Reliable and Energy-Efficient Routing Protocol for Wireless Networks

Hamdollah Ghamgin[1]

[1]Instructor, Department of Engineering, Payame Noor University, po Box 19395-3697, Tehran, IRAN

Abstract
Today, designing high energy-efficient protocol for WSN becomes more challenging. Some of military applications, many civilian software, trading and environmental applications, considered the use of sensor networks. WSN routing protocols can be classified into two main components: Central and hierarchical data. However, in order to choosing browses and keeping the branches, hierarchical protocols need more computation. Therefore, our research is focused primarily on the central data categories, and also all of the communications were established based on metadata (The data that have been named based on its attributes). In this paper, we're going to design a valid and efficient protocol, which is called REEP. The aim of this study is developing a reliable and high efficient protocol, which can increase the needs of fault tolerance with increasing of network lifetime.

Keywords:
Wireless sensor networks, fault tolerance, design REEP, reliability

1. Introduction

Typically, a wireless sensor network would be consists of significantly large number of sensor nodes which are arranged on the same level in a compact way. Generally, a WSN algorithm is somehow a distributed algorithm. The sensor nodes are able to work with each other and they have the ability to estimate their environment (e.g. light, temperature, sound and vibration) as well. Then sensory estimates will be converted into digital signals and they will be processed in order to present some features of the phenomena of the sensors. The gathered data would be used as a feedback for users and are pathd through non-substructure multi-station construction of WSN. Important factors of a design has some influence on designing a wireless sensor network which is considered as an instruction for designing a protocol or an algorithm, and operates for any type of sensor network. These factors include scalability, production costs, environment factor, sensor network topology, hardware limitations, transmission medium, fault tolerance and power consumption [AS02]. Among these factors fault tolerance and power consumption have significant importance for designing a WSN energy efficient protocol.

Manuscript received October 5, 2017
Manuscript revised October 20, 2017

2. Previous works

Recently many researchers presented different routing protocols for MANET. These are categorized as follows [LK03]:

- Pre-active routing Protocol (Routing With the Table), such as: DSDV (Sequenced Distance Vector - Destination), WRP (Wireless Routing Protocol), FSR (Fisheye Type Routing) [IK03], AWDS (Ad hoc Wireless Distribution Service) – Layer 2 and Babel wireless network routing protocol – types of AODV with faster deviation, HSR (Hierarchical State Routing), TBRPF (Diffusion topology based on routing protocol transmitted in the opposite direction) [WIKI].
- Hierarchical routing protocols that are categorized by zone are: ZRP (Zone Routing Protocol), HARP (Hybrid ad hoc routing protocols), and ZHLS (Hierarchical Zone linkage State Routing) [LK03].
- Cluster routing protocols includes: CGSR (Cluster Gateway Switch Routing), HSR (Hierarchical State Routing), CBRP (Cluster Based Routing Protocol) [LK03], DART (Dynamic Address Routing), DDR (Distributed Dynamic Routing), and GSN (General State Routing) [WIKI].
- Routing protocol based on central nodes include: LANMAR (Landmark Ad hoc Routing), CEDAR (Centered Adaption Distribution Routing Protocol), and OLSR (Optimized Linkage State Routing) [LK03].

2.1 REEP designing

In this paper the proposed protocol for wireless sensor network is designed with respect to DD protocol. Therefore data-centric routing technique used in DD is would be reused in the REEP. The purpose of this study is to provide a viewpoint for structure and design of REEP processing, which include: Design features, design elements, design process, and usability of the designed scenarios. Before describing the design steps, it is necessary to first introduce the design choices of REEP. There are five design elements that are used in different stages of REEP protocol. The energy threshold value is one of these elements, and selecting this value varies depending on different

requirements of different applications. REEP have five important elements. These include: 1. the sense event, 2. the INFO events, 3. request event, 4. the energy threshold value, 5. the FIFO queue. A sense event is a kind of query which pushes each node to start sensing. This sense event is generated at the sink of central node, and is supported by a sensor network for acquiring information. The response of this query is the INFO event, which is generated at source node. Request event are defined in the destination node and are used for path setup to retrieve the real data. Real data which are collected from each sensor network would be used for processing any physical phenomenon. Each node in REEP uses the energy threshold value by checking whether a node agrees or denies participating in any future activities. This will result in a more reliable transmission of any event information or any real data; the FIFO queue is used at each node to track the sequence of received INFO event from different neighbors.
Each node selects the first neighbor through the request sent for path setup. The above described elements are a particular category of sensory network applications.
The steps of REEP design consists of designing naming scheme and designing different events propagation. These events propagation represent the main functionalities of REEP. They can be divided into three stages as you shown table 4.1. In the next section we will present an explanation about how these events are generated and processed in the REEP protocol.

Table 1. Operational procedures of REEP

Steps	Name
1st	*Sense event* propagation (Section 4.3.3)
2nd	*Info event* propagation (Section 4.3.4)
3rd	*Request event* propagation (Section 4.3.5)

2.2. Simplified schematic of REEP

Let's consider a simple form of WSN application in order to get an overall view of how the REEP works. Assume that a large geographic area has a large number of sensor nodes which are used for security purpose. As shown in the Figure, assume that the user send a request through task management node. This query will be sent to all sensors so that they could sense it in the specific time and collect information from the environment as well. Nodes sense any moving object – such as human, animal, or a vehicle – and then send this information to the user (they are not real data). If multiple sources detect objects from different locations, then this user will receives a list of detected objects and matches it with the information of the source node location. In order to have detailed data of the detected object, the user can choose one or more source from the list. Then, the user will send the real data through the central node. Therefore, the central node sends this request to the particular neighbors in order to setup the path for real data transmission (see FIG 4.1(c)). Intermediate nodes do the same task as central node for path setup. Once the sources receive their corresponding request for the real data, they complete the path and start to send data along the dedicated path (see FIG 4.1 (d)). Thus, the user start receiving the real data of an individual path for every task.

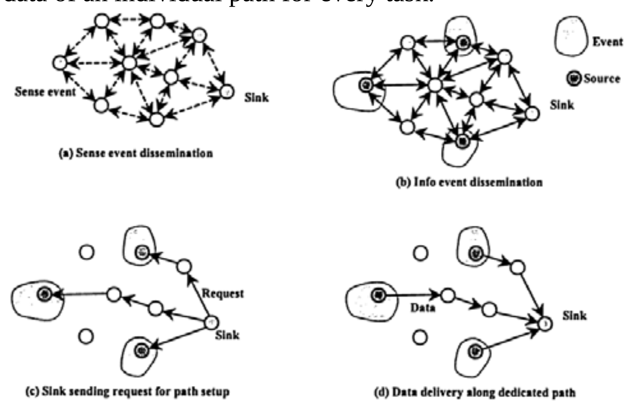


Fig 1. A brief explanation of REEP operations

One of the most important features of REEP is maintenance of an energy threshold value in each node; this value should be checked at any time, and in any node, and will be transferred only when it receives the request event node and the real data. Designing the energy threshold value depends on the application. When the energy of path node and data transmission is avoided, only control data would be transferred. Nodes with less amount of energy can be replaced before getting destroyed. In some situations such as nodes or link failure, nodes needs to get an alternative set up for the path; in such cases, REEP nodes do not need to perform extensive research in the network or do periodic flooding in order to find the best next path for data transmission or it is not necessary for REEP nodes to maintain some of the alternative path {GG01}. Rather, they can replace the alternative path set up with the next best neighbor from the queues. Therefore, in each node all of the required events are saved for future use. REEP protocol is designed based on a method that allows it to work with different scenarios. For example, such scenarios are used when an individual source detect a single or multiple object, or when multiple sources detect a single or multiple object. These scenarios are described in the following. When in the entire network only one source node find an object, it create an entry for that object and flood this information through the entire network. If the user sends a request for this object, then the central node sends a request for set path set up. Once the path is established, the source node starts sending real data through this path; this scenario is shown in the Fig 2.

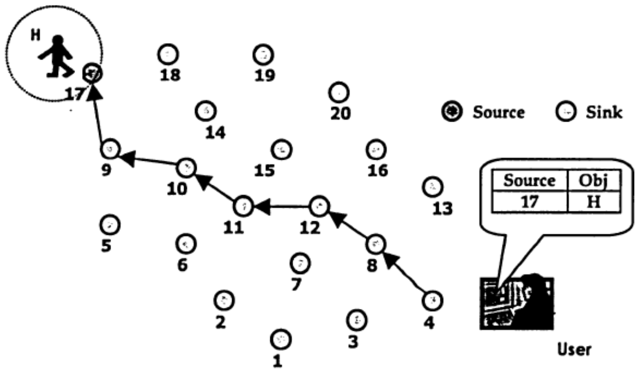


Fig 2. a source detects an object.

This is a common scenario in wireless communication, which a pair of sensor nodes detects the same object (see Fig 3). This happens when the object belongs to the area where sensing events occur in multi-layer multiple-node regions. In such situations, different source nodes create different entries for the same object and transfer and defuse this information to various sources. This case is a type of data redundancy in memories of node caches, but does not affect the overall performance of the network. The reason is that when the application start working on this subject, it shows a list of sources that detected the same object, and the location of these source nodes are belongs to a small area and user can quickly realize that there is only one object. Then, the user is able to choose any source node for requesting the data based on the amount of additional data available for each task. Once source node receives this request, it starts sending the data along the path toward the destination node. Fig 3 shows this scenario.

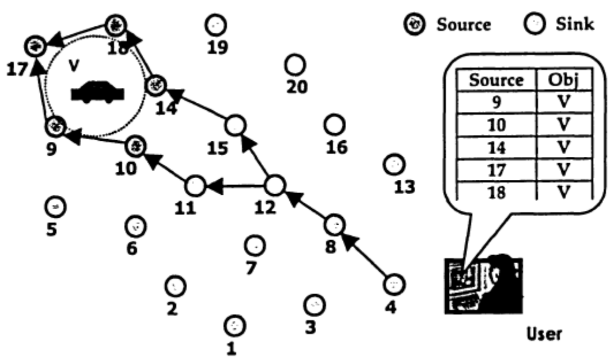


Fig 3. Several sources finds one object

2.3 Fault tolerance in REEP

Wireless sensor communication probably has some failure nodes or links because of source limitations and irreplaceable energy of sensor nodes. This type of failure causes disconnected path and breaks the communication between sensor nodes. There are other types of failure that can be experienced by unexpected environmental factors on the communication path. Examples of such failure include lightning, strong wind, and any obstacle in the path, you should know that repairing or reconstructing a failed path need to resolve the problem. Network protocols are designed in a way that can work with such failures and keep the communication among sensor nodes despite the existing obstacle and problems. This is called fault tolerance. These kinds of problems are considered when designing REEP. In the next section we will discuss about fault tolerance and how to handle it in REEP.

Some of the most important features of REEP protocol are introduction of FIFO queue and energy threshold value. These features give good shape to REEP protocol and have some advantage for us. The Energy checking mechanism keep the control of remaining energy in every node and results in a complete path in the network which will be used for data transmission. Furthermore, maintaining energy threshold in each node allows data to be transmitting from each node to the next node which is in the path and has low energy. Therefore, in REEP energy loss is very low. In order to apply alternative path setup in a node or on a path failure in a network, there is no need for REEP to flood low rate events or to maintain the proactive alternative paths in an advances way which leads to more computation and more energy consuming. In such situations, REEP simply pick the next best neighbor nodes from the FIFO queue in order to request an alternative path setup without causing any flooding. REEP works well with multiple objects as well as multiple node sources. The reason for this is that REEP is comparable based on both the type of detected object and the information about the location of source node that find that object. Usually random emission moves forward, and REEP provides some information about this path. The next sources create some info events about the detected object and send them to all the nodes. Then the user can understand the path of moving objects if he observes the changes in the data location of the source node for a specific object. When it is necessary for a user to send a queue into a network, there is no need to learn about the existing object or about the area that it works, the user can achieve an overall view about all of the detected objects and also gain some information about the relative position of those by sending sensor events to the sink node in order to scan the whole area of the network. In these cases REEP cannot do all the tasks at the same time. Instead, it gives the user the right to choose one task after completing INFO events. The remaining activities such as path setup and data transmission are completed based on user's choices. User only chooses the tasks from the list that he needs them. These features help to avoid additional and unnecessary processes in the network. It is likely that all of the data-centered protocols in REEP are for applications and its sensor nodes, and we are going to discuss about it in the next section. These restrictions have some limitation on the usage of the protocols. Therefore, REEP is not suitable for applications that need dynamic changes of network topology. In REEP, due to high consumption of energy in the entire network, the more repetitive sensory function of pre-organized object or started sensory function by a user, the greater changes would occur for defected nodes and data losses in the network, and you should know that without accurate designing of pre-organized sensory functions, REEP cannot work properly. The necessary requirements and non-performance of presented protocols should be qualified. First, we present qualitative comparison of REEP and DD, and then we provide quantitative comparison of both protocols based on multiple metrics, which includes computational complexity, the bond transfer mean, average dissipated energy, energy interruptions, ratio of dissipated energy. Performance analyzes of both network arrangement and random arrangement of sensory nodes is done in the sensor networks.

The simple network topology that we used in our simulation consists of a large area which is occupied by a large number of sensor nodes, and is scattered randomly and have the ability to perform collaborative tasks as well. We should mention here that our intended application needs the whole sensory area to study available objects, because the user may not know all of objects that exist in the network. Then, based on the results on study, the user chooses an object or an exploring source node for retrieving the actual data.

2.4 Propagation comparison between REEP and DD

We made some changes in REEP propagation process designing, to make a comparison with DD emission process. Unlike DD, the gradient setup technique in REEP is not followed in the first stages of sensory event's emission. This reduces information overload. In INFO event propagation phase, all information gained before participation of an event will be saved.

Request event propagation phase maintain the same energy consumption in every node. Then we compare the difference between REEP and DD propagation phase which is presented in the following. In DD, during the interest propagation phase (the first step), two methods of gradient establishment are done between each pair of neighboring nodes.

3. Implementation

We used MATLAB 7.4 software for implementing REEP and DD. Propagation of each event stimulated as an individual packet transmission, i.e. interests, reply, reinforcement and data packets are used as event transmission in DD, and sense, info, request and data packets are used ad event propagation in REEP. In order to implement REEP, we must make some assumption about communication range; techniques used for node placement and different data structure details. We assume that all the nodes are stationary in our sample network application, means that they are not sensing all the time.

Communication range: we assume that in our system network the distance between two horizontal or vertical nodes is 10 meter and each center node (A node in Fig 4) can be surrounded at most with 8 neighbor's node. According to this assumption each node's radio range is about 14.142 m (FIG 4). This value of radio range is constant in all of our experiments. In this paper, different communication range such as sensing, reception and transmission range of each sensor have been assumed equal which is because of observation both protocol in term of energy efficiency.

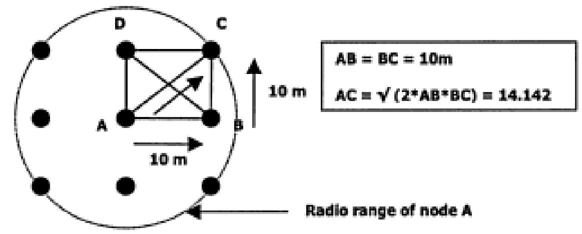


Fig 4: radio range of each sensor node

Node placement in topological level: we placed each sensor node within a rectangular area in a pseudo-random fashion (FIG 5). This arrangement is referred to as "pseudo-random" because a node can be placed anywhere within a rectangular region or cell, but the location of that cell is fixed. The topology area (i.e. width and height) of the network is an input in our simulation. We need to follow network topology to place the nodes in a way that the distance between tow nodes is not fixed. Moreover, a random number is within the range. For example, if the average distance between the nodes is 10, then the distance between tow nodes can vary from 5 to 15, which is based on this calculation: (x or y value in network system) + (random 15). This topology allows the avoidance of any sensor node isolation from all its neighbors. A totally random arrangement of sensor nodes in topological level can result in increasing of the congestion in some subareas or it can leads to separation of area from isolated or dispersed sensor nodes. If any intermediate node does not have any reachable neighbor, then path cannot be constructed between entry and destination and as a result the data will be lost. Our pseudo-random nature of node placement does not necessitate having a fixed number of neighbors for each node; rather it varies from node to node. Sensor nodes can have different range limits for different activities depending on the requirements of any application. We have assumed that sensing, transmitting and receiving range of every sensor nodes are equal. Therefore, neighbor list of each node includes all those neighbor nodes that belong to its radio range (see FIG 4)

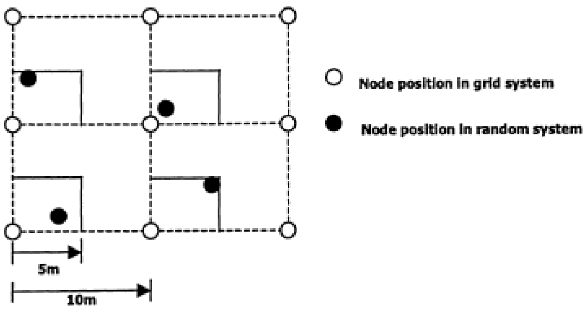
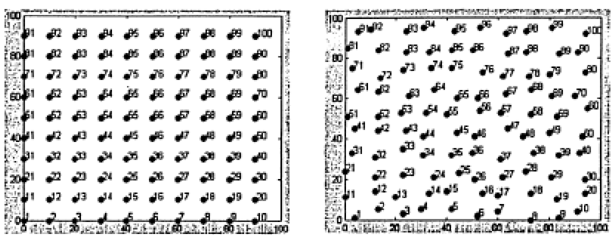


Fig 5: Node placement in a pseudo-random model

According to above placement technique, we used a setup of 100 nodes in a 100 by 100 squared meter area (see FIG 6). Node placement can be either in a network arrangement (see FIG 6 (a)) or in a random arrangement (see FIG 6 (b)) system.



(a) Grid placement **(b) Random placement**

Fig 6: placement of 100 nodes in a 100 ×100 m square area

Structure of a sensor node: The structure of a sensor node is at the top of hierarchy of all structures in our simulation, so that we can store the state information.

Structure of an Object Type: This data structure is used to store all the detailed information about object detected by each source node.

4. Stimulations

In order to analyze the performance of REEP and DD as a function of size and energy network, we have stimulated a variety of different sized sensor area with different setup for variety of scenarios. These scenarios reflect the behavior of both protocols, and some of them highlight the advantages of REEP compared to DD for some specified scenarios and application types. In all our simulations, we considered the farthest distance between the source and the central node, and we provide only snapshot of output files, rather than the complete output files.

4.1 Stimulation input

The users input to our sensor network simulation are as followed:

Total – sensor – nodes refers to the number of total sensor nodes used in the sensor network. Energy is the total initial energy available at each node. Area – Width is the width of topological area. RECT area and the object type are used for defining interest in DD. Energy threshold value is used for finding reliable path in REEP. Maximum data indicates the maximum number of data which are generated in the source node and Maximum Object refers to the maximum number of detected objects. There are other specific inputs in the simulation as well, which are calculated based on the user's inputs. These are:

NODES_IN_ROW = ceil (sqrt (TOTAL_SENSOR_NODES)) (4)

SINK = TOTAL_SENSOR_NODES – mod (TOTAL_SENSOR_NODES, NODES_IN_ROW) (5)

NODE_HOR_VERT_DISTANCE = AREA_WIDTH / NODES_IN_ROW (6)

4.2 Performance metric

We use four metric to analyze and compare the performance of DD and REEP and for considering specific application. These metrics are as followed:

$$\frac{(P_r+P_t)/2}{N\times T}$$ (٧)

Average packet transmission: This value measures the average number of packet transmissions per node, per task and is determined by above equation.

In the above equation P_r indicate the total number of received packet and P_t denotes the total number of packet transmission in the network. The sum of P_r and P_t has been divided by two, because one transmission includes the reception and transmission of each packet. Here N is the total number of sensor nodes and T indicates the total number of tasks. The lower value of this metric indicates the lesser number of packet transmission by each node, less energy consumption and better performance.

Average dissipated energy: This metric indicates the average dissipated energy for each individual task. An increase in the average dissipated energy indicates more power consumption by each node. This metric is computed with the following equation:

$$\frac{\sum_{i=1}^{N} (IEi - REi)}{N\times T}$$ (8)

In this equation, N is the total number of sensor node and T indicates the total number of tasks. For each node i, the used energy is the difference between the IE (the total energy available in the node i) and the RE (the remaining energy in node i after simulation).

4.3 Simulation results

Different simulation results are shown here for different cases. In every experiment conducted in this paper with the aforementioned types of application indicates that REEP performance was better that DD performance. Since the gradient setup and low rate flooding of exploratory events (in case of alternative path discovery) are not followed in REEP, we can observe better performance of REEP in terms of average packet transmission and average dissipated energyin all graphs. Flooding rates has been reduced in REEP, which contributes to the performance in terms of average delay in all graphs. The technique used to handle the fault tolerance issue in REEP, where the data packet are not getting lost in low energy nodes, contributes to the performance in terms of average data loss ratio in all graphs. Note that all the mentioned values in all graphs are the average of ten simulation runs.

4.4 Performance of Grid arrangement of nodes

In our experiment, the performances of DD and REEP are compared in terms of the average packet transmission, average dissipated energy, average delay, and average data loss ratio. We have simulated five different network sizes, with an increment of 100 nodes each time, ranging from 100 to 500 nodes. The sensor area has been generated by placing all nodes in network mode within a square area, and by scaling and keeping the communication range constant. If we do not keep the density constant, the performance will be affected by performance effect and as the result of increased network size which leads to increased network connectivity. The initial available energy at each node is 150 in cases of average packet transmission, average dissipated energy, average delay, and 100 nodes have been used to in case of average data loss ratio. In a grid system network, the values of average packet transmission (FIG 7 (a)) and average dissipated energy (FIG 7 (b)) are not affected by the increase network size.

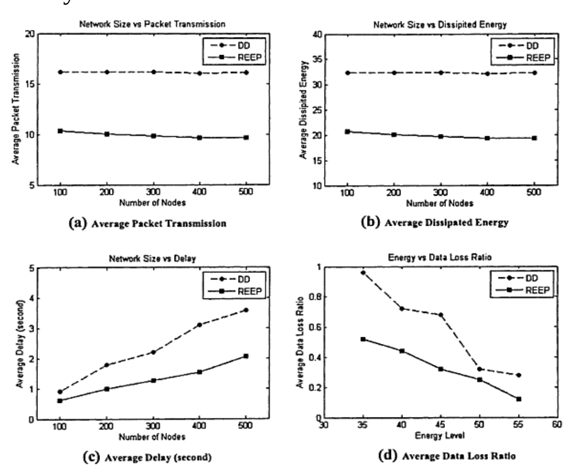


Fig 7: performance in grid arrangement

Since we have kept the communication range constant for every node, the number of neighbors for each node is the same in each increased size of network. Thus, the number of packet transmission remains almost the same at each node. Thus, the dissipated energy also remains similar in every network.

The average delay (FIG 7 (c)) increases with an increased network size because the hop count increases on the path. When there are 100 nodes in the network, the difference between DD and REEP is small in terms of average delay, but this difference increases with the increased number of nodes. This behavior is reflected inversely in terms of average data loss ratio (see FIG 7 (d)).

4.5 Performance in random arrangement of nodes

Node placement in random fashion can results in the FIG 8. We realize that with the increase in the number of nodes, the average packet transmission (FIG 8 (a)) and the average dissipated energy (FIG 8 (b)) will decrease. Also, the data values are smaller in this random arrangement network nodes compared to the grid arrangement case. Both protocols perform similarly in terms of average delay (FIG 8 (c)) in both grid and random systems leads to increasing of network size. The values of the average data loss ratio (FIG 8 (d)) are very close for both DD and REEP.

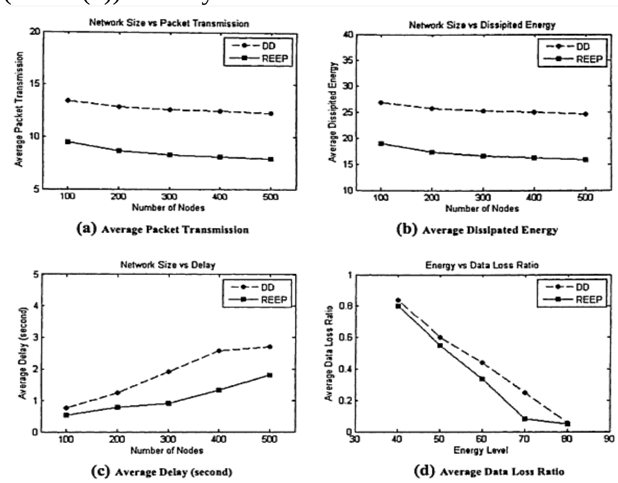


Fig 8: performance in random arrangement

4.6 Fault tolerance in REEP

FIG 9 shows the performance of REEP with and without the fault tolerance. We have simulated 100 nodes to show how data loss ratio varies with the increased energy level. When a node fails and fault tolerance issue is absent, then the alternative paths are not created and therefore data loss becomes higher. Data loss cannot be ignored in wireless communication, but can be reduced by implementing fault tolerance in routing protocols.

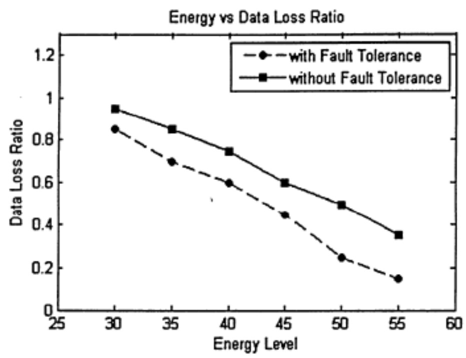


FIG 9: Fault tolerance in REEP

An analysis of our experimental results shows that REEP performs better than DD both with respect to energy efficiency and fault tolerance, for specified application types and scenarios. In order to maximize the lifetime of a wireless sensor network, energy resources of each individual sensor node must be spent in an effective way. In order to maximize the node's lifetime as well as the entire network lifetime, unnecessary flooding of control information should be avoided in REEP. The network density is another important parameter that can significantly alter the performance of a protocol. For example, a dense network introduces more connectivity among nodes with a large number of neighbors. We introduced constant density of a network. Results of our study have shown a clear view of the protocol performance in the studies usability scenarios. We believe that these performance criteria should be reflected in a similar manner in most of the usability scenarios. Most of the sensor network protocols are application specific, so are data- centric protocols. As long as we follow DD design structure, it is obvious that both DD and REEP are suitable for the same type of applications. But, for large network sizes REEP will work better than DD.

5. Conclusion

Although we have mentioned several times in our work that our proposal protocol has been designed following DD approach, but you should know that, there are som other design technique that highlighted as follows:

- The design idea of the first phase (i.e. sense event propagation) of REEP, in which all the sensor nodes sense for a specified time to scan the whole network for available objects.
- The design idea of the second stage (i.e. info event propagation) of REEP, that events related information are set after saving all necessary information from that event.
- Checking energy value during request event and data transmission.
- Maintaining FIFO queue for each task in every node. Although using this queue makes some looping problem, we overcome this problem by manipulating the queue in such a way that it removes specific neighbor information from the queue.
- A technique for reducing the data loss. In this technique, the data packet is sent to the next neighbor by changing the status of the package as negative. This technique has been validated through our experimental result.

Reference

[1] Manjeshwar and D. P. Agrawal. TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In 1st International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, April 2001

[2] T.N. Qureshi a, A.H. Khan ‘ Enhanced Developed Distributed Energy-efficient Clustering for Heterogeneous Wireless Sensor Networks The 4th International Conference on Ambient Systems, Networks and Technologies (ANT 2013), the 3rd International Conference on Sustainable Energy Information Technology (SEIT-2013)

[3] Changling Liu, Jörg Kaiser ‘A Survey of Mobile Ad Hoc network Routing Protocols’ University of Ulm Tech.Report Series, Nr. 2003-08

[4] Parul Saini, Ajay.K.Sharma, “E-DEEC- Enhanced Distributed Energy Efficient Clustering Scheme for heterogeneous WSN”, in: 2010 1st International Conference on Parallel, Distributed and Grid Computing (PDGC - 2010).

[5] Navid Nikaein, Houda Labiod,’ distributed dynamic routing algorithm for mobile ad hoc networks’institu Eurecom april 2000

[6] W. B. Heinzelman. “Application-Specific Protocol Architectures for Wireless Networks”. PhD thesis, Massachusetts Institute of Technology, June 2000

[7] L. Qing, Q. Zhu, M. Wang, “Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor network”, ELSEVIER, Computer Communications 29, 2006, pp 2230- 2237.

[8] K. Sadek , Z. Han, and K. J. Ray Liu, “A distributed relay-assignment algorithm for cooperative communications in wireless networks”, in Proc. of IEEE International Conference on Communications, Istanbul, Turkey, June 2006.

[9] L. M. Feeney and M. Nilsson, “Investigating the energy consumption of a wireless network interface in an ad hoc networking environment,” in Proc. of IEEE INFOCOM, Anchorage, AK, Apr. 2001.

[10] J. N. Laneman, D. N. C. Tse, and G. W. Wornell, “Cooperative diversity in wireless networks: efficient protocols and outage behaviour,” IEEE Trans. Inform. Theory, vol. 50, pp. 3062-3080, Dec. 2004.

[11] J. Elson and D. Estrin. “An Address-Free Architecture for Dynamic Sensor Networks”. Technical Report 00-724, Computer Science Department, USC, January 2000.

[12] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, “An ApplicationSpecific Protocol Architecture for Wireless Microsensor Networks,” IEEE Trans. Wireless Commun., vol. 1, no. 4, Oct. 2002, pp. 660–70.