Enhanced PEGASIS using Dynamic Programming for Data Gathering in Wireless Sensor Network

Mohammad Robihul Mufid, M. Udin Harun Al Rasyid, Iwan Syarif

Politeknik Elektronika Negeri Surabaya
Kampus PENS, Jl. Raya ITS, Sukolilo Surabaya
E-mail: mufid@pasca.student.pens.ac.id, {udinharun, iwanarif}@pens.ac.id

Abstract

A number of routing protocol algorithms such as Power-Efficient Gathering in Sensor Information Systems (PEGASIS) and Low-Energy Adaptive Clustering Hierarchy (LEACH) have been proposed to conquer energy efficiency issues in Wireless Sensor Network (WSN) technology. PEGASIS is a perfection of the LEACH protocol, where all nodes are in PEGASIS protocol active during data transfer rounds thus limiting the lifetime of the WSN. This study aims to propose improvements from the previous PEGASIS version by giving the name Enhanced PEGASIS using Dynamic Programming (EPDP). EPDP uses the Dominating Set (DS) idea in selecting a node in coverage same area to be activated and using dynamic programming based optimization in forming chains from each node. There are 2 topology nodes that this research use, namely random and static. Then for Base Station (BS) placement, it will be placed in outside, in the corner, and in the middle of the network. Whereas to determine the performance between EPDP, PEGASIS and LEACH, an analysis of die nodes, alive nodes, and remaining of energy were analyzed. From the experiment result, it was found that the EPDP protocol had better performance compared to the PEGASIS and LEACH protocols in regard to the number of die nodes, alive nodes, remaining of energy, latency average, and number of data received at BS. Whereas the best BS placement in the network is in the middle and uses static node distribution topologies to save more energy.

Keywords: Wireless Sensor Network, EPDP, LEACH, PEGASIS, dynamic programming.

1. INTRODUCTION

The development of Micro Electro Mechanical Systems (MEMS) technology led to the improvement of miniature sensor nodes capable of sensing parameters such as temperature, humidity, pressure by maintaining wireless connectivity between each other to form the Wireless Sensor Network (WSN). WSN technology plays an important role in various types of domains, for example in environmental monitoring systems [1,2]. For implementation in environmental monitoring systems, WSN components are

divided into 2 namely Base Station (BS) and sensor nodes. Then on account of the separation between the sensor nodes and the BS for the activity and the power supply at the sensor nodes is just provided by the battery, the energy reserves are limited. So that the problem obtained in the WSN is about energy consumption [3,4]. One of the most popular and widely used routing protocol algorithms is Low-Energy Adaptive Clustering Hierarchy (LEACH). In LEACH, the system is separated into clusters, and then every cluster will select a cluster head node. After that, each node in the cluster will send the detecting data obtained to the cluster head. Then cluster head performs a fusion of data packets got into one bundle and afterward sent to BS. LEACH has several weaknesses including poor bandwidth utilization, because it uses Code Division Multiple Access (CDMA) to settle interference between contiguous clusters [5]. The development of the famous LEACH algorithm and the pioneer of chain-based hierarchical protocols is Power Efficient Gathering in Sensor Information Systems (PEGASIS). In PEGASIS, all nodes are arranged into linear chains for data transmission and data aggregation. Every node in the chain coordinates its own information to the information got from its antecedent and sends amassed information to its successor along the chain with the exception of the terminal node (BS). The chain leader (CL) is chosen at the location of the chain randomly to transmit the final aggregate data to the BS. The benefit of the nearest connected information isn't taken by PEGASIS and the majority of the information is gathered too much to the BS. As a consequence, all nodes are live during data transfer rounds that limit WSN longevity [6,7].

Based on the above problems, this research propose an upgrade to the previous PEGASIS version by giving the name Enhanced PEGASIS using Dynamic Programming (EPDP). EPDP uses the idea of Dominating Set (DS) in selecting a node in the same area coverage to be activated in one round. The use of DS aims to reduce energy consumption between nodes because it only activates several nodes in the same area coverage. After that this research use dynamic programming based optimization in forming a chain of each node. Dynamic programming has a more optimal solution compared to greedy algorithm, brute force, and ant colony optimization, because it always finds global optimum, faster, and saving memory because it only holds a partial path once. As for the selection of the Chain Leader (CL), it is chosen based on the amount of energy remaining and its proximity to the BS. The use of EPDP aims to reduce the energy consumption of each node and extend the life of the node. EPDP can also handle WSN deployments randomly where there is a possibility that some sensor nodes have the same coverage.

2. RELATED WORKS

In overcoming the problem of energy consumption in the WSN, a number of algorithms have been proposed including Heinzelman et al. [8] who became the pioneer of introducing LEACH for the first time. LEACH is a hierarchical protocol that has the main characteristic of dividing nodes into clusters randomly. Then each cluster will pick a node to turn into a cluster head. After that, every node in the cluster will send the detecting data

obtained to the cluster head. Cluster head then fuses data packets into one bundle and then sends them to BS. The results of his research indicate that LEACH can increase the life of the system.

Baranidharan et al. [9] who proposed a new clustering algorithm called Genetic Algorithm Based Energy Efficient Clustering Hierarchy (GAECH). GAECH is proposed to increase the time round from First Node Die (FND), Half Node Die (HND) and Last Node Die (LND) by using a new fitness function. The results of the experiments conducted showed that GAECH's performance was better than the Genetic Clustering Algorithm (GCA), Energy-Aware Evolutionary Routing Protocol (EAERP), and LEACH algorithms in both time rounds on FND, HND, and LND.

Lindsey et al. [10] who proposed an improvement of the LEACH algorithm which calls PEGASIS for the first time. Where all nodes are orchestrated into linear chains for data transmission and aggregation. Every node just speaks with the closest node alternately to transmit data to the BS. This is done so as to decrease the measure of energy spent per round. Simulation results demonstrate that PEGASIS performs superior to anything LEACH for various system sizes and topologies.

Mishra et al. [11] who proposed an increase in the process of forming the PEGASIS chain to promote increase its age, and called it PEGASIS with Improved Network Lifetime (PEGASIS-INL). Unlike the native PEGASIS protocol, in PEGASIS-INL if a node is within the strong communication scope of the BS, then the node will be selected as the leader. The experimental results show that the PEGASIS-INL algorithm shows better performance than PEGASIS in terms of delay and energy consumption.

Wang et al. [12] proposed an energy efficiency strategy to increase lifetime at the WSN based on PEGASIS and MECA. This algorithm intends to limit and adjust energy utilization for all cluster heads and sensor nodes. The routing algorithm used shows the results that the performance is better than LEACH in the field of energy consumption and lifetime.

Ghosh et al. [13] presented an enhanced version of PEGASIS (E-PEGASIS) which can overcome the shortcomings of PEGASIS and energy efficiency. The simulation results using Matlab 12 show that E-PEGASIS extended the life span of WSN compared to PEGASIS, PEGASIS Binary and LBEERA. E-PEGASIS uses the concept of dominance set (DS) in selecting a subset of nodes that are placed, to be activated in rounds that are close to the same scope as in the original PEGASIS.

To explain the comparison and development of this study with existing related work, then in table 1, it explains the comparison of routing protocol algorithms that have been done before. Starting from the LEACH, PEGASIS, and modification algorithms with several criteria such as protocol, measured parameters, node topology, platform, year, number of nodes, and number of rounds.

Tabel 1. Comparison of routing protocols

Criteria / Researcher	Protocol name	Perameter Measured	Topology Node	Platform	Year	Number of Nodes	Number of Rounds
Heinzelman et al. [8]	LEACH	Total amount of data, Average energy dissipated	Static	NS2	2002	100	1000
Baranidharan et al. [9]	GAECH	Dead nodes, FND, HND, LND, Energy Consumption	Random / Static	Matlab	2015	100	2500
S.Lindsey et al. [10]	PEGASIS	Dead nodes	Random	NS2	2002	100	1200
Mishra et al. [11]	PEGASIS-INL	Energy consumption, Remaining Energy, alive nodes, dead nodes	Random	Castalia-3.2	2015	500	2500
J. Wang et al. [12]	PEGASIS-MECA	Energy consumption, Network lifetime	Static	Matlab	2015	100	5000
S. Ghosh et al. [13]	E-PEGASIS	Alive nodes, FND, HND, LND	Random / Static	Matlab 2012	2016	100	1200
this research proposal	EPDP	Remaining Energy, alive nodes, dead nodes, number of data received at BS, latency average	Random / Static	Matlab	2019	100	2700

3. ORIGINALITY

WSN technology plays an important role in various types of domains, for example in environmental monitoring systems. But because in its implementation there is separation between the sensor nodes and the BS for the activity and the power supply at the sensor nodes is just provided by the battery, the energy reserves are limited. So the problem that is obtained in the WSN is regarding energy consumption. A number of routing protocol algorithms such as LEACH and PEGASIS are proposed to overcome the problem of energy consumption. But LEACH and PEGASIS also have weaknesses. Therefore this study proposes improvements to the PEGASIS protocol that has better achievement than LEACH. At PEGASIS the advantage of the closest correlated data is not taken and all of the data is collected excessively into the BS. As a result, all nodes are active during data transfer rounds that limit WSN longevity. The contribution of this research is to

propose improvements from the previous PEGASIS version by giving the name Enhanced PEGASIS using Dynamic Programming (EPDP). EPDP uses the Dominating Set (DS) idea in selecting a node in coverage same area to be activated in a round. After that this research uses dynamic programming based optimization in forming chains from each node. For the selection of the Chain Leader (CL), it is chosen based on the amount of remaining energy and its proximity to the BS. The use of EPDP aims to reduce the energy used of each node and extend the lifetime of the node, and reduce the latency needed. The fact that most WSN deployments are random and dense with areas covered by multiple sensor nodes can also be exploited by EPDP.

4. SYSTEM DESIGN

In this study, it will be divided into several stages, including the following: (1) Defining the required parameters, (2) Performing a simulation of the PEGASIS, EPDP, and LEACH algorithms in matlab, (3) Creating a sensor node and BS placement scenario (4) Testing the algorithm and displaying the experimental graph, (5) Analyzing the results of the experiment. Figure 1 shows the system design of this study, where each step in the design will be explained in more detail in section 4.1-4.5.
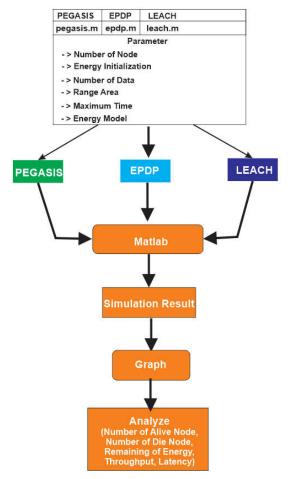

Figure 1. Research system design

4.1 Defining the required parameters

The first step in this research is to define the parameters needed for both the PEGASIS, EPDP, and LEACH algorithms. The formation of this parameter aims to make all the algorithms to be tested using the same parameters. Among the parameters needed are the number of nodes to be used (Number of nodes), the number of energy the nodes need to operate in a single-use network (Energy initialization), The number of data sent in each process or round by each node (Number of data), area based on the cartesiyan axis (Range area), maximum time from simulation in the form of round (Maximum time), and energy model. One round is the process starting from the formation of an algorithm simulation, until each node can send its data to BS in one shipment. The energy model is a parameter used to perform modeling calculations of electronic energy for node activity (E_{elec}) used to send (E_{Tx}) or energy to take in (E_{Rx}) message bits (m) depending on the distance between the transmitter and the receiver in equation (1) or (2), whether less than or greater than the threshold distance d_0, each free space model (fs) or multi - path model (mp).

$$E_{Tx}(m,d) = E_{elec} * m + \varepsilon_{fs} * m * d^2 \qquad \text{if } d < d_0$$

$$= E_{elec} * m + \varepsilon_{mp} * m * d^4 \qquad \text{if } d \geq d_0 \qquad (1)$$

$$E_{Rx}(m) = E_{elec} * m \qquad\qquad\qquad (2)$$

Table 2 shows the initialization of the parameters that this study going to use.

Table 2. Parameter for this research simulation

Parameter for this research simulation	value
Location of the base station	With scenario
Node positioning	Random / Static
Initialization of energy (joule)	0.25 J
Node placement area (meter)	100 m X 100 m
Number of node	100
Package size (bits)	4000 bits
Obstacles/ Barriers	No
Maximum time available (round)	1200
ε_{mp} (pico Joules/bit/m2)	0.0013
ε_{fs} (pico Joules/bit/m2)	10
E_{elec} (nano Joules/bit)	50

4.2 Performing a simulation of the PEGASIS, EPDP, and LEACH algorithms in matlab

After determining the required parameters, an algorithm will be formed starting from LEACH, PEGASIS, and EPDP. Following is the algorithm concept.

4.2.1 LEACH

In this section, will define the LEACH protocol concept, which will be implemented in matlab, and as a confrontation protocol of the results acquired from the PEGASIS and EPDP protocols. In LEACH, the network is split into clusters, and then every cluster will select a cluster head node. Figure 2 shows cluster formed using LEACH.

Figure 2. Cluster formed using LEACH

The stages of forming the LEACH algorithm consist of 2 main parts, namely setup-phase and steady-state. Figure 3 shows the LEACH algorithm consisting of setup-phase and steady-state. The setup phase is the phase in which cluster formation occurs. Starting from determining the cluster head (CH) which is determined based on the information of the residual energy level of every node, until the joining of the non-CH node to each cluster head is based on the advertisement message (ADV) which contains information that the node is a cluster head and continued making scheduling using TDMA and sending the schedule to each cluster member. The next phase is steady-state, where schedule creation and the data transmission process occurs, starting from sending data in the cluster to sending it to the BS by the cluster head. For this purpose, every cluster head broadcasts ADV advertising messages using CSMA which contains the header and node ID that differentiates this message as the message of an announcement. Each non-CH node defines its cluster based on minimum signal strength, which is obtained from the cluster head. Then each node will decide which cluster to enter, and notify the cluster head node by sending a join request message (Join_REQ) using CSMA that it will become a cluster member. Then based on all messages received in the cluster, CH will randomly select the CSMA code, create a TDMA schedule, and broadcast TDMA tables to cluster members. After that the data transmission phase starts the cluster member node to CH, then CH collects this data and sends it to BS [5].
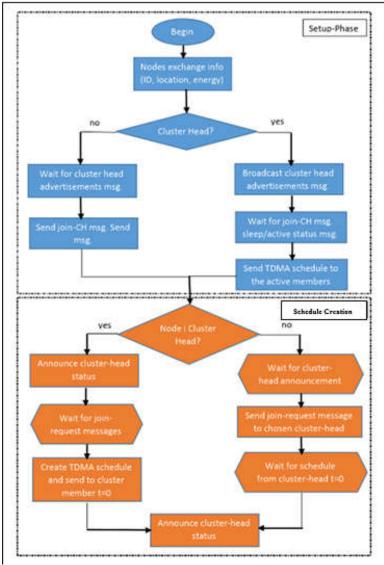
Figure 3. Flowchart of LEACH algorithm in setup-phase and schedule creation in
steady-state

4.2.2 PEGASIS

In this section, will explain the concept of the PEGASIS protocol which will be used as a comparison to the analysis of the performance of the EPDP algorithm proposed in this study. The improvement of the well known LEACH algorithm and the pioneer of chain-based hierarchical protocols is Power Efficient Gathering in Sensor Information Systems (PEGASIS). In PEGASIS, all nodes are arranged into linear chains for data transmission and data aggregation as shown in algorithm 1 about the PEGASIS algorithm. The process of forming a chain in PEGASIS starts from the most distant node from the base station using the Greedy algorithm. Then the node that has joined will select the next node based on the signal strength of the neighboring node to enter into the chain. Nodes cannot be reviewed if the node is already connected to a chain, and the chain cannot be branched. This process goes on until the last node and will produce a single chain consisting of all network

nodes. Then each node in the chain coordinates its own information to the information it receives from its predecessor and sends it to its successor along the chain with the exception of the base station (BS). The chain leader (CL) is chosen at the location of the chain randomly to transmit the final aggregate data to the BS [6]. Figure 3 shows a conceptual description of the PEGASIS protocol.

Algorithm 1. PEGASIS Algorithm

Input : node location (xi,yi), base station (x,y), total node, maximum round, energy, packet size to be sent.
Output : graph of number of die node graph, graph of number of alive node, graph of remaining of energy.

Step 1 : Creation of the random / static sensor network
Step 2 : Calculate the distance of all nodes from base station
Step 3 : Look for the node closest to the base station to be chosen as the leader node
Step 4 : Calculate the matrix to find distance of all nodes from each node
Step 5 : Make a chain of all nodes, starting from the leader node
Step 6 : Run the code to send the data to each node continuously
Step 7 : Starts the data transfer from the end node to the BS
Step 8 : Checking of any dead node after a single round
Step 9 : Reconstruction of Chain
Step 10 : Forming of graph of number of die node graph, graph of number of alive node, graph of remaining of energy

---

Figure 4 shows the chain formed in the PEGASIS algorithm. To provide a more detailed explanation of the description of chain formation in the PEGASIS algorithm.
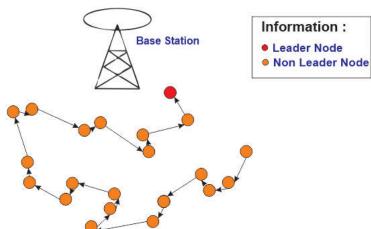

Figure 4. Chain formed using PEGASIS

4.2.3 EPDP

The concept is almost the same as PEGASIS, but in EPDP to avoid long chains and close to optimal chain formations this study uses the concept of Dynamic Programming. The concept of EPDP is as follows.

a) Leader Node Selection and Forming Cluster

In real-world implementation, the Base Station (BS) is assumed to have a high transmission range, BS sends transmissions to all nodes. Then by using an RSSI signal, the sensor node measures the signal strength of the transmission from the BS. If RSSI is measured by a node from a broadcast message that has a value higher than the other node, then the node will be selected as the leader node. But in this EPDP simulation process, leader node selection is done by selecting the distance node ($distNode(x, y)$) closest to the BS based on the calculation of the distance of the BS coordinate (x_i, y_i) with node coordinates (x_j, y_j) as in equation 3. In this EPDP also use dominating but cluster-based set concepts [14,15]. As for cluster formation, the first is to calculate the distance from one node to another node. Then look for at least 5 nodes that have the shortest distance with the node to represent the sensing data in the same area and the distance must be below the specified threshold. Equations 4 and 5 are functions to determine the distance threshold ($Tdist$) and the number of nodes that have the shortest distance ($Nshortest$) by paying attention to the total number of nodes ($Ntotal$). After that the labeling process for each node starts from the node closest to the leader node and continues to all vertices. In algorithm 2, the process of selecting the leader node and forming clusters will be explained.

$$distNode(x, y) = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (3)$$

$$Tdist = \frac{Ntotal}{Nshortest} \qquad (4)$$

$$Nshortest = \frac{Ntotal}{Tdist} \qquad (5)$$

In algorithm 2 will be explained about the process of selecting leader nodes and forming clusters.

Algorithm 2. Leader Node Selection And Cluster-Based Dominating Set Concept in EPDP

Input : node location (xi,yi), base station (x,y), total node, maximum round, energy, packet size to be sent.
Output : leader node, cluster.

Step 1 : find the distance of all nodes from base station
Step 2 : look for the node closest to the base station to be chosen as the leader node
Step 3 : calculate the distance of a node to each node, then save the results of a circle in a matrix.
Step 4 : find a node that has the 5 shortest distances from that node
Step 5 : determine the threshold distance used for the boundary of defining clusters
Step 6 : give a cluster label to each node starting from the node closest to the leader node based on the shortest distance and the specified threshold

---

Figure 5 shows the cluster node formed, and the selected node leader. To provide a more detailed explanation of the description of leader node selection and the formation of this cluster.
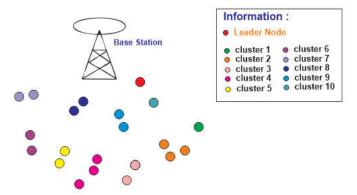

Figure 5. Cluster nodes are formed and leader nodes are selected

b) Chain Formation
   The process of creating an EPDP chain is to use dynamic programming algorithms and starts from the node closest to the leader node up to the farthest node from the base station. Where the concept of dynamic programming proposed is by calculating the probability of the distance between adjacent nodes based on cluster sequences, Figure 6 shows the probability of distance based on the cluster order.
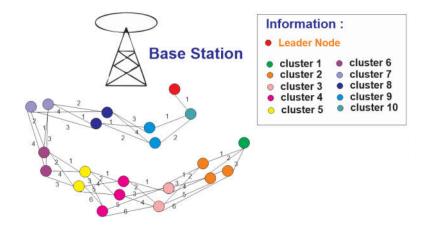
Figure 6. Distance probability each node

For the above completion process you can use backward dynamic programming [16], for example $x_1, x_2, ..., x_n$ are the nodes visited at stage k (k = 1, 2, 3, 4, ..., the number of clusters). Then the route passed is $1 \rightarrow x_1 \rightarrow x_2 \rightarrow x_3 \rightarrow x_n$ which in this case 1 = leader node, and x_n = the farthest node. Stage (k) is the process of selecting the next destination node, where the amount depends on the number of clusters. While the status (s) are the nodes in the graph that are before that stage begins. Equation 3 is a function to calculate the minimum distance value starting from the farthest node, where c_{sxk} is the weight or distance from s to x_k (nodes that are at stage k). While for equation 4 is a recursive function to calculate the shortest path from s to x_n for use in each stage k.

$$f_{x_n}(s) = c_{sx} \qquad (6)$$

$$f_k(s) = \min(c_{sx} + f_{k+1}) \qquad (7)$$

After the process of calculating dynamic programming algorithms for each node in each cluster, the chain will be formed as seen in Figure 7.
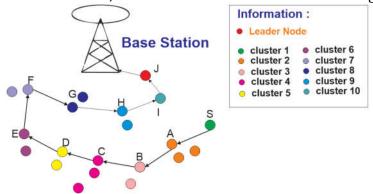

Figure 7. Chain formed using EPDP

c) Data Agregation

Data collection and fusion begins after choosing the leader nodes and chain formation is complete. Data collection is done using an approach based on tokens (small control packages). The token will be passed by the leader node along the chain to the last node. Then the last node sends data and tokens to the next node along the chain after receipt of the token is complete. Each node that is passed will merge data by combining its own data with neighboring data into one package of the same length. This process continues until the data reaches the leader node. After reaching the leader node, the data packet will be transmitted to BS (Base Station). For example in figure 6, the leader node is J which is at the beginning of the chain. Node A passes a token (small control package) to the end node S along the chain. After receiving a token, the node S transmits its sensing data and token to the next node in the chain, namely node A. Node A when receiving packet data and tokens from node S, merges data with the received packet and its own sensing data and transmits it to the next node namely node B. Similarly node B sends the packet to node C, D to E, E to F, F to G, G to H, H to I and I to J. Finally, the Leader node J receives a packet from node I and brings together its own sensing data with the packet received and then send it to the Base Station.

4.3 Creating a sensor node and BS placement scenario

In this section, will explain about making scenarios for the positioning of nodes and base stations (BS). Where the scenario for the positioning of nodes in this research will be done on a random (the location coordinates of each node are determined when the code simulation algorithm is executed using a random function) and static (the location coordinates of each node are determined first, before the code simulation algorithm is executed). Static and random node placement is done with the aim to see the difference in energy consumption for each scenario. And one scenario that has significant energy consumption efficiency is BS's location. Therefore, the location of the BS will be split into three scenarios including the BS in corner of the network with coordinate (100,100), the BS is placed in the middle of the network with coordinate (50,50), and the BS is placed outside the network with coordinate (125,55). Figure 8 shows the scenario of placing different BS.
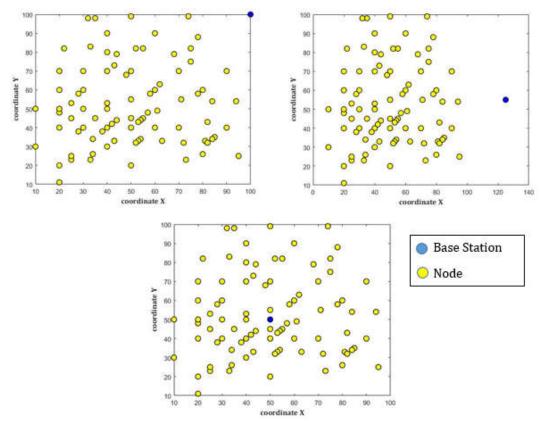
Figure 8. The base station is at the network corner (A), the base station is at the network outside (B), the base station is at the network middle (C).

4.4 Testing the algorithm and displaying the experimental graph

The next step after defining the required parameters and maker of node and BS placement scenarios is to simulate starting from the LEACH, PEGASIS, and EPDP algorithms. In this study uses Matlab 2016 to simulate it. Every algorithm that has been implemented will be compiled and the results will be obtained for further processing. After the results of the compile are obtained, graphs including the number of nodes that die, the number of nodes that alive, remaining of energy, latency average, and number of data received at BS will be obtained.

4.5 Analyzing the results of the experiment

The process of analyzing how effective LEACH use, PEGASIS, and EPDP algorithms is done after all parameters are obtained and also the graph has been generated. The experimental scenario used to analyze the performance of LEACH, PEGASIS, and EPDP is to analyze the results of the BS placement (at the network corner, at the network middle, and at the network outside) using static topology nodes first. Then proceed with analyzing the results of the placement of BS (at the network corner, at the network middle, and at the network outside) by using topology nodes randomly, if the experiments with topology nodes are statically completed.

5. EXPERIMENT AND ANALYSIS

In this section, will explain the analysis of the protocol that submitted under the EPDP name compared to the LEACH and PEGASIS protocols. In this study uses Matlab 2016 to simulate it. Some of the parameters that test are about the number of nodes that die, the number of nodes that alive, remaining of energy, latency average, and number of data received at BS will be obtained.

5.1 The Number of Nodes That Alive

One of the measurements used to analyze the lifetime of the network is the number of alive nodes in each round (The Number of Nodes That Alive). The measurement of the number of nodes that alive is used to measure the reliability of an algorithm both LEACH, PEGASIS and EPDP, the way is to see the number of live nodes, indicating that the node still has the energy to transmit data in every round. Figure 9 shows a graph of the number of nodes that alive based on the static node topology and the BS placement scenario. It can be concluded from the graph that the number of nodes that alive of the EPDP algorithm is better than the PEGASIS and LEACH algorithms, where the triangle symbol line, circle symbol line, and minus symbol line of the graph are always higher than the other lines. EPDP has better performance because in EPDP not all nodes are activated in one round with the strategy of using the dominating set concept, where the node to be activated is only 1 node in one cluster, so it can save energy consumption in each round. Besides that, in EPDP also uses dynamic programming algorithms for chain formation, so that it can get the best path compared to using the Greedy algorithm in PEGASIS. From the graph, it can also be concluded that the placement of BS, has better performance for LEACH if it is placed at the network middle. As for PEGASIS and EPDP, the BS that is located outside the network has better performance.
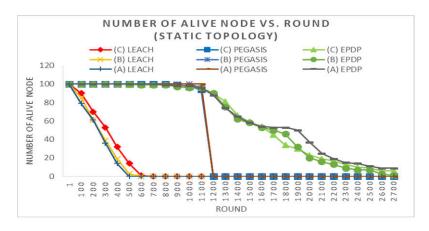

Figure 9. The Number of Nodes That Alive (Static Topology) where the BS is outside the network (A), The Number of Nodes That Alive (Static Topology) where the BS is corner the network (B), The Number of Nodes That Alive (Static Topology) where the BS is middle the network (C)

From the graph in figure 9 it also shows that PEGASIS has decreased sharply in the rounds of 1100 - 1200, this is because PEGASIS activates all nodes in every round, so that the node will die almost simultaneously. Figure 10 is a graph that shows the decline of PEGASIS starting from round 900 to 1200. Figure 11 is an alive node number graph based on BS location with random node topology. It can be concluded from the graph that the number of nodes that alive of the EPDP algorithm is better than the PEGASIS and LEACH algorithms, where the triangle symbol line, circle symbol line, and minus symbol line of the graph are always higher than the other lines. From the graph, it can also be concluded that the placement of BS, has better performance for LEACH, PEGASIS, and EPDP if it is at the network middle. Where on a line with a triangle symbol, up to a round of 2700 the number of living nodes is still left, compared to EPDP with BS which is at the network outside or at the network corner that is represented by a line with a circle and minus symbol.
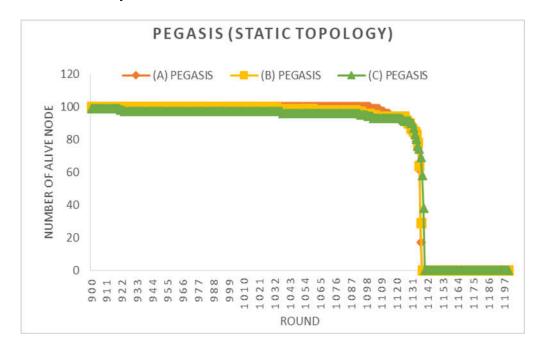

Figure 10. The Number of Nodes That Alive in PEGASIS (Static Topology) where the BS is outside the network (A), The Number of Nodes That Alive in PEGASIS (Static Topology) where the BS is corner the network (B), The Number of Nodes That Alive in PEGASIS (Static Topology) where the BS is middle the network (C).
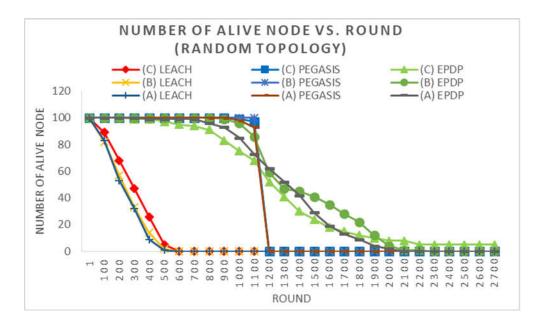
Figure 11. The Number of Nodes That Alive (Random Topology) where the BS is outside the network (A), The Number of Nodes That Alive (Random Topology) where the BS is corner the network (B), The Number of Nodes That Alive (Random Topology) where the BS is middle the network (C)

5.2 The Number of Nodes That Die

The Number of Nodes That Die measurement is used to measure the reliability of an LEACH, PEGASIS and EPDP algorithm by looking at the number of dead nodes, which indicates that the node has run out of energy and cannot transmit data in each round. In the measurement of Number of Die Node will be carried out into several scenarios, including based on the placement of BS (at the network corner, at the network middle, and at the network outside). And based on node topology, namely topology with nodes randomly and static. Figure 12 shows a graph of The Number of Nodes That Die based on static node topologies and BS laying scenarios. It can be concluded from the graph that The Number of Nodes That Die of the EPDP algorithm is better than the PEGASIS and LEACH algorithms, where the line with the symbol triangle, the symbol circle, and the minus symbol of the graph are always lower than the other lines. From the graph, it can also be concluded that the placement of BS, has better performance for LEACH if it is at the network middle. As for PEGASIS and EPDP, the BS that is located outside the network has better performance.
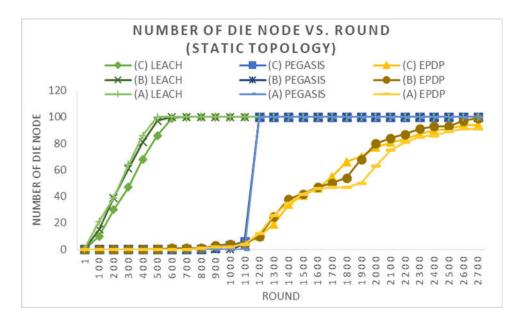
Figure 12. The Number of Nodes That Die (Static Topology) where the BS is outside the network (A), The Number of Nodes That Die (Static Topology) where the BS is corner the network (B), The Number of Nodes That Die (Static Topology) where the BS is middle the network (C)

Figure 13 is an die node number graph based on BS location with random node topology. It can be concluded from the graph that The Number of Nodes That Die of the EPDP algorithm is better than the PEGASIS and LEACH algorithms, where the line with the triangle symbol, the circle symbol, and the minus symbol of the graph display below the other lines. From the graph it can also be concluded that the placement of BS, has better performance for LEACH, PEGASIS, and EPDP if it is at the network middle. Where on a line with a triangle symbol, up to a round of 2700 the number of dead nodes does not reach 100%, compared to EPDP with BS which is at the network outside or in the corner of the network that is represented by a line with a circle and minus symbol.
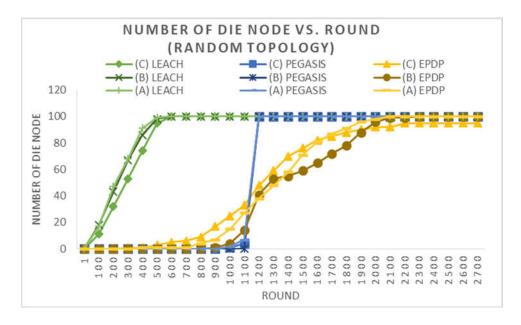
Figure 13. The Number of Nodes That Die (Random Topology) where the BS is outside the network (A), The Number of Nodes That Die (Random Topology) where the BS is corner the network (B), The Number of Nodes That Die (Random Topology) where the BS is middle the network (C)

5.3 Remaining of Energy

This Remaining of Energy measurement is used to measure the reliability of an algorithm both LEACH, PEGASIS, and EPDP by looking at the remaining energy from all nodes to transmit data in each round. The Remaining of Energy measurement will be carried out in several scenarios, including based on the BS location, which is in the middle, on the corner, and outside the network. And based on node topology, namely topology with nodes randomly and static. Figure 14 shows a graph of the Remaining of Energy based on the static node topology and the BS laying scenario. It can be concluded from the graph that Remaining of Energy of the EPDP algorithm is better than the PEGASIS and LEACH algorithms, where the line with the triangle symbol, the circle symbol, and the minus symbol of the graph is always higher than the other lines. From the graph it can also be concluded that the placement of BS, has better performance for LEACH, PEGASIS, and EPDP if it is at the network middle.
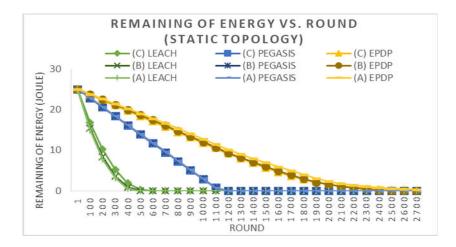
Figure 14. Remaining of Energy (Static Topology) where the BS is outside the network (A), Remaining of Energy (Static Topology) where the BS is corner the network (B), Remaining of Energy (Static Topology) where the BS is middle the network (C)

Figure 15 is an remaining of energy graph based on BS location with random node topology. It can be concluded from the graph that Remaining of Energy of the EPDP algorithm is better than the PEGASIS and LEACH algorithms, where lines with triangles symbol, circle symbols, and minus symbols of the graph show higher than other lines. From the graph, it can also be concluded that the placement of BS, has a better performance for LEACH if it is at the network middle. Whereas for PEGASIS and EPDP, the BS that is located on the network corner has better performance.
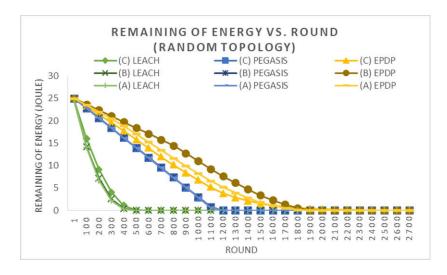

Figure 15. Remaining of Energy (Random Topology) where the BS is outside the network (A), Remaining of Energy (Random Topology) where the BS is corner the network (B), Remaining of Energy (Random Topology) where the BS is middle the network (C)

5.4 Latency Average

In addition to analyzing network lifetime, in this research will also analyze several Quality of Service (QoS), including Latency Average (the average time spent from all nodes to send data to BS in each round). This Latency Average measurement is used to measure the processing time of an algorithm both LEACH, PEGASIS, and EPDP. Figure 16 is a graph that shows the average latency of the LEACH, PEGASIS, and EPDP algorithms based on the number of nodes starting from 50 nodes, 100 nodes, and 150 nodes. From the graph, it can be seen that the average latency of the algorithm is the largest compared to PEGASIS and EPDP. The average latency of EPDP with the number of nodes 50 is smaller than the others, but at the number of nodes 100 and 150, the average latency of EPDP is greater than EPDP. This is because in the EPDP there is a calculation process for dynamic programming that is longer than the greedy algorithm in PEGASIS.
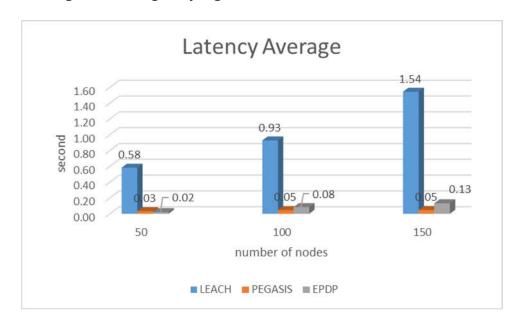

Figure 16. Latency Average

5.5 Number of Data Received to BS

Another parameter used to analyze the efficiency of the algorithm is the number of data received at BS. This parameter is used to determine the efficiency of an algorithm in the process of collecting data to BS in each round. Figure 16 is a graph showing the number of data received at BS from the LEACH, PEGASIS, and EPDP algorithms based on the BS location in the middle, at the corner, and outside the network. From the graph, it can be concluded that the number of data received at BS of the EPDP algorithm has better results compared to LEACH and PEGASIS. Where in EPDP, up to round 2400 there is a node that still sends data to the BS, while the LEACH algorithm is only up to round 600 and PEGASIS is only up to round 1200.
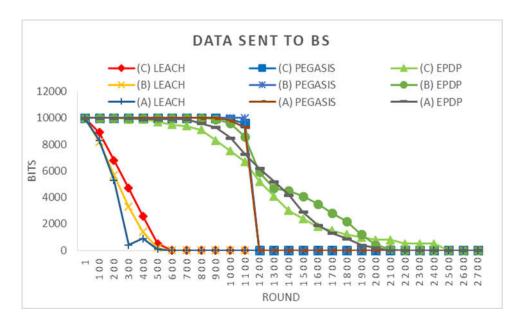
Figure 17. Number of data received at BS

6. CONCLUSION

A number of routing protocols such as LEACH and PEGASIS has been suggested to overcome the problem of energy efficiency in the WSN. But LEACH and PEGASIS also have weaknesses. At PEGASIS the data is collected excessively into the BS. As a result, all data are active during data transfer rounds that limit WSN longevity. In this research, this study proposes the PEGASIS version uses Dynamic Programming (EPDP). EPDP uses the Dominating Set (DS) idea in selecting a node in coverage same area to be activated in a round. After that, this study uses dynamic programming based optimization in forming chains from each node. For the selection of the Chain Leader (CL), it is chosen based on the amount of residual energy and its proximity to the BS. There are 2 topology nodes that this study use, namely random and static. Then for Base Station (BS) placement, it will be placed in outside, in the corner, and in the middle of the network. It was produced from the experiments that the EPDP protocol implementation was performed better than the LEACH and PEGASIS protocols. Whereas to save more energy, the best placement of the BS is at the network middle, and using the topology of static node deployment. For further research, EPDP trials should also be conducted on other versions of PEGASIS like PEGASIS-INL, PEGASIS-MECA or E-PEGASIS and tested also on its throughput.

REFERENCES

[1] Frank L. Lewis, **Wireless sensor networks**, *Smart environments: technologies, protocols, and applications*, pp. 11-46, 2004.

[2] M. Robihul Mufid, M. Udin Harun Al Rasyid, Amang Sudarsono, **Allocation Strategy Guaranteed Time Slots (GTS) on Real Hardware Wireless Sensor Network (WSN)**, *International Conference on Information & Communication Technology and System (ICTS)*, pp. 187-192, Oct. 2017.

[3] IEEE 802.15.4, **Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)**, *IEEE Standard for Local and metropolitan area networks*, Sep. 2011.

[4] M. Udin Harun Al Rasyid, Bih-Hwang Lee, Iwan Syarif, Mokhammad Muqoffi Arkham, **LEACH Partition Topology for Wireless Sensor Network**, *International Conference on Consumer Electronics – Taiwan (ICCE-TW)*, pp. 1-5, May 2018.

[5] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan, **Energy efficient communication protocol for wireless microsensor networks**, *International Conference on Hawaii International Conference in System Sciences (HICSS)*, pp. 3005-3014, Jan. 2000.

[6] Stephanie Lindsey, Cauligi Raghavendra, and Krishna M. Sivalingam, **Data gathering algorithms in sensor networks using energy metrics**, *IEEE Trans. Parallel Distrib. Syst.*, vol. 13, no. 9, pp. 924–935, Sep. 2002.

[7] Mohammad Robihul Mufid, M. Udin Harun Al Rasyid, Iwan Syarif, **Performance Evaluation of PEGASIS Protocol for Energy Efficiency**, *International Electronics Symposium on Engineering Technology and Applications (IES-ETA)*, pp. 241-246, 2018.

[8] Wendi Rabiner Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan, **An application-specific protocol architecture for wireless microsensor networks**, *IEEE Trans. Wireless Communications*, vol. 1, no. 4, pp. 660- 670, Oct. 2002.

[9] Balakrishnan Baranidharan, and B. Santhi, **GAECH: Genetic Algorithm Based Energy Efficient Clustering Hierarchy in Wireless Sensor Networks**, *Journal of. Sensors*, 2015.

[10] Stephanie Lindsey and Cauligi S. Raghavendra, **PEGASIS: Power-efficient gathering in sensor information systems**, *In Aerospace conference proceedings*, vol. 3, pp. 3-3, 2002.

[11] Alekha Kumar Mishra, Rukshan Ur Rahman, Rahul Bharadwaj, and Rohit Sharma, **An Enhancement of PEGASIS Protocol with Improved Network Lifetime for Wireless Sensor Networks**, *Communication and Information Technology Conference (PCITC)*, pp. 142-147, 2015.

[12] Jin Wang, Jiayi Cao, Yiquan Cao, Bin Li, and Sungyoung Lee, **An improved energy-efficient clustering algorithm based on MECA and PEGASIS for WSNs,** *International Conference on Advanced Cloud and Big Data*, pp. 262-266, 2015.

[13] Saurav Ghosh, Sanjoy Mondal, and Utpal Biswas, **Enhanced PEGASIS using ant colony optimization for data gathering in WSN**,

International Conference on Information Communication and Embedded Systems (ICICES), pp. 1-6, 2016.

[14] Azrina Abd Aziz, and Y. Ahmet Şekercioğlu. **A distributed energy aware connected dominating set technique for wireless sensor networks**, *International Conference on Intelligent and Advanced Systems (ICIAS)*, vol. 1, pp. 241-246, Jun. 2012.

[15] Kui, X., Sheng, Y., Du, H., & Liang, J. (2013). **Constructing a CDS-based network backbone for data collection in wireless sensor networks**, *International Journal of Distributed Sensor Networks*, vol. 9, no. 4, pp. 258081, 2013.

[16] Ronald A. Howard, **Dynamic Programming and Markov Processes**, *MIT Press and Wiley*, New York, 1960.