EEQR: An Energy Efficient Query-Based Routing Protocol for Wireless Sensor Networks

Shohreh Ahvar[1], Mehdi Mahdavi[2] *

(1), (2) Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

s.ahvar@ec.iut.ac.ir; m_mahdavi@cc.iut.ac.ir

Received: 2011/07/11 ;Accepted: 2011/07/28 Pages: 25-38

Abstract

Routing in Wireless Sensor Networks (WSNs) is a very challenging task due to the large number of nodes, their mobility and lack of proper infrastructure. Since the sensors are battery powered devices, energy efficiency is considered as one of the main factors in designing routing protocols in WSNs. Most of energy-aware routing protocols are mere energy savers that attempt to decrease the energy consumption. But it is vital to balance the energy consumption in the network to avoid network partitioning. This paper proposes an energy efficient query-based routing protocol called EEQR. This protocol is able to operate as both energy saver and energy balancer. In addition, it is intended for contexts in which geographic routing criteria are not applicable. The results of EEQR simulations are compared with some well-known query-based routing protocols. Results show that EEQR achieves significant improvements in terms of energy saving and energy balancing.

Keywords: *Energy Aware, GlomoSim, Query-based, Sensor network*

1. Introduction

Sensor is a device that senses physical quantities such as pressure, temperature, humidity and converts them into continuous (analog) or discrete (digital) electrical quantities. In fact, the sensor is an electrical device that measures the chemical or physical changes and converts them into electrical signals. A sensor network consists of a large number of sensors which are spread throughout the environment to collect data. These sensor nodes are spread randomly which makes it possible to put them in dangerous or inaccessible places. WSNs have drawn considerable attention in recent years due to their potential applications [1]. These applications include monitoring active volcano eruption, monitoring the stability of dams, bridges and roads, monitoring sensitive areas of military battlefield, and planetary exploration projects such as lunar exploration to probe for water/ice. Since these nodes are randomly dispersed throughout the environment and are usually placed in harsh or inaccessible places, switching the nodes with depleted energy is not possible. Thus, the issue of energy consumption should be addressed in the design of routing protocols.

Routing protocols in sensor networks are generally divided into 3 categories based on the network structure; namely flat, hierarchical, and location-based [2]. In another classification approach, the protocols are classified into the followings categories based on the protocol operation: multipath-based, query-based, negotiation-based, QoS-based, and coherent-based [2].

While in flat networks all nodes play the same role and have the same importance, hierarchical protocols aim at clustering the nodes such that cluster heads are capable of data aggregation and reduction in order to save energy. Location-based protocols utilize the position information to relay the data to the desired regions rather than the whole network.

The second classification approach includes routing methods which vary according to the approach used in the protocol as follows: Multipath-based protocols maintain multiple paths between the source and the destination. In these protocols, network reliability can be increased at the expense of increased overhead of maintaining the alternate paths. Query-based protocols usually are intended for contexts in which geographic routing information are not available. In these protocols, the Station sends query packets to find specific events. Negotiation-based protocols use high-level data descriptors in order to eliminate redundant data transmissions through negotiation. QoS-based routing protocols, aim at data quality when delivering data to the BS through satisfying certain metrics, e.g., delays, energy, bandwidth, etc. In coherent routing, the data is forwarded to aggregators after short processing which typically includes tasks like time stamping, duplicate suppression, etc.

In recent years, various energy efficient routing protocols have been proposed to increase the networks' lifetime from different points of view [3,5]. Methods for increasing lifetime of sensor networks through designing energy aware routing protocols are divided into two general categories: reducing overall energy consumption and distributing energy consumption in sensor networks.

Most of energy aware routing protocols are designed to reduce overall energy consumption; however, it is vital to balance the energy consumption to avoid network partitioning. The main purpose of this paper is to increase lifetime of sensor networks through reducing overall energy consumption and distributing energy consumption in sensor networks. This paper proposes an energy aware query-based Routing algorithm named EEQR. It is intended for contexts in which geographic routing criteria are not applicable because a coordinate system is not available or the phenomenon of interest is not geographically correlated. Since the positioning systems consume energy, this feature helps to save energy. Also positioning systems can work outdoors and cannot work in the presence of any obstruction. Moreover positioning systems receivers are expensive and not suitable in the construction of small cheap sensor nodes.

The remainder of this paper is organized as follows: the next section reviews the previous works in the area of query-based energy aware routing protocols in WSNs. Section 3 introduces some useful definitions for better comprehension of the following sections. Section 4 describes our proposal for Routing algorithm. In section 5, we present the simulation details and present the numerical results for evaluation of the proposed method in section 6. Finally, section 7 concludes the discussion on EEQR.

2. Related work

In general, query-based algorithms use two methods to find event; namely, pull and push-pull methods. Pull method sends query to network to find event. Push-pull method, in addition to sending queries, gives notice of happening of an event as a node witnesses such event. Flooding the entire network with Query is the first attempt in pull method. In this method, each node sends query packet to all of its neighbors. This process continues until the TTL field of the packet expires or reaches the destination.

This method requires no routing algorithm and there is no need to maintain network topology; however, it tends to deplete network energy more quickly. This algorithm is useful when the number of events compared to the number of queries is very large. In improved versions of flooding algorithm, to reduce the number of transfers, each node sends packet only to one of its neighbors by a particular policy. As mentioned above, another mechanism used in query-based approaches is push-pull method. In this method, in addition to sending queries (pull), gives notice of happening of an event as a node witnesses such event (push). Push methods generally send a high-level description of data instead of the actual data. In this method, the nodes assign the high level descriptions to their relevant data. One of the first algorithms based on this method is Sensor Protocols for Information via Negotiation (SPIN) [2]. SPIN is a query-based and negotiation-based routing algorithm. In this algorithm, nodes send high-level descriptions of their data to their neighbors. After receiving data, each node will be able to send meta-data to its neighbors as well. In later versions of this algorithm, in order to consider the energy parameter, the algorithm is modified to incorporate the energy threshold. When a node becomes aware that its power is under the energy threshold, it does not run the protocol. For example, when a node receives new data, it drops the packet in case it does not have enough energy to complete the protocol. Because the meta-data are much smaller than the data, this method has small energy consumption. The main disadvantage of this method however, is that the nodes between the sender and the applicant should be interested in data. Push-pull methods use a combination of push and pull methods. Rumor is one of the known algorithms that are based on the query in which a mechanism is designed for combining push and pull methods [7]. The idea is to create paths leading to event whenever a node witnesses an event. In this method, when a query is generated, instead of flooding it throughout the network, it can be sent on a random walk until it finds the event's path. As soon as the query discovers the event path, it can be routed directly to the event. If the path is not found, the application can try re-submitting the query or flooding it, as the last resort. In Rumor, each node maintains a list of its neighbors as well as an events table. Whenever a node witnesses an event, it inserts its description into its event table. It also generates an agent. An agent is a long-lived packet which travels along the network and propagates information about local events to distant nodes. It contains an events table, similar to that of nodes, which is synchronized with every visited node. The agent travels the network for some number of hops and then destroys. Any node may generate a query which then should be routed to a particular event. If the node has a route to the event, it will transmit the query; otherwise, it will forward the query in a random direction. This process continues until the query TTL expires, or the query reaches a node which has observed the target event or the event path. If the node which had originated the query determines that the query did not reach its destination, it would try retransmitting, giving up, or flooding the query.

The next research efforts to improve the performance of these protocols have been made in recent years. These researches are based on the assumption that the probability of two lines intersecting in a bounded rectangular region is more than that of two cursive lines [7].

The first attempts to improve Rumor routing based on this method are Straight Line Routing (SLR) [8] in 2005 and Directional Rumor routing (DRR) [9] in 2007. They try to keep the routing path straight through using geographical system. They aim at propagating the agent path and the query path in straight lines centered at the source

point and the sink point. In DDR, when a node senses an event, it makes a number of event agents and propagates them through the network along some linear paths, forming a star-like propagation trajectory. In other words, if there are N agents numbered as $i \in \{1,...N\}$, they try to route the agent in a line coming out from the source node with angle $\theta_i = (i-1) \times \frac{2\pi}{N}$. If there is not such a node, the packet is misrouted to a neighbor which is the farthest from the source node. The query path is generated in this way as well. Suppose that in SLR algorithm, node *b* is the newest hop, node *a* is the pre-hop of node *b*, (obviously, the direction of the path is from *a* to *b*) and the distance between them is $R/2$, where R is the radio distance. It defines the first dimension as the distance from node *a* and the second dimension as the distance from node *b*. It is clear that the node whose location is $(R, R/2)$ or near$(R, R/2)$ is the most suitable node to be the next hop.

In 2009, DRR [10] was improved in two respects. First, a high performance geographical routing was used after locating the source of the event. This improves the overall performance of the routing protocol especially in high volumes of data transfer. The latter improvement proposed a method to eliminate the location information devices.

Later, Zonal Rumor Routing (ZRR) was proposed [11]. ZRR algorithm enables the Rumors to be spread to a larger part of the network with high energy efficiency by partitioning the network into different zones. It improves the query delivery percentage and requires fewer transmissions, thus reducing the total energy consumption in a sensor network.

The other problem in Rumor is stopping the paths by the network boundary. In 2009, Appointment Rumor Routing (ARR) was designed in [12]. In order to find the event, sink propagates one query agent that moves like an event agent except that it is not stopped by the network boundary. Query agent tries to rotate over the network edge by walking on the boundary nodes until hitting the event agent. The researchers continued to work in this area in 2010 in papers as [13] and [14]. Furthermore, Clustering Rumor Routing Protocol (CRR) was proposed in 2010[15]. The advantage of this method is that it effectively avoids winding. The main drawback of CRR however, is the use of clustering structure, selective next-hop scheme, and double variable thresholds for rotation of cluster-heads which lead to enormous energy wasting [16]. The next improvement of Rumor in 2010 was Novel Rumor Routing (NRR). It is designed based on the Ant Colony Optimization (ACO) technique and attempt to resolve the loop route established by Rumor routing. One of the most recent efforts in this area was proposed in 2011 [17]. It proposes a new version of appointment Rumor routing. The main difference of this method with ARR is that it set the appointment in the center of network instead of the edge.

Most of the proposed query-based routing protocols assume that the sensors are either equipped with global positioning system (GPS) receivers or use some localization technique to learn their locations [20] while the EEQR is intended for contexts in which geographic routing criteria are not applicable because a coordinate system is not available or the phenomenon of interest is not geographically correlated. EEQR saves energy in the same manner as Rumor.

3. Definitions

In this section definitions which needed in algorithm design are presented.

Def.1: Query. A query is a request packet for receiving information on a particular event. Each query is associated with a time to live (TTL) determines the number of hops the query can make. it also maintains a list of visited nodes in a history list [11].

Def.2: Agent. An agent is a packet that is responsible for spreading rumors about the events in the network. Each agent is associated with the time to live (TTL) that determines the number of hops the agent can traverse before it dies. Also An agent maintains a history list [11].

Def.3: Event List. This list stores the event names and the distance to the events. Agent nodes maintain their respective event lists. Each node maintains an event list.

Def.4: Neighbor List. Neighbor list includes a list of ID, energy level and time of receipt of Hello packet from neighbors. Each node maintains a Neighbor list.

Def.5: History List. The history list stores the node ids of the visited nodes.

Def.6: Energy Consumption: The energy consumption means average energy consumption of each node for routing by each routing protocol.

Def.7: An event is an abstraction, identifying anything from a set of sensor readings to the node's processing capabilities.

Def.8: energy threshold (ET_i) and Distance threshold (D_i): each sender node (e.g. node "i"), computes its energy and Distance threshold according to equations 1 and 2.

$$ET_i = \frac{\sum_{allj} E_j}{N} \quad (1)$$

$$D_i = \frac{\sum_{allj} T_j}{N} \quad (2)$$

"j" is a neighbor node of node "i" and E_j is it's energy level. N is number of node "i" neighbor nodes and T_j is receipt time of "Hello" packet from node "j".

4. Proposing routing algorithm

EEQR routing protocol is an energy aware routing protocol that consists of three major steps:

- Neighbors discovery
- Agent forwarding
- Query forwarding

4.1 Neighbors discovery

At the beginning, each node simply broadcasts its id and energy level by a Hello packet. When each node receives the Hello packet, it adds an entry to its "neighbor list", including "neighbor id", "energy level" and "receipt time" to its "neighbor list". Let us define I, J and K as follows.

$$\begin{cases} I = \{i \mid i \in \text{Neighbors of current node}, E_i \geq ET_{\text{current node}}\} \\ J = \{j \mid j \in \text{Neighbors of current node}, T_j \geq D_{\text{current node}}\} \\ K = \{k \mid k \in I \cap J\} \end{cases}$$

"I" shows the set of nodes whose energy levels are above energy threshold. "J" shows the set of nodes whose receipt times are above distance threshold. "K" shows nodes whose energy levels are above energy threshold and receipt times are above distance threshold.

4.2 Agent forwarding

When a node witnesses an event, it creates an agent packet and forwards it to the next hop based on forwarding policy described in section 4.4. agent forwarding is shown in figure 1.

```
1)    Start
2)    If sensor NODE detects an event
3)      Update the node's event table based on event information
4)      Create agent pkt with hop count=1
5)      While (Hop count ≤ agent-path-Hop count)
6)        Hop count++
7)        Update the node's event table based on event information
8)        Mark the node as agent node
9)        I = {i| i ∈ Neighbors of current node, E_i ≥ ET_current node}
10)       J = {j| j ∈ Neighbors of current node, T_j ≥ D_current node}
11)       K = {k| k ∈ I ∩ J}
12)       If (K = ø)
13)         Select a node from I randomly as the next hop
14)       Else
15)         Select a node from K randomly as the next hop
16)       End if
17)       Forward agent packet to the next hop
18)     End While
17)   End if
18)   End
```

Figure 1. Agent send phase pseudo code

The receiver node inserts information of the Agent packet into its "Event List" (line 7 in figure 1). Then, it regenerates an Agent packet and forwards it to the next hop based on the forwarding policy (line 9-16 in figure 1). This procedure will continue until N hops, where N represents the length of the Agent path.

When a packet is sent to a particular destination, all nodes in its transmission range will be received. Until now there is not any mechanism that explains how some of these packets would not receive. It should be noted that all neighbor nodes receive the Agent packet by overhearing technique [18]. They only extract information about the witnessed event from the Agent packet, insert it into their Event Lists, and ultimately drop it. As such, they are known as nodes of the agent traversal path.

4.3 Query forwarding

To collect data from the network, when the geographic information is available, the best route is the shortest path which is obtained with the help of Geographic Information and there is no need to broadcast. But if the nodes in the spatial information are not available, queries are distributed across the network to discover the best route to event. When the query reaches its destination, the data can be sent to the producer of query.

Any node may generate a query, Def.1, which should be routed to a particular event. If the node has a route to the event, it will transmit the query. If it does not, it will forward the query-based on the forwarding policy (section 4.5). This continues until the query TTL expires, or until the query reaches a node that has observed the target event.

In EEQR, the query is created a maximum of 3times. Each path starts from the origin. If the first route fails to find event, the second path and then the third path are created. If data are never returned to the origin after a specified time, the path has failed. Finally, if the third path does not find the event, the query will be broadcast on the network (lines 38-40 in figure 2). Query forwarding is shown in figure 2.

```
1)  Start
2)  Query source Create query pkt with hop count=1
3)  While (number of Query send< 3 and event is not found)
4)    While (Hop count ≤ query-path-Hop count)
5)      Hop count++
6)      Search query source in its neighbors
7)      If (event detector node has been found in neighbors)
8)          Forward the Query pkt to the event detector node
9)      End if
10)     If (Only one agent node has been found in neighbors)
11)         Forward the Query pkt through the Agent path to event detector node
12)     End if
13)     If (find more than 1 agent node in neighbors)
14)         I = {i| i ∈ Neighbors of current node, E_i ≥ ET_current node}
15)         J = {j| j ∈ Neighbors of current node, T_j ≥ D_current node}
16)         K = {k| k ∈ I ∩ J}
17)          If (K = ø)
18)             Select a agent node by min hop count to event from I
19)         Else
20)             Select a agent node by min hop count to event from K
21)         End if
22)          Forward the Query pkt through the Agent path to event detector node
23)     End if
24)     If (there is no agent node in neighbors)
25)         I = {i| i ∈ Neighbors of current node, E_i ≥ ET_current node}
26)         J = {j| j ∈ Neighbors of current node, T_j ≥ D_current node}
27)         K = {k| k ∈ I ∩ J}
28)          If (K = ø)
29)             Select a node from I randomly as the next hop
30)         Else
31)             Select a node from K randomly as the next hop
32)         End if
33)         Forward query pkt to the next hop
34)     End if
35)   End While
36)   Query send++
37) End While
38) If (event is not found)
39)     Broadcast query in network
40) End if
41) End
```

Figure 2. Query send phase pseudo code

4.4 Agent forwarding policy

The policy of choosing between the neighbors in order to forward agent packet can be implemented randomly or it can choose nodes with the most remaining energy or other methods can be used. If it always selects a particular node from its neighbors, the

energy of this node will drain faster than the rest of the nodes. So it is better to select a set of the best options. Then, among these, a node is randomly selected as the next step. For example, if anode with the highest residual energy between neighbors is always selected, the energy of this node will drain faster than the rest of the nodes. So, the number of nodes with more remained energy should be chosen under a particular policy. Then, among these, a node is randomly selected as the next step. For this, affixed threshold can be used in all steps. This method can somewhat balance energy consumption in the network. But if there was no node with this profile in neighbors, Performance of this algorithm is reduced. So the best solution is dynamically selecting threshold in each node (each step). EEQR selects agent next hop based on energy level and distance of neighbors to prevent network partitioning and increase agent path to improve time of finding path.

In the proposed method, each sender node computes its energy and Distance threshold. The energy threshold of each sender node is average energy level of its neighbors. Since the hop count between the node and its neighbors is one, the delay of returning back the packet from node's neighbors is caused only by the distance. So the average of the received time can be used instead of Distance threshold.

Then, the next hop is selected randomly from nodes whose energy levels are higher than the energy threshold and their received times are above distance threshold. After that, the agent is forwarded to the selected next hop (lines 9-15 in figure 1).

Since the overhearing technique is used in the EEQR, if the farther neighbor is selected, the path length will be larger and will visit more nodes. So the delay of finding event is decreased and the probability of finding event is increased.

4.5 Query forwarding policy

When a packet is sent to a particular destination, all nodes in its transmission range will receive the packet. There is no mechanism for not receiving that packet. It is called overhearing. In the process of sending query packet, all nodes in transmission range will receive the query packet but drop it. Only the destination of the packet processes it. Event witnessed node or agent path nodes may be among the overhearing nodes and the algorithm will pass by but not visit them. Although searching for overhearing nodes increases the amount of data delivery, energy consumption and delay should not be increased. In EEQR, at first, the query sender node broadcasts a one-hop Information Request (INFREQ) packet to find if there is agent or event witnessed node or not.

- (first mode) If it receives one reply from the event witnessed node, it selects that node as a next hop (lines 7-9 in figure 2).
- (second mode) If it receives one reply from an agent node, it selects the node as a next hop (lines 10-12 in figure 2).
- (third mode) If it receives replies from some agent nodes, it computes average energies and distances (delays) of the agent nodes. Then, it selects an agent node whose energy and delay are above the average and also its hop count from event mode is the lowest (lines 13-23 in figure 2).
- (fourth mode) If there is not any agent or event node in its neighborhood, it selects the next hop according to the Agent forwarding policy (section 4.4) and forwards to it (lines 24-34 in figure 2).

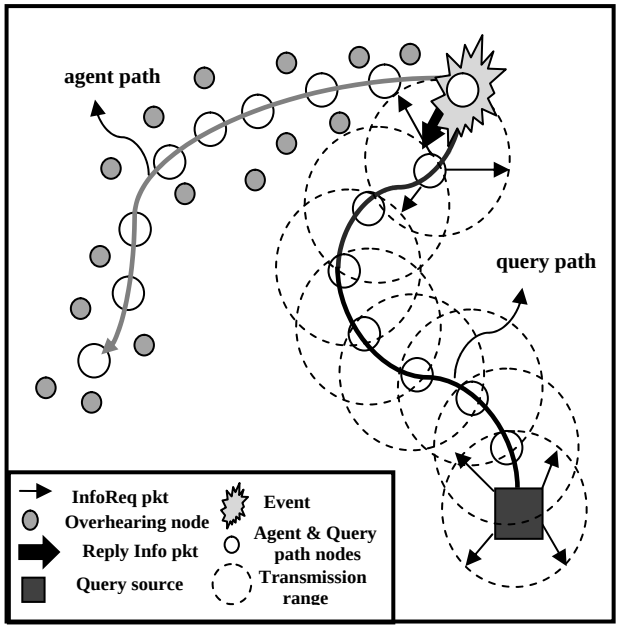

Figure 3. The query finds the event detector node

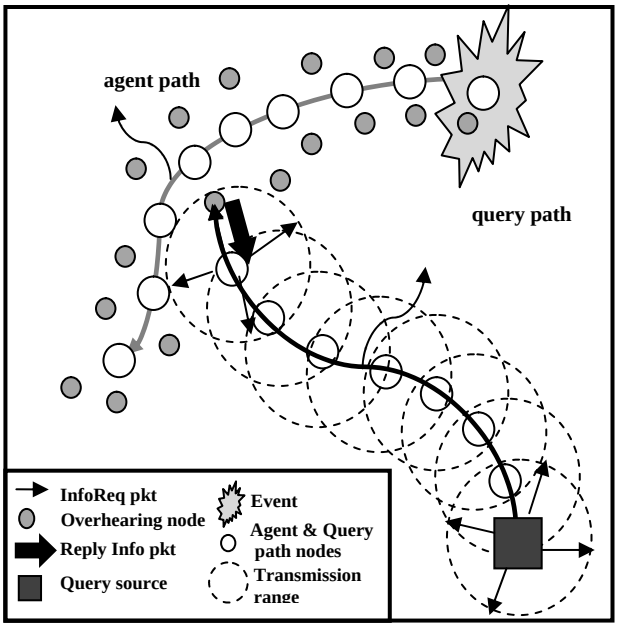

Figure 4. The query finds overhearing node in agent path

5. Simulation

To compare the routing protocols, a parallel discrete event-driven simulator, GloMoSim, was used. GloMoSim (Global Mobile Information System Simulator) is a simulation tool for large wireless and wired networks [19].

Since EEQR combines pull and push methods, it will be compared with Rumor, DRumor and Flooding algorithms. To evaluate the proposed algorithm, several experiments have been simulated. In all experiments, the network size is 1000m× 1000m, MAC layer protocol is 802.11, agent and query hop counts are 10 steps and the

number of nodes is 400. Also by changing seed, five topologies were generated. Moreover, the placement of the sensors in the train was randomly chosen. It is worth highlighting that, even though the placement of the routes was random, once it was set it remained fixed for the rest of trails to obtain comparable result across experiments.

5.1 Experiments

In order to examine two factors that improved the algorithm, two groups of tests were designed. The first batch of experiments shows the success rate in reducing the number of sending queries for finding event. Reducing energy consumption is also measured in this way. Second group of tests is designed to evaluate the performance of energy distribution on the network and increase network lifetime.

First group of experiments:

At the beginning of simulation, due to the number of nodes, 100 nodes are randomly selected as query source or the Station to send query packets. Since the process of finding event and returning back the data to query source take a maximum of 50 minutes, every 50 minutes, an event is detected by a randomly selected node. So100 events are detected. In order to compare the number of sending queries for finding event, three charts are displayed.

First chart: This chart shows the percent of finding event in the first time of sending query.This parameter shows which algorithm can find the event faster by a fewer number of query sending and less energy consumption. This chart is shown in figure 5.

Second chart: In this chart, the percent of finding event up to 3 times query sending is shown. All three algorithms, Rumor, DRumor and EEQR broadcast the query in the network if they are unable to find event in the maximum of 3 times of sending query. The algorithm which needs fewer flooding queries for finding event saves more energy compared to the others.

Third chart: This chart compares the number of flooding queries. It shows the Number of the queries which find event in the first, second and the third posts, as well as the flooding query that represents the total number of query posts. Therefore, this diagram is complementary to the previous chart. The diagram is shown in Figure 6.

Second group of experiments:

Test1: The number of Events in this experiment and the simulation continue until the energy of 20% of the nodes will finish. When energy of 20% of the nodes finishes, the performance of network will not be acceptable.

First chart: The time during which 2, 4,…, and 20% of nodes fails is shown in this chart. The time of failure node is related to energy management. Energy balancer protocols should have better node failure time than the other protocols. Therefore, this test evaluates energy management of each protocol. The diagram in Figure 7 is shown.

Test 2: At the beginning of simulation, due to the number of nodes, 100 nodes are randomly selected as Query source or the Station to send query packets. Since the process of finding event and returning back the data to Query source takes a maximum of 50 minutes, every 50 minutes, an event is detected by a node that is randomly selected. So, 100 events are detected. In order to compare the number of sending queries for finding event, three charts are displayed.

The main goal of EEQR is increasing lifetime of network. So EEQR tries to balance and reduce energy consumption in network. In order to compare the energy balancing, at the end of the simulation, the variance of energy consumption in the network is shown.

The diagram in Figure 8 is shown. Also, in order to compare the reduction of energy consumption, the average total energy consumption of network nodes is displayed. The diagram in Figure 9 is shown. The algorithm that has the lowest average and variance energy consumption increases the network lifetime further.

6. Result

The results of simulations in Figures 5 to 10 are shown.

The results of first group experiments:

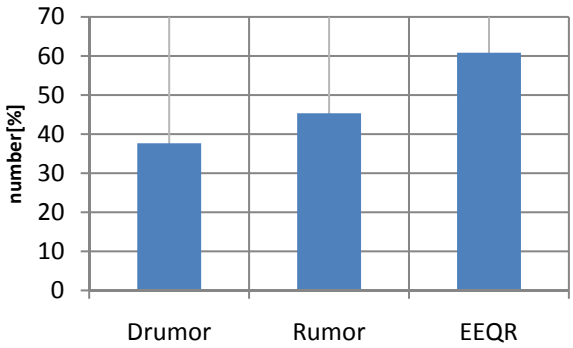

Figure 5. The Percentage of Events found in the first query forwarding

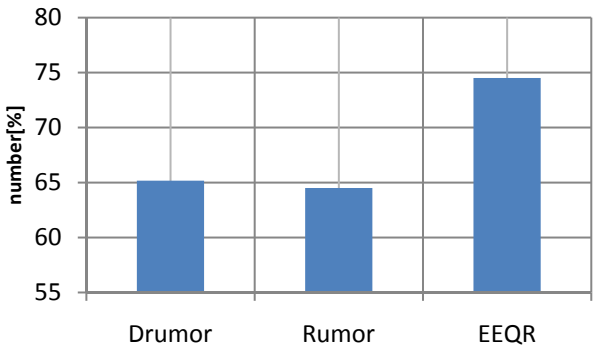

Figure 6. The percent of finding event in max 3 query sending

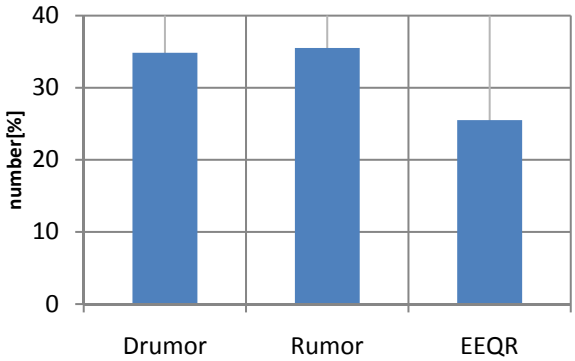

Figure 7. The percent of flooding query

Figure 3 shows the results of Experiment 1. With regard to the figure 5, the average of 5 topologies shows that the proposed algorithm raised the percentage of success in the first query sending and it leads to a lower number of query sending. Thus, it reduces energy consumption further. Figure 6 and Figure 7 show the proposed algorithm. It shows the fewer flooding queries and thus further reduction of energy consumption.

The results of second group of experiments:

Test1: Figure 8 shows that the average energy consumption of our algorithms is lower than the other algorithms available in the literature. This reduces energy consumption and increases network lifetime. Also, this experiment shows that the speed of finding event without increasing energy consumption could be increased. According to figures 9 and 10, algorithm further increases the network lifetime.

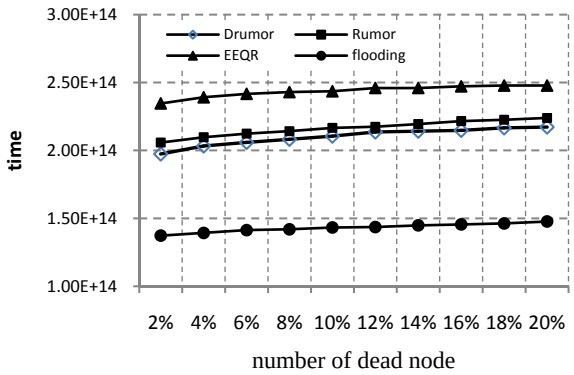

Figure 8. times until nodes fail

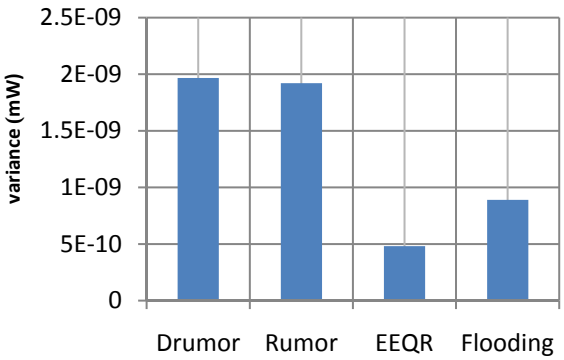

Figure 9. The variance energy consumption of all nodes in the network, at transmission mode

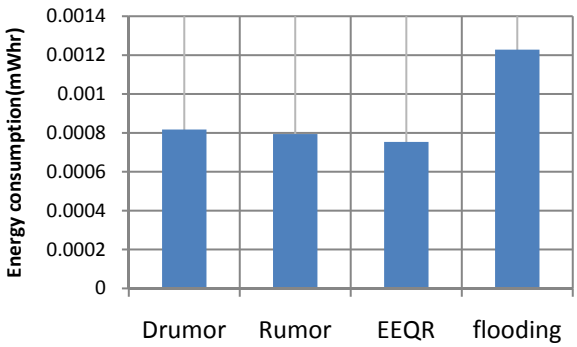

Figure 10. The average energy consumption of all nodes in the network, at transmission mode

7. Conclusion

One of the main challenges in sensor networks is increasing their lifetimes. Many studies have been done to increase the lifetime of these networks. One method for increasing the lifetime of sensor networks is designing energy-aware routing algorithms. However, many energy-aware algorithms in these networks are designed to reduce energy consumption. Balancing energy consumption should also be considered. This paper presented and evaluated a query-based routing protocol. The results of EEQR simulations were compared with some well-known query-based routing protocols. Simulation results showed that the proposed protocols improve network lifetime compared to other query-based routing algorithms in sensor networks.

8. References

[1] S. K. Singh, M.P. Singh, and D.K. Singh, "Applications, Classifications, and Selections of Routing Protocols for Wireless Sensor Networks", International Journal of Advanced Engineering Sciences and Technologies (IJAEST), vol. 1, issue no. 2, pp. 85-95 (2010).

[2] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey", IEEE Wireless Communications, Volume 11, No. 6 (2004).

[3] E. Ahvar, A. Pourmoslemi, M. J. Piran, "FEAR: A Fuzzy-based Energy-Aware Routing Protocol for WSN ", International Journal of grid computing and applications, Volume.2, No.2, pp 32-45 (2011).

[4] E. Ahvar, M. Fathy, "BEAR: A Balanced Energy-Aware Routing Protocol for Wireless Sensor Networks", Wireless Sensor Network Journal, vol. 2, pp. 768–776 (2010).

[5] S. Fedor, "Cross-Layer Energy Optimization of Routing Protocols in Wireless Sensor Networks", Phd thesis, Dublin City University (2008).

[6] M. Zuniga, Ch. Avin and M. Hauswirth, "Querying Dynamic Wireless Sensor Networks with Non-Revisiting Random Walks", 7th European Conference on Wireless Sensor Networks (EWSN), Coimbra, Portugal (2010).

[7] D. Braginsky and D. Estrin, "Rumor Routing Algorithm for Sensor Networks", In Proc. First ACM Workshop on Sensor Networks and Applications, Atlanta, GA, USA, pp. 22–31 (2002).

[8] C Chou, J. J. Su, and C Chen, "Straight Line Routing for Wireless Sensor Networks", 10th IEEE Symposium on Computers and Communications, pp. 110-115 (2005).

[9] H. Shokrzadeh, A. T. Haghighat, A. Nayebi, "Directional Rumor Routing in Wireless Sensor Networks", Third IEEE International Conference in Central Asia on Internet The Next Generation of Mobile, Wireless and Optical Communications Networks (ICI) (2007).

[10] H. Shokrzadeh, A. T. Haghighat, A. Nayebi, "New Routing Framework Base on Rumor Routing in Wireless Sensor Networks", Computer Communications journal (ComCom), Vol 32**.** Page 86-93 (2009).

[11] T. Banka, G. Tandon, and A. P. Jayasumana, "Zonal Rumor Routing for Wireless Sensor Networks", In Proc. IEEE International Conference on Information Technology: Wireless Ad Hoc/Sensor Networks and Network Security (ITCC), Las Vegas, NV (2005).

[12] H. Shokrzadeh, M.N. Fesharaki, " ARR: Appointment-base Rumor Routing in Wireless Sensor Networks", The 5th International Conference on Information & Communication Technology and Systems (ICTS). Surabaya, Indonesia (2009).

[13] H. Shokrzadeh, P. Saadatmandy, N. Forouzideh, A. Broumandnia, "Single-Link Serial Directional Rumor Routing in Wireless Sensor Networks", International Conference on Information and Network Technology (ICINT). Shanghai, China (2010).

[14] H. Shokrzadeh, A.M. Rahmani, A. T. Haghighat, N. Forouzideh, "SDRR: Serial Directional Rumor Routing in Wireless Sensor Networks", International Conference on Networking and Information Technology (ICNIT). Manila, Philippines (2010).

[15] Z. Wang et al, "Energy efficient clustering rumor routing protocol for wireless sensor networks", workshops on Ubiquitous, Autonomic and Trusted Computing, china (2010).

[16] Y. Cui et al, "A Novel Rumor Routing for Wireless Sensor Network", 4th Conference on Genetic

and Evolutionary Computing (2010).
[17] H. Shokrzadeh, A.T. Haghighat, P. Saadatmandi, "Rumor Routing by Appointment in Center Of Gravity in Wireless Sensor Networks", The International Conference on Information Networking(ICOIN) ,Malaysia (2011).
[18] S. Biswas, S. Datta., "Reducing overhearing energy in 802.11 networks by low-power interface idling", IEEE PCC (2004).
[19] X. Zeng, R. Bagrodia, and M. Geria, "Glomosim: A Library for Parallel Simulation of Large Scale Wireless Networks", Proceedings of the 12th Workshop on Parallel and Distributed Simulations, pp. 154-161 (1998).