard

The proposed methods follow a message sequence as shown in table 1. When SN wants to introduce the longest edge of the chain, it constructs a MAC frame that contains its target address of SR. It puts message code 1 which is of length 1 byte within payload, as in table 1, and sends this message to SR. After finding Sj, node SR sends a message to Sj. The payload of this message consists of a message code of length 1 byte (that it is number 2) and the address of SL which is of length 2 bytes. The remaining of the chain correction process continues through the messages shown in Code 3 & 4 in table 1. As seen, the length of sent frames for the chain correction operations is much smaller than the data messages collected by the network for transmitting to the sink because the payload field is small. This has demonstrates that the energy overhead to send control messages in the chain correction operations are negligible in comparison to network operation.

6. THE RESULTS OF SIMULATIONS

In this study, we developed a graphical, object-oriented software package in C# based on .NET4 to simulate and compare the proposed methods with PEGASIS and intra-grid PEGASIS. Table 2 presents the simulation parameters. The energy required to fuse the data in each signal is equal to EDA=5nJ/bit/signal and each data package consists of 2000 bits. In all simulations the energy overhead of the proposed method is considered in the calculations (the energy overhead is less than consumed energy in a round when all of nodes are alive.).

Table 1. The Part of payload content of the transmitted messages among nodes in the first proposed method

Code	Explanation of Message	Event	Content of Message (byte)
1	The last node of the chain sends a message to the node that located in the end of the longest edge. ($S_{SI} \rightarrow S_R$)	The first chain is constructed.	Message code 1 (1)
2	S_L finds a suitable auxiliary node and informs to it. ($S_R \rightarrow S_j$)	A suitable auxiliary node was found in the amplitude of $S_L S_R$.	Message code 2 (1) Address of S_L (2)
3	S_j sends a message to S_{j-1}. ($S_j \rightarrow S_{j-1}$)	S_j received a message of S_R. (message code: 2)	Message code 1 (1) Address of S_L (2)
4	S_{j-1} sends a message to S_L. ($S_{j-1} \rightarrow S_L$)	S_{j-1} received a message of Sj. (message code: 3)	Message code 3 (1)

Table 2. The used properties in scenarios one and two in this study

Scenario name	Network size (m^2)	Sink Location (m)	Number of nodes	Initial energy of nodes (J)
First scenario	50 × 50	(25,150)	100	0.25
Second scenario	100 × 100	(50,300)	100	0.25
Large scale	100 × 100	(50,300)	2000	2

In each of these scenarios, one thousand random deployments of the network nodes were simulated for each method compared. Fig. 11 shows the results of the simulations. The time of first node death is recorded within 10 ranges, each of which is 100 rounds in duration and covering from 1 to 1000 rounds in total. From these simulations we can observe the frequency of the first node death for each method. As seen in Fig 11, PEGASIS, the proposed method with correction of the longest edge of the chain (m=l) and the proposed method with chain correction of five iterations (m=5) have been compared. In the proposed method (m=5) the network efficient lifetime exceeds 900 rounds in 80% of the simulations while for PEGASIS only 2% last as long. For network efficient lifetime exceeds in excess of 800 rounds; (m=5) has 99.6% simulations while PEGASIS has 12.5%.

Figure. 12 shows the results for the same number of deployments for scenario 2. In the chain correction method with five iterations (m=5), the network efficient lifetime exceeds 400 rounds for 99.4% of the simulations, while only 13.7% of PEGASIS simulations achieve this. As can be seen, the chain correction has helped to improve the performance with respect to network efficient lifetime.

The quantity of energy consumed in the whole network until the time of death of the first node in first and second scenarios is shown in figures 13 and 14. This quantity scenario 1 for the proposed methods with m=1, m=5, m=15 and m=5 without knowledge is 74.8, 87.5, 89.1 and 87.3 percent of whole energy of the network, respectively. The quantity of energy consumed in PEGASIS until the death time of the first node is 58%.

In scenario 2 the quantity of energy consumed for m=1, m=5 and m=15 and m=15 without knowledge is respectively 53.9, 73.1, 75.8 and 72.8 percent of whole energy of the network while it is 37.2% in PEGASIS. It is worth mentioning that the death time of the first chain (the first chain exhaustion) in PEGASIS occurs sooner than that in the proposed methods because in PEGASIS there are longer edges in the chain than in the proposed methods.

The methods presented in (Meghanathan et al. 2010; Chen et al. 2012; Abdulameer et al. 2016; Tang et al. 2012) are grid–based schemes which have good performance when the nodes are distributed uniformly in the network and there is at least one node in all putative grids. Figure 15 compares the lifetime of the proposed methods, PEGASIS and intra-grid PEGASIS. In this simulation, it is assumed that nodes are distributed uniformly in the network so that there is at least one node in each of the 100 grids (the size of each grid is 10m×10m) of the network. This figure has been obtained from the average of the lifetime of each method in six network deployments of scenario 2.

Figure 16 is obtained by the same procedures as previous simulation for the proposed methods, PEGASIS and intra-grid PEGASIS. In contrast to the simulation reported in Figure 15, this simulation's nodes are not distributed uniformly but are distributed randomly. In this case the grid-based method creates the chain in the network line by line therefore it is possible that it will produce long edges in the chain. In this simulation, the network efficient lifetime by intra-grid PEGASIS method is less than 500 rounds (in case of the uniform distribution, as in figure 15, this amount is close to 600 rounds) while the network efficient lifetime for the proposed methods with m=5 and m=10 average more than 500 rounds in both scenarios.

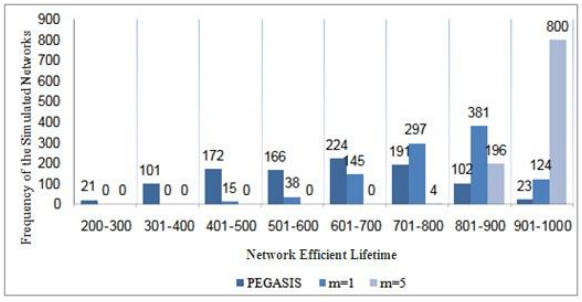

Figure 11. Frequency of the first node death in PEGASIS and proposed methods with m=1, m=5 in scenario

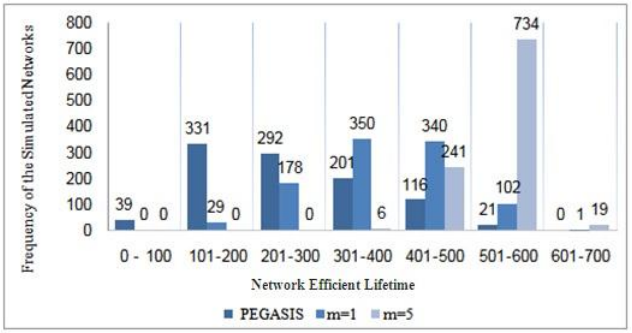

Figure 12. Frequency of the first node death in PEGASIS and proposed methods with m=1, m=5 in scenario2.

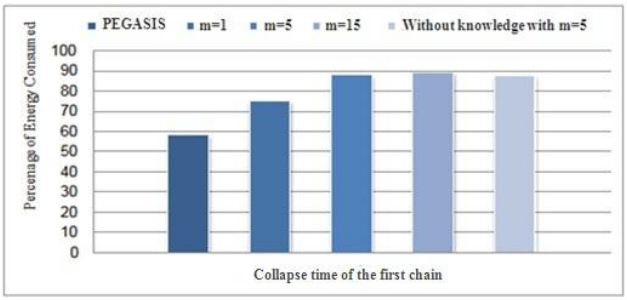

Figure 13. Quantity of energy consumed until collapse time of the first chain in scenario 1.

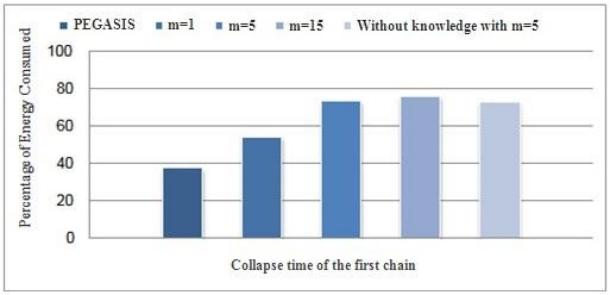

Figure 14. Quantity of energy consumed until collapse time of the first chain in scenario 2.

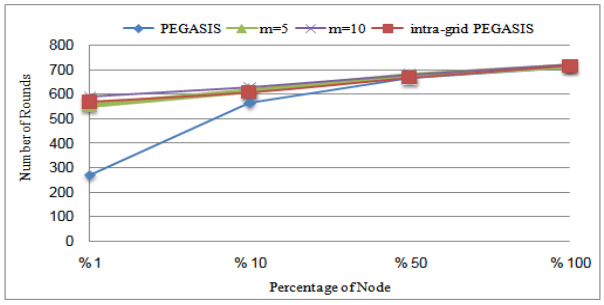

Figure. 15. Comparison of the lifetimes of proposed methods, PEGASIS and intra-grid PEGASIS when the nodes are distributed uniformly.

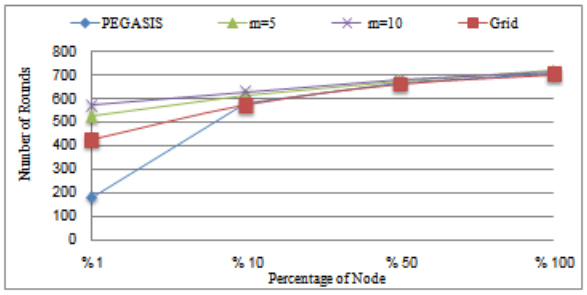

Figure. 16. Comparison of the lifetimes of proposed methods, PEGASIS and intra-grid PEGASIS when the nodes are distributed randomly.

Table 3. Comparison of the lifetimes of proposed methods m=10, PEGASIS and DCBRP and CCM with . Number of rounds when 1%, 10%, 50%, and 100% nodes die. The initial energy value for nodes is 2J.

Protocol	1%	10%	50%	100%
PEGASIS	1980	9260	9360	9457
DCBRP	7679	8971	9068	9292
proposed method m=10	7919	9262	9382	9513

Table 3 shows Comparison of the lifetimes of proposed method, PEGASIS and grid approaches (include DCBRP (Abdulameer et al. 2016) and CCM (Tang et al. 2012) have same performance) when 2000 sensors are distributed uniformly (large scale scenario).

7. CONCLUSION

In this research, we present three methods for the chain correction in the PEGASIS protocol. One of the main points of weakness of the PEGASIS protocol is the probability of early death of some nodes in the network. The main reason for this is the existence of the long edges in the constructed chain. The probability of formation of these long edges exists because a greedy algorithm is used in the chain construction.

The proposed methods in this research prevent the long transmission between sensors by using distributed algorithms and correcting the long edges of the chain. Also, we presented a chain correction method in the networks in which nodes perform their operations without awareness of their own locations or that of other network. The results of simulating thousands of WSN deployments showed that in the proposed methods there are significant improvements in the network efficient lifetime, the energy consumption in the whole network before the death of the first node and the variance of sensor death. The complexity of our algorithms is O(m) its efficient chain-based algorithm for large-scale networks. Finally, these improvements result in an increased security in the monitored field due to extended network integrity.

8. REFERENCES

A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, J. Anderson, (2002) Wireless sensor networks for habitat monitoring, In Proceedings of ACM International Workshop on Wireless Sensor Networks and Applications (WSNA), 88–97.

E. Cayirci, T. Coplu, (2007) SENDROM: sensor networks for disaster relief operations management, Wireless Networks 13 (3) 409–423.

G. Werner-Allen, K. Lorincz, M. Welsh, O. Marcillo, J. Johnson, M. Ruiz, J. Lees, (2006) Deploying a wireless sensor network on an active volcano, IEEE Internet Computing 10 (2) 18–25.

Y. Zhu, W. Wu, J. Pan, Y. Tang, (2010) An energy-efficient data gathering algorithm to prolong lifetime of wireless sensor networks, Computer Communications 33, 639–647.

W. Guo, W. Zhang, G. Lu, (2010) PEGASIS Protocol in Wireless Sensor Network Based on an Improved Ant Colony Algorithm, In Proceedings of the Second International Workshop on Education Technology and Computer Science (ETCS) 3 64-67.

A. Acharya, A. Seetharam, A. Bhattacharyya, M. K. Naskar, (2009) Balancing Energy Dissipation in Data Gathering Wireless Sensor Networks Using Ant Colony Optimization, In Proceedings of the 10th International Conference on Distributed Computing and Networking (ICDCN ‘09) 5408 437-443.

B.

S. Lindsey, C. Raghavendra, K. M. Sivalingam, (2002) Data Gathering Algorithms in Sensor Networks Using Energy Metrics, IEEE Transactions on Parallel and Distributed Systems (TPDS) 13 (9) 924-935.

S. Lindsey, C. S. Raghavendra, (2002) PEGASIS: Power-Efficient Gathering in Sensor Information Systems, In Proceedings of the IEEE Aerospace Conference 3 1125-1130.

M. Ding, X. Cheng, G. Xue, (2003) Aggregation Tree Construction in Sensor Networks, In Proceedings of the IEEE 58th Vehicular Technology Conference (VTC) 4, 2168-2172.

R. Rohankar, C. P. Katti, S. Kumar, (2015), Comparison of Energy Efficient Data Collection Techniques in Wireless Sensor Network, In Proceedings of the 3rd International Conference on Recent Trends in Computing (ICRTC) 57, 146-151.

H. O. Tan, I. Korpeoglu, (2003) Power Efficient Data Gathering and Aggregation in Wireless Sensor Networks, ACM SIGMOD Record Newsletter 32 (4) 66-71.

N. Meghanathan, (2010) Grid Block Energy Based Data Gathering Algorithm for Wireless Sensor Networks, International *Journal of Communication Networks and Information Security* (IJCNIS) 2 (3) 151-161.

H. Rana, S. Vhatkar, M. Atique, (2014) Comparative Study of PEGASIS Protocols in Wireless Sensor Network, In Proceedings of the IOSR *Journal of Computer Engineering* (IOSR-JCE) 16 25-30.

J. Rana, S. Vhatkar, M. Atique, (2015) Comparative Study of PEGASIS and PDCH Protocols in Wireless Sensor Network, IJCA Proceedings on International Conference and Workshop on Emerging Trends in Technology (ICWET) 2 13-18

Z. Gengsheng, L. Xiaohua, H. Xingming, Z. Weidong, (2009) The Research of Clustering Protocol Based on Chain Routing in WSNs, In Proceedings of the Asia-Pacific Conference on Computational Intelligence and Applications (PACIIA) 1 292-295.

S. Pal, D. Bhattacharyya, T. H. Kim, (2010) Chain Based Hierarchical Routing Protocol for Wireless Sensor Networks, In Proceedings of Conference on the Security-Enriched Urban Computing and Smart Grid (Springer) 782010 482-492.

J. Shin, C. Suh, (2008) Energy-Efficient Chain Topology in Ubiquitous Sensor Network, In Proceedings of the 10th *International Conference on Advanced Communication Technology* (ICACT) 3 1688-1693.

H. Min, S. YiS, J. Heo, Y. Cho, J. Hong, (2008) Energy-Efficient Data Aggregation Protocol for Location-Aware Wireless Sensor Networks, In Proceedings of International Symposium on Parallel and Distributed Processing with Applications (ISPA '08) 751-756.

A. Avizienis, J. C. Laprie, B. Randell, (2001).Fundamental Concepts of Dependability, Research Report No 1145, LAAS-CNRS

Y. L. Chen, J. S. Lin, (2012) Energy Efficiency Analysis of a Chain-Based Scheme via Intra-Grid for Wireless Sensor Networks, *Computer Communications Journal* (Elsevier) 35 (4) 507-516.

W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, (2000) Energy-Efficient Communication Protocol for Wireless Microsensor Networks, In Proceedings of the 33rd Annual Hawaii International Conference on System Sciences (HICSS '00) 2 10.

W. R. Heinzelman, (2000). Application-Specific Protocol Architectures for Wireless Networks, PHD thesis, Massachusetts Inst. of Technology.

R. P. Mahapatra, R. K. Yadav, (2015) Descendant of LEACH Based Routing Protocols in Wireless Sensor Networks, *In Proceedings of Computer Science* 57 1005-1014.

K. Manikandan, P. Kanmani, M. M. Sulthana, (2015) Energy Efficient Algorithms for Wireless Sensor Network, In Proceedings of the International *Journal of Advanced Research in Computer and Communication Engineering* 4 342-346.

IEEE Standard, Part 802.15.4: (2006) Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs).

ZigBee Alliance, (2008). ZigBee Document 053474r17, ZigBee Alliance Board of Directors.

Microchip Technology Inc, MRF24J40MA Data Sheet 2.4 GHz IEEE Std. 802.15.4™ RF Transceiver Module.

H. Abdulameer Marhoon, M. Mahmuddin and S. Awang Nor, (2016) DCBRP: a deterministic chain based routing protocol for wireless sensor networks, *SpringerPlus* 5: 2035

F. Tang, I. You, S. Guo, M. Guo, Y. Ma, (2012) A chain-cluster based routing algorithm for wireless sensor networks, **Journal of Intelligent Manufacturing** 23 (4) 1305-1313.