

An Efficient Energy Aware Routing Protocol for Real Time Traffic in Wireless Sensor Networks

Amir Hossein Mohajerzadeh

Department of Computer Engineering, Ferdowsi
University of Mashhad, Iran
e-mail: ah.mohajerzadeh@stu-mail.um.ac.ir

Mohammad Hossein Yaghmaee

Department of Computer Engineering, Ferdowsi
University of Mashhad, Iran
e-mail: hyaghmae@ferdowsi.um.ac.ir

Abstract—Wireless Sensor Networks (WSN) have recently been extensively deployed and researched. They are composed of a high number of small and simple nodes where most of them have to function as a router in an ad hoc manner. Because of limited energy sources in sensor network node, routing protocols should save the energy as much as possible. Energy consumption has a direct influence on network lifetime. From the Quality of Service (QoS) point of view, in many applications such as real time one, it is necessary to consider application QoS requirements. In this paper we propose an energy aware routing protocol for real time traffic in wireless sensor networks. The proposed protocol considers both energy and delay metric to find an optimal path with minimum energy consumption and minimum end to end delay. Simulation results show that the proposed protocol is successful in low energy consumption and satisfying low end to end delay which makes it suitable for real time applications.

Keywords- Energy aware routing protocols; Real time traffic; Wireless sensor networks; Quality of service

I. INTRODUCTION

In the recent years, many researches have been conducted on wireless sensor networks. A wireless sensor network consists of sensor nodes that communicate with each other using wireless links. Wireless sensor networks contain hundreds or thousands of sensor nodes that can both send and forward data [1, 2]. The WSNs are used to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations. Each node in WSNs consists of different parts including: sensors, processor unit (usually a small microcontroller), energy source (usually a battery) and communication unit. The communication part of a sensor node uses wireless communication devices, to be able to send and forward data using a wireless link.

During past few years, WSNs have found many different applications. Typical applications of WSNs include monitoring, tracking, and controlling. Some of the specific applications of WSNs are: habitat monitoring, object tracking, nuclear reactor controlling, fire detection and traffic monitoring. Small sensor nodes could also be used for medical applications [3], e.g., for the surveillance of

elderly people. In this application, sensor devices monitor vital function and report them to the family doctor or directly to the ambulance in case of an emergency like a heart attack. Some sensor nodes could also be implanted into the body in order to detect diseases like cancer in an early. WSNs are also used in commercial and industrial applications to monitor data that would be difficult or expensive to monitor using wired sensors. In the field of home automation [4] sensor nodes could be located in every room to measure the temperature. Sensor nodes could at the same time monitor more than only temperature. They could also detect movements within rooms and report this information to the alarm equipment in case of absent occupants. Sensor networks have also been used for many real time applications. Each application has unique QoS requirements [5]. For example, real time applications need low delay in data delivery [6]. Sensor network's protocols should run appropriate algorithm to satisfy application QoS requirements. Routing protocols also should use appropriate algorithm to find routes with ability to satisfy application QoS requirements. The sensor nodes in WSNs have many limited sources of energy and computing. The main constraint of these networks is the amount of energy consumption. The lifetime of a sensor network depends on its node's energy. In most of sensor networks there is no way to charge node's battery; therefore efficient use of available energy sources is essential. With respect to all above mentioned points, the protocols in wireless sensor network should consider energy constraint in all network's layers. Also routing protocols should use efficient algorithms that consume energy optimally. Due to the inherent characteristics that distinguish WSNs from other networks, routing in wireless sensor networks is very challenging [7]. Some of the routing challenges and design issues that affect the routing process in WSNs are: node deployment, energy consumption without losing accuracy, data reporting method, node/link heterogeneity, fault tolerance, scalability, network dynamics, transmission media, connectivity, coverage, data aggregation and quality of service.

Recently, many new routing protocols have been proposed for the routing in WSNs. These routing

mechanisms have taken into consideration the inherent features of WSNs along with the application and architecture requirements. The task of finding and maintaining routes in WSNs is nontrivial since energy restrictions and sudden changes in node status cause frequent and unpredictable topological changes. To minimize energy consumption, energy aware routing techniques have been proposed in the different literatures. They employ some well-known routing tactics such as data aggregation, in-network processing, clustering, different node role assignment and data-centric methods. In real time applications, data should be delivered within a certain period of time from the moment they are sensed, or it will be useless. Therefore, bounded latency for data delivery in real time applications is too important. However, in many applications, conservation of energy, which is directly related to network lifetime, is considered relatively more important than the quality of data sent. As energy is depleted, the network may be required to reduce the quality of results in order to reduce energy dissipation in the nodes and hence lengthen the total network lifetime. Hence, energy-aware routing protocols are required to capture this requirement.

In this paper an efficient energy aware routing protocol for real time traffic in wireless sensor networks has been proposed. The proposed routing protocol is energy aware so its main goal is to consume energy optimally. The proposed routing protocol can find the best route which not only has the optimal energy consumption but also has the minimum end to end delay. The routing algorithm in the proposed protocol, considers a cost function which helps the algorithm to assign a cost to each route. This cost function could be determined based on the application requirements. The proposed algorithm finds the best route depends on its cost. By using a cost function, the proposed routing algorithm selects an optimal route with possible lowest cost. The cost function is based on energy consumption and end to end delay. The end to end delay consists of transmission delay and queuing delay. In the proposed algorithm, there is an attempt to minimize end to end delay by minimizing transmission delay. As transmission delay is directly related to the route length, the minimum transmission delay can be achieved by minimizing route length between source and sink nodes. The proposed routing protocol uses a neighbor discovery algorithm to find its neighbors uniquely. As most of routing algorithms need to send data to a specific neighbor, neighbor discovery is very important. The proposed neighbor discovery algorithm uses three input parameters includes: node identifier (ID), received signal strength and a random number. Simulation results show that it can discover the neighbor uniquely.

The reminder of this paper is organized as follow. The second section, discusses the related researches in this field. Section 3 describes the proposed energy aware routing protocol in details. In section 4 the proposed neighbor discovery mechanism is explained. Section 5 is dedicated to the simulation results. Finally section 6 concludes the paper.

II. RELATED WORKS

During past few years, many routing protocols have been developed for wireless sensor networks. As the energy is an important constraint of the WSNs, so the energy aware routing algorithms are too important. In the rest of this section we review some of the general routing protocols proposed for wireless sensor networks. Directed Diffusion [8] is a well known routing algorithm for wireless sensor networks. This algorithm is not complicated and directly diffuses the data related to sensor nodes. This procedure guarantees high data delivery rate and low delay for communications. Directed Diffusion consumes more energy to forward and receive redundant data. Sink sends interests to each network nodes and determines their job. When a node senses an event, it sends appropriate event related information to sink. SPEED [9] is a well known algorithm for transmitting real time traffic in wireless sensor networks. It considers energy consumption in its routing procedure. SPEED is a highly efficient and scalable protocol for sensor networks where the resources of each node are scarce. SPEED can be used in both data link and network layers. It is a flat routing algorithm. By guaranteeing data forwarding rate, it can support real time communications. It acts locally and uses neighbor information for routing. SPEED uses interesting mechanism layer for route maintenance and recovery that uses both data link and network layer. Reactive Energy Decision Routing Protocol (REDRP)[10] is another routing algorithm for WSNs that its main goal is optimal energy consumption. This algorithm attempts to distribute traffic in the entire network fairly. Using this mechanism, it decreases total network energy consumption. REDRP is routing reactively, and uses residual node energy in routing procedure. It uses local information to routing, but nodes have a global ID which is unique for the entire network. This algorithm is divided into 4 steps. In the first step, sink sends a control packet to all network nodes. The nodes estimate their distance to sink relatively by using this packet. Next step is route discovery. Routing is performed on demand in REDRP. This means that the routes are established reactively. After route establishment in route discovery step, data are forwarded to sink by using those routes. In route recovery step if a route is damaged, it will be recovered or a new route will be established. Real time Power Aware framework (RPTAW) [11] considers both energy and QoS metrics. This algorithm acts hierarchically. By changing cluster structure and creating new node which is called Relay Node that its job is forwarding information from cluster to sink, its goals are achieved. This algorithm claims that by using data aggregation functions, energy consumption is reduced. Furthermore it can manage quality of service depending on efficiency of routing protocol used. In [12-15], different reliable energy aware routing protocols for wireless sensor networks have been proposed.

III. PROPOSED PROTOCOL

In this section, we describe the proposed energy aware routing protocol in details. The proposed protocol uses a

flat routing algorithm [7] which is done proactively. This means that the routes are established before traffic transmission. The algorithm is run to find the least cost route between the source and sink. The proposed routing algorithm is divided into 3 phases which are: route discovery phase, data transmission phase and route recovery phase. The last phase is only done when the topology has been changed. Each node has a unique identifier (ID) which is determined in the route discovery phase. The nodes also have a routing table which includes 3 fields: ID, signal strength and route cost. There is a record for each neighbor of a node in its routing table. The routing table is created in the route discovery phase. This table is used in data transmission phase to send traffic from source to sink. In the following subsections, we describe the functions of each phase in details.

A. Route discovery phase

The sink node as the initiator of this phase broadcasts a packet to all its neighbors. This packet is called Route Discover packet. The structure of Route Discover packet is shown in figure 1. As shown in this figure, each Route Discover packet consists of three fields which are: message type, sender ID and best route cost. The message type field determines the type of packet. The sender ID field determines the value of sender's ID. The best route cost field determines the cost of optimal route between sender node and sink.

Message type	Sender ID	Best route cost
--------------	-----------	-----------------

Figure1. Structure of Route Discover packet

Usually the value of sender ID field in all Route Discover packets which are sent by the sink node is equal to zero. As the cost of optimal route between sink node and itself is always zero, so the value of best route cost field is also equal to zero.

After receiving the Route Discover packet, each node follows these steps:

1. The node increments the value of sender ID field in the received packet and compares the result with its ID. If the result is bigger than the node's ID, the received packet is dropped. Otherwise the node's ID is replaced by the value of the result. When the node doesn't have any ID, the node's ID is equal to: sender ID +1. If packet is accepted the steps are continued as follows:
2. The node creates a new record for new received packet in its routing table. The ID field of the routing table is set to the value of sender ID field of the received packet.
3. The forwarding cost of packet which is sent directly from node i to j is calculated using the following cost function:

$$Cost_{ij} = \alpha.F(distance_{ij}) + \beta.G(energy_j) \quad (1)$$

where F, G are two functions which their inputs are equal to the distance between node i and j ($distance_{ij}$) and the residual energy of node j ($energy_j$), respectively. Furthermore, α, β are two constant coefficients. By adding the value of best route cost field in the received packet with the value of $Cost_{ij}$, the node could be able to obtain the new value of route cost. Each node to transmit its data toward sink, selects the optimal route which has least cost. If the new discovered route has a lower cost than the existing least cost route, the node replace the new discovered route as its best route to the sink. In this case, the sender of packet is chosen as the next hop node. As the routing strategy is hop by hop, so each node only stores information about its next hops.

4. To determine the distance between sender and receiver, the signal strength of received packet is measured. This value is stored in the signal strength field of the routing table. Using the signal strength, the distance between two nodes could be determined. Furthermore, using the distance, the transmission delay between two nodes is obtained. The end to end delay is determined using the transmission delay.
5. If in steps 2, 3 and 4 any changes occur in the values of node properties, the node should send a Route Discover packet to its neighbors containing the new value of the parameters.

In the proposed algorithm, each node receives Route-Discover packet from all its neighbors. It selects the lowest neighbor's ID as its ID. When all nodes send the Route-Discover packets, the value of best cost route field in their routing table is set to the value of least cost route. At the end of route discovery phase, each node knows the cost of sending data from itself to the sink node.

To make Route Discovery phase more clear an example is given in the following. Consider a wireless sensor network with a random topology. Suppose that there is a unique sink node in the network. The sink node as the initiator of this phase sends Route Discover packet to all its neighbors. The sink ID field in these packets is set to zero; therefore as describe above, the value of ID for all neighbors will be equal to 1. In figure 2, the node ID of all 3 sink's neighbors will be set to 1. For each neighbor, the cost of route between it and sink node depends on its distance to sink.

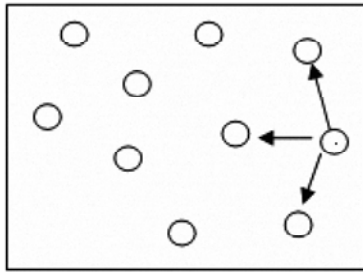


Figure 1. Sink as the initiator node of route discovery phase, sends first packets

As shown in figure 3, suppose that the upper neighbor of sink receives the Route Discover packet. The node obtains its ID from received packet's ID and then sends the Route Discover packet to its neighbors. As the next hop for this node is the sink node, so the value of route cost field in the Route Discover packet is equal to the cost between it and the sink node. As the Route Discover packets are broadcast to all neighbors, so the sink node will also receive this packet from its upper neighbor, but as the sender ID of the received packet is bigger than the sink's ID, the sink node doesn't process the received Route Discover packet. Now consider the Square node shown in figure 4. When it broadcasts Route Discover packet to its neighbors, the Diamond node will receive this packet.

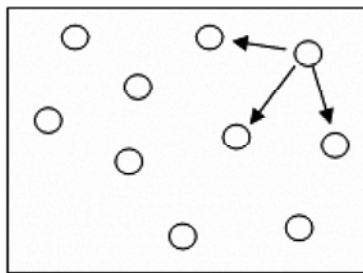


Figure 2. The upper node sends Route Discover packet to its neighbors

The Diamond node has already received the Route Discover packet from the Triangle node. So its current ID is equal to 2. For the Diamond node, the next hop node in the least cost route toward sink is the Triangle node. So when Diamond node receives the Route Discovery packet from the Square node, its ID is not changed. But the least cost route between Diamond node and sink node may be changed. The cost of route between Diamond node and sink node is equal to the sum of cost between Diamond node to Square node and Square node to sink node.

If cost of route from Square node is less than that of Triangle node, the least cost route and next hop node of Diamond node will be changed. Note that in the route discovery phase the cost of routes is propagated between all nodes using Route Discover packets. This procedure is continued while all the nodes obtain their least cost route.

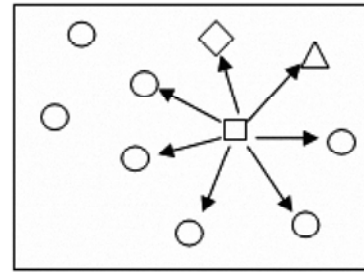


Figure 3: The Square node sends Route Discover packet to its neighbors

B. Data transmission phase

When a node detected an event, it should send data related to that event to the sink. As mentioned before, the routes are established in the route discovery phase. All nodes know their least cost route to the sink. So, using the optimal path the node will be able to send its data to the sink. Each node knows its next hop in its least cost route. When a node detected an event or received any data, it sends them to the sink node via its next hop node.

C. Route recovery phase

This phase is executed periodically. The length of time periods depends on the node's mobility. If a node dies, it will never participate in the routing procedure in the next period. Therefore, the dead nodes are not belonging to any established route. If the next hop node is failed, the data are sent using a backup node. All nodes in the network know the cost of forwarding information through their neighbors. When the least cost route is failed then the node forwards data using the second least cost route. As the information about all the possible routes from a node to sink is stored in the node routing table, so it is easy to find the first and second least cost routes. When the reminding energy of a node is less than a predetermined threshold, it will inform this situation to all its neighbors. If a node realizes that its next hop node doesn't have any sufficient energy, it uses its second least cost route to send its data.

IV. PROPOSED NEIGHBOR DISCOVERY PHASE

In this section, we explain the operation of proposed neighbor discovery phase. Most of routing algorithms need to send data to a specific neighbor. In essence, wireless links are broadcast links which means, when a node sends a packet, all the nodes placed in its communication range will receive it. In this situation, every node needs a mechanism that makes it enable to send data to a particular neighbor so that the other neighbors wouldn't process those data.

All energy aware routing protocols need neighbor discovery mechanism. Proposed approach uses a hop by hop routing algorithm; route to the sink is selected by each node via choosing next hop, meanwhile different routes are picked out by considering different next hop nodes. Most of routing algorithms use hop by hop strategy which is more efficient. All nodes which use hop by hop routing algorithm need information only about their next hop, which means

they just need local information. When an algorithm needs to have a global view of the entire network, it absolutely must pay much more in contrast with the situation with only local view. Neighbor discovery algorithms collect local information about node's neighbors. To distinguish nodes from each other, we can assign a unique identifier to each node. This identifier makes enable the other nodes to select one node uniquely. By considering this deployment, all the node's neighbors will receive the data, but only one node that is identified by the packet destination identifier field will process it. The node's identifier could be local or global. When a node's identifier is global, the node could be identified by the other nodes uniquely. But as we mentioned before, this type of identifying is too expensive. When a node uses local identifier, it can only distinguish its neighbors. As the proposed algorithm needs network's nodes to distinguish their neighbors uniquely, so it doesn't need global identifier and the local identifier is sufficient. In the following, we propose a new neighbor discovery mechanism for distinguishing node's neighbors.

In the proposed neighbor discovery mechanism, each node estimates its distance to the sender using received signal strength. This parameter can be used for distinguishing node's neighbors. In the route discovery phase many packets are transmitted between nodes. Using the signal strength of these packets, the receiver can estimate its distance to the sender node. Therefore at the end of route discovery phase all nodes know their distance to their neighbors. As discussed in section 3, in the route discovery phase an ID is assigned to each node. This ID is not unique in the entire network. The nodes with equal ID have the same number of hops to the sink. The proposed neighbor discovery algorithm uses both node ID and received signal strength to distinguish neighbors with a suitable accuracy rate. We believe that by using the distance between two nodes and the node ID, we can distinguish the neighbors with a high accuracy. If by using these two parameters, the node couldn't distinguish all its neighbors, this means that some of its neighbors have the same distance and ID. In this case, the proposed mechanism uses a random number to discern them.

When a node detects a collision, this means that it has more than one neighbor with the same distance and ID. It sends a Collision Recovery packet to the neighbors. In this packet the sending node advertises that only the nodes which detected any collision should process it and the other neighbors should ignore it. When the nodes which detected collision receive this packet, they create a random number between 0 and MAX (usually MAX is a big number, e.g, 100000) and send it for the node that has sent the Collision Recovery packet, using Collision Recovery Reply packet. Both sender and receiver, store this random number in their routing table in an appropriate record. This random number makes distinguishing action complete. By using distance (signal strength), ID and if needed the random number, it is possible to distinguish neighbors from each other. When a node wants to send a packet to one of its neighbors, it

should use all of 3 mentioned parameters in the packet. All neighbors receive the packet, but only the neighbor which can find a match and has the same properties will process the packet and the other nodes will ignore it. To evaluate the performance of the proposed neighbor discovery phase, we implemented it in a simulator. Table 1, shows the simulations results. As mentioned before, the collision is only occurred when a node has more than one neighbor with the same distance and ID.

Table 1. Number of collision occurred in different experiments.

Simulation trials	Area (squam m)	Number of nodes	Communication range (m)	Number of detected collision
1	50*50	200	5	0
2	100*100	200	5	0
3	100*100	500	5	2
4	500*500	1000	20	23
5	10*10	100	1	12

The results shown in table 1, confirm that by increasing the density of nodes in the network, the probability of collision is also increased. We should emphasis here that, by using the third parameter (random number) as explained earlier, the distinguishing rate may be reach to 100%. When the mechanism uses the first two parameters (node ID and signal strength) the overhead is always zero, but when the third parameters is applied, only 2 extra packets should be transmitted in the network, that can be disregarded relative to number of packets transmitted in other phases.

IV. SIMULATION RESULTS

In this section, using computer simulation, we evaluate the performance of the proposed energy aware routing protocol with that of SPEED protocol. Before evaluating the performance, we describe the environment of our simulation. After that we analyze the simulation results and compare the performance of the proposed protocol with that of SPEED protocol.

A. Environment setup

We developed a simulation software using . To compare the performance of both protocols, we implemented the proposed protocol as well as the SPEED protocol in our simulation software. The simulated network topology consists of 100 fixed sensor nodes which are randomly deployed in a 200m × 200m area. Each node is able to send data in a range of 40m. There is one sink node at point (0, 0). The location of sink node can be changed in many scenarios. We consider many different scenarios to evaluate different aspect of the proposed algorithm.

B. Results analyze

In figure 4, for both proposed algorithm and SPEED algorithm, the average energy consumption of all nodes is plotted versus number of events. Less energy consumption means longer lifetime for the network. Horizontal axis shows the number of events which are occurred in the network terrain. Events are occurred in a random place in the network. Vertical axis shows the average energy consumption of the network nodes. The scale of vertical axis is 0.00005 J . Based on results shown in figure 4, it is obviously observable that the average energy consumption of the network nodes in the proposed protocol is less than that of SPEED protocol. When the number of events is less than 200, the energy consumption of two protocols is nearly equal. But when number of events is more than 200, the proposed protocol consumes less energy than SPEED. It is necessary to note here that the main goal of proposed algorithm is to decrease energy consumption.

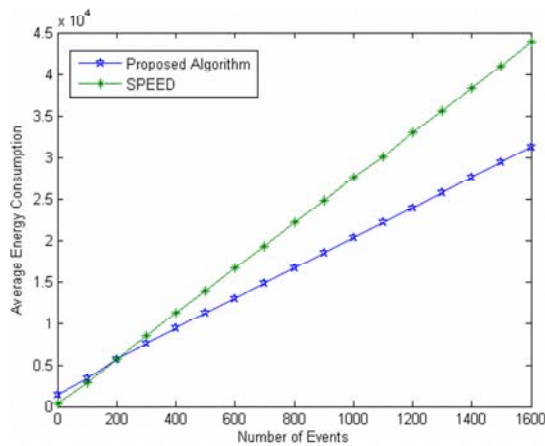


Figure 4. The comparison of average energy consumption between two protocols, number of nodes =100.

In the next trial, we increased the number of nodes to 200. The results are shown in figure 5. As shown in this figure, when the number of events is less than 500, the average energy consumption of the SPEED protocol is less than the proposed protocol. Note that the proposed protocol uses a proactive routing algorithm; it means that the routes are established in advance before data transmission. So, when the number of events is low, the average energy consumption of the proposed protocol is more than SPEED.

In figure 6, for a network with different amount of nodes, the energy consumption of two protocols is shown. Horizontal axis shows the amount of network nodes and the vertical axis shows total network energy consumption. Total network energy consumption is calculated as the sum of energy consumption in all network nodes. It could be seen in figure 6 that for different number of nodes, the total energy consumption of the proposed protocol is less than that of SPEED protocol. Furthermore, it is obviously observable that by increasing the number of nodes, the performance of proposed protocol is also increased.

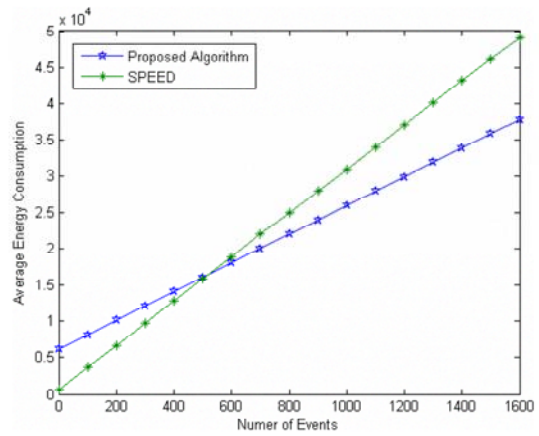


Figure 5. The comparison of average energy consumption between two protocols, number of nodes =200.

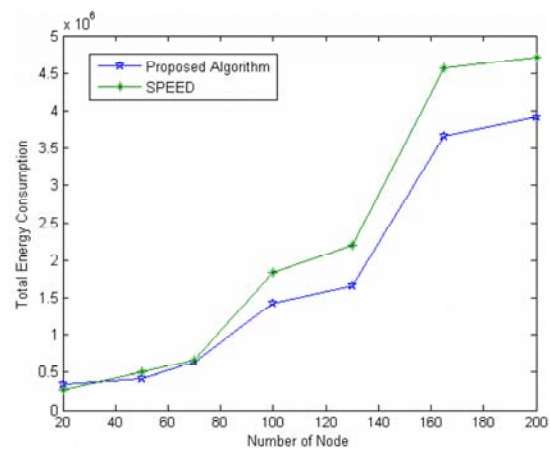


Figure 6. The total network energy consumption for different amount of network nodes.

In figure 7, for both two protocols, the number of dead nodes is plotted versus the number of events. If the energy of a node is finished, it will be dead. When a wireless sensor network has high number of alive nodes, it will live longer. Note that a wireless sensor network with higher number of nodes can perform its functions better. Figure 7 shows that the number of dead nodes in the proposed protocol is always less than that of SPEED protocol. Based on results shown in figures 4-7, it is clear that the proposed protocol has better performance in comparison with the traditional SPEED protocol.

In the next simulation trials, we evaluate the delay performance of the proposed protocol. As we mentioned earlier, by decreasing the transmission delay, it is possible to decrease the end to end delay. Figure 8 shows the simulation results related to delay performance of both protocols.

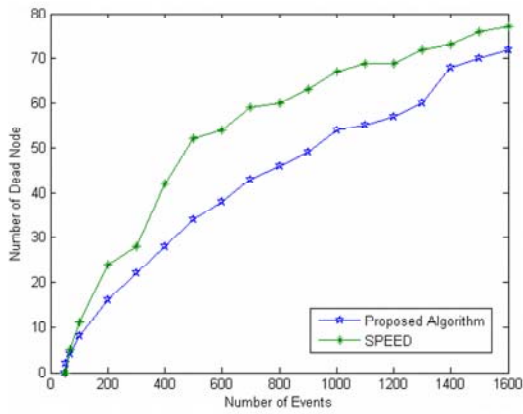


Figure 7. Comparing the number of dead nodes between two protocols

Results show that the path traversed by packet using proposed protocol has less delay than that of SPEED protocol. Horizontal axis illustrates the place where the event has occurred. Note that events occur in points with equal width and length. For example, number 200 in horizontal axis means that the event has occurred at point (200,200). The vertical axis shows the path delay which is related to the length of the route that a packet traverses between source and sink. Results shown in figure 8 clear that the end to end delay of the proposed protocol is less than that of SPEED protocol. Figure 9 shows the end to end delay, in the case that the number of nodes in network has been increased to 200 nodes. Based on results shown in figures 8,9, it is clear that the proposed protocol has better delay performance than the SPEED protocol so it is more suitable for real time applications. As discussed in section 1, two main objectives of the proposed protocol are to minimize energy consumption and to choose a route with minimum end to end delay. The simulation results show that the proposed protocol has achieved to both of its goals.

C. Real time traffic

To make the proposed routing protocol more suitable and more scalable for real time traffic, we used the clustering techniques. In this case, the network is divided to some clusters using a clustering technique presented in [17]. For this purpose, sensor nodes are grouped into clusters controlled by a single cluster head. Sensor nodes are only responsible for probing the environment to detect an event. Every cluster has a cluster head that manages the other members in the cluster. Clusters can be formed based on many criteria such as communication range, number and type of sensors and geographical location. We assume that all nodes are stationary and the cluster head is located within the communication range of all the cluster members. The routing process in hierarchical routing protocols is divided into two parts. First, routing between cluster members and cluster head, and second, routing between cluster heads and sink. In this section we emphasis on the first routing type.

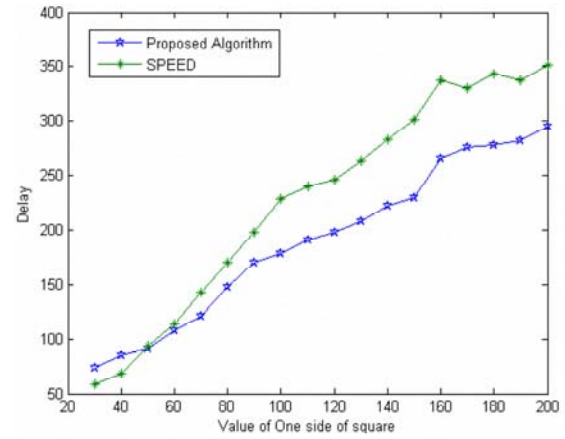


Figure 8. End to end delay(ms) between source and sink (number of nodes=100 nodes)

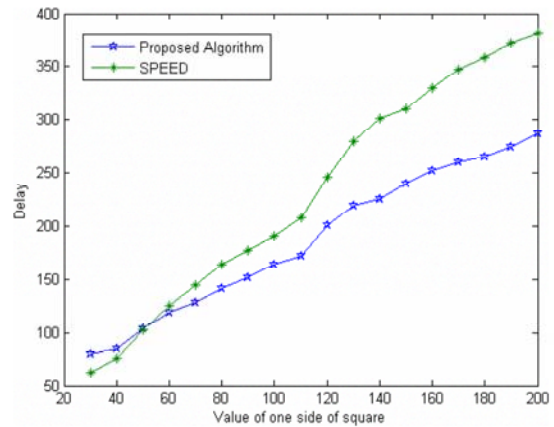


Figure 9. End to end delay(ms) between source and sink (number of nodes=200 nodes)

To forward traffic toward sink node, each cluster head should be able to route data to other cluster heads. The cluster head is responsible to find best route for all its members in terms of energy consumption and the end to end delay requirement. We consider two types of traffic: real time and non-real time. Real time traffic has hard constraint on the value of end to end delay while non real time traffic doesn't have any specific delay constraint. Both real-time and non-real-time traffic coexist in the network. As delay constraints are associated only with real-time data, the cluster head is responsible to find the best path for this kind of traffic so that the end to end delay requirement is satisfied for real-time traffic. Each sensor node uses different queues for the two different types of traffic. Furthermore, each node has a classifier, which checks the type of the incoming packet and sends it to the appropriate queue. There is also a scheduler, which determines the order of packets to be transmitted from the queues. The cluster heads use the proposed cost function to find the best path which not only can meet the delay requirement but also

consumes the minimum energy. We use a modified version of Dijkstra routing algorithm. The cluster head is responsible to find the best route for all of its members. It selects the more suitable route that has enough resources for transmitting real time traffic from nominee routes with lowest energy consumption. After finding the best route, the cluster head sends the routing information to all of its members. So, each sensor node in each cluster knows the best path between itself and its cluster head and uses it for transmitting real time traffic. The cluster heads use an existing routing algorithm to transmit traffic toward the sink node. Non real time traffic is sent using Gossiping [16] algorithm. In the simulation, we set the size of each cluster to 40m*40m. Each cluster consists of some nodes. In figures 10(a,b,c), for both SPEED and the proposed protocols and for a cluster with 20 nodes, the average energy consumption, the energy consumption per node and the queuing delay are given. In this experiment, real time traffic is produced with a constant rate but non real time traffic have variable rate. Figure 10(a) shows the average energy consumption versus number of events occurred in the cluster. It can be seen that the proposed routing protocol consumes less energy in comparison with the traditional SPEED protocol. In figure 10(b), the energy consumption of each node is given. In figure 10(c), the queuing delay of real time traffic is plotted versus different number of non real time packets. As the geographic area of a cluster is very small, so the queuing delay is major part of the end to end delay. As it can be seen, for the proposed routing protocol, the increasing in non real time traffic density doesn't have any serious effect in the delay performance of real time traffic. In the next simulation trial, we increased the number of sensor nodes in a cluster to 40 nodes. In this scenario both real time and non real time traffic has variable rate. In figure 11(a,b,c), the results are shown. Based on results given in figures 10,11, it is clear that the proposed routing protocol has better delay and energy consumption performance than the traditional SPEED protocol.

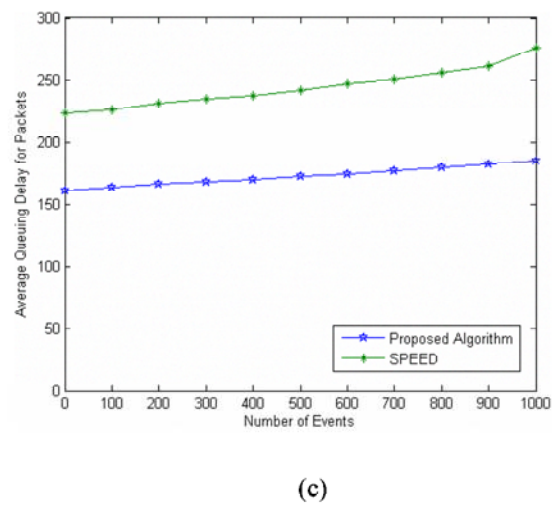
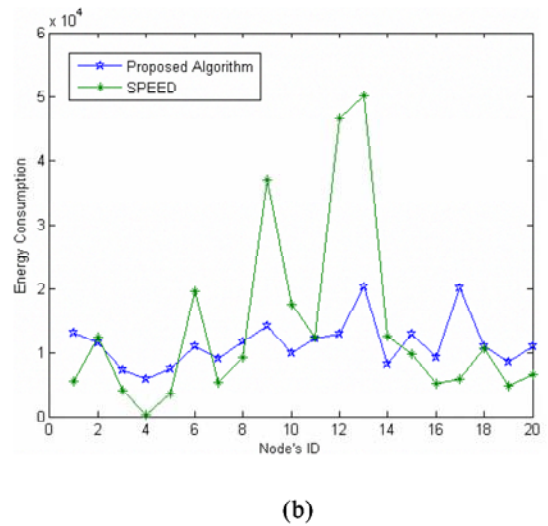
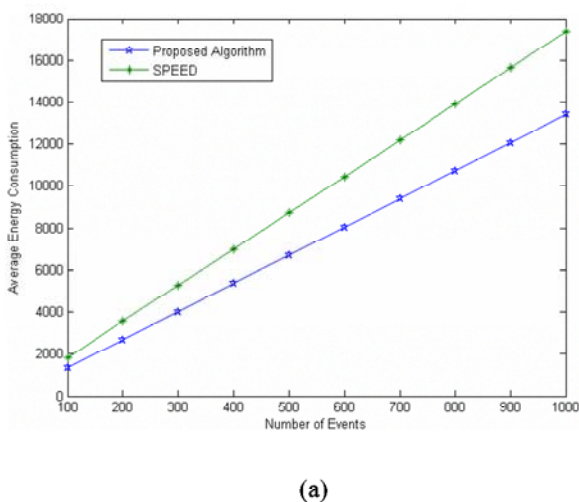


Figure 10. Performance evaluation of two protocols for real time traffic: (a) average energy consumption, (b) energy consumption of each node and (c) queuing delay (ms) (number of nodes in cluster = 20 nodes)

V. CONCLUSION

Energy aware routing is most challenging issue in wireless sensor networks. Current research on routing of sensor data mostly focused on protocols that are energy aware to maximize the lifetime of the network, scalable for large number of sensor nodes and tolerant to sensor damage and battery exhaustion. In this paper an efficient energy aware routing protocol was proposed. The proposed routing protocol has two major goals which are low power consumption and high delay performance. We evaluated the performance of the proposed protocol under different scenarios. Simulation results confirmed that the proposed protocol is more efficient in energy consumption in comparison with the traditional SPEED protocol. Furthermore, the proposed routing protocol can find the optimal path with a low end to end delay. We believe that, by using data aggregation techniques the higher performance will be achievable. Also using other cost